ON THE INTERACTION OF THERMAL BUCKLING AND DEBONDING OF PATCHED STRUCTURES

by

PAMELA MARIE CARABETTA

A dissertation submitted to the

Graduate School – New Brunswick

Rutgers, the State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Mechanical and Aerospace Engineering

written under the direction of

Professor William J. Bottega

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2011

ABSTRACT OF THE DISSERTATION

On the Interaction of Thermal Buckling and Debonding of Patched Structures

by PAMELA MARIE CARABETTA

Dissertation Director: Professor William J. Bottega

The widespread use of patched assemblies in engineering structures creates the need for better fundamental understanding of the failure phenomena of such structures to ensure their safe and effective usage. A structure possessing two substructures, the "patch" and "baseplate", with mismatched coefficients of thermal expansion is studied under thermal loading for two extreme edge conditions. The composite structure is taken as initially flawed, such that partial separation exists at the edges of the patch between the two substructures. Two relevant failure mechanisms are sling-shot buckling and edge debonding (separation). Sling-shot buckling, first observed for perfectly intact structures under thermal loads, occurs when the structure dynamically slings from one equilibrium configuration to another in an opposite sense of deflection at a critical load. It is desired to be able to predict and characterize buckling of the structure, and its coupling with debonding. The geometrically nonlinear problem is formulated via a variational formulation, which allows the boundaries of the domains of the structure to vary in addition to the displacements. The vanishing of the first variation of the potential energy yields governing equilibrium equations, boundary/matching conditions, and transversality conditions. The transversality condition yields a Griffith type delamination criterion. A closed-form solution is obtained after recasting the problem in a mixed formulation. A stability analysis is performed using the second variation of the potential energy

functional. The partially debonded structure is seen to possess a “dual nature”, where it is structurally different depending on the deflection, due to the existence of the initial flaw. It experiences sling-shot buckling, which, coupled with the structure’s duality leads to what we refer to as “Buckle Trapping.” It is suggested that, for certain critical temperature fields, the structure oscillates dynamically between unstable equilibrium configurations. In addition, examination of the energy release rate against the bond size reveals a variety of possible behaviors, dependent on flaw size, temperature difference, and bond strength. It is observed how and when separation leads to thermal buckling, and vice versa. Through this analysis, the onset, extent, and stability of debonding, as well as the relationship with buckling, are diagnosed and characterized.

Acknowledgement

I came to Rutgers University as a naïve high school graduate, and chose to major in mechanical engineering based on my interest in physics and mathematics. That was more than ten years ago, and I am finally ready to move on after a long, difficult, yet rewarding odyssey. It was by no means easy, as there were plenty of times when the pressure was overwhelming. A lot of people have been influential to me throughout this portion of my life, and here I will recognize them.

The first person I wish to thank is none other than my advisor, Dr. William J. Bottega. I met Dr. Bottega when I was in his undergraduate Vibrations class in the Fall of 2003. He encouraged me to continue my education, and the next thing I knew, I was his Master's student. I knew all along that I was going to follow through with my goal of obtaining a Ph.D, and I stuck to that commitment, both to Dr. Bottega and myself. To put it simply, Dr. Bottega is the most brilliant person I have ever met. His knowledge is incredibly vast, and he has an answer to every question you could imagine. He is always very honest with me, and has always treated me like an equal, even at times where I was not sure I deserved the honor. To support myself, I had to work as a TA, and he was very patient with me when I struggled to find balance between my responsibilities and my research. Students are sometimes intimidated by Dr. Bottega, due to his reputation for being very demanding of them. It is a shame they are, because if they got to know him they would see that he is very caring, and has a great sense of humor. I will forever cherish what I learned from him, from a research perspective, as well as a life perspective.

I would also like to thank Dr. Haim Baruh and Dr. Haym Benaroya, for serving on my dissertation committee. I met both of them while I was an undergraduate at Rutgers. It was Dr. Baruh who had originally accepted me into the graduate program. Dr. Benaroya welcomed me to take an office space in his Center for Structures in eXtreme Environments (CSXE). I felt like I was one of his own, and I really enjoyed the camaraderie that they had established there. Dr. Baruh and Dr. Benaroya are both quite easygoing, and they have always been helpful when I needed it. I really appreciate all they have done for me.

In addition, I'd like to cite my "out-of-department" committee member, Dr. Yook-Kong Yong of the Rutgers Civil Engineering Department. I have only recently met him, but I am very grateful for his willingness to serve on my committee. He has shown genuine interest in my work, and has been very constructive in his commentary.

During my time as a teaching assistant, I worked for several people, including Dr. Peng Song, Dr. Kook Pae, and Dr. Mitsunori "Mitch" Denda. I spent many semesters involved with the Statics course, and for my last one, the Department Chairman Dr. Yogesh Jaluria asked me to handle the actual lecture. I truly appreciated the opportunity and it was a great experience. I would like to give special recognition to Dr. Alberto Cuitiño and Dr. Andrew Norris. I worked with both of them very closely as I began to accept a more authoritative role in managing the class. They both treated me like a true colleague and not as a subordinate.

There are times when difficult situations arise while teaching a class. It was in these instances that I would seek the counsel of Dean Donald Brown and Dean Ilene

Rosen of the Office of Student Development. They were always able to assist me; not to mention, I always enjoyed their company.

The Mechanical Engineering Department has always had the finest administrative staff that you could ask for in Helene Press, Virginia Dare, Aiesha Long, Carmen Elsabee, Shefali Patel, and Cindy Cartegna. These women have been able to help me no matter what I needed, and they always seemed so happy to do so. Staff member John Petrowski has also always been able to help me, particularly when I needed it most during my undergraduate years. The department could not function without the contributions of these individuals.

I would be remiss if I did not mention Trisha Mazzuco, who had first worked for the department, and later worked with us in CSXE. She was the heart and soul of the Center, and she worked very hard there. But more importantly, she was a good friend to me. Working in CSXE, I also met Dr. Yuriy Gulak, who seems to know everything there is to know about computers. I wish to thank him for all the things he has done for me during my time here. Not only is he brilliant, but he is very kind.

I'd like to mention my good friend, Dr. Elan Borenstein. Elan was my classmate, and we went through all of the rigors of graduate school together: classes, teaching assistantships, and even the qualifying exam. His desk was right next to mine in CSXE, and we would spend plenty of time talking about everything that was on our minds at the moment. It's important to mention that he is also very good with computers as well as programming. Some simple programming advice from him really helped me when I needed it, and I am forever grateful for that.

The other students I met at CSXE are Dr. Jason Florek, Paola Jaramillo, Dr. Subramanian "Subbu" Ramakrishnan, Tushar Saraf, Nir Pony, and Paul Bonness. They are all great people, and it was fun to be around them.

In my time as a TA, I worked alongside Dr. Pedro Romero, Dr. Zhanyu Sun, Dr. Ke Xu, Dr. Yi Pan, Jingru Zhang, and Alexey Titovich. No matter who it was, we always worked well as a team. The amity was always helpful in getting through what could occasionally be a challenging and stressful job. In addition, many times, visiting lecturer Dr. Frank Pavlovich was a part of our team. The man is somewhat of an enigma; however, I took the time to get to know him more than most people did. What I found is that he is very intelligent, and quite passionate about what he does. While he and I did not always see eye-to-eye, mutual respect and understanding allowed us to maintain a very cordial working relationship. I'd also like to thank a most unlikely source of inspiration to me: my Statics students. You learn to expect certain things as a teacher, and for the all the students that drive you crazy, there are plenty of the opposite, who make the job worthwhile. They let me know that they trusted me and believed in me, which always was very encouraging.

I want to pay homage to some of the many other students whose paths I have crossed during my graduate education: Kevin Gomes and Joe VanderVeer (both of whom were in my undergraduate class at Rutgers as well), David Sindall, Megan Smith, Alisar Tuncer, Srinivas Iyer, Jon Doyle, Mohamed Sadik, Dhaval Dadia, Cassandra D'Esposito, Christel Seitel, Kellie Anderson, Mona Golbabaie, Po-Ting "Tim" Lin, Gulsad Kucuk, and Huihui "Sonya" Qi. For a brief time (too brief) Sonya's desk was next to mine. It was interesting to me help her become acclimated to life in the United States. She is a

wonderful person, and I always appreciated her company. Finally, to Prakash Rao, Jill Mesonas, Dr. Michael Muller, and the rest of the Center for Advanced Energy Systems (CAES) softball team: may you always have at least one win per season. Hey, we have to keep our goals realistic!

Finally, I'd like to thank my family. I know how proud of me they all are. In fact, one of my final memories of my Grandpa Hulley was him boasting to the nurse in the hospital, in his waning moments, that my sister and I were going to be doctors. I can only wish that Grandpa Carabetta could have been around long enough to know what was in store for my future. I know he would have been proud. I'm glad I still have my two grandmothers, and I appreciate everything they have ever done for me.

My parents provided me with a wonderful childhood, and I always had everything I ever needed. They still take care of me when I need it, unconditionally. I know I generally don't tell them much about what is going on with my work. I guess that is because, to me, there usually wasn't anything exciting to report, and I thought they would find it boring. But I should realize that they think everything I do is amazing and interesting. I know they are my number one fans, even though I am a huge pain in their sides!

And then there is my big sister, or, as she is formally known: Dr. Valerie Carabetta. I have always looked up to her. The way Val operates is incredible – she is almost robotic in her work ethic and time management. I've always wanted to be more like that, and believe me, I keep trying! It is impossible for me to put into writing how much I want to thank her for all she has done for me. We're Best Friends For Life.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The layered structure is a key part of the aerospace industry. Aircraft panels experience the wear and tear of regular use in harsh conditions, resulting in their physical breakdown. Small cracks on the fuselage/wing panels can (and occasionally do) have devastating results. From an economic standpoint, repair is more pragmatic than replacement. One prevailing method of repair is the use of a "patch" to adhere over the damaged panel. This transfers the load to the patch and hence reduces the stress on the bottom panel.

The utility of the patched assembly extends well beyond the aerospace industry to reaches such as "smart" electronic materials and structures. Of course, adding a structural element presents new challenges which must be addressed. It is not uncommon for such a structure to be employed in an environment of varying temperature. Should the individual layers possess differing thermal properties, a structure will transversely deflect upon being exposed to a temperature difference. It is known that a fixed patched structure can experience thermal buckling under certain loading circumstances. But this is not the only bane of its existence, as edge debonding is also of great concern when using layered structures. Stress is concentrated at the edges of the bonded patch; hence separation failure is most likely to occur at those points. The stiffness of the structure is compromised by the onset of debonding, which alters the behavior of the structure. These phenomena, buckling and debonding, will ultimately render the patch useless.

This can lead to dire consequences, which, depending on the application, could even be life-threatening. It is of the utmost importance to be able to predict and understand such behavior in the quest of safety. From here we find the motivation to pursue the characterization of the combined buckling and debonding behavior of a patched assembly under an applied temperature difference.

In the following section we review some of the literature discussing the phenomena of thermal buckling and/or debonding.

1.2 A Survey of Related Literature

The use of patched assemblies in modern technology has been a crucial issue in engineering structures. Especially in the aircraft industry, the structural integrity of repair patches is a primary safety concern. Baker et al. (2009) note that when using a bonded repair patch, it is difficult to be able to predict the structural integrity, which can be compromised by debonding and fatigue. They resort to monitoring "structural health" using strain gauges in several case studies. Naboulsi and Mall (1997) use a two-dimensional finite element analysis (FEA) to predict the most appropriate type of repair patch for use in a thermal environment. It is seen that the appropriate material mismatch between the layers depends on the configuration of patch, and if it is one-sided or two-sided. While such sample works focus on designing and monitoring repair patches, it is necessary to characterize the failure mechanisms for these structures.

The classic work of Timoshenko (1925) remains a hallmark of thermal studies of layered structures. Timoshenko presents a thorough discussion on the behavior of a bi-metallic strip subjected to a uniform thermal load. The problem is formulated analytically from first principles. It is seen that the maximum stresses during heating occur on the

bearing surface mating the two materials. The materials possess different thermal coefficients of expansion, which causes bending during heating. For a strip that has an initial curvature, Timoshenko notes that upon heating, the strip begins to deflect, and when the load reaches a critical level, the strip experiences “sudden buckling”. The structure deflects (in the same direction) through the flat configuration into a configuration of reverse concavity. This is commonly referred to as *snap-through buckling*. During cooling of a structure which has buckled in this manner, it will “reverse buckle”, that is, it will undergo sudden buckling back to its original concavity. This is a *snap-back* phenomenon. This behavior is discussed in the classic two-part work of Wittrick (1953) and Wittrick et al. (1953). A similar problem is formulated, however the bi-metal structure studied is an axisymmetric disk. As in Timoshenko’s analysis, the structure is considered as possessing an initial curvature; hence the disk has the form of a spherical cap. The stability analysis of the load-deflection curves suggests snap-though and snap-back phenomena when the structure is heated and cooled, respectively.

Several different techniques are used to model and analyze buckling behavior of layered structures today. A popular approach to studying buckling and post-buckling behavior is via numerical methods and FEA, such as that performed by Lee et al. (2002), who study post-buckling behavior of laminated cylindrical panels. The panels are subjected to a uniform as well as a nonuniform (varying through the thickness) temperature gradient. They arrive numerically at a nonlinear solution, and conclude that important factors affecting post-buckling behavior are the thickness of the patch as well as the lamination angle. The panel is found to function better with a patch on the outside rather than inside. They observe snap-through and snap-back behavior. Özben (2009)

also uses FEA, along with analytical methods to predict the critical bifurcation buckling loads for a fiber-reinforced composite orthotropic plate in compression (no thermal load). The buckling loads are predicted for varying support conditions and layup configurations. The analytical method begins with an expression for potential energy. The Ritz method is employed to obtain the critical buckling parameter values. The two methods are shown to have a favorable consistency for low buckling modes, but diverge at higher buckling modes. Also using the Ritz method is Aydogdu (2007), who performs an analysis of thermal buckling behavior on cross-ply laminated beams. A three degree-of-freedom shear deformable theory is used in conjunction with a shape function to fulfill geometric and material constraints. The displacement components are each assumed as a series of polynomials. The buckling temperature depends on many factors, and buckling is seen to occur for heating and/or cooling of the structure. Pi and Bradford (2008) examine thermal buckling and develop a method for calculating the shear center and effective centroid for fixed beams loaded with a linear temperature gradient field. The beam experiences combined uniform compression and bending. There are multiple buckling possibilities predicted, including bifurcation in an out-of-plane lateral torsional mode, or an in-plane flexural mode. They suggest that as the temperature difference increases, the thermal bending become increasingly more significant than thermal compression. The dominant factor for in-plane flexural buckling is shown to be the critical temperature at the effective centroid. Zenkour and Sobhy (2010) use a sinusoidal shear deformation plate theory to model thermal buckling phenomena of sandwich plates. Various configurations of the functionally graded material (FGM) plates are used, as well as different types of thermal loading. The mathematical formulation used is attributed to

Reddy (2000). A comparison of numerical solutions corresponding to various mathematical plate models is presented, and it is seen that general agreement of the methods depends on the aspect ratio of the plates. They also characterize several parameters affecting the critical buckling load, such as plate aspect ratio, core thickness, and material. Yang et al. (2006) explore post-buckling behavior of FGM cylindrical panels. The panels undergo an axial load as well as an applied temperature difference. A classical shell theory with von Karman-Donnell nonlinearity is used as the basis of the mathematical model. The critical buckling load is predicted semi-analytically and via an iterative process. Factors affecting predicted buckling load level include boundary conditions, pre-stress, and initial geometric imperfections. Song and Li (2007) perform a buckling analysis considering an Euler-Bernoulli beam on an elastic Winkler foundation. The beams are pinned-fixed and experience a static temperature rise. The geometrically nonlinear problem is formulated analytically, and is solved numerically using a shooting method. It is seen that the parameters of the foundation affect the post-buckling behavior.

Classical Euler bifurcation buckling occurs for thin structures possessing a free end that are acted on by compressive loads. Upon reaching a critical buckling load threshold, multiple equilibrium configurations exist, and hence the load-deflection path bifurcates. Snap-through buckling is a phenomenon associated with transversely loaded shallow arches. Upon reaching a critical load value (limit point), the structure deflects dynamically (in the same direction) to an alternate configuration. However, it has been shown by Rutgerson and Bottega (2004) [as well as Rutgerson (2001)] that such structures will experience pre-limit point buckling. In this current work we are

examining the buckling behavior of a fixed flat patched structure under applied thermal loading, which is known to undergo *sling-shot buckling*. Figure 1 presents a schematic of the load-deflection behaviors that display three types of buckling phenomena.

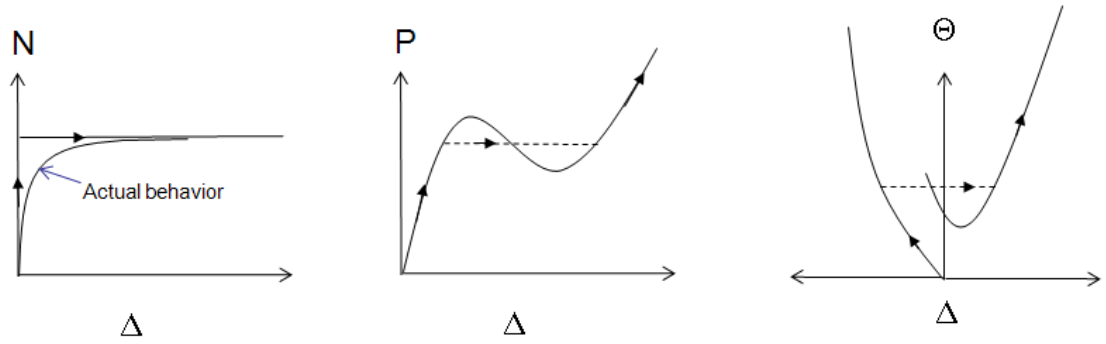

Figure 1. Various depictions of buckling phenomena. Left-to-right: compressive load *N* vs. characteristic deflection Δ showing bifurcation buckling, transverse pressure *P* vs. characteristic deflection Δ showing snap-through buckling, and thermal load Θ vs. characteristic deflection Δ showing sling-shot buckling

Hence, the relevant mechanism is sling-shot buckling, which was first observed and explained by Karlsson and Bottega (2000). They study a perfectly intact structure, that is, no debonding exists between the patch and the baseplate. Upon experiencing and increase in thermal load, the structure deflects in one direction, until the loading reaches a predicted critical value. Beyond the critical load, the structure dynamically "slings" to an alternate equilibrium configuration in the opposite sense of deflection. A thorough review of this phenomenon may also be found by Karlsson (1999). Sling-shot buckling has been shown to also occur in curved structures [Rutgerson and Bottega (2002)] loaded with a uniform temperature field. The combined findings for flat structures and shell structures have been united by Bottega (2006). An important and unique feature of these works of Bottega, Karlsson, and Rutgerson is the variational approach taken to formulate the problem. This technique for formulating the problem is used in the current work, and will be described in Section 1.3.

The edge debonding problem may also be approached in the same spirit, as originally established by Bottega (1983). Bottega (1995) explores the debonding behavior of various layered assemblies, including a lap joint and a patched plate. Mechanical loading types considered are in-plane tension, three-point loading, and applied transverse pressure. A growth criterion based on the energy release rate is used, and this generates what is known as a *growth path* for debonding. (Later Bottega (2003) decomposes energy release rates on the structural scale based on mode of debonding, for flat and curved layered structures. A breakdown of energy release rate by mode is also performed by Williams (1988), however, it is done on the local scale in the vicinity of the crack tip.) A growth path represents a threshold curve of equilibrium positions associated with the moving boundary of the bond between the layers. The analysis performed in 1995 is linear. The work is further expanded upon by Karlsson (1999), and Karlsson and Bottega (1999) who explore the phenomena of edge contact. Carabetta and Bottega (2008) demonstrate that geometric nonlinearities should be retained in order to more accurately capture the debonding behavior a patched structure loaded with transverse pressure. It is shown that the geometrically nonlinear solution deviates significantly from the linearized solution before any debonding behavior occurs. The significance of geometric nonlinearities on edge debonding is also seen by works (such as Brusat or Erdogan and Joseph) included in the “round robin” by Johnson (1987). Carabetta (2007) analyzes the influence of an applied temperature difference on a pressure loaded patched structure. It is seen that the temperature has a strong effect on the debonding threshold. The mathematical model is formulated from first principles, as applied to a local force balance on an element of the structure. The same problem is then re-formulated using a

variational approach by Bottega and Carabetta (2009), who also study the influence of applied temperature on a pressure loaded patched structure with edges that are free to translate in-plane.

The aforementioned works of Bottega and Carabetta include the possibility of the presence of a contact zone, a region where the primitive structures are not bonded, yet remain in sliding contact. It is seen that this existence has major implications on the overall behavior of patched structures. This notion is also reinforced by the work of Comiez et al. (1995) who study delamination buckling of a composite plate using a combined analytical and experimental approach. The approximate analytical model is adopted from the literature and solved numerically. It is noted that the possibility of a delaminated portion of the layers remaining in contact should not be excluded from any analytical model, and the solution should dictate if contact occurs or not.

Other studies of debonding behavior focus on the stress fields within the components, such as the classic work of Goland and Reissner (1944). The stresses in the adhesive layer are calculated to be the highest at the joint edges, confirming that debonding is most likely to occur at the edges. Similar results are found by Delale and Erdogan (1981) as well as Bigwood and Crocombe (1989).

Like the thermal buckling problem, a variety of techniques are employed to study the debonding of a layered structure. Davies et al. (2006) used FEA to model debonding and the consequent propagation for a composite structure, subject to various loading, such as a point load or uniform pressure. They develop a special interface element in an attempt to avoid use of a more fine mesh. The model is applied to one-dimensional as well as two-dimensional configurations, and experimental results are planned for

validation of the FEA model. Alderliesten et al. (2006) use the energy release rate approach in an effort to model delamination in a fiber-metal laminate. The crack growth rate is predicted by using an empirical Paris relation, based on stress intensity factor. The relationship between crack growth rate and energy release rate is also derived experimentally. Byrd and Birman (2006) examine the effect of temperature on debonding of z-pinned joints. The z-pinned joints are intended to curb delamination, however, their effectiveness is compromised by the presence of a temperature gradient. The focus is to illustrate an approach to calculating residual thermal stresses. A double-cantilever beam specimen is used as a model. The criterion for failure, established previously by the authors, is based on an analysis of deflection of the debonded portion. It is determined that the temperature difference can lead to two modes of failure, but that in certain cases may actually be used to reinforce the joint. Also exploring delamination due to thermal effects are Hsueh et al. (2006). They study mode-I edge debonding, yet are unable to arrive at an exact analytical solution for peeling stresses, even for a bilayer system. Instead, the peeling moments are derived analytically, and they are able to provide suggested design guidelines for strips composed of specific materials. Moore (2005) formulates a general beam model from first principles using a force balance approach. The model is used to calculate the peeling moment arising from delamination of the structure under a uniform temperature difference. The structures have multiple layers with mismatched thermal properties. The model is then favorably compared with FEA, and is physically interpretable. Rabinovitch (2010) also considers the effect of temperature on debonding, specifically of beams strengthened with a bonded composite. The method is a higher order stress analysis. Energy release rates are calculated through

the *J*-integral. The model is validated using experimental results from the literature. There is seen to be a nonmonotonic dependency of debonding failure on temperature, which is attributed to the thermal impact on the interfacial stresses. A reduction in debonding strength is seen to correlate positively with increasing temperature.

Wu et al. (1998) present a comprehensive review of failure mechanisms of layered structures, including debonding and buckling. The structures considered are a flat composite panel subject to in-plane compression, and a layered shell subject to transverse pressure. The goal is to attempt to predict the dominant failure mode in the interest of design. FEA is used to achieve the results. The delamination considered is an internal delamination; however it is noted that regardless, the propagation of delamination adversely affects the structure's ability to carry a compressive load, thus affecting critical buckling behavior. For the flat structure, it is seen that the energy release rate increases rapidly during the post-buckling phase, which may cause delamination growth. Yin (1998) studies the thermal post-buckling behavior of an internally delaminated strip. The problem is formulated analytically, and the thermal effects are characterized by "thermal forces". It is shown that a temperature load increases the energy release rate and may aggravate delamination, post-buckling.

The details of the present problem are discussed in the following section.

1.3 Current Objective

We wish to model thermal buckling behavior as well as edge debonding behavior of a patched structure. The patch (top layer) is partially adhered to the baseplate (bottom layer); hence there is a portion of the plates that is not bonded. The aspect ratio of the

two isotropic constituent layers is such that they can be modeled using a thin structure theory. A geometrically nonlinear strain-displacement relationship is used.

The mathematical foundation of the problem is established by Bottega and Carabetta (2009), who approach it as a moving boundaries problem in the calculus of variations, as performed similarly by Bottega (1995). The formulation begins with the construction of a potential energy functional. The problem is restated in a mixed formulation (in terms of transverse deflection and membrane force) to facilitate the analysis as well as the interpretation. Applying the Theorem of Stationary Potential Energy, it is the vanishing of the first variation of the potential energy functional which provides the governing equations, constitutive relations, boundary and matching conditions, as well as *transversality equations*. After extracting this information, and upon developing a compatibility condition, the problem is fully posed.

The structures of interest are patched assemblies which are fixed at the edges to prevent in-plane translations. Two versions are considered, the first being fixed structures which are hinged at the edges to allow for rotation, and the second being fixed structures which are clamped at the edges to prevent rotation. Multiple equilibrium configurations at a common load value will exist for such structures, and hence we must determine the most favorable. This is done by evaluation of the potential energy as well as the stability of each configuration. We adopt the approach of Karlsson and Bottega (2000), who assess stability of an equilibrium position by checking the second variation of the potential energy functional for positive-definiteness. Through analysis of the load-deflection behavior along with consideration of stability, we can thoroughly diagnose thermal buckling behavior.

To predict the onset of edge debonding, we employ a growth law established by Bottega (1983), who develops a general growth law for propagation using variational principles. Delamination growth is governed by a Griffith (1920) criterion, (and is restricted to monotonic growth). The energy required to generate a unit of disbond (the bond strength) is a material property of the bonding agent. Growth occurs if the sum of the jump in strain energy density of the structure at any given point (on the load-deflection path) is greater than twice the bond strength (twice, because there are two surfaces created upon debonding). The equations describing this condition mathematically are the transversality equations arising from the vanishing of the first variation of the potential energy. These emerge because we allow the bond zone boundary to vary, along with the deflections.

Numerical simulations based on the aforementioned mathematical preliminaries are performed. From the data generated we observe and interpret the load-deflection behavior of the structure. Critical buckling parameters are also calculated. We can thus comment on the salient behavior and unique buckling phenomena observed. Finally, we examine the energy release rates of the structure and compare with the observed buckling behavior, to make inferences on the combined effect.

1.4 Outline of the Dissertation

The dissertation is presented in seven chapters. Chapter 2 will detail the variational formulation of the mathematical model of the problem. Chapter 3 will describe the analysis of the mathematical model established in Chapter 2. In Chapters 4 and 5 we explore the thermal buckling behavior for fixed partially debonded structures with two different edge support conditions, hinged and clamped, respectively. The

coupled phenomena of thermal buckling and edge debonding will be discussed in Chapter 6. Finally, Chapter 7 offers a thorough discussion on the findings of this work.

Chapter 2

Formulation

2.1 Introduction

We formulate the problem for the combined effects of debonding and thermal buckling. The plates are subject to a uniform temperature difference, and two different edge support conditions will be examined. The analytical model to be employed is that of Bottega and Carabetta (2009), who formulate the problem from first principles, using a thin structure theory for each constituent element. They approach the problem as a moving boundaries problem in the calculus of variations. Taking the appropriate variations of the potential energy functional will yield all relevant governing equations and conditions. In this chapter we present a replication of that formulation. We then discuss the normalization and formulate a potential energy functional. The remainder of the chapter focuses on that which emanates from the first and second variations of the potential energy functional.

2.2 Geometry

We begin by establishing the pertinent geometry. The overall full span of the beam-plates is shown in Figure 2a-b, for either edge support condition. Both edges of the structure are fixed with respect to in-plane translation and either hinged or clamped with respect to rotation.

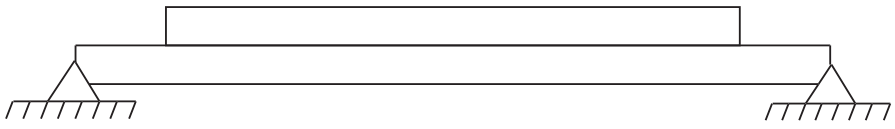

Figure 2a. Full span depiction of patched structure showing hinged-fixed edges

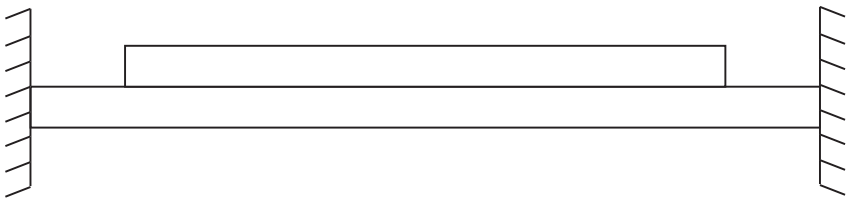

Figure 2b. Full span depiction of patched structure showing clamped-fixed edges

Figure 3 depicts half of the span of the patched beam-plates. It shows normalized parameters, where the details of the normalization can be found in Section 2.3.1.

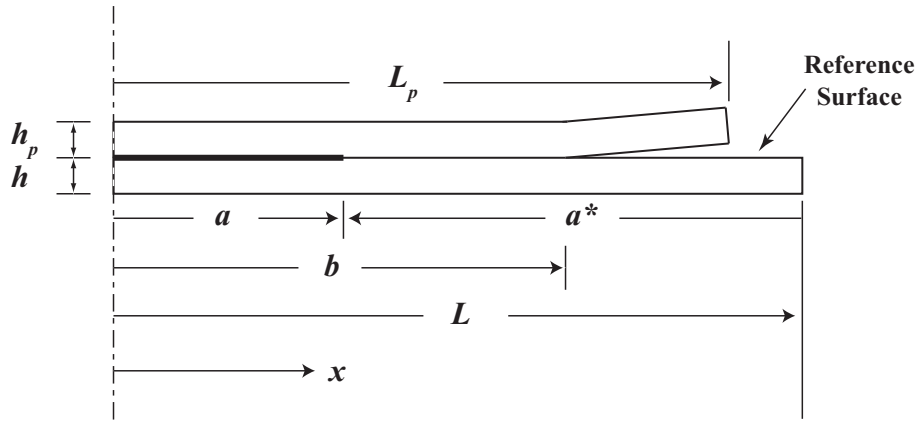

Figure 3. Half-span of the patched structure

The coordinate x (and hence in-plane displacement $u(x)$) is measured positive outward from the center of the structure. Positive transverse deflection, $w(x)$, is taken downward. Four domains will be defined: $S_1 : 0 \le x \le a$ is the "bond zone," where the patch and baseplate are perfectly intact; $S_2 : a \le x \le b$ is the "contact zone," where the plates are no longer adhered, yet remain in sliding contact (the mating surfaces are taken as frictionless); and $S_3 : b \le x \le L$ is the "region of separation," where the plates are no longer in contact. For completeness, the domain of the lifted portion of the patch, the "lift zone", is denoted as $S_{p3} : b \le x \le L_p$. It is seen from Figure 3 that a represents the length of the bonded region, and we define the *conjugate bond length* $a^* = L - a$. In

addition, the parameter *b* defines the boundary between the contact zone and lift zone, while L_p defines the length of the patch. (Throughout the formulation, all quantities containing a subscript "*p*" will refer to the patch.) The transverse thicknesses of the plates are designated as $h << 1$ and $h_p << 1$ for the baseplate and patch, respectively. Curvature is taken as $\kappa = w''$, where primes represent total derivatives with respect to *x*. We employ a nonlinear strain-displacement relationship, $e = u' + \frac{w'^2}{2}$ to evaluate relative elongation. The reference surface is taken as the mating surface of the baseplate and the patch. The strains and corresponding axial displacements evaluated at the reference surface (denoted by stars) are as follows

$$e_i^* = e_i + \tfrac{1}{2} h \kappa_i' \quad \text{and} \quad e_{pi}^* = e_{pi} - \tfrac{1}{2} h_p \kappa_{pi}' \tag{2.1}$$

$$u_i^* = u_i + \tfrac{1}{2} h w_i' \quad \text{and} \quad u_{pi}^* = u_{pi} - \tfrac{1}{2} h_p w_{pi}' \tag{2.2}$$

where the index *i* has the value of 1, 2, or 3, depending on the appropriate domain. We now proceed to a discussion of the relevant parameters.

2.3 Parameters

2.3.1 Normalization

The normalization of parameters is consistent with that established by Bottega (1995). Length scales are normalized with respect to the dimensional half-span length, $\bar{L}$, of the baseplate. Quantities marked with an over-bar represent dimensional quantities. Hence,

$$L = \frac{\bar{L}}{\bar{L}} = 1, \; x = \frac{\bar{x}}{\bar{L}}, \; u = \frac{\bar{u}}{\bar{L}}, \; w = \frac{\bar{w}}{\bar{L}} \tag{2.3}$$

describe the nondimensional half-span of the baseplate, in-plane coordinate, in-plane displacement, and transverse displacement, respectively. Further, the normalized thicknesses of the baseplate and patch are, respectively,

$$h = \frac{\bar{h}}{\bar{L}},\ h_p = \frac{\bar{h}_p}{\bar{L}} \tag{2.4}$$

In addition, we define the useful thickness ratio

$$h_0 = \frac{h_p}{h} \tag{2.5}$$

We also define a modulus ratio, which is the ratio of the elastic properties of the patch to that of the baseplate. For plane stress and plane strain, respectively,

$$E_0 = \frac{\bar{E}_p}{\bar{E}},\ E_0 = \frac{\bar{E}_p / \left(1-\nu_p{}^2\right)}{\bar{E} / \left(1-\nu^2\right)} \tag{2.6}$$

The parameters $\bar{E}$ and $\bar{E}_p$, are the Young's Moduli for the baseplate and patch, respectively. Also, ν and ν_p are the Poisson's ratios for the baseplate and patch, respectively. The preceding parameters, as well as the forthcoming results, correspond to either a state of plane stress or plane strain, respectively, depending on the interpretation of the modulus ratio. The bending stiffnesses (D and D_p) and membrane stiffness (C and C_p) are normalized with respect to the dimensional bending stiffness of the baseplate. Hence,

$$D = 1,\ C = \frac{12}{h^2},\ D_p = E_0 h_0^3,\ C_p = C E_0 h_0 \tag{2.7}$$

Finally, the normalized membrane force and bond strength are, respectively,

$$N = \frac{\bar{N}\bar{L}^2}{\bar{D}},\ \gamma = \frac{\bar{\gamma}\bar{L}^2}{\bar{D}} \tag{2.8}$$

2.3.2 Temperature Scale

The temperature scale used will be that of Rutgerson and Bottega (2002), who define the non-dimensional temperature change

$$\tilde{\Theta} = \alpha\Theta = \alpha \frac{\bar{\Theta} - \bar{\Theta}_0}{\bar{\Theta}_0} \tag{2.9}$$

where $\bar{\Theta}$ represents the current dimensional temperature of the system, and $\bar{\Theta}_0$ represents a reference temperature at which the system undergoes no thermal strains or deflections. The parameters α and α_p are defined as non-dimensional coefficients of thermal expansion of the baseplate and patch, such that

$$\begin{aligned} &\alpha = \bar{\alpha}\bar{\Theta} \quad \text{and} \quad \alpha_p = \bar{\alpha}_p\bar{\Theta} \quad \text{(plane stress)}, \\ &\alpha = \bar{\alpha}\bar{\Theta}(1+\nu) \text{ and } \alpha_p = \bar{\alpha}_p\bar{\Theta}(1+\nu) \text{ (plane strain).} \end{aligned} \tag{2.10}$$

It is convenient to define the ratio

$$\alpha_p^0 = \frac{\alpha_p}{\alpha} \tag{2.11}$$

2.4 Potential Energy Functional

The formulation begins with construction of a potential energy functional

$$\Pi = \sum_{i=1}^{3} \left\{ U_B^{(i)} + U_{Bp}^{(i)} + U_M^{(i)} + U_{Mp}^{(i)} + U_T^{(i)} + U_{Tp}^{(i)} \right\} - \Lambda + \Gamma \tag{2.12}$$

The energy functional contains the bending (*B*) and membrane (*M*), as well as the thermal energies (*T*) of the patch and the baseplate, to account for the total free energy of the system. The bending energies in each domain are, for the present model,

$$U_B^{(i)} = \int_{S_i} \tfrac{1}{2} D\kappa_i^2 dx, \; U_{Bp}^{(i)} = \int_{S_i} \tfrac{1}{2} D_p \kappa_{pi}^2 dx \quad (i = 1-3) \tag{2.13}$$

while the membrane energies are

$$U_M^{(i)} = \int_{S_i} \tfrac{1}{2} C \left(e_i - \alpha \Theta \right)^2 dx,\ U_{Mp}^{(i)} = \int_{S_i} \tfrac{1}{2} C_p \left(e_{pi} - \alpha_p \Theta \right)^2 dx \quad (i = 1-3) \qquad (2.14)$$

The associated thermal energies are

$$U_T^{(i)} = \int_{S_i} \left[c_\sigma - (1+\Theta) c_e \right] \Theta \, dx,\ U_{Tp}^{(i)} = \int_{S_i} \left[c_{\sigma p} - (1+\Theta) c_{ep} \right] \Theta \, dx \quad (i = 1-3) \qquad (2.15)$$

The parameters c_σ, c_e, $c_{\sigma p}$, and c_{ep} appearing in Eq. (2.15) are the normalized specific heats for the baseplate and the patch under constant stress and constant strain. The variation of the thermal terms given by Eq. (2.15) will vanish identically since the thermal load is to be prescribed. The energy functional of Eq. (2.12) also contains the constraint functional

$$\Lambda = \sum_{i=1}^{2} \int_{S_i} \sigma_i \left(w_{pi} - w_i \right) dx + \int_{S_i} \tau \left(u^*_{p1} - u^*_1 \right) dx \qquad (2.16)$$

The parameters $\sigma_i \left\{ i = 1, 2 \ (\sigma_2 < 0) \right\}$ and τ are Lagrange multipliers which ultimately represent normal and shear stresses. From this functional it can be seen that the transverse displacements of the baseplate and patch are matched in the bond zone and contact zone. Axial displacements (evaluated at the reference surface) of the baseplate and patch are matched in the bond zone only. Finally, the potential energy functional includes the debonding energy given by the functional

$$\Gamma = 2\gamma (a^* - a^*_0) \qquad (2.17)$$

The term γ is the normalized bond energy (bond strength), which is a property of the common interface. The term a^*_0 represents some initial value of the conjugate bond size a^*. This represents the energy required to reduce the bond zone length by one unit.

By invoking the Principle of Stationary Potential Energy, for equilibrium it is required that the first variation of the potential energy functional vanish, or $\delta \Pi = 0$. The

bond zone and contact zone boundaries, *a* and *b*, respectively, are allowed to vary in addition to the displacements. Taking appropriate variations, and eliminating the Lagrange multipliers yields the governing differential equations and constitutive relations, as well as the boundary and matching conditions. It also yields what are referred to as *transversality conditions*, from which the locations of the moving bond zone and contact zone boundaries are found for equilibrium configurations of the evolving structure. We thus generate a self-consistent set of equations.

2.4.1 Governing Differential Equations/Constitutive Relations

The problem will be posed in a mixed formulation; that is, in terms of the transverse deflection and the membrane force (rather than in-plane deflection). This renders the resulting equations physically recognizable and analytically solvable. The following relationships are established via the constraint functional Eq. (2.16)

$$w^*_i(x) \equiv w_i(x) = w_{pi}(x) \qquad (x \in S_i; i = 1,2) \tag{2.18}$$

$$\kappa^*_i(x) \equiv \kappa_i(x) = \kappa_{pi}(x) \qquad (x \in S_i; i = 1,2) \tag{2.19}$$

$$u^*_1(x) = u^*_{p1}(x) \qquad (x \in S_1) \tag{2.20}$$

Recall that the starred terms refer to quantities evaluated at the reference surface. The governing differential equations resulting from the vanishing of the first variation of the potential energy functional are

$$M^{*\prime\prime}_i - \left(N^*_i\, w^{*\prime}_i\right)' = 0, \;\; N^{*\prime}_i = 0 \quad (x \in S_i; i = 1,2) \tag{2.21}$$

$$M_3'' - \left(N_3 w_3'\right)' = 0, \;\; N_3' = 0 \quad (x \in S_3) \tag{2.22}$$

$$M_{p3}'' = 0, \;\; N_{p3}' = 0 \quad (x \in S_{p3}) \tag{2.23}$$

Equation (2.21) arises from adding the governing equations (obtained from the vanishing of the first variation) for the constituent layers (the patch and baseplate) together for both regions S_1 and S_2. The resultant membrane forces in the baseplate and patch are, respectively,

$$N_i(x) = C\left[e_i(x) - \alpha\Theta\right],\ \ N_{pi}(x) = C_p\left[e_{pi}(x) - \alpha_p\Theta\right] \qquad (i = 1-3) \tag{2.24}$$

In the bond zone, the resultant normalized membrane force is found by adding together those of the constituent layers and incorporating the equalities described in Eqs. (2.18) - (2.20),

$$\begin{aligned} N^*_1(x) &= N_1 + N_{p1} \\ &= C^* e^*_1(x) + B^* \kappa^*_1(x) - n^*\tilde{\Theta} \\ &= C^*\left[e^*_1(x) - \alpha^*\tilde{\Theta}\right] + B^*\left[\kappa^*_1(x) - \beta^*\tilde{\Theta}\right] \end{aligned} \tag{2.25}$$

We define the following stiffness parameters of the composite structure, expressed in terms of the stiffnesses of the constituent layers:

$$B^* = \frac{h_p}{2}C_p - \frac{h}{2}C,\ \ C^* = C + C_p \tag{2.26}$$

The parameter B^* is a bending/stretching coupling stiffness, while the parameter C^* is the membrane stiffness. The parameters

$$\alpha^* = \alpha_1 - \rho^*\beta^*,\ \ \beta^* = m^*/D^* \tag{2.27}$$

are equivalent to the thermal expansion coefficients of the intact portion of the composite structure and represent the thermally induced membrane strain at the reference surface and the associated curvature change, respectively, per unit normalized temperature change for a free unloaded structure. They are also established in terms of the properties of the constituent layers, as

$$\alpha_1 = n^*/C^*,\ \ m^* = \mu^* - \rho^* n^*,\ \ \mu^* = \frac{h_p}{2} C_p \alpha_p^0 - \frac{h}{2} C,\ \ n^* = C_p \alpha_p^0 + C \tag{2.28}$$

The thermal expansion coefficient α_1 represents the corresponding strain per unit temperature at the centroid of the intact segment of an unloaded composite structure. The quantities *m** and *n** represent the total bending moment per unit temperature change and the membrane force per unit temperature change, respectively. The parameter μ^* is the induced thermal moment (about the reference surface) in the composite structure per unit of temperature change. It can be shown that

$$\rho^* = B^*/C^* \tag{2.29}$$

The normalized bending moment in the bond zone (acting on a cross section of the composite structure) is found by adding the bending moments of the individual layers, and once again incorporating the relationships described by Eqs. (2.18) - (2.20), which results in

$$\begin{aligned} M^*_1(x) &= A^* \kappa^*_1(x) + B^* e^*_1(x) - \mu^* \tilde{\Theta} \\ &= A^* \left[\kappa^*_1(x) - \beta^* \tilde{\Theta} \right] + B^* \left[e^*_1(x) - \alpha^* \tilde{\Theta} \right] \\ &= D^* \left[\kappa^*_1(x) - \beta^* \tilde{\Theta} \right] + \rho^* N^*_1 \end{aligned} \tag{2.30}$$

Here we see that the quantity ρ^* is seen to locate the centroid of the composite structure with respect to the reference surface (which is taken as the mating surface of the plates). The parameters

$$A^* = D + D_p + \left(\frac{h}{2} \right)^2 C + \left(\frac{h_p}{2} \right)^2 C_p,\ \ D^* = A^* - \rho^* B^* \tag{2.31}$$

Represent bending stiffnesses and are expressed in terms of their counterparts from the primitive structures, the patch and baseplate. In the contact zone, the resultant membrane force is

$$N^*_2(x) = N_2 + N_{p2} \tag{2.32}$$

and the bending moment is

$$M^*_2(x) = D_c \kappa^*_2(x) + \left[\frac{h_p}{2} N_{p2} - \frac{h}{2} N_2 \right] \tag{2.33}$$

The bending stiffness of the debonded segment of the composite structure in the contact zone is denoted found to be

$$D_c = D + D_p \tag{2.34}$$

Finally in the region of separation and the lift zone (the sub-regions S_3 and S_{p3}), the bending moments are, respectively

$$M_3(x) = D\kappa_3(x) - \frac{h}{2} N_3 \tag{2.35}$$

and

$$M_{p3}(x) = D_p \kappa_{p3}(x) + \frac{h_p}{2} N_{p3} \tag{2.36}$$

With the governing equations established, we proceed to the boundary and matching conditions. These conditions also emanate from the vanishing of the first variation of the potential energy functional, as well as physical considerations.

2.4.2 Boundary and Matching Conditions

The following is a list of the boundary and matching conditions for each domain of the structure

$$\begin{aligned} &u^*_1(0) = 0,\ \ w^{*\prime}_1(0) = 0,\ \left[M^{*\prime}_1 - N^*_1 w^{*\prime}_1 \right]_{x=0} = 0, \text{ (symmetric)} \\ &u^*_0(0) \equiv u^*_1(0) + \rho^* w^{*\prime}_1(0) = 0,\ \ w^*_1(0) = 0,\ \ \kappa^*_1(0) = 0, \text{ (anti-symmetric)} \end{aligned} \tag{2.37}$$

$$u^*_1(a) = u^*_2(a) = u^*_{p2}(a),\ \ N^*_1(a) = N^*_2(a) \tag{2.38}$$

$$w^*_1(a) = w^*_2(a), \;\; w^{*\prime}_1(a) = w^{*\prime}_2(a) \tag{2.39}$$

$$M^*_1(a) = M^*_2(a), \;\; \left[M^{*\prime}_1 - N^*_1 w^{*\prime}_1 \right]_{x=a} = \left[M^{*\prime}_2 - N^*_2 w^{*\prime}_2 \right]_{x=a} \tag{2.40}$$

$$u^*_2(b) = u^*_3(b), \;\; N_2(b) = N_3(b) \tag{2.41}$$

$$u^*_{p2}(b) = u^*_{p3}(b), \;\; N_{p2}(b) = N_{p3}(b) \tag{2.42}$$

$$w^*_2(b) = w_3(b) = w_{p3}(b), \;\; w^{*\prime}_2(b) = w_3'(b) = w_{p3}'(b) \tag{2.43}$$

$$\begin{aligned} &M^*_2(b) = M_3(b) + M_{p3}(b), \\ &\left[M^{*\prime}_2 - N^*_2 w^{*\prime}_2 \right]_{x=b} = \left[M_3' - N_3 w_3' \right]_{x=b} + \left[M_{p3}' - N_{p3} w_{p3}' \right]_{x=b} \end{aligned} \tag{2.44}$$

$$N_{p3}(L_p) = \kappa_{p3}(L_p) = \left[M_{p3}' - N_{p3} w_{p3}' \right]_{x=L_p} = 0 \tag{2.45}$$

$$u_3(L) = 0 \quad \text{or} \quad N_3(L) = 0 \tag{2.46}$$

$$w_3(L) = 0, \quad \text{and} \quad w_3'(L) = 0 \quad \text{or} \quad \kappa_3(L) = 0 \tag{2.47}$$

Upon integrating the governing differential equations, we impose these conditions. However, we choose to recast the problem statement in terms of membrane force in order to simplify our efforts.

2.4.3 Unknown Membrane Force

The segment of the debonded portion of the patch in the lift zone is unloaded, and it follows from Eqs. (2.23) and (2.45) that

$$N_{p3}(x) = \kappa_{p3}(x) = M_{p3}'(x) = 0 \tag{2.48}$$

By integrating the differential equations for membrane force in each domain, Eqs. (2.21) and (2.22), and applying the appropriate boundary/matching conditions it is seen that

$$N^*_1 = N_2 = N_3 = -N_0 = \text{ constant, and } N_{p2} = 0 \tag{2.49}$$

The membrane force N_0 will be defined as positive compressive since we are considering structures whose edges are fixed with respect to in-plane translation. The problem is now stated in terms of the unknown transverse deflection *w* and the unknown membrane force, N_0.

2.4.4 Transverse Loading Parameter

Examination of the matching condition (Eq. (2.40)) which relates the resultant bending moments of Regions S_1 and S_2 at the bond zone boundary *a*, $M^*_1(a) = M^*_2(a)$, reveals that this is where the prescribed temperature difference enters the formulation. Using Eqs. (2.30) and (2.33) this condition can be expanded to the form

$$\left(D^* \kappa^*_1 - m^* \tilde{\Theta} - \rho^* N_0 \right)\Big|_{x=a} = \left(D_c \kappa^*_2 + \frac{h}{2} N_0 \right)\Big|_{x=a} \tag{2.50}$$

Equation (2.50) may be rewritten as

$$\left(D^* \kappa^*_1 - D_c \kappa^*_2 \right)\Big|_{x=a} = M_\lambda \tag{2.51}$$

where

$$M_\lambda \equiv m^* \tilde{\Theta} + \left(\rho^* + \frac{h}{2} \right) N_0 \tag{2.52}$$

is identified as the *transverse loading parameter* for the beam-plate assembly. It may be interpreted as a moment applied at *a* that results from a thermal property mismatch in the constituent beam-plates and a discontinuity in the centroidal surface at the bond zone boundary. It has been shown by Karlsson and Bottega (2000) that this parameter is crucial to the understanding of the sling-shot buckling phenomenon. The first and second terms of the right-hand side of Eq. (2.52) are seen to correspond to a thermally induced moment and a mechanical moment, respectively. It is observed that, depending on the

sign of the thermal moment term $m^*\tilde{\Theta}$, these effects are either competing with one another or reinforcing each other.

2.4.5 Transversality Condition

We arrive at the *transversality conditions* via the variational principle of Section 2.4, under the assumption that the propagating bond zone and contact zone boundaries (*a* and *b*, respectively) may arbitrarily vary, as well as the displacements. The equations denote equilibrium positions associated with those moving boundaries. When contact occurs (i.e. Region S_2 is present), the transversality condition associated with the moving bond zone boundary is

$$G\{a\} = \frac{1}{2}\left[D_c \kappa^{*2}_2 - D^* \kappa^{*2}_1 + \frac{N_0^2}{C_e} - 2N_0\left(1-\alpha_1\right)\tilde{\Theta} + \frac{\eta\tilde{\Theta}^2}{2} \right]_{x=a} = 2\gamma, \ (b \geq a^+) \quad (2.53)$$

When the unbonded patch flaps lift off the baseplate and contact does not occur, the transversality condition for $x = a$ takes the form

$$G\{a\} = \frac{1}{2}\left[D\kappa_3^2 - D^* \kappa^{*2}_1 + \frac{N_0^2}{C_e} - 2N_0\left(1-\alpha_1\right)\tilde{\Theta} + \frac{\eta\tilde{\Theta}^2}{2} \right]_{x=a} = 2\gamma, \ (b = a) \quad (2.54)$$

where

$$\frac{1}{C_e} \equiv \frac{C_p / C}{C^*} \quad (2.55)$$

and

$$\eta \equiv C + \left(\alpha_p^0\right)^2 C_p - \alpha_1^{\ 2} C^* \quad (2.56)$$

Using a Griffith-type delamination criterion, $G\{a\}$ is identified as the *energy release rate*. Recall that the parameter γ is the bond strength for the patch-baseplate interface (a material property). We thus have the following delamination growth criterion: *if* $G\{a_0\} <$

2γ, where a_0 is the initial value of the bond zone boundary a, then delamination will not occur. That is, the length of the bond zone will remain unchanged. If $G\{a_0\} > 2\gamma$, then delamination will propagate, and continue to do so until $G\{a\} = 2\gamma$.

At the contact zone-region of separation boundary, *b*, the transversality condition reduces to a statement of continuity of curvature

$$\kappa^*_2(b) = \kappa_3(b) = \kappa_{p3}(b) \tag{2.57}$$

This concludes what we can extract from the vanishing of the first variation of the potential energy functional (Eq. (2.12)).

2.4.6 Integrability Condition

The use of a mixed formulation offers mathematically convenient governing equations describing the system, yet omits related information regarding in-plane deflections. We must enforce continuity among the in-plane deflections in each domain and the membrane force N_0. The *integrability condition* to be presented is obtained by integrating the strain-displacement relationships and imposing the appropriate boundary and matching conditions on the in-plane displacements along each domain of the structure. Doing this results in the (compatibility) condition

$$\Delta_L \equiv u_3(1) = \frac{N_0}{C_a} + \left(1 - a + a\alpha_1\right)\tilde{\Theta} - rw'(a) - \sum_{i=1}^{3}\int_{Si}\frac{1}{2}w_i'^2 dx \tag{2.58}$$

where

$$\frac{1}{C_a} = \left(\frac{a^*}{C} + \frac{a}{C^*}\right) \tag{2.59}$$

and

$$r = \frac{h}{2} + \rho^* \tag{2.60}$$

Equation (2.58) offers the relationship between the in-plane edge displacement and the applied load. If the edges of the structure are free, we may readily solve for the in-plane edge displacement. Should the edges be fixed, as considered in this work, the in-plane edge displacement is zero and we thus have a relationship between the prescribed thermal load and the resulting membrane force.

2.4.7 Stability

Up to this point we have formulated the mathematical problem for the displacement of the patched structure under a prescribed temperature difference. Thermally loaded structures for which the edges are fixed with respect to in-plane motion tend to possess multiple equilibrium configurations. Hence, it must be established which position the system "prefers," i.e. which of the multiple equilibrium configurations is energetically favorable, as well as which are *stable*. A stable configuration is taken as one for which the system stays in the vicinity of, or returns to, upon experiencing a perturbation in transverse deflection and/or membrane strain. This is determined by examination of the second variation of the potential energy functional. For stability, in the aforementioned context, it is required here that the second variation be positive-definite. In the absence of disbond growth, upon perturbing w and u (and hence e), the second variation assumes the canonical quadratic form (Greenberg)

$$\delta^2\Pi = \zeta\left(\delta N_0\right)^2 + \frac{\widehat{F}}{N_0}\left(\delta M_\lambda\right)^2 \tag{2.61}$$

where, for all cases presented here

$$\zeta = \frac{1}{2}\left(\frac{a}{C^*} + \frac{a^*}{C}\right) \tag{2.62}$$

In matrix form:

$$\delta^2\Pi = \begin{bmatrix} \delta N_0 & \delta M_\lambda \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & \dfrac{\widehat{F}}{N_0} \end{bmatrix} \begin{bmatrix} \delta N_0 \\ \delta M_\lambda \end{bmatrix} \tag{2.63}$$

According to the Sylvester criterion, the quadratic form is positive-definite if all of the leading principal minors of the square matrix are positive (Gelfand and Fomin). Thus,

$$\zeta > 0, \begin{vmatrix} \zeta & 0 \\ 0 & \dfrac{\widehat{F}}{N_0} \end{vmatrix} > 0 \tag{2.64}$$

For this reason, we require that ζ and $\widehat{F}/N_0$ are positive. It is apparent that ζ is always positive. The solution is consistent with N_0 being positive as well. Thus, to satisfy positive definiteness, we require that $\widehat{F} > 0$. Hence, *an equilibrium position is said to be stable if* $\widehat{F} > 0$. We will refer to $\widehat{F}$ as the *stability function*.

The problem is now fully posed and we may proceed to the analysis, which is detailed in Chapter 3.

Chapter 3

Analysis

3.1 Introduction

The complete mathematical problem has been formulated in Chapter 2. In this chapter we present a nonlinear analysis. The problem is geometrically nonlinear, via the strain, which is manifested in the unknown membrane force in the governing differential equations. It is by retaining the geometric nonlinearities that we are able to capture the salient buckling behavior. It has also been shown by Carabetta and Bottega (2009) that geometric nonlinearities should not be discarded when discussing delamination behavior, even when buckling is not an issue. In the sections that follow we discuss reduction of the governing equations, critical parameters, the specific solutions for deflection, and the details of the numerical analyses to be performed.

3.2 General Solution

The problem has been recast in terms of the membrane force along with the transverse deflection. In Section 2.4.3 it was deduced that the membrane force in the unloaded portion of the patch is zero while the membrane force in the composite portion as well as in the baseplate, is an unknown constant. Applying Eq. (2.49) along with the resultant bending moment expressions (Eqs. (2.30), (2.33), (2.35), and (2.36)), the governing differential equations derived in Chapter 2, Eqs. (2.21) - (2.23) may now be reduced to the forms

$$
\begin{aligned}
&D^{*}w^{*}_{1}{}^{iv} + N_0 w^{*}_{1}{}'' = 0 \\
&D_c w^{*}_{2}{}^{iv} + N_0 w^{*}_{2}{}'' = 0 \\
&D w_3{}^{iv} + N_0 w_3{}'' = 0 \\
&D_p w_{p3}{}^{iv} = 0
\end{aligned}
\tag{3.1}
$$

where N_0 is the unknown membrane force. Further, Eq. (3.1) may be rewritten as

$$
\begin{aligned}
&w^{*}_{1}{}^{iv} + k_1{}^2 w^{*}_{1}{}'' = 0 \\
&w^{*}_{2}{}^{iv} + k_2{}^2 w^{*}_{2}{}'' = 0 \\
&w_3{}^{iv} + k_3{}^2 w_3{}'' = 0 \\
&w_{p3}{}^{iv} = 0
\end{aligned}
\tag{3.2}
$$

where

$$
k_1 = \sqrt{\frac{N_0}{D^{*}}},\ k_2 = \sqrt{\frac{N_0}{D_c}},\ k_3 = \sqrt{\frac{N_0}{D}} \tag{3.3}
$$

Integrating Eqs. (3.2) gives the general analytical solution

$$
\begin{aligned}
&w_i = A_i \cos(k_i x) + B_i \sin(k_i x) + C_i x + D_i,\ \ (i = 1-3) \\
&w_{p3} = \frac{A_{p3} x^3}{6} + \frac{B_{p3} x^2}{2} + C_{p3} x + D_{p3}
\end{aligned}
\tag{3.4}
$$

where $A_{(p)i}$, $B_{(p)i}$, $C_{(p)i}$, and $D_{(p)i}$ are constants of integration, to be found from the boundary and matching conditions described in Section 2.4.2. We then find the membrane force N_0 as a function of prescribed temperature difference Θ via the integrability condition Eq. (2.58). These analytical solutions allow for identifying critical parameters of the system, as described in the next section.

3.3 Critical Buckling Parameters

It will be seen, that for *symmetric deformation*, the closed-form analytical solutions for the deflections in Regions S_1 – S_3 are each of the general form

$$w(x) \sim \frac{M_\lambda}{N_0 H(N_0)} \Psi(x; N_0) \tag{3.5}$$

where M_λ is the transverse loading parameter described in Section 2.4.4, Eq. (2.52), and $\Psi(x; N_0)$ is some trigonometric function. The specific form of the trigonometric function $H(N_0)$ depends on the edge support conditions of the structure. When $H(N_0)$ vanishes, the deflection is singular, signifying critical behavior. The roots of this function are seen to correspond to the critical membrane forces for bifurcation buckling. The deflection is in an undefined state when the numerator and denominator vanish simultaneously, which also signifies a critical condition. That is, when M_λ and $H(N_0)$ both vanish. The vanishing of $H(N_0)$ identifies the critical membrane forces, the first (lowest) of which is then related to the critical thermal load via the vanishing of the loading parameter, such that

$$\tilde{\Theta}_{cr} = -\frac{\left(\rho^* + \frac{h}{2}\right)}{m^*} N_{cr} \tag{3.6}$$

This is the critical sling-shot buckling temperature difference found by Karlsson and Bottega (2000). In the following section we will give the explicit forms of the deflections generally described by Eq. (3.5).

3.4 Solutions

We obtain the specific analytical solutions for the deflections of each domain of the structure by applying the boundary and matching conditions of Section 2.4.2 to Eqs. (3.4) and solving the resulting algebraic systems. Two extreme end support conditions have been examined. The first is when the edges of the baseplate are hinged with respect to rotation and the second is when the edges are clamped with respect to rotation. In both

cases the edges are fixed with respect to in-plane translation. Mathematically, each case differs by one boundary condition, Eq. (2.47).

In this work we consider a patched structure which possesses partial debonding at the edges. Karlsson and Bottega (2000) performed a thermal buckling analysis for a perfectly intact patched structure, i.e. the limiting case for the current structure of interest, when $L_p = b = a$. Those results will be repeated here, since it can be shown that they equivalently represent the case when the unbonded patch flaps fully lift off the baseplate. These are presented alongside original results allowing for the presence of a contact zone and lift zone. It can be shown that the solution including those domains reduces to the solution for the intact structure in the limiting case. This is equivalent to the former minus Regions S_2 and S_{p3}. What follows is a symmetric solution for transverse deflection. In addition, we present the function H and the stability function $\widehat{F}$ for each case considered.

3.4.1 Hinged: Intact Structure

This section includes a reproduction of the results for the case of hinged supports produced by Karlsson and Bottega (2000). The transverse deflections for an intact structure with hinged-fixed edges are, in each domain

$$w^*_{1(in)}(x) = \frac{M_\lambda}{N_0 H_{h(in)}}\left[H_{h(in)} - \sqrt{\frac{D^*}{D}}\cos(k_3 a^*)\cos(k_1 x)\right] \tag{3.7}$$

and

$$w_{3(in)}(x) = -\frac{M_\lambda}{N_0 H_{h(in)}}\sin(k_1 a)\sin(k_3(1-x)) \tag{3.8}$$

where

$$H_{h(in)}(N_0) = \sqrt{\frac{D^*}{D}}\cos(k_1 a)\cos(k_3 a^*) - \sin(k_1 a)\sin(k_3 a^*) \tag{3.9}$$

Recall that the parameters k_1 and k_3 contain the membrane force N_0. The roots of the function $H_{h(in)}$ give the critical membrane forces for bifurcation buckling, hence

$$H_{h(in)}(N_{cr}) = 0 \tag{3.10}$$

The stability function[1] for this case is

$$\widehat{F} = \frac{1}{4}\left(f_A^2 k_1 \sin(2k_1 a) - f_B^2 k_3 \sin(2k_3 a^*)\right) \tag{3.11}$$

where

$$f_A = \frac{-\sqrt{\frac{D^*}{D}}\cos(k_3 a^*)}{H_{h(in)}}, \quad f_B = \frac{\sin(k_1 a)}{H_{h(in)}} \tag{3.12}$$

Recall that for an equilibrium configuration to be considered stable, it is required that $\widehat{F} > 0$. The following section details the specific solution and corresponding equations for the current structure of interest, which is partially debonded.

3.4.2 Hinged: Partially Debonded Structure

Here we present original results for a partially debonded structure which includes a contact zone and a lift zone. The transverse deflection in Region S_1 is thus

$$w^*_1(x) = \frac{M_\lambda}{N_0 H_h}\left[H_h + Z_1 \cos(k_1 x)\right] \tag{3.13}$$

where

$$Z_1 = \sqrt{\frac{D^*}{D_c}}\sin(k_3(1-b))\sin(k_2(b-a)) - \sqrt{\frac{D^*}{D}}\cos(k_3(1-b))\cos(k_2(b-a)) \tag{3.14}$$

The transverse deflections in Regions S_2 and S_3 are found to be, respectively,

[1] It is noted that Eq. (3.11) contains a correction to Eq. (37) from Karlsson and Bottega (2000), which is presented with the second term positive rather than negative for the hinged-fixed case. This correction, however, does not affect the results shown in that work.

$$w_2(x) = -\frac{M_\lambda}{N_0 H_h}\sin(k_1 a)\Big[\sin(k_3(1-b))\cos(k_2(b-x)) + \sqrt{\frac{D_c}{D}}\cos(k_3(1-b))\sin(k_2(b-x))\Big] \tag{3.15}$$

$$w_3(x) = -\frac{M_\lambda}{N_0 H_h}\sin(k_1 a)\sin(k_3(1-x)) \tag{3.16}$$

And finally, the transverse deflection in Region S_{p3} is

$$w_{p3}(x) = \frac{M_\lambda}{N_0 H_h}\sin(k_1 a)\left(Z_2 x + Z_3\right) \tag{3.17}$$

where

$$Z_2 = k_3\cos(k_3(1-b)),\ Z_3 = -\sin(k_3(1-b)) - Z_2 b \tag{3.18}$$

The function H_h appearing in Eqs. (3.13), (3.15) - (3.17) has the form

$$\begin{aligned} H_h(N_0) = &-\sin(k_1 a)\sin(k_3(1-b))\cos(k_2(b-a)) \\ &+\sqrt{\frac{D^*}{D}}\cos(k_1 a)\cos(k_3(1-b))\cos(k_2(b-a)) \\ &-\sqrt{\frac{D_c}{D}}\sin(k_1 a)\cos(k_3(1-b))\sin(k_2(b-a)) \\ &-\sqrt{\frac{D^*}{D_c}}\cos(k_1 a)\sin(k_3(1-b))\sin(k_2(b-a)) \end{aligned} \tag{3.19}$$

The critical membrane forces for bifurcation buckling are found as those satisfying

$$H_h(N_{cr}) = 0 \tag{3.20}$$

The corresponding stability function for the hinged partially debonded structure is

$$\begin{aligned} \widehat{F} = \frac{1}{4}\Big\{ &f_A^2 k_1\sin(2k_1 a) - f_B^2 k_3\sin(2k_3(1-b)) \\ &+ f_C k_2\sin(2k_2(b-a)) + f_D k_2\left[1-\cos(2k_2(b-a))\right]\Big\} \end{aligned} \tag{3.21}$$

where

$$f_A = \frac{Z_1}{H_{hPD}},\ f_B = \frac{\sin(k_1 a)}{H_{hPD}},\ f_C = f_B^2\left[\sin^2(k_3(1-b)) - \frac{D_c}{D}\cos^2(k_3(1-b))\right]$$
$$f_D = 2 f_B^2 \sqrt{\frac{D_c}{D}}\sin(k_3(1-b))\cos(k_3(1-b)) \tag{3.22}$$

The next two sections will parallel the previous two sections, for the case when both edges of the structure are clamped. We begin with the previously established solution for a fully intact structure.

3.4.3 Clamped: Intact Structure

We present a replication of the solution for the intact structure by Karlsson and Bottega (2000). Evaluation of the solution for the case of clamped supports requires use of one different boundary condition at the edge of the baseplate. The transverse deflections are found to be

$$w^*_{1(in)}(x) = \frac{M_\lambda}{N_0 H_{c(in)}}\left[H_{c(in)} - \sqrt{\frac{D^*}{D}}\sin(k_3 a^*)\cos(k_1 x) - \sin(k_1 a)\right] \tag{3.23}$$

and

$$w_{3(in)}(x) = -\frac{M_\lambda}{N_0 H_{c(in)}}\sin(k_1 a)\left[1 - \cos(k_3(1-x))\right] \tag{3.24}$$

where

$$H_{c(in)} = \sqrt{\frac{D^*}{D}}\cos(k_1 a)\sin(k_3 a^*) + \sin(k_1 a)\cos(k_3 a^*) \tag{3.25}$$

Like the hinged-fixed case, the roots of the function $H_{c(in)}$ are the membrane forces corresponding to bifurcation buckling, such that

$$H_{c(in)}(N_{cr}) = 0 \tag{3.26}$$

The stability function for this case is

$$\widehat{F} = \frac{1}{4}\left(f_A^2 k_1 \sin(2k_1 a) + f_B^2 k_3 \sin(2k_3 a^*)\right) \tag{3.27}$$

where

$$f_A = \frac{-\sqrt{\frac{D^*}{D}}\sin(k_3 a^*)}{H_c},\ f_B = \frac{\sin(k_1 a)}{H_c} \tag{3.28}$$

We now proceed to an original solution obtained for the present work for a partially debonded structure with clamped supports.

3.4.4 Clamped: Partially Debonded Structure

Once again, the corresponding analytical solutions for a partially debonded structure allow for the existence of a contact zone and include the lift zone. These results are original to the current work. The transverse deflection in the bond zone is found as

$$w^*_1(x) = \frac{M_\lambda}{N_0 H_c}\left[H_c - Z_1 \cos(k_1 x) - \sin(k_1 a)\right] \tag{3.29}$$

where

$$Z_1 = \sqrt{\frac{D^*}{D_c}}\sin(k_2(b-a))\cos(k_3(1-b)) + \sqrt{\frac{D^*}{D}}\cos(k_2(b-a))\sin(k_3(1-b)) \tag{3.30}$$

The corresponding solutions in the contact zone and region of separation are determined to be, respectively,

$$\begin{aligned} w_2(x) = \frac{M_\lambda}{N_0 H_c}\sin(k_1 a)\Big[&\cos(k_3(1-b))\cos(k_2(b-x)) \\ &-\sqrt{\frac{D_c}{D}}\sin(k_3(1-b))\sin(k_2(b-x)) - 1\Big] \end{aligned} \tag{3.31}$$

$$w_3(x) = -\frac{M_\lambda}{N_0 H_c}\sin(k_1 a)\left[1 - \cos(k_3(1-x))\right] \tag{3.32}$$

In the lift zone, the transverse deflection is found as

$$w_{p3}(x) = \frac{M_\lambda}{N_0 H_c} \sin(k_1 a)\left(Z_2 x + Z_3\right) \tag{3.33}$$

where

$$Z_2 = k_3 \sin(k_3(1-b)), \quad Z_3 = \cos(k_3(1-b)) - (1+Z_2 b) \tag{3.34}$$

We find that, for the case of clamped edges

$$\begin{aligned} H_c(N_0) = {} & \sin(k_1 a)\cos(k_3(1-b))\cos(k_2(b-a)) \\ & + \sqrt{\frac{D^*}{D}}\cos(k_1 a)\sin(k_3(1-b))\cos(k_2(b-a)) \\ & - \sqrt{\frac{D_c}{D}}\sin(k_1 a)\sin(k_3(1-b))\sin(k_2(b-a)) \\ & + \sqrt{\frac{D^*}{D_c}}\cos(k_1 a)\cos(k_3(1-b))\sin(k_2(b-a)) \end{aligned} \tag{3.35}$$

As described previously, the vanishing of H_c yields the critical membrane forces for bifurcation buckling

$$H_c(N_{cr}) = 0 \tag{3.36}$$

The corresponding stability function for the partially debonded structure with clamped supports is found to be

$$\begin{aligned} \widehat{F} = \frac{1}{4}\Big\{ & f_A^2 k_1 \sin(2k_1 a) + f_B^2 k_3 \sin(2k_3(1-b)) \\ & + f_C k_2 \sin(2k_2(b-a)) + f_D k_2 \left[\cos(2k_2(b-a)) - 1\right]\Big\} \end{aligned} \tag{3.37}$$

where

$$\begin{aligned} & f_A = \frac{Z_1}{H_{cPD}}, \; f_B = \frac{\sin(k_1 a)}{H_{cPD}}, \; f_C = f_B^2\left[\cos^2(k_3(1-b)) - \frac{D_c}{D}\sin^2(k_3(1-b))\right] \\ & f_D = 2 f_B^2 \sqrt{\frac{D_c}{D}}\sin(k_3(1-b))\cos(k_3(1-b)) \end{aligned} \tag{3.38}$$

Now, with all relevant analytical solutions established for both edge support conditions, we advance to the basis of the numerical simulations to be performed in later chapters.

3.5 Method

With the exact equations derived, we seek to generate data via a computational analysis, which will reveal characteristic behavior. The numerical portion of the analysis is performed using Matlab. The goal is to describe the behavior of the evolving structure as prescribed thermal loading is applied. There are two phenomena occurring, sling-shot buckling and edge debonding of the two layers. We may calculate the critical values for the membrane force and temperature difference associated with sling-shot buckling. To examine the debonding behavior, we calculate the strain energy release rate of each equilibrium position for the structure. The initial algorithm is run considering *full contact*, where the entire debonded portion of the patch remains in sliding contact. This assumption must be checked during the process, as explained in the following section.

To find the critical buckling values of both the membrane force and prescribed thermal load, possible values of the bond size, a, and the membrane force, N_0, are cycled while the Matlab root finder "fzero" is used on the appropriate H function, as defined by Eqs. (3.9), (3.19), (3.25), or (3.35). For each case there are generally multiple roots, the lowest of which is taken as the first critical buckling membrane force. Once the critical membrane force has been found for each value of a, the critical temperature difference is solved for using Eq. (3.6).

The load-deflection behavior of the structure is paramount in this analysis. To generate the load-deflection paths, we begin by cycling values of the bond size, a, and the prescribed thermal load, $\tilde{\Theta}$. The length of the patch is chosen. Then, depending on the edge supports, the appropriate transverse deflections described in Section 3.2 are substituted into the integrability condition, Eq. (2.58). This provides a relationship

between N_0 and $\tilde{\Theta}$. Since the edges are fixed, the left-hand side of Eq. (2.58) is zero, and hence we may use "fzero" to extract the roots (values of N_0) which satisfy the equation for given values of $\tilde{\Theta}$. This process generates an ordered triple of parameters (a, $\tilde{\Theta}$, N_0) defining an equilibrium configuration. The roots are sorted in ascending order, based on N_0.

Having the parameter triples allows for calculation of all other relevant information for each equilibrium configuration, such as the center span deflection (transverse deflection in Region S_1 at the point $x = 0$) and the total potential energy. To check the stability of an equilibrium configuration, the stability function (Eqs. (3.11), (3.21), (3.27), or (3.37) depending on the conditions) is then tested for positive-definiteness. To assess the debonding behavior of the loaded structure, we calculate the energy release rates, the left-hand sides of Eqs. (2.53) or (2.54), depending on the presence or absence of Region S_2. This brings us to the question of the physical validity of the contact zone's existence, which must be diagnosed.

3.6 Existence of a Contact Zone

Mathematically, we include the presence of a contact zone. However, it is imperative that we examine whether or not the mathematical solution is indeed physically realizable. This is achieved by examining the curvatures in the bond zone and contact zone. Previous work by Bottega (1995) excludes the possibility of the presence of a contact zone for a patched structure with hinged supports. However, that has been determined for purely mechanical loading (upward applied transverse pressure to the bottom surface of the baseplate) and hence does not consider the case of downward deflection which is known to result from thermal loading (applied temperature

difference). According to Bottega and Carabetta (2009), if an inflection point (or pseudo-inflection point) occurs on $a < x < L_p$ such that

$$\kappa_2^*(b) = \kappa_3(b) = 0, \quad \kappa_3(b+) > 0, \ (b < L_p) \tag{3.39}$$

then an intermediate contact zone exists. This means that only a portion of the unbonded patch flap is in contact with the baseplate. Furthermore, if it exists, a slight alteration in the Matlab procedure is necessary. The bond zone size, *a*, is chosen. The contact zone boundary, *b*, is then cycled instead, and the rest of the algorithm proceeds as described in Section 3.5. Only solutions which satisfy Eq. (3.39) are considered valid. If the conditions of Eq. (3.39) are not satisfied, either no contact zone is present $(b = a)$ or a full contact zone is present $(b = L_p)$. The case of absent contact zone, when the patch flaps have fully lifted from the baseplate, is equivalent (in terms of stiffness and energy) to the fully intact case studied by Karlsson and Bottega (2000), for equal bond zone size. The presence of a full contact zone requires that the lift zone, Region S_{p3} does not exist. It is further required that

$$\kappa_2(a^+) < 0 \quad (b = L_p) \tag{3.40}$$

in order for the contact zone to exist. Schematics depicting intermediate contact and full contact are shown in Figure 4a-b.

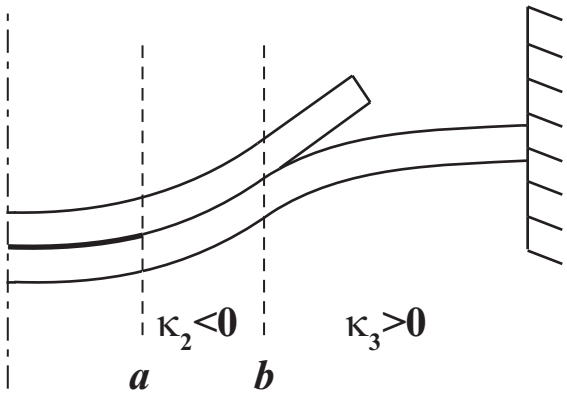

Figure 4a. General depiction of an intermediate contact zone

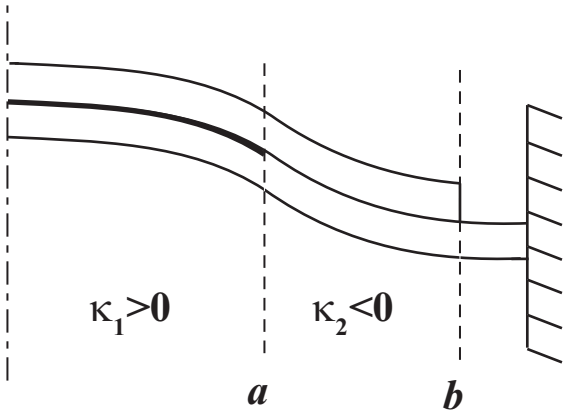

Figure 1b. General depiction of a full contact zone

Results based on the analytical expressions derived in this chapter, in conjunction with the numerical simulations, will be presented in the chapters that follow. We will first interpret and discuss thermal buckling phenomena for the hinged-fixed edge support conditions in Chapter 4. That will be followed by a corresponding discussion of thermal buckling behavior for the clamped-fixed case in Chapter 5. In Chapter 6 the coupled debonding and thermal buckling behavior will be discussed for both edge support conditions. An overall general discussion will be presented in Chapter 7. Numerical codes may be found in the Appendix.

Chapter 4

Thermal Buckling of the Hinged Structure

4.1 Introduction

With the analytical model established, we proceed to evaluating the resulting data generated from the numerical simulation. In this chapter, we will first discuss the thermal buckling behavior of a perfectly intact structure that is hinged with respect to rotation and is fixed with respect to in-plane translation at both of its ends. This case has been characterized by Karlsson and Bottega (2000), and serves as a precursor to the discussion of the thermal buckling behavior for the current structure, which is partially debonded. A partially debonded structure may or may not possess a contact zone, which will be determined. Conclusions are then made about the overall behavior, including a thorough discussion of an unexpected phenomenon.

4.2 Sling-Shot Buckling of an Intact Patched Structure

Karlsson and Bottega (2000) performed a thermal buckling analysis for a patched structure for the limiting case when the entire patch is bonded to the baseplate. From our geometry defined in Section 2.2, this corresponds to the situation when $L_p = b = a$. It is from this analysis that they first observed what is now known as sling-shot buckling. The thermally loaded structure deflects in one direction, and upon reaching a critical temperature difference, dynamically "slings" to an alternate equilibrium configuration in the opposite direction. Here we present a reproduction of the results for hinged edge supports, for the parameter values $\alpha_p^0 = 0.5$, and $a = 0.8$, remaining consistent with the literature. In addition, for the purpose of isolating the thermal effect, the height ratio and

Young's Modulus ratio are taken to be $h_0 = 1$ and $E_0 = 1$, respectively. The results are shown in Figure 5a-d.

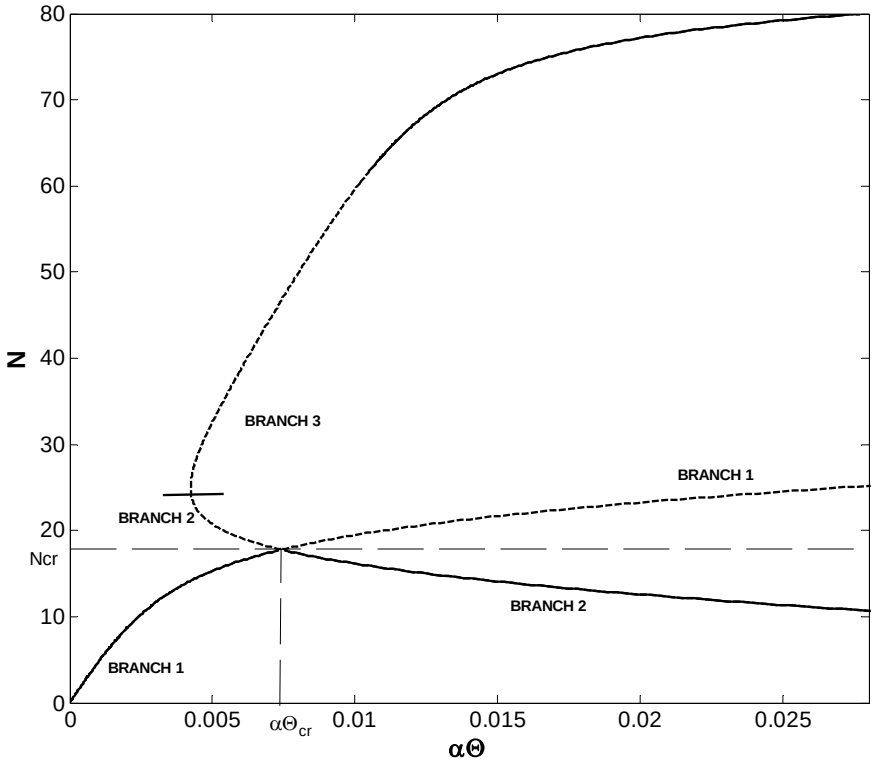

Figure 5a. Membrane force vs. applied temperature difference, hinged-fixed, intact structure, *a* = 0.8. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

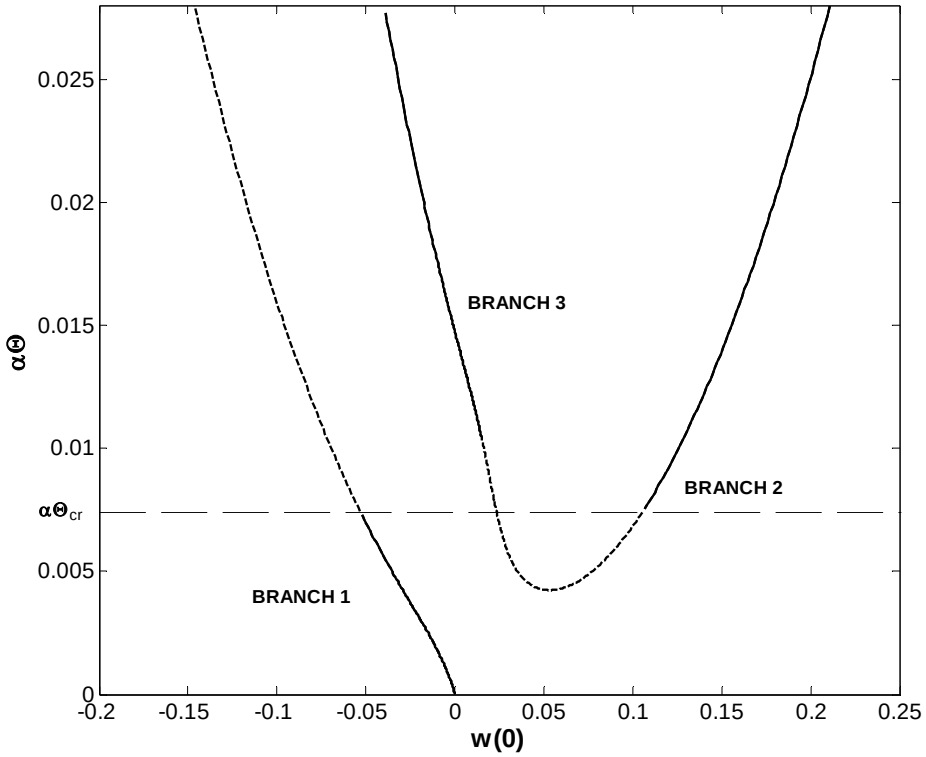

Figure 5b. Applied temperature difference vs. centerspan deflection, hinged-fixed, intact structure, *a* = 0.8. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

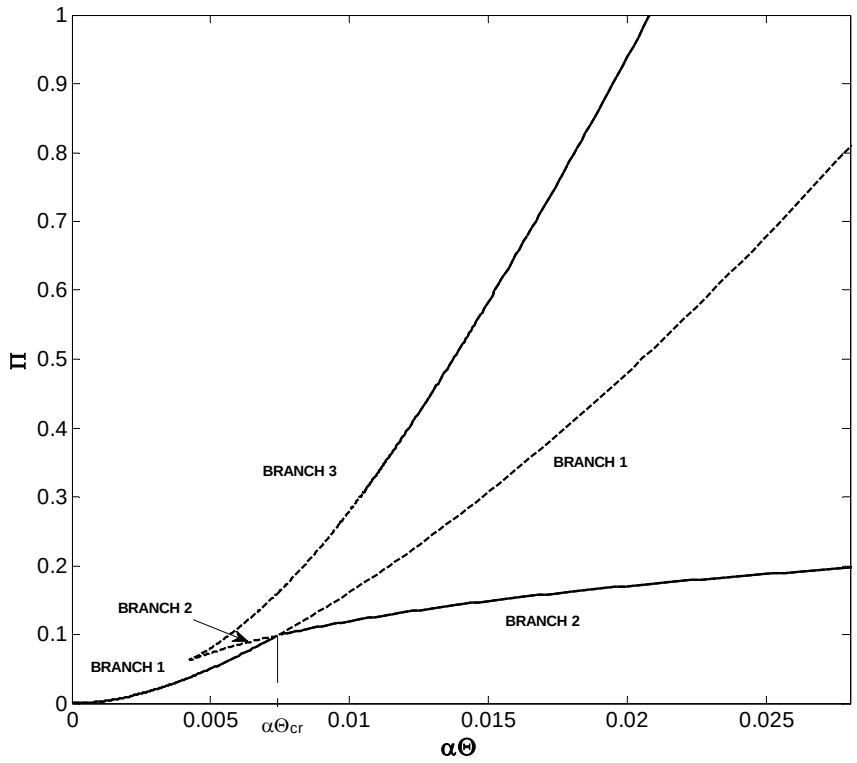

Figure 5c. Potential energy vs. applied temperature difference, hinged-fixed, intact structure, *a* = 0.8. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

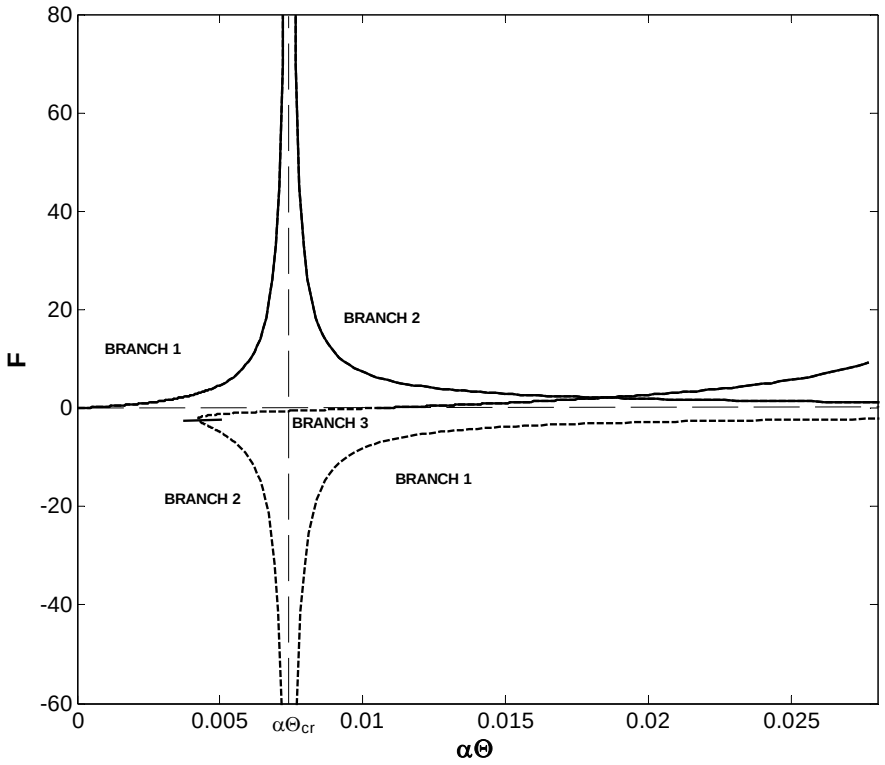

Figure 5d. Stability function vs. applied temperature difference, hinged-fixed, intact structure, *a* = 0.8. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

Figure 5a shows the membrane force as a function of the applied temperature difference. The path is recognizable in three branches, as indicated. Points on the solid line portions of the paths correspond to equilibrium configurations which qualify as stable, based on the sign of the stability function $\hat{F}$, which is displayed in Figure 5d. Points on the dashed portions of the paths correspond to unstable equilibrium configurations. It is apparent that critical behavior occurs at the point on the graph representing the ordered pair $(\alpha\Theta_{cr}, N_{cr})$ for the specific value of the bond size *a*. For the particular structure under consideration, the calculated values are $\tilde{\Theta}_{cr} = \alpha\Theta_{cr} = 0.0074$ and $N_{cr} = 17.78$. Before loading to this point, Branch 1 is stable

and Branch 2 is unstable. Beyond this critical point, the reverse is true. It is noted that there is a third branch, with a portion of it being stable. This trend is also reflected in Figure 5d. Examination of Figures 5b and 5c gives insight into the general behavior.

Figure 5b shows the load-deflection behavior. The characteristic displacement is taken as the centerspan deflection of the structure. Recall the sign convention employed for transverse displacement: positive deflection is taken as down. The structure, for the given set of parameters, deflects up when a temperature difference is increased from zero. The first branch of the path is seen to be stable below the critical temperature difference, and is unstable beyond it. The second branch of the solution is seen to be unstable below the critical temperature difference, and stable above it. As in the previous figure, the third branch also displays stable equilibrium configurations.

We now turn to Figure 5c, which shows the potential energy as a function of the applied thermal load, in order to determine the most energetically favorable position among multiple stable configurations. The figure shows that while Branch 3 does contain stable equilibrium configurations, they are at higher energy levels, and hence not the most favorable. The system remains at the lowest possible potential energy, and hence follows Branch 1 until the temperature difference reaches the critical value, as predicted by the vanishing of the appropriate function H as well as the loading parameter M_λ. Upon achieving the critical point, the system takes the path shown in Branch 2. This behavior is what Karlsson and Bottega (2000) first referred to as *sling-shot buckling*.

4.3 Sling-Shot Buckling of a Partially Debonded Structure

Section 3.4.2 presents a solution for a partially debonded structured which accounts for the existence of a contact zone. In particular, we examine the case of full

contact. That is, when $L_p = b \neq a$. The parameters chosen are $h_0 = 1$, $E_0 = 1$, to isolate the thermal effect, and $\alpha_p^0 = 0.5$, and $a = 0.8$, to be consistent with the previously established findings. The patch length is taken as $b = 0.9$ and for this case, $\tilde{\Theta}_{cr(cz)} = 0.0078$ and $N_{cr(cz)} = 18.79$. The structure is stiffened by the presence of a contact zone (in comparison to if it were absent). Figure 6a-d shows the results obtained for a partially debonded structure possessing a full contact zone. The qualitative interpretations of the figures follow from the discussion in the preceding section. Once again, the solid line portions of the path indicate stable equilibrium configurations, while dashed line portions indicate unstable equilibrium configurations.

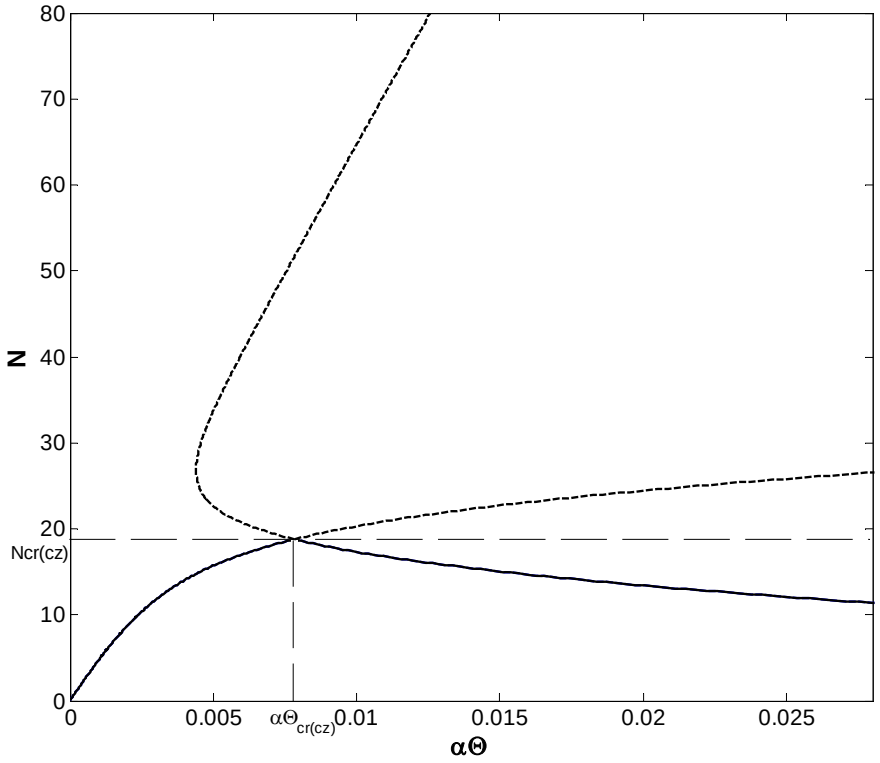

Figure 6a. Membrane force vs. applied temperature difference, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

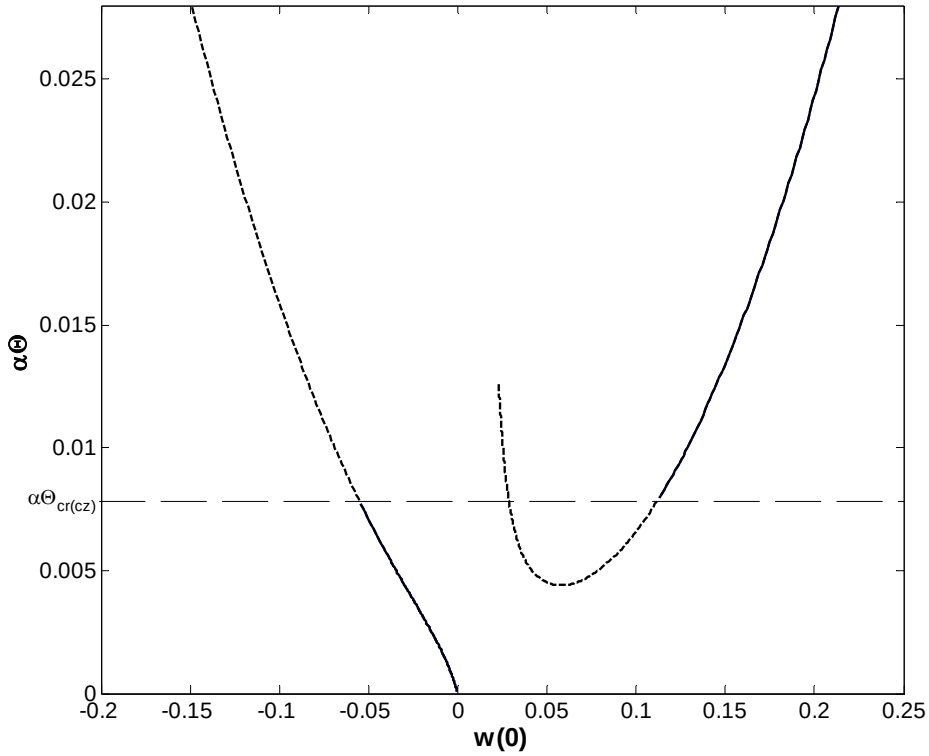

Figure 6b. Applied temperature difference vs. centerspan deflection, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

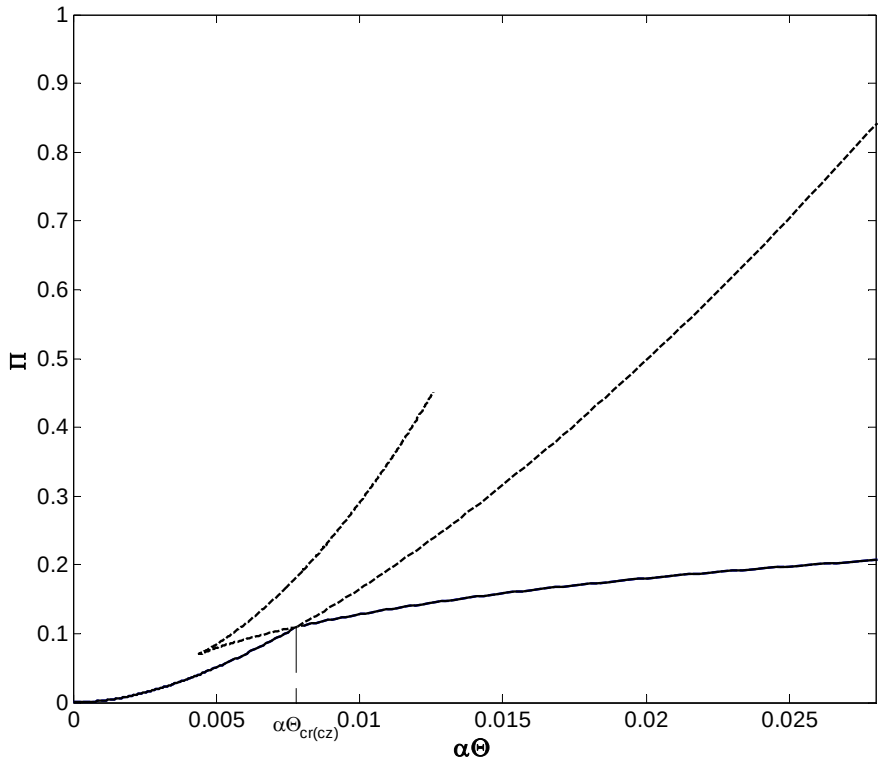

Figure 6c. Potential energy vs. applied temperature difference, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

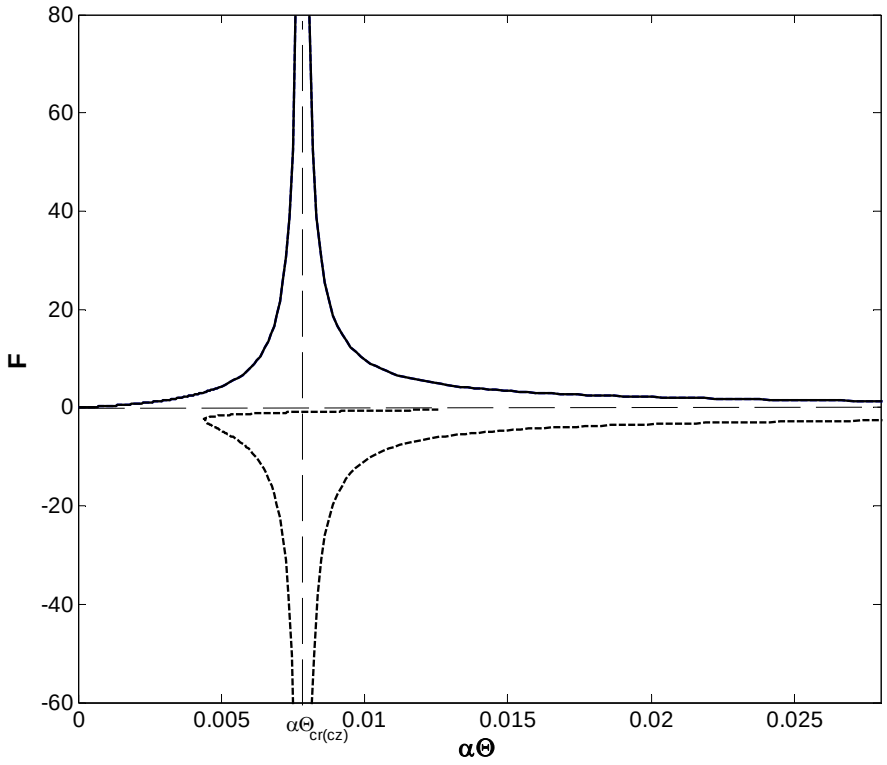

Figure 6d. Stability function vs. applied temperature difference, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

Our mathematical model accommodates the possibility of a contact zone. However, we must diagnose the physical validity of the solution. This is done by performing the checks on the curvatures described in Section 3.6. Figure 7a-d specifies the configurations on the paths (shown in Figure 6a-d) for the structure which are physically realizable (i.e. do not violate the conditions on the curvatures which allow for full contact) by highlighting them in red.

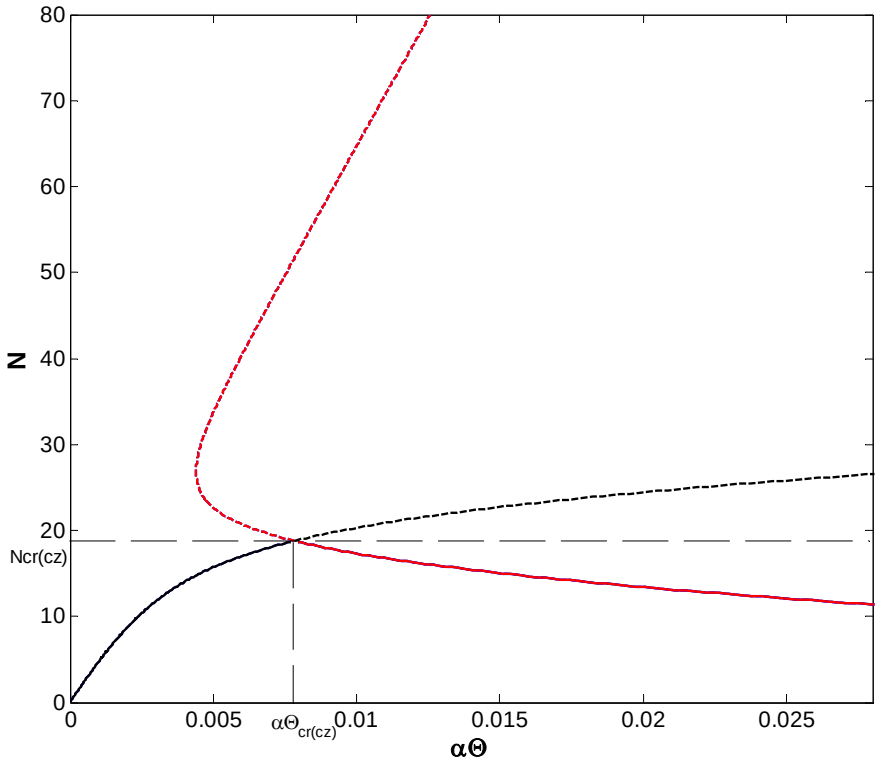

Figure 7a. Membrane force vs. applied temperature difference, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

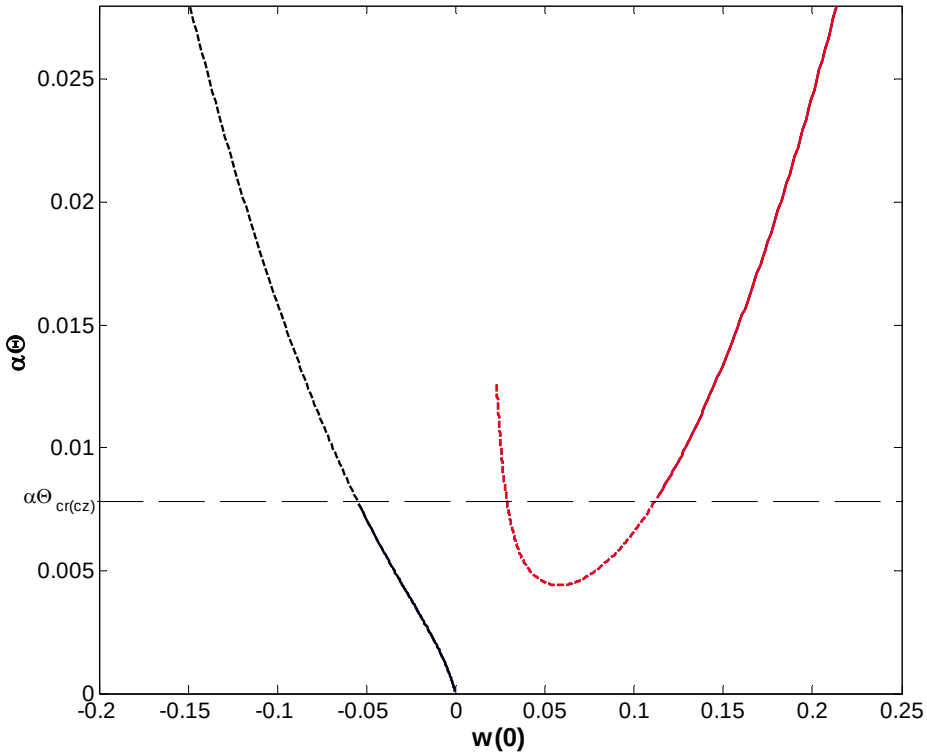

Figure 7b. Applied temperature difference vs. centerspan deflection, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

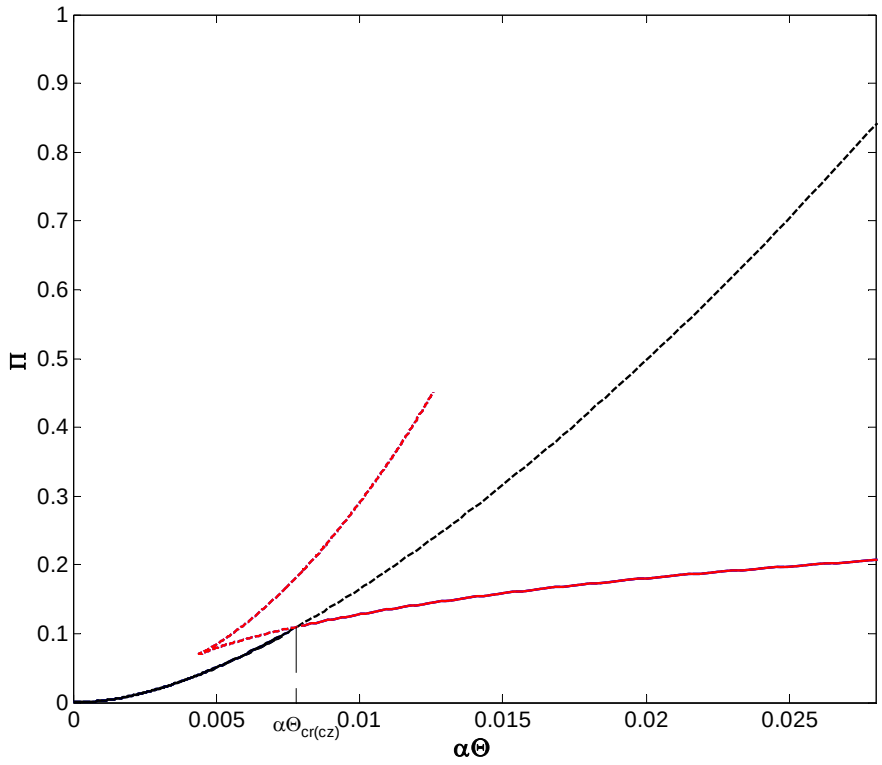

Figure 7c. Potential energy vs. applied temperature difference, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

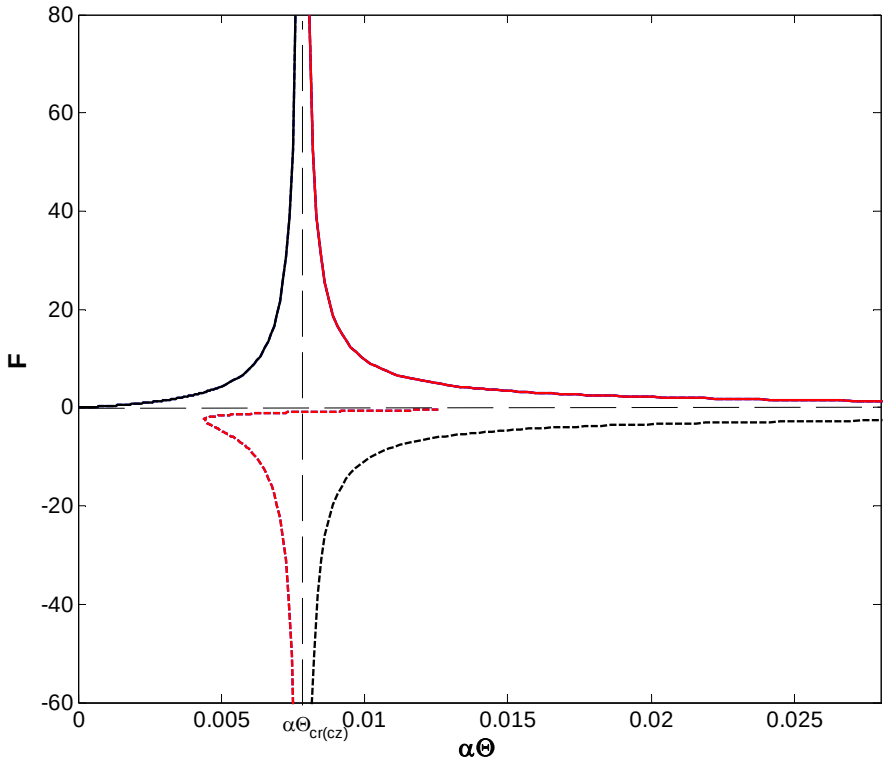

Figure 7d. Stability function vs. applied temperature difference, hinged-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

This reveals that, for a fixed partially debonded structure with hinged ends, the contact zone is physically present post sling-shot buckling. More generally, it is present only when the hinged structure deflects downward, no matter what the stability of the configuration. This is pictured in Figure 8.

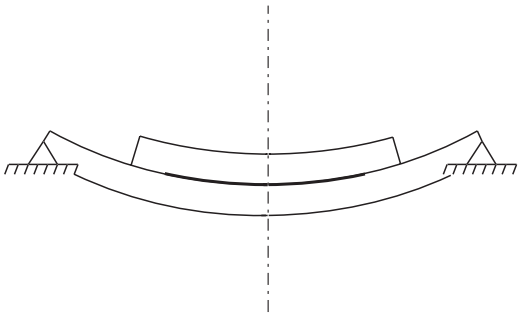

Figure 8. Downward deflection of a partially debonded structure with hinged-fixed ends

We observe this behavior, regardless of patch size or bond zone size. The mathematical solution for the partially debonded structure possessing a full contact zone is artificial while it deflects up; essentially, it is forced. On physical grounds, the free portions of the patch (the flaps), will lift off the baseplate and will not bend in this configuration, since they are unloaded (see Section 2.4.3). Although it physically possesses the flaps in such configurations, in terms of stiffness and energy, it is as if they are not present. This is shown in Figure 9.

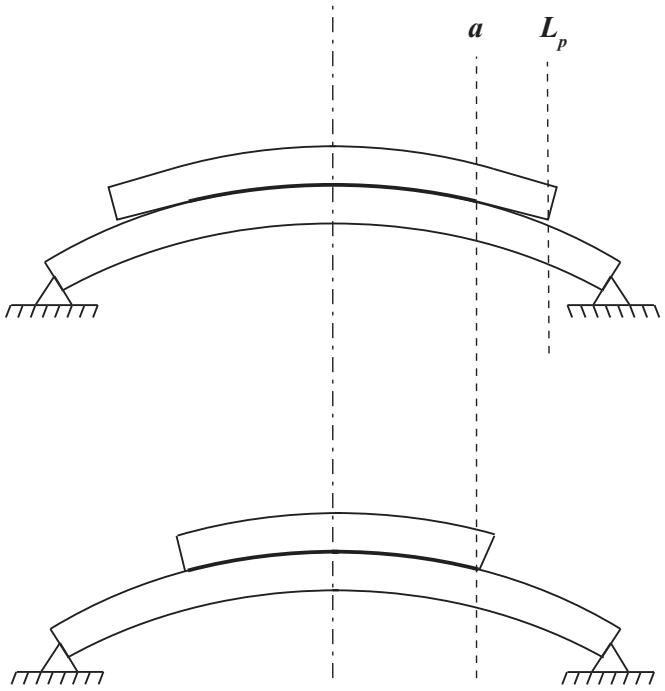

Figure 9. Schematic depicting the equivalence of the upward deflected partially debonded structure to that of an intact structure with the same bond zone size.

Hence the salient behavior can be captured using the appropriate model for an intact structure (of the same bond zone size), which is given in Chapter 3. A partially debonded hinged structure possesses a *dual nature*, depending on whether the flaps have fully lifted or are in full contact. The global stiffness and strain energy of the structure are different, depending on the position of the unbonded patch flaps. To assess the overall behavior, we superimpose the physically valid results of the solutions for when the flaps are in full contact zone and when the flaps are fully lifted. The latter will be represented by the model for an equivalent intact structure.

4.4 The Phenomenon of "Buckle-Trapping"

The dual nature of the partially debonded structure significantly affects its behavior. When the flaps lift off, the structure actually buckles at the lower buckling threshold predicted by the model of a fully intact structure (for equivalent values of *a*). In the following figure we compare the solution for the case when the flaps are fully lifted to the solution for the case when the flaps are in full contact (a = 0.8, b = 0.9). Figure 10a-b shows the results for the two cases superimposed. The physically valid full contact zone configurations are indicated in red. Black corresponds to the physically valid configurations with lifted flaps. Again, we consider a structure for which $h_0 = 1$, $E_0 = 1$ and $\alpha_p^0 = 0.5$.

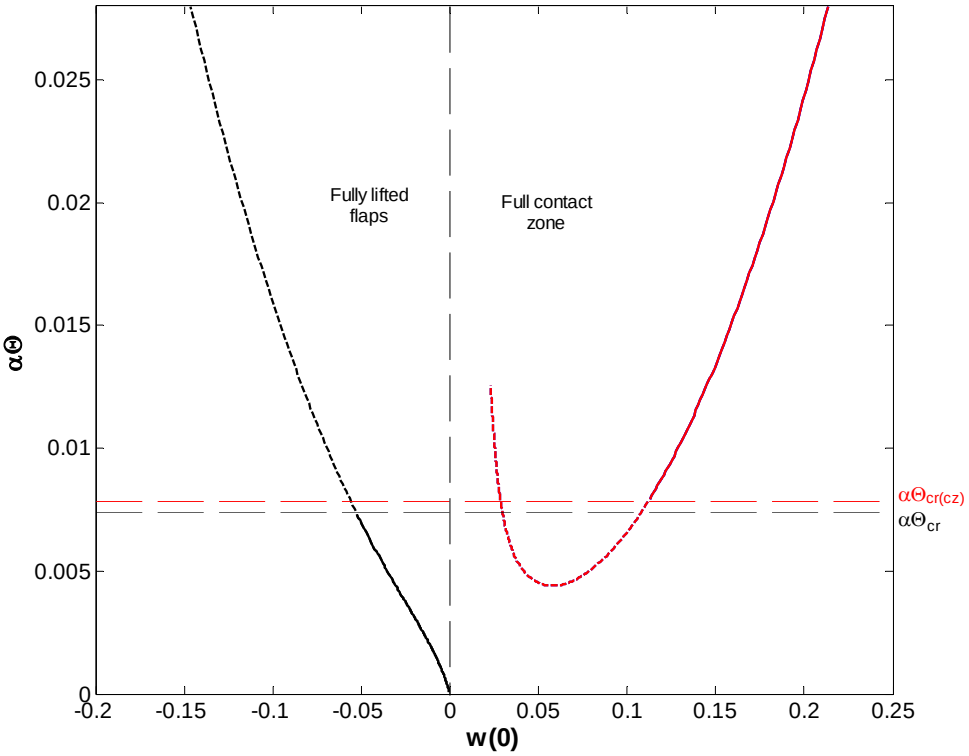

Figure 10a. Applied temperature difference vs. centerspan deflection, Hinged-fixed, a = 0.8, L_p = 0.9 partially debonded structure with full contact zone (red) or fully lifted flaps (black). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

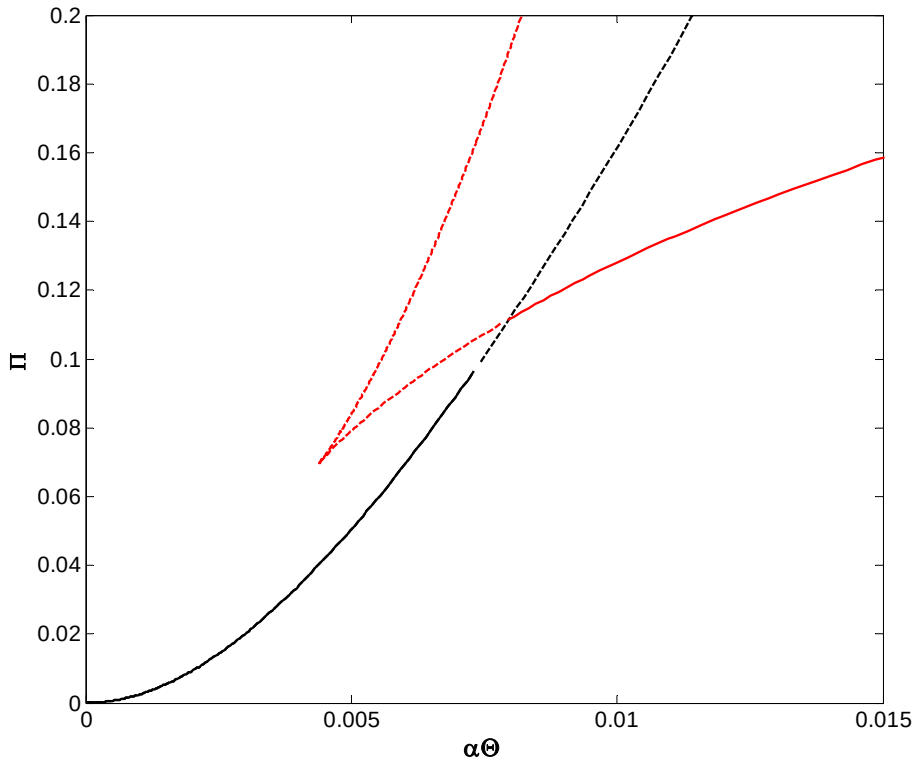

Figure 10b. Potential energy vs. applied temperature difference, Hinged-fixed, a = 0.8, L_p = 0.9 partially debonded structure with full contact zone (red) or fully lifted flaps (black). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

Examination of Figure 10a immediately reveals the discrepancy in predicted critical sling-shot buckling threshold for the prescribed loading. When a full contact zone is present, the model predicts a higher threshold in comparison to the case when a contact zone is absent. However, the solution including the existence of a full contact zone is not physically realizable while the structure deflects up, since the flaps lift off. Hence, sling-shot buckling occurs at the lower of the two predicted critical values of the temperature difference. When the structure buckles, physically speaking, the lifted flaps will come back into contact with the baseplate as soon as the structure passes through vanishing transverse deflection. At the flat configuration, the flaps touch the baseplate, however,

they are not deformed and do not possess strain energy. Hence, they are in *passive contact*. During sling-shot buckling, the structure deflects downward, beyond the flat configuration, so the flaps become engaged and the stiffness increases. As can be seen in the figures, there is no stable equilibrium configuration for the structure to assume at the critical buckling threshold. Figure 11 presents a close up of Figure 10a.

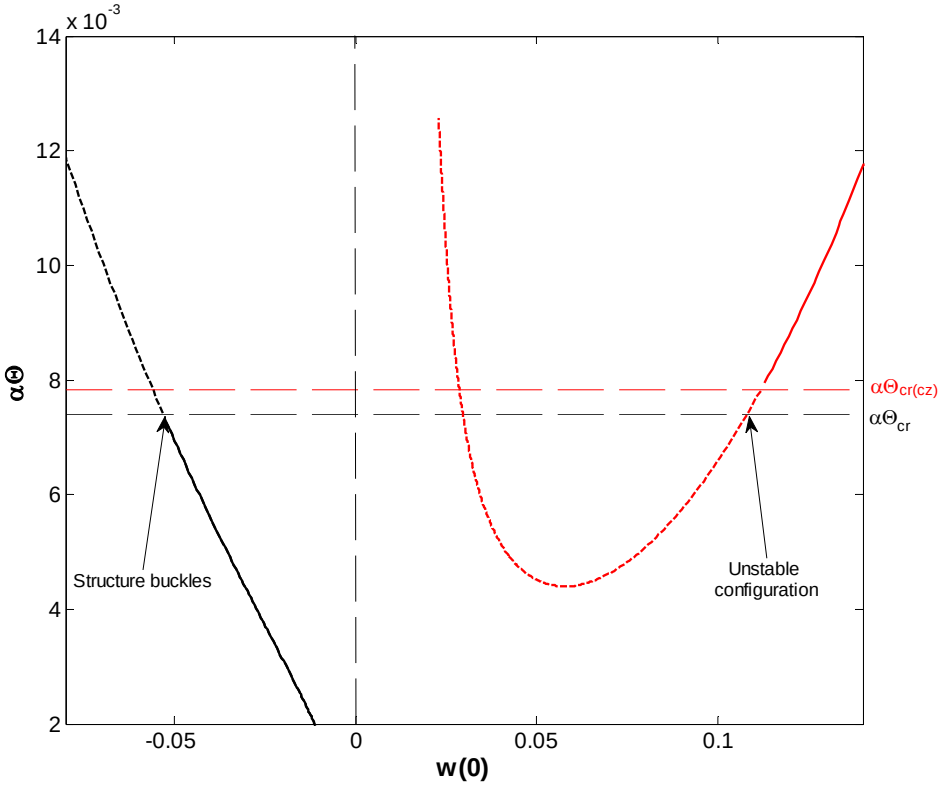

Figure 11. Close up of applied temperature difference vs. centerspan deflection, Hinged-fixed, a = 0.8, L_p = 0.9 partially debonded structure with full contact zone (red) or fully lifted flaps (black). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

This creates a paradox. When the structure is deflected above the zero level, the contact zone lifts off. At zero deflection, the flaps are in passive contact, meaning there is no compressive stress at the mating surface. When the structure is deflected below the zero level, the contact zone is present. There is a dynamic effect taking place. Under the

given parameters, a partially debonded structure buckles at the lower predicted threshold, since the flaps are initially disengaged. However, upon passing through zero deflection, the flaps come in contact, thus changing the nature of the structure. It will not go to the equilibrium position offered by the solution accounting for a full contact zone. This is because the equilibrium configuration for that structure, at that load level, is unstable. At that load level, for a structure with full contact, the associated stable equilibrium position is deflected upward. Hence the deflection will reverse directions and proceed to sling back to an upward configuration. But once it passes through the flat position on the way back up, the flaps will lift off, again changing the structure. With fully lifted flaps, the structure's "preference" is to buckle to a configuration with downward deflection, and thus the process repeats, as an undamped oscillator bounded by two unstable equilibrium configurations. We shall refer to this phenomenon as *Buckle-Trapping*. Since actual systems possess some amount of structural damping, the oscillatory "behavior" will likely decay, such that the structure eventually comes to rest at some intermediate configuration, not predicted by conventional analysis.

To identify this configuration, we make the following argument: the Theorem of Stationary Potential Energy locates equilibrium positions as stationary points of a smooth, continuously differentiable potential energy functional. However, the analyses of a partially debonded structure with fully lifted flaps or with a full contact zone are each performed separately. The structure has a dual nature, that is, when deflected down, the flaps are in full contact with the baseplate and when deflected up, the flaps are lifted off the baseplate. Thus, the duality implies that its potential energy is *piecewise defined*. Further, the intersection of the potential energies for each of the two structures (fully

lifted flaps or full contact zone) must occur at a common configuration, when both structures exist simultaneously. This is so only for the flat configuration (zero deflection). If this is, in fact, an equilibrium configuration then, from the Theorem of Stationary Potential Energy, it must correspond to an extremum of the potential energy functional of the system. Further, if stable, it must also correspond to a minimum. Since the potential energy functional for the dual structure is only piecewise smooth, it is postulated that an energy cusp (rather than a smooth minimum) exists for this configuration. This cusp exists where the potential energies corresponding to the two different structures intersect. Thus, the structure becomes trapped in the flat position.

A further increase in load would cause similar alternating behavior about the flat position. Once the higher of the two critical buckling temperature differences is reached, $\tilde{\Theta}_{cr(cz)}$, the structure will be released from the trap, and finally completely buckle and remain in the downward deflected position predicted by the solution for the structure with a full contact zone. Upon unloading, the structure will get caught in the buckle trap when the prescribed loading becomes less than $\tilde{\Theta}_{cr(cz)}$. When the loading becomes less than $\tilde{\Theta}_{cr}$, the structure is released from the trap and assumes the path for the fully lifted flap configuration. The range of temperature differences for which buckle-trapping is possible increases as the bond zone size *a* decreases (and the overall patch length remains the same).

In Chapter 5 we will explore the ramifications of including the contact zone in the thermal buckling analysis for the clamped-fixed case.

Chapter 5

Thermal Buckling of the Clamped Structure

5.1 Introduction

We turn our attention to the structure which is clamped with respect to rotation and fixed with respect to in-plane translation at its ends. The completely intact version of this case was also simulated by Karlsson and Bottega (2000). For an intact structure, the behavior is qualitatively the same as that outlined for the hinged-fixed case in Section 4.2. Upon examination of a clamped partially debonded structure, we find that the range of behaviors is more robust than that for the hinged case of Chapter 4. As in the previous chapter, we consider the example case when $h_0 = 1$, $E_0 = 1$, $\alpha_p^0 = 0.5$, with a patch length of $L_p = 0.9$. However, the contact zone boundary *b* and the patch length are not necessarily equal. It is under these support conditions that we may see intermediate contact propagate. That is, the structure does not always possess a full contact zone. Here we present three representative examples of the different types of behavior, based on the bond zone size *a*.

5.2 Example 1: *a* = 0.8

When a full contact zone is present, the model for a partially debonded structure predicts that $\tilde{\Theta}_{cr(cz)} = 0.0159$ and $N_{cr(cz)} = 38.2256$. Figure 12a-d presents the various graphical representations of the behavior for this case. Solid lines indicate equilibrium configurations which qualify as stable according to the sign of the appropriate stability function, while dashed lines indicate unstable equilibrium configurations.

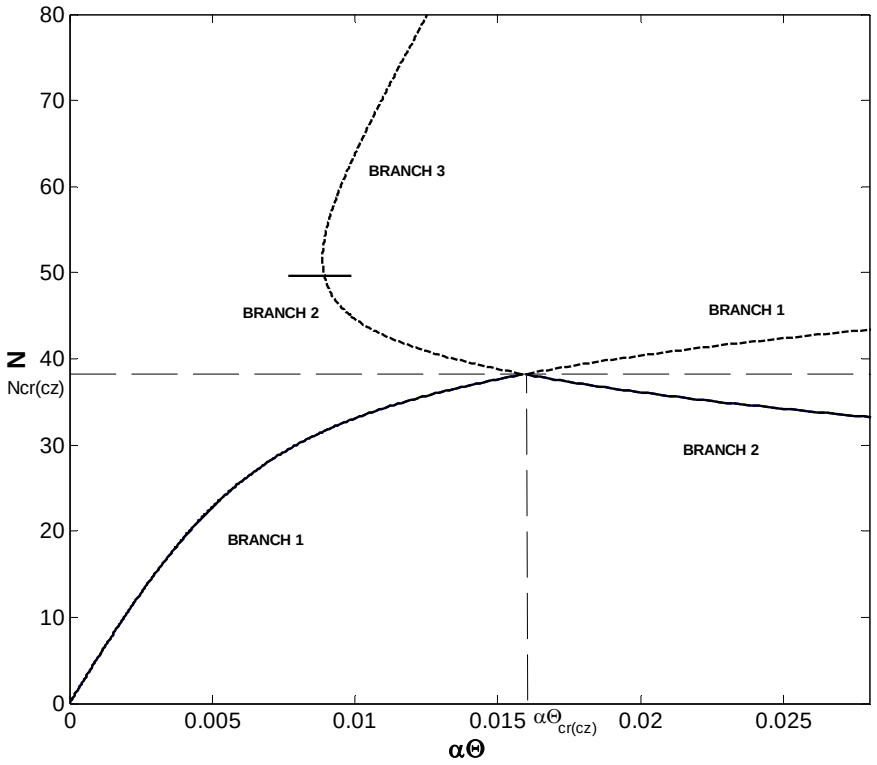

Figure 12a. Membrane force vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

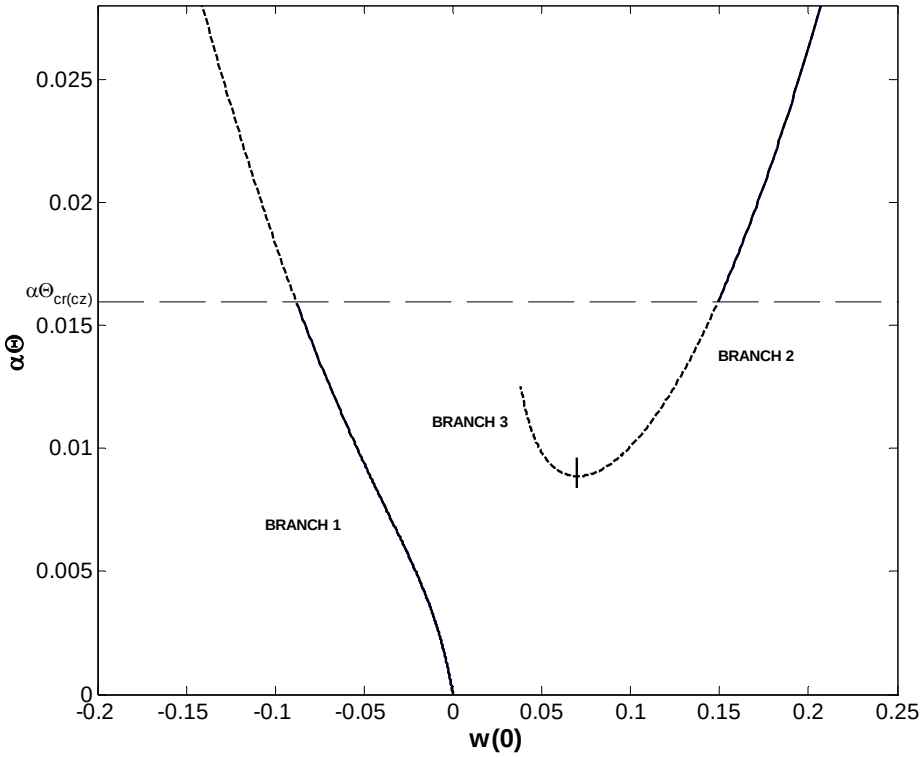

Figure 12b. Applied temperature difference vs. centerspan deflection, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

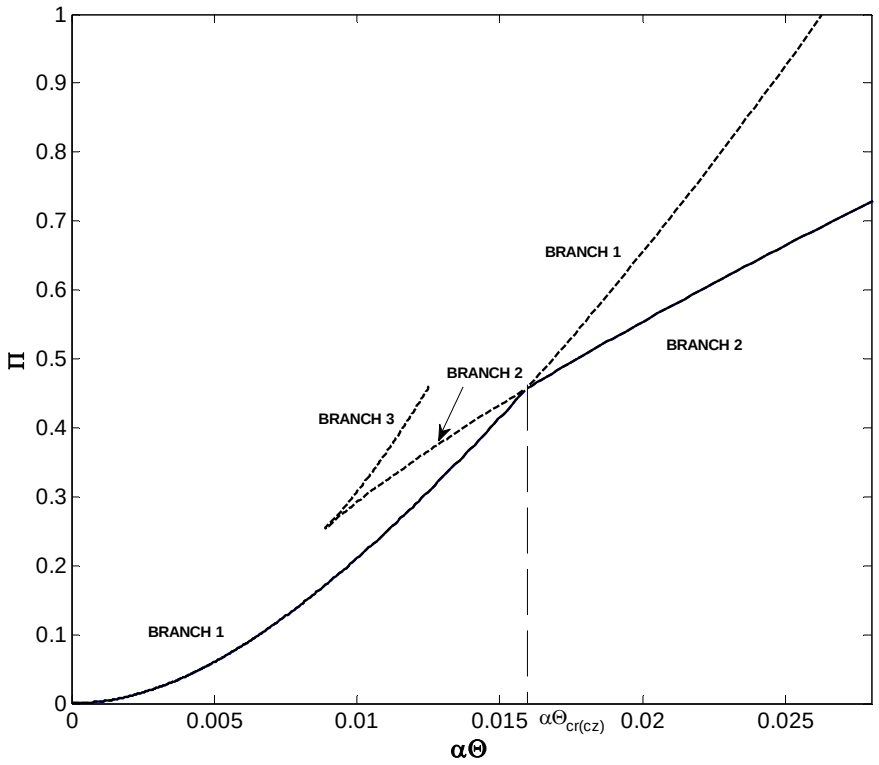

Figure 12c. Potential energy vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

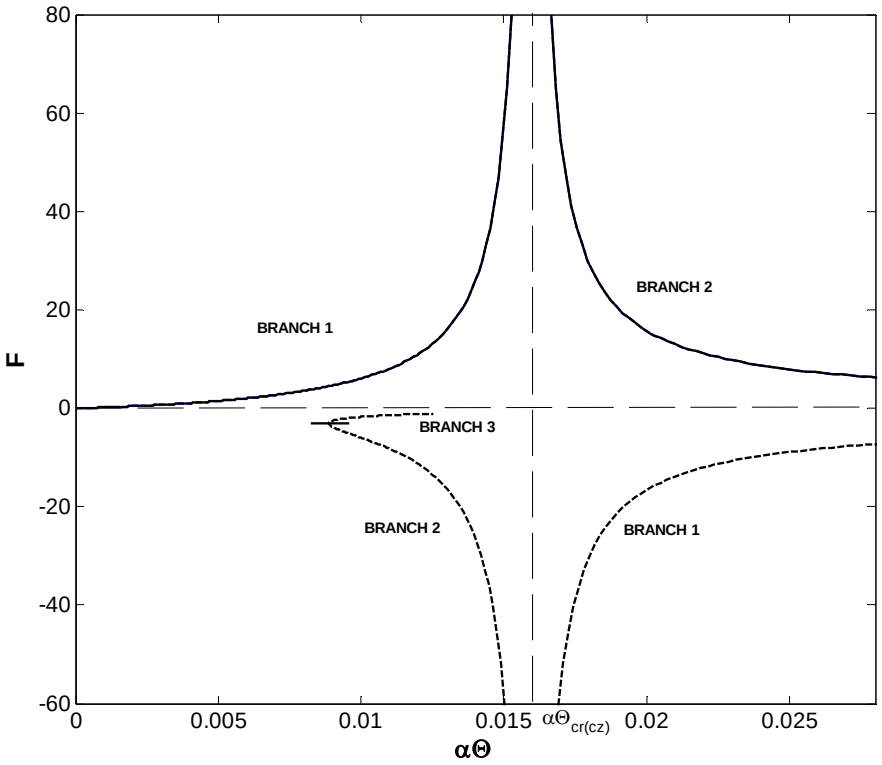

Figure 12d. Stability function vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

The interpretation follows from that for the hinged case presented in Section 4.2. As noted for the hinged case, the mathematical model for the partially debonded structure allows for a full contact zone, yet is not always physically valid. Figure 13a-d shows which configurations from Figure 12a-d correspond to those for which a full contact zone is physically realizable by denoting them in red. (Solid lines continue to indicate stable configurations, regardless of color.)

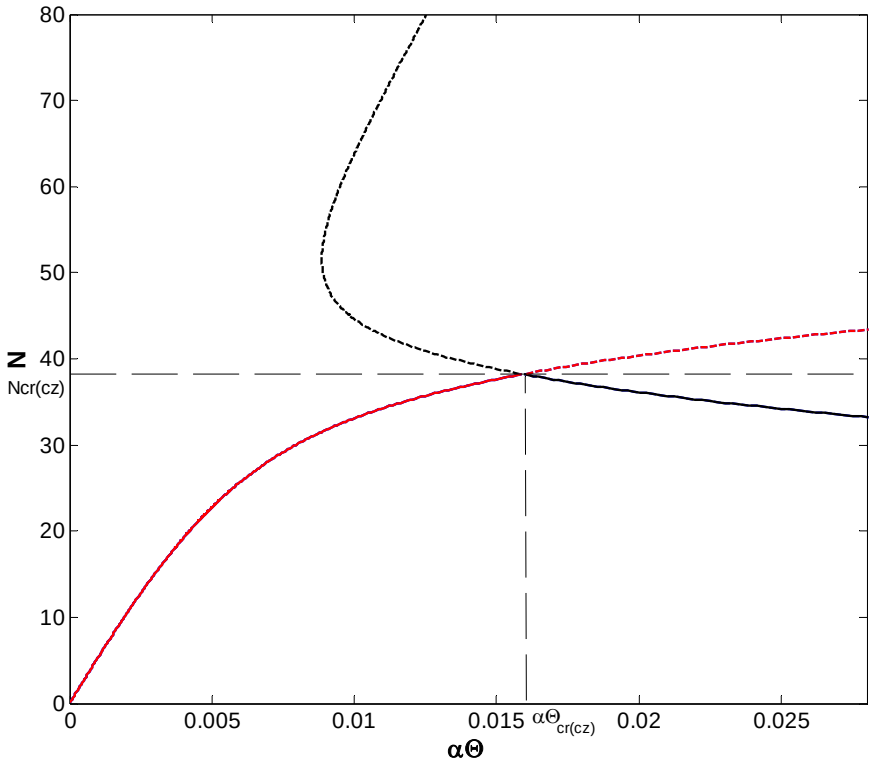

Figure 13a. Membrane force vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

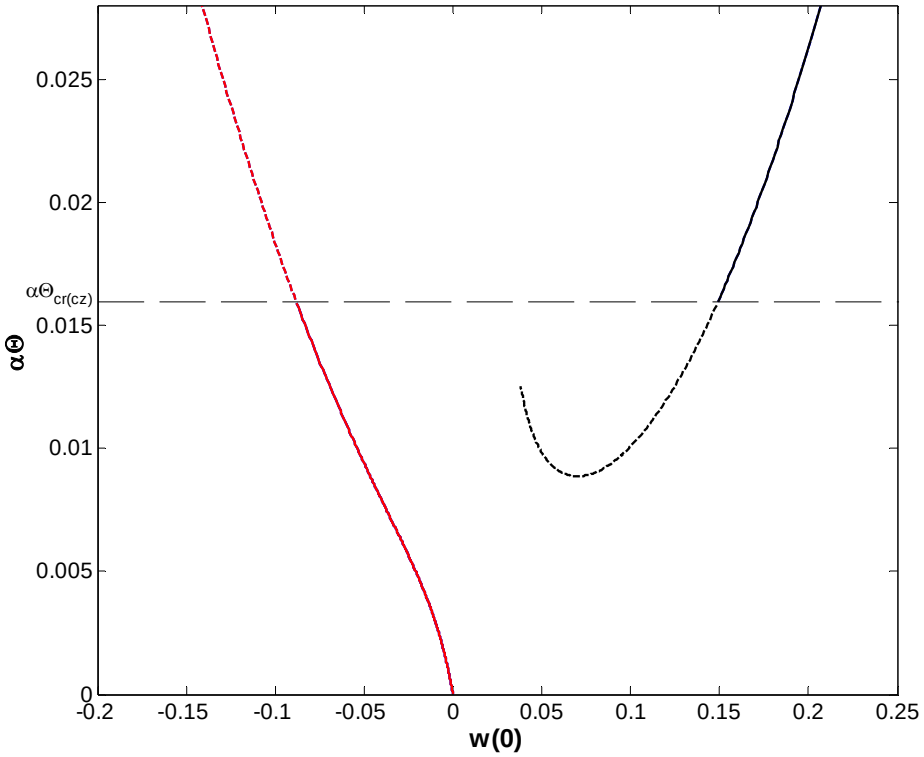

Figure 13b. Applied temperature difference vs. centerspan deflection, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

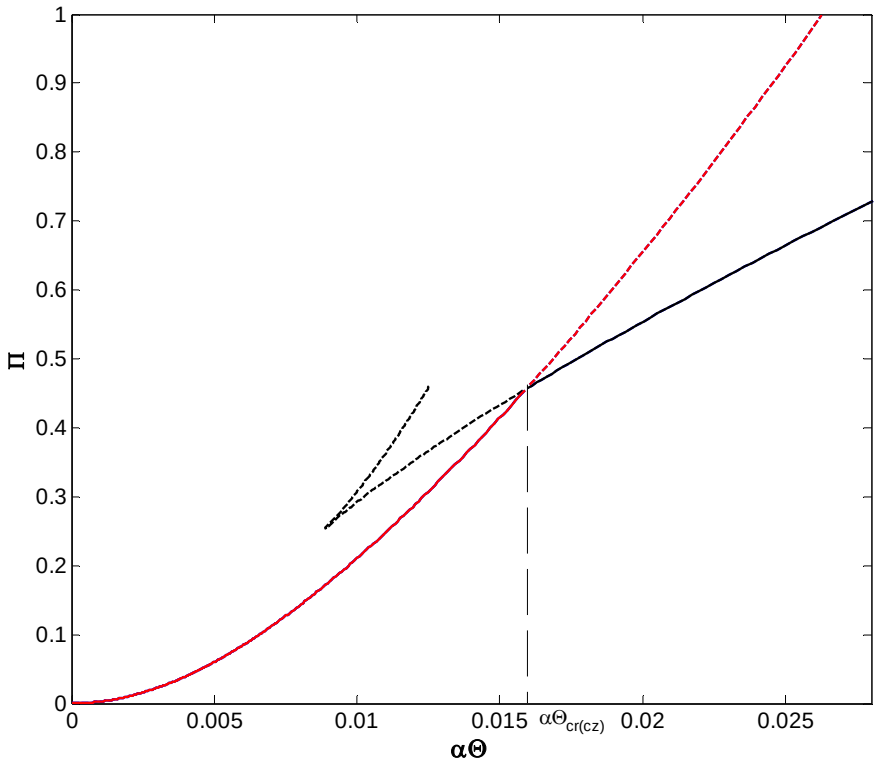

Figure 13c. Potential energy vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

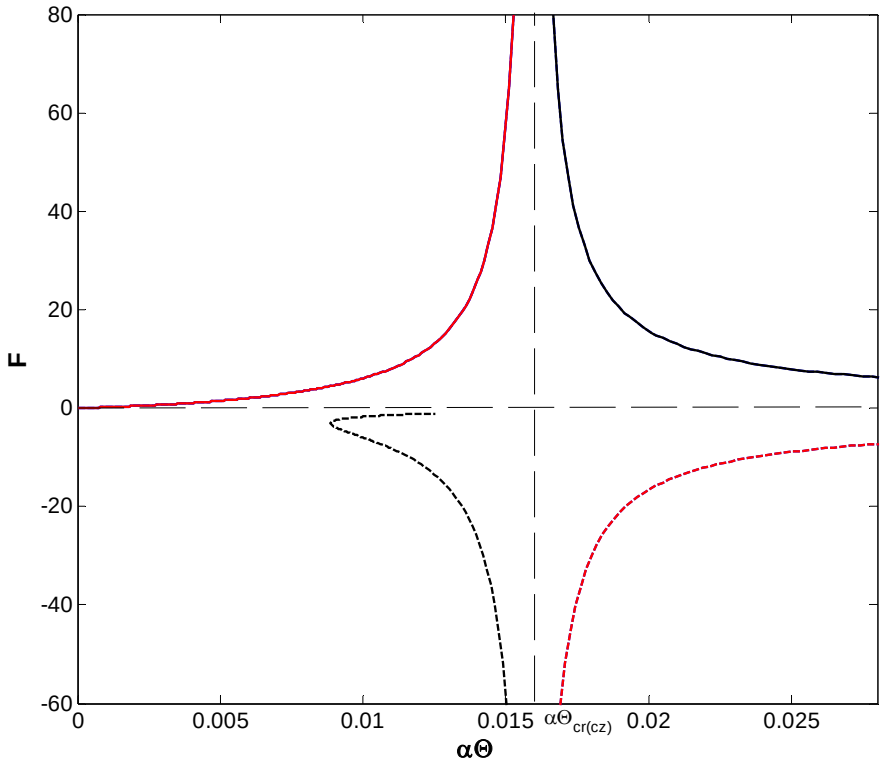

Figure 13d. Stability function vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.8, L_p = 0.9. The physically valid portion is shown in red. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

It is apparent that the behavior of the fixed patched structure when the edges are clamped differs from that when the edges are hinged. Here we see that a full contact zone exists for the pre-buckled state of the structure. Upon sling-shot buckling, the flaps lift off, such that the contact zone is no longer physically present. Hence, in this case, the deflected structure with lifted flaps in a post-buckled configuration can be represented by the model for an intact structure possessing the same bond zone size. The scenario is depicted in Figure 14.

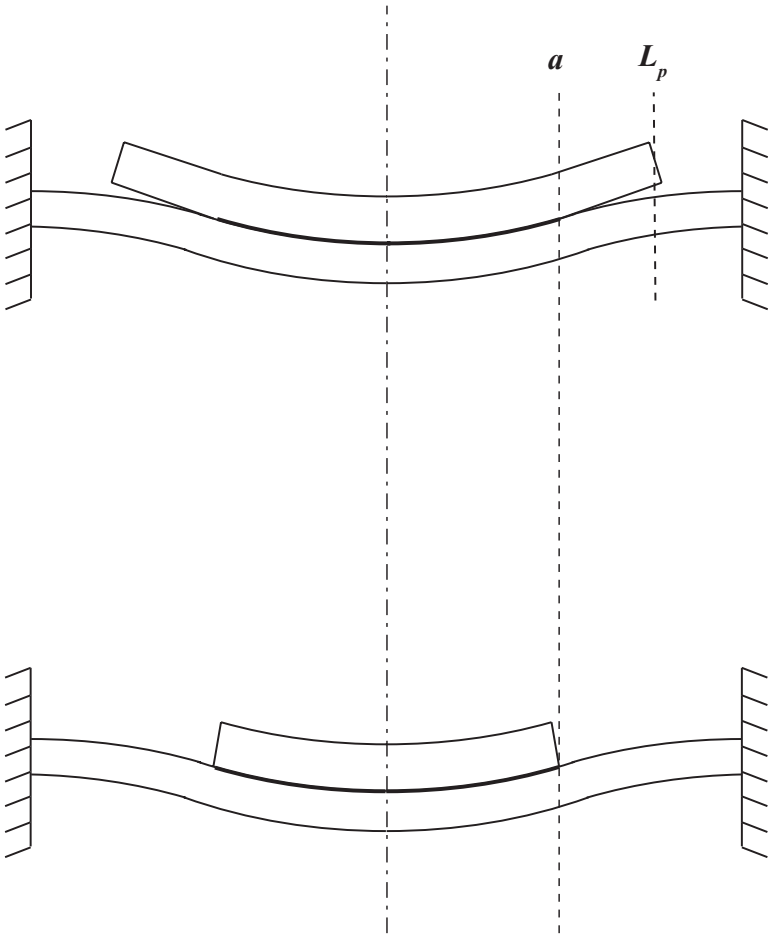

Figure 14. Schematic depicting the equivalence of the downward deflected partially debonded structure to that of an intact structure with the same bond zone size.

There is a duality of the clamped structure as well. To gain better insight into the overall behavior, we superimpose the results of the two cases: fully lifted flaps and full contact, shown in Figure 15a-b. For a structure with fully lifted flaps, it is predicted that $\tilde{\Theta}_{cr} = 0.0151$ and $N_{cr} = 36.3430$. As discussed in Section 4.4, the physically valid full contact zone configurations are denoted by red. Black indicates the physically valid solution for the structure with fully lifted flaps.

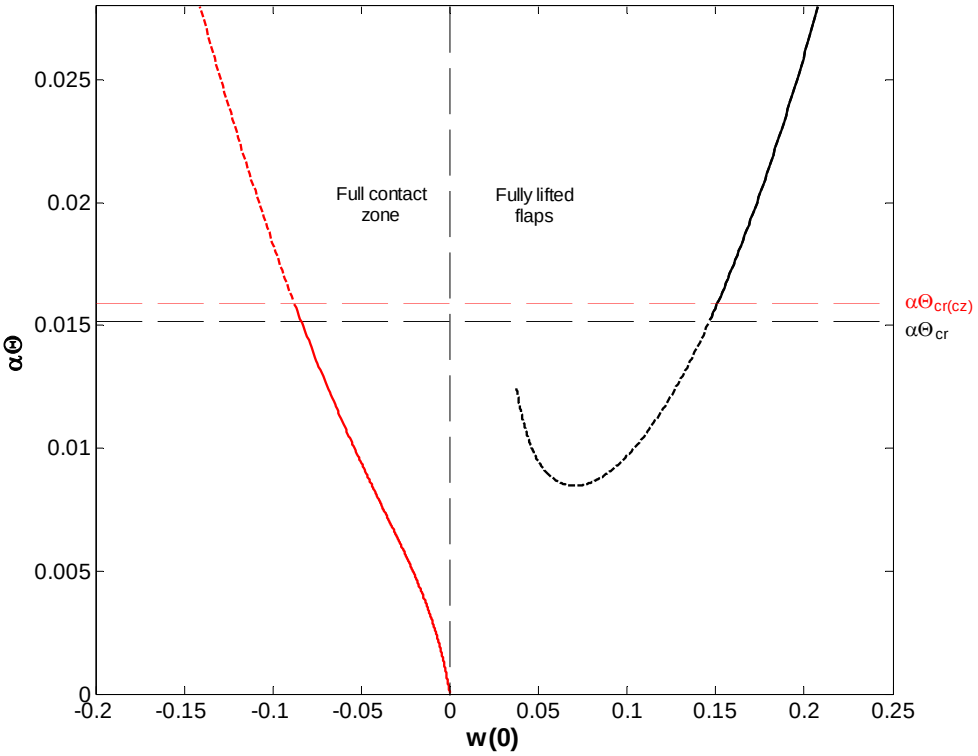

Figure 15a. Applied temperature difference vs. centerspan deflection, clamped-fixed, a = 0.8, L_p = 0.9 partially debonded structure with full contact zone (red) or fully lifted flaps (black). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

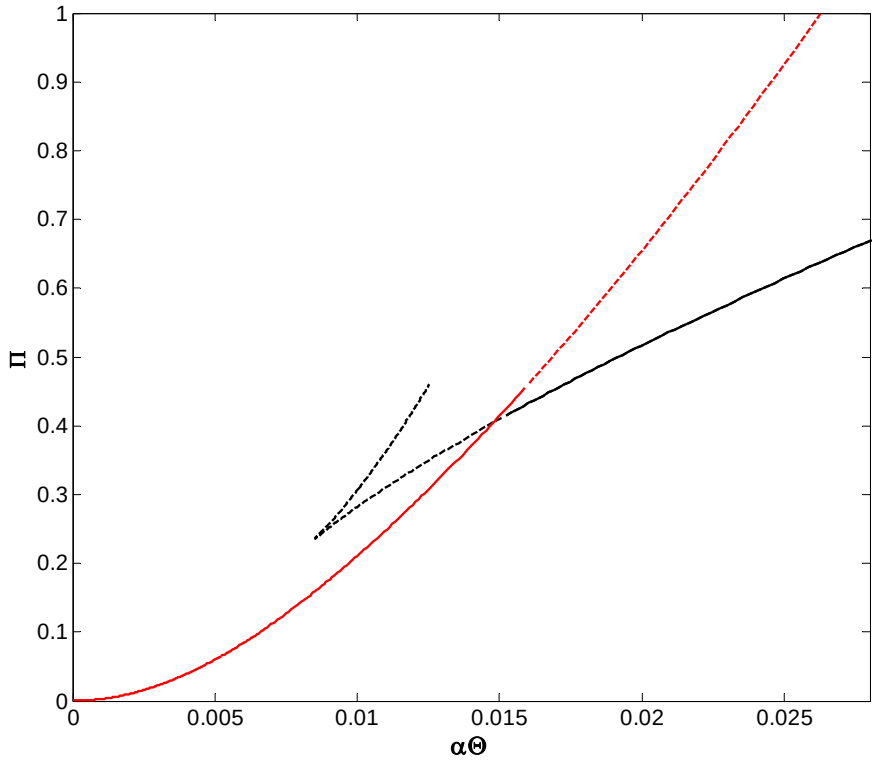

Figure 15b. Potential energy vs. applied temperature difference, clamped-fixed, a = 0.8, L_p = 0.9 partially debonded structure with full contact zone (red) or fully lifted flaps (black). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

A close-up of Figure15a is presented in Figure 16 to elucidate the behavior at the critical sling-shot buckling temperature difference.

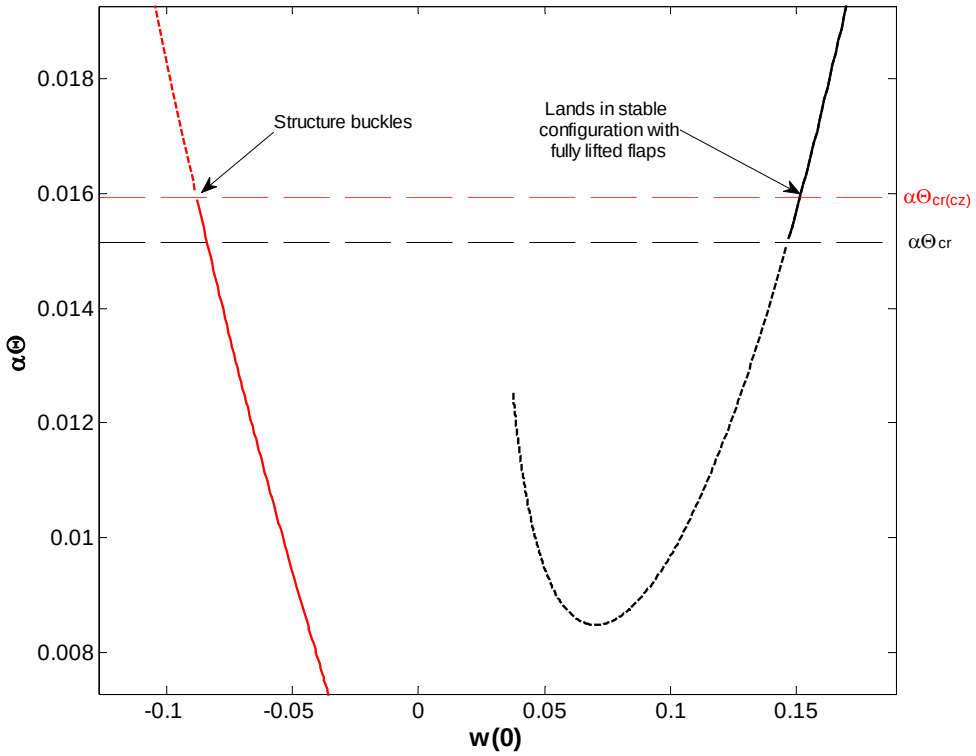

Figure 16. Close up of applied temperature difference vs. centerspan deflection, clamped-fixed, a = 0.8, L_p = 0.9 partially debonded structure with full contact zone (red) or fully lifted flaps (black). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

The predicted critical thermal buckling threshold is higher when the flaps of the patch remain in full contact, as expected by the added stiffness they provide. Since the full contact zone remains physically present, the structure will buckle at the higher level predicted by the appropriate model. Once the structure buckles, the flaps lift off, rendering the structure more compliant. It buckles to a stable equilibrium configuration where the flaps are lifted. This behavior occurs for structure which possess a bond zone size of $a > 0.72$. After buckling, the deflection continues to increase in the same sense as $\tilde{\Theta}$ is increased. However, upon "unloading," the structure will not sling back until $\tilde{\Theta}_{cr} < \tilde{\Theta}_{cr(cz)}$ is achieved. This differs from the case of a perfectly intact structure.

5.3 Example 2: $a = 0.7$

When the bond zone size is slightly smaller, we observe different behavior as compared with the example described in Section 5.2. Here we consider a structure such that $a = 0.7$ and $L_p = 0.9$. For a structure with fully lifted flaps, the model predicts that $\tilde{\Theta}_{cr} = 0.0150$ and $N_{cr} = 36.1197$. For a structure which includes a full contact zone, the model predicts that $\tilde{\Theta}_{cr(cz)} = 0.0159$ and $N_{cr(cz)} = 38.0463$. In this case, like the one discussed in the previous section, a full contact zone is present in pre-buckled configurations. However, for the present case, a small intermediate contact zone remains (approximately $b = 0.74$) immediately after the structure buckles. The flaps are not fully lifted, and hence the structure in this configuration cannot be modeled by considering an equivalent intact structure. As loading is increased post-buckling, the remaining contact zone shrinks to the point where $b = a$ and it is no longer present. This is referred to as *propagating contact*. At the point of vanishing contact, the flaps have completely lifted off (and the behavior can once again be captured by the model of an equivalent intact structure). Figure 17a-b shows the load-deflection plot as well as the potential energy, for a partially debonded structure. Highlighted in green are configurations where the structure exhibits a contact zone which is not full. (The intermediate critical buckling temperature is seen to be $\tilde{\Theta}_{cr(in)} = 0.0150$.) The configurations of valid full contact are again indicated by red.

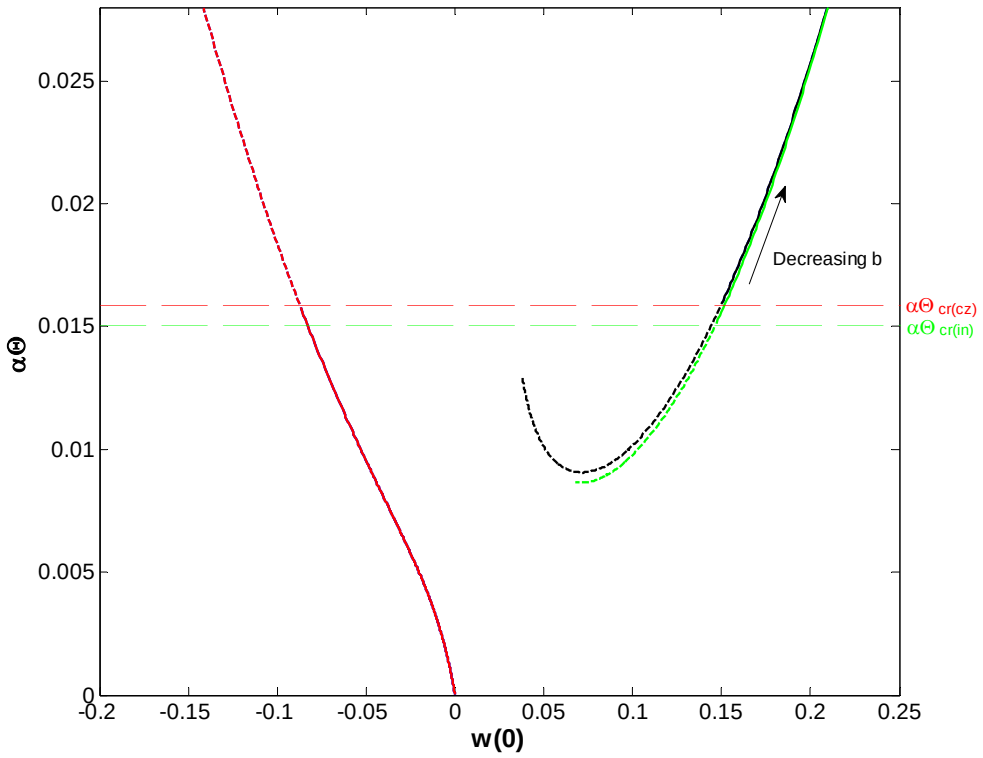

Figure 17a. Applied temperature difference vs. centerspan deflection, clamped-fixed, partially debonded structure with full contact zone, a = 0.7, L_p = 0.9. The physically valid portion is shown in red. The region of propagating contact is shown in green. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

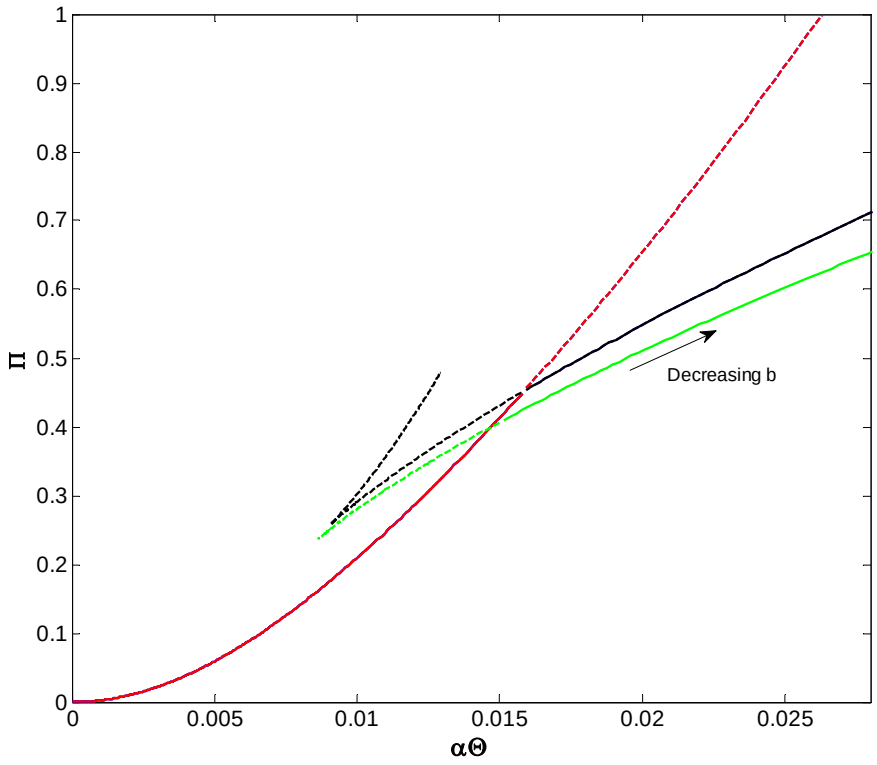

Figure 17b. Potential energy vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.7, L_p = 0.9. The physically valid portion is shown in red. The region of propagating contact is shown in green. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

Figure 18a-b displays the physically valid results for the partially debonded structure with a full contact zone (red) as well as propagating contact zone (green), superimposed on the results for the case of fully lifted flaps (black).

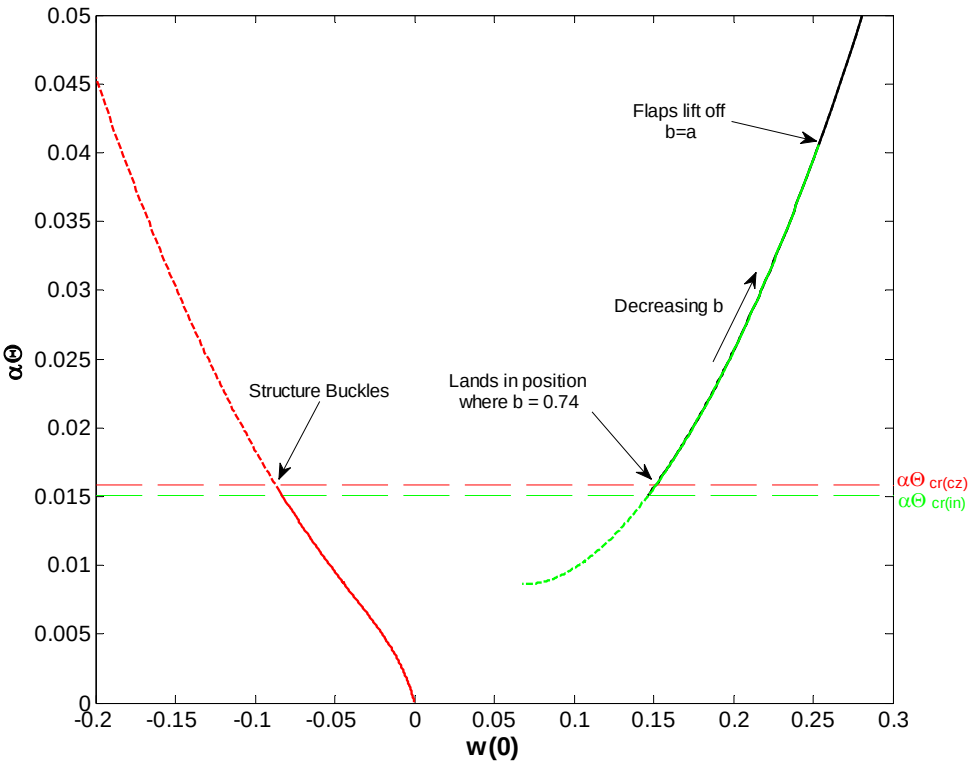

Figure 18a. Applied temperature difference vs. centerspan deflection, clamped-fixed, a = 0.7, L_p = 0.9 partially debonded structure with full contact zone (red), fully lifted flaps (black), or intermediate contact zone (green). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

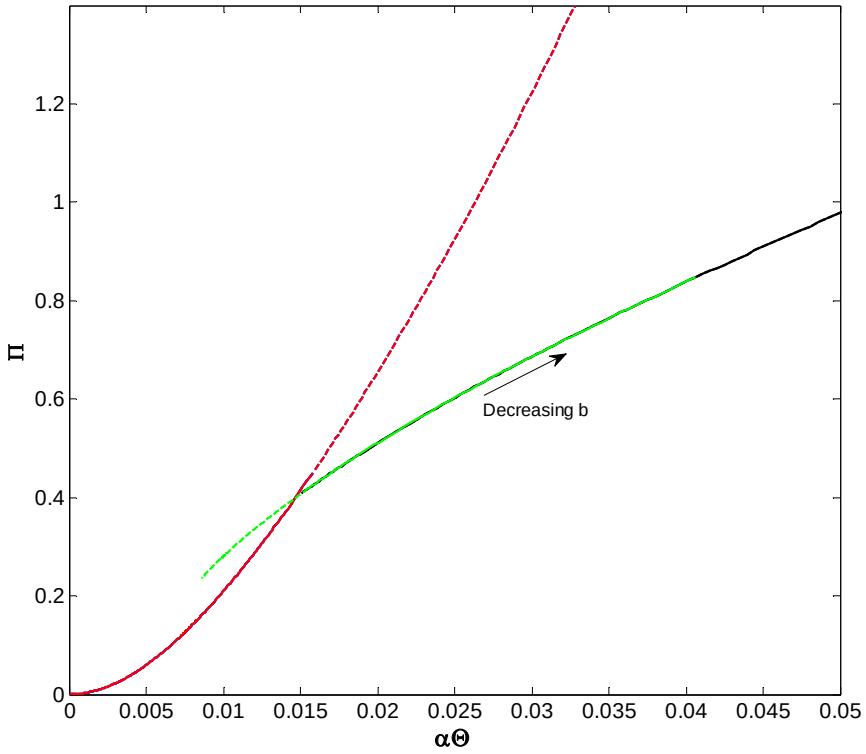

Figure 18b. Potential energy vs. applied temperature difference, clamped-fixed, a = 0.7, L_p = 0.9 partially debonded structure with full contact zone (red), fully lifted flaps (black), or intermediate contact zone (green). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

The partially debonded structure possesses a full contact zone until it reaches its predicted critical sling-shot buckling loading level. Upon buckling, it continues to possess a contact zone, albeit not a full one. As the load further increases, the contact zone propagates (i.e. b approaches a) until the flaps have completely lifted off the baseplate. We observe this type of behavior when the bond zone size is in the range $0.62 < a < 0.73$. Upon unloading, the structure follows the configurations denoted by the green path. For this particular case, sling back occurs at $\tilde{\Theta}_{cr(in)} < \tilde{\Theta}_{cr(cz)}$, which is similar to the scenario described in Section 5.2.

5.4 Example 3: a = 0.6

We now look at a structure whose bond zone size is a = 0.6 and patch size is L_p = 0.9. In the case of fully lifted flaps, the model predicts that $\tilde{\Theta}_{cr} = 0.0143$ and $N_{cr} = 34.2235$. For the structure which includes full contact, it is predicted that $\tilde{\Theta}_{cr(cz)} = 0.0156$ and $N_{cr(cz)} = 37.4650$. Observation of Figure 19a-b reveals the behavior for a partially debonded structure possessing either full or propagating contact.

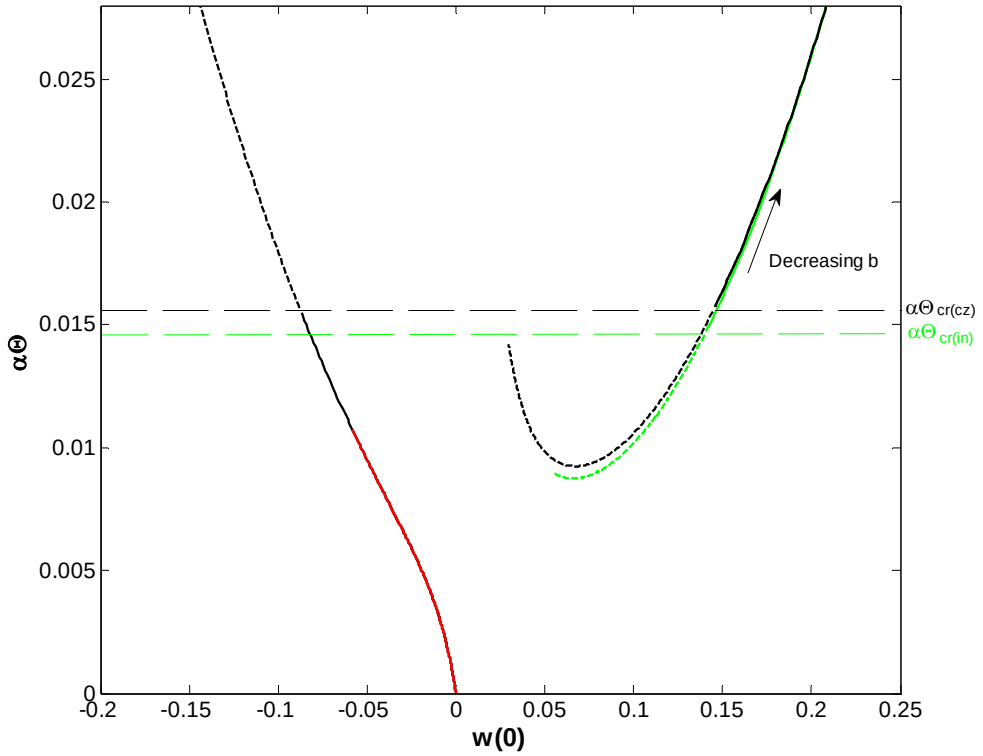

Figure 19a. Applied temperature difference vs. centerspan deflection, clamped-fixed, partially debonded structure with full contact zone, a = 0.6, L_p = 0.9. The physically valid portion is shown in red. The region of propagating contact is shown in green. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

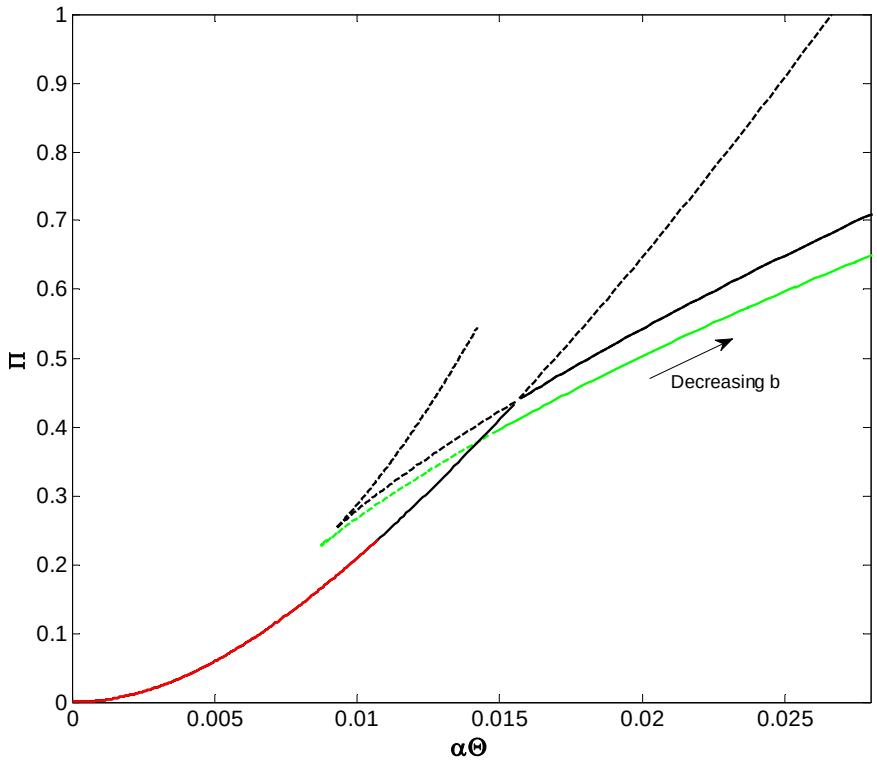

Figure 19b. Potential energy vs. applied temperature difference, clamped-fixed, partially debonded structure with full contact zone, a = 0.6, L_p = 0.9. The physically valid portion is shown in red. The region of propagating contact is shown in green. The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

A major difference (in comparison to the previously described examples) is noted. The full contact zone's existence does not remain valid up until the critical sling-shot buckling threshold is reached. Rather, the full contact zone is physically present until $\tilde{\Theta} = 0.0107$, at which point the flaps lift off, as indicated by the figure. This occurs because the pseudo inflection point shifts outside of the bond zone. Post-buckling, we find configurations corresponding to propagating contact, for temperature differences greater than an intermediate critical level of $\tilde{\Theta}_{cr(in)} = 0.0147$. The physically valid results corresponding to the partially debonded structure possessing either full contact or

propagating intermediate contact as well as those corresponding to the structure with fully lifted flaps are shown in Figure 20a-b.

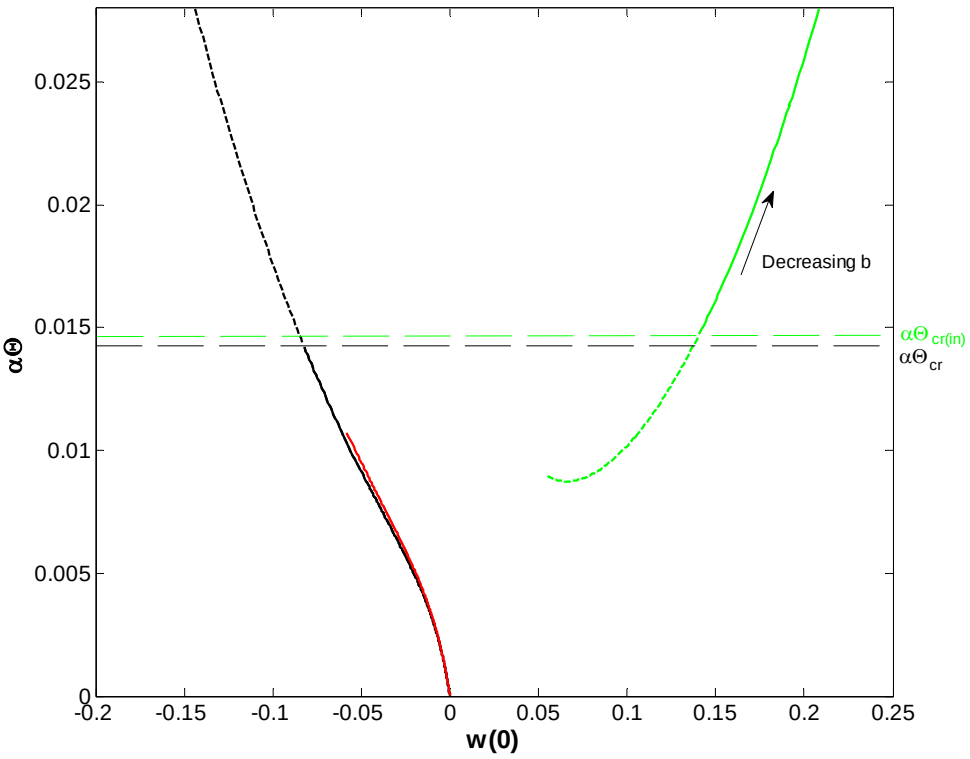

Figure 20a. Applied temperature difference vs. centerspan deflection, clamped-fixed, a = 0.6, L_p = 0.9 partially debonded structure with full contact zone (red), fully lifted flaps (black), or intermediate contact zone (green). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

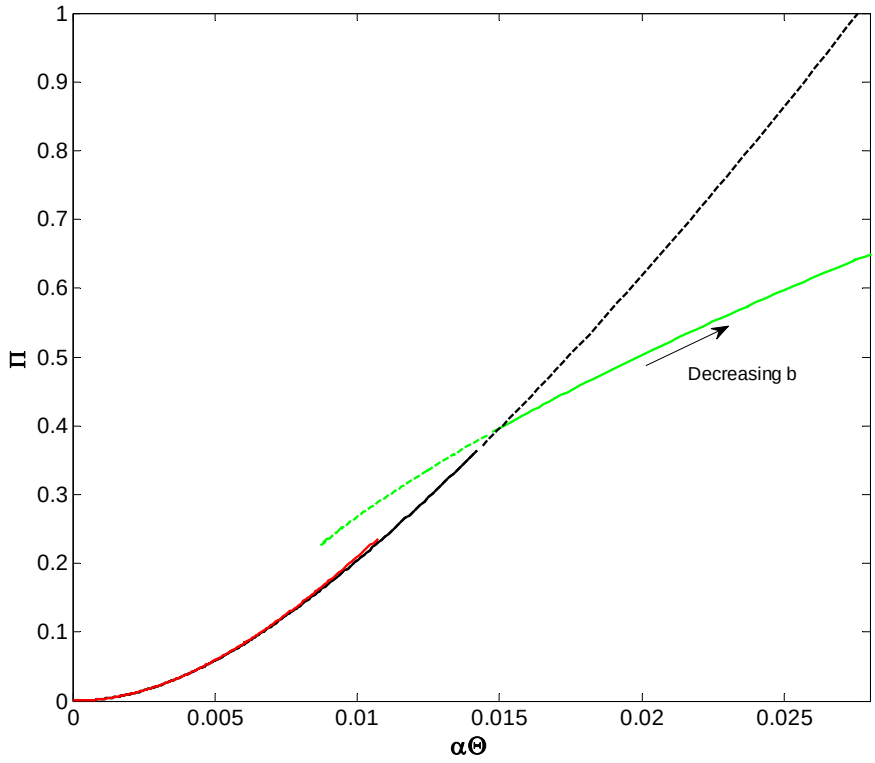

Figure 20b. Potential energy vs. applied temperature difference, clamped-fixed, a = 0.6, L_p = 0.9 partially debonded structure with full contact zone (red), fully lifted flaps (black), or intermediate contact zone (green). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

This bears a resemblance to what was observed for hinged structures. Once the flaps lift off, the structure is rendered more compliant, and hence the buckling threshold is lowered. Post-buckling, the flaps come into intermediate contact, and hence propagating contact occurs. Unlike the case described in the preceding section, however, there is no stable intermediate contact configuration after buckling at the predicted sling-shot buckling load level. Hence, as first described in Section 4.4, we observe the buckle-trapping phenomenon. A close up of Figure 20a is given in Figure 21.

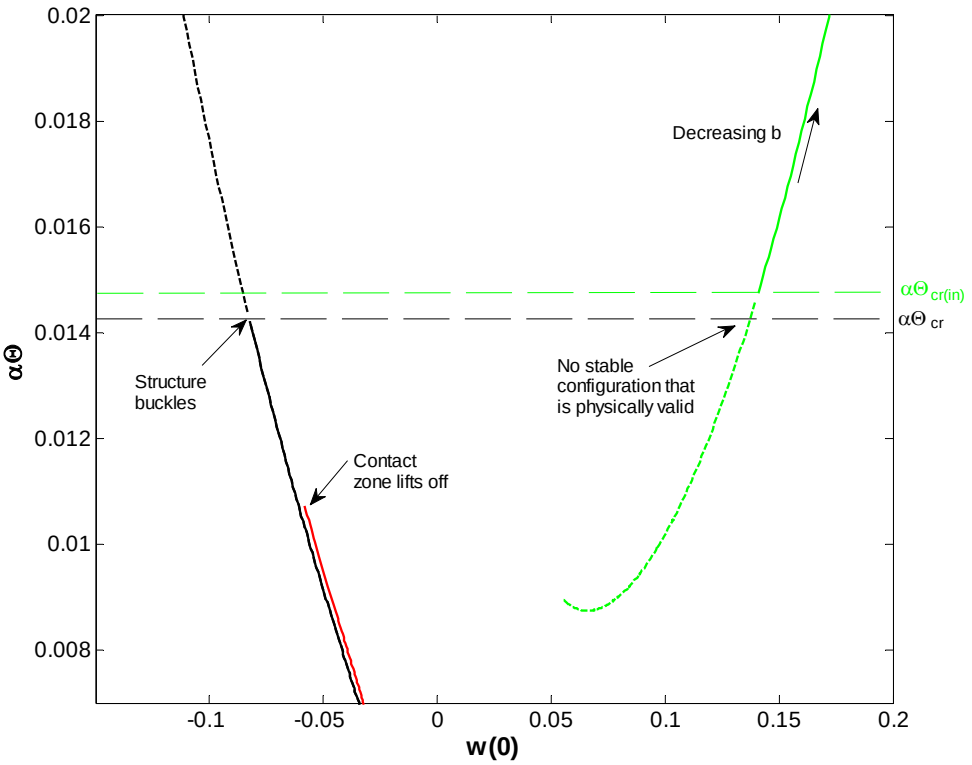

Figure 1. Close-up of applied temperature difference vs. centerspan deflection, clamped-fixed, *a* = 0.6, L_p = 0.9 partially debonded structure full contact zone (red), fully lifted flaps (black), or intermediate contact zone (green). The solid portion of the path represents stable equilibrium configurations, while the dashed portion of the path indicates unstable equilibrium configurations.

The buckle-trapping phenomenon is observed when the bond zone size is less than approximately $a = 0.63$. For larger bond zones, the overall structure is stiffer, preventing such behavior. As the bond zone size decreases, the structure becomes increasingly compliant, and the pseudo inflection point moves toward the supports. Hence, the observed buckling phenomena mimic that of the structure possessing hinged ends.

This concludes our assessment of sling-shot buckling behavior of a partially debonded patched structure under thermal loading. Two different edge support conditions were examined: hinged (in Chapter 4) and clamped, representing two

extremes. The structures are fixed with respect to in-plane translation. The hinged structure is seen to display a unique and unexpected phenomenon: buckle trapping. There is a duality to the structure, which experiences a dynamic change in stiffness when undergoing thermal buckling. The clamped structure exhibits a duality as well, but different behaviors, dependent on the bond zone size. When the bond zone size is small enough, the pseudo inflection point moves closer to the edges (and outside the bond zone). This causes a clamped structure to behave like its more compliant counterpart, a hinged structure. Thus, it exhibits the buckle-trapping phenomenon.

In Chapter 6 we discuss failure by debonding, and the manner in which it is coupled with thermal buckling.

Chapter 6

Edge Debonding Phenomena

6.1 Introduction

In addition to thermal buckling, separation failure (i.e. debonding) is a major issue with layered structures. The peeling apart of layers changes the overall stiffness of the structure, which can vastly affect desired performance. In particular, the debonding failure of an aircraft composite repair patch has devastating consequences. Complete separation of a panel during operation would significantly weaken the original structure to which it was adhered, rendering its structural integrity unacceptable.

Here we explore the coupled phenomena of edge debonding and thermal buckling. When the stiffness of the structure is compromised by edge debonding, it will directly affect the threshold for thermal buckling. In addition, a structure that buckles may experience an increase in energy release rate, which can lead to edge debonding. These are important considerations for design and use of such structures. Debonding phenomena is governed by the energy release rate, which mathematically arises from the vanishing of the first variation of the potential energy functional, as described in Section 2.4.5. Recall that the (Griffith-type) debonding criterion dictates that a structure will debond when the energy release rate exceeds the bond energy (a material and interface property). In the following section we discuss the general interpretation of the data to be presented in this chapter.

6.2 Interpretation

To assess debonding behavior, we compare the energy release rate against the conjugate bond zone size, *a**. Recall from Section 2.2, that conjugate bond zone size is defined as $a^* = 1 - a$, where *a* is the length of the bond zone. It is interpreted as a measure of the separation between the two layers: as *a** is increasing, the structure is debonding. Figure 22 shows a generic plot of energy release rate, *G*, vs. conjugate bond zone size, *a**. Each point on the path shown represents the energy release rate calculated for a structure at each particular bond zone size (or conjugate bond zone size) at a constant temperature. The paths are generated for several temperatures, to demonstrate thermal effects on debonding.

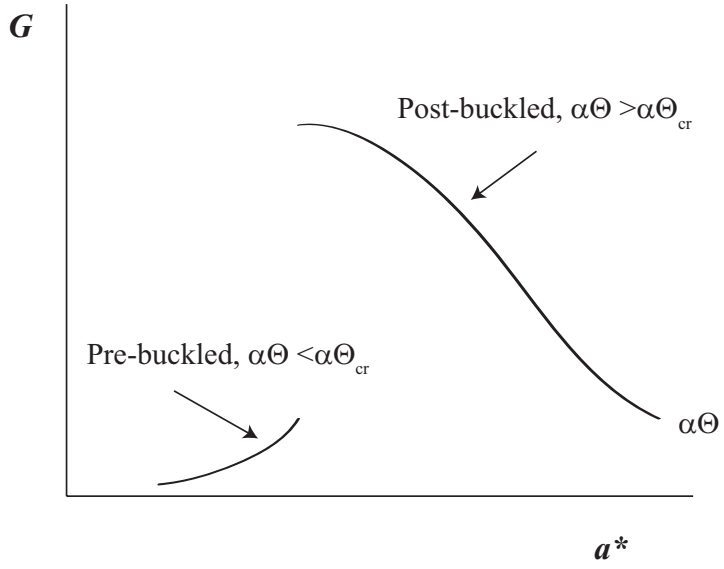

Figure 22. General schematic of *G* vs *a**

The figure shows a path for one constant temperature. Points lying on the portion of the path to the left of the discontinuity are identified as pre-buckled configurations, such that the temperature of the contour is less than the critical sling-shot buckling temperature for the bond zone sizes indicated. Points lying on the portion of the path to the right of the

discontinuity represent configurations where the temperature is greater than the critical sling-shot buckling temperature for those bond zone sizes. Hence, points on this path correspond to post-buckled configurations.

When the energy release rate exceeds 2γ, the structure will begin to separate. Figure 23 depicts an example of a possible debonding scenario.

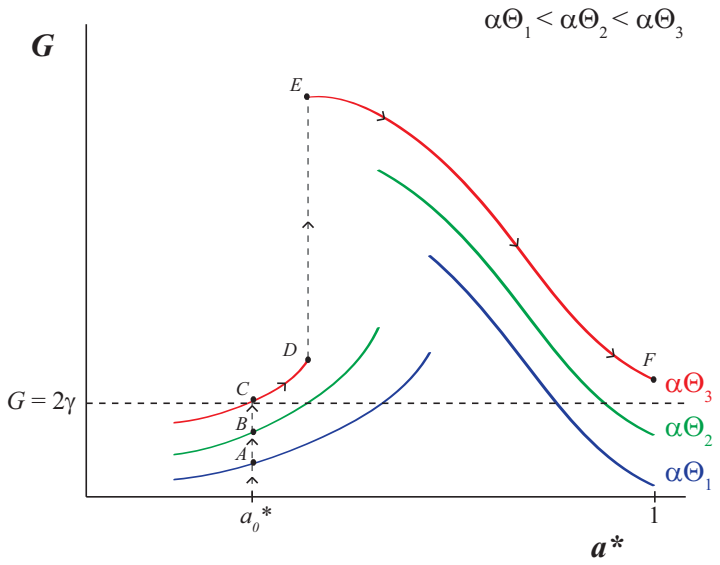

Figure 23. Sample catastrophic debonding scenario

The thermal load is increased on some structure possessing an initial bond zone size of a_0 (and hence an initial conjugate bond zone size of a_0*). The critical energy release rate is indicated by the horizontal dashed line. Three temperature differences are represented in the figure. As the temperature difference is increased from $\tilde{\Theta}_1$ (blue) to $\tilde{\Theta}_2$ (green) to $\tilde{\Theta}_3$ (red), the energy release rate is described by points *A*, *B*, and *C*, respectively. At point *C*, the temperature is such that the current energy release rate just exceeds the critical energy release rate, as per the debonding criterion. When $\tilde{\Theta}_3$ is achieved, the structure will debond in an unstable manner, and continue to follow the contour (increasing a*) until the energy release rate is less than or equal to the critical value. This

is referred to as *unstable debonding*, since it continues without any additional load. Beyond the conjugate bond zone size corresponding to point *D*, the structure becomes compliant enough, globally, to buckle at the prescribed temperature difference. Point *E* represents the counterpart of point *D* in the buckled configuration. The post-buckled structure continues to debond all the way to point *F*; at this point the entire patch has detached. This is referred to as *catastrophic debonding*. In this case, it is noted that debonding precipitates thermal buckling. That is, the debonding of the structure eventually reduces the overall stiffness enough so that the given temperature exceeds the critical sling-shot buckling temperature for the particular bond zone size.

Another possible debonding scenario is depicted in Figure 24.

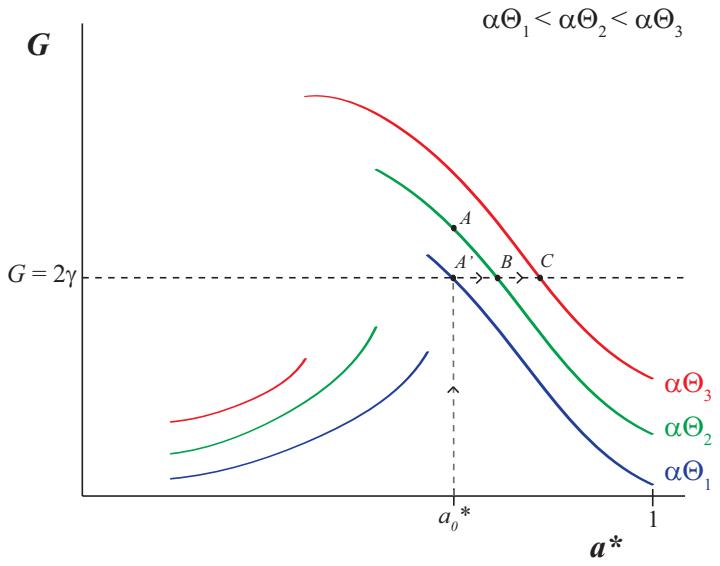

Figure 24. Sample stable debonding scenario

For example, the thermal load is increased, from zero, to $\tilde{\Theta}_2$ on a structure whose initial conjugate bond zone size is a_0*. The temperature contours on the right side of the figure represent post-buckled configurations, meaning that in this case, the structure buckles first, before the critical energy release rate is reached and any debonding takes place.

The structure does not reach the configuration at point *A*, which is associated with the final temperature $\tilde{\Theta}_2$. The structure will achieve the critical energy release rate when the temperature reaches $\tilde{\Theta}_1$, at point A'. As $\tilde{\Theta}$ is increased, incrementally, say to $\tilde{\Theta}_2$, the structure debonds (a^* increasing) to the conjugate bond zone size indicated by point *B*. If the temperature difference were further increased to $\tilde{\Theta}_3$ for example, the structure would debond again until reaching point *C*. An increment in temperature is required to generate an increment in debonding; hence we refer to this as *stable debonding*.

Such examples illustrate the interpretation of the data to be presented in the following section.

6.3 Results

6.3.1 Hinged Ends

Our presentation of the results pertaining to the structure with hinged ends begins with a look at the critical buckling temperature differences for each conjugate bond zone size, which is shown in Figure 25.

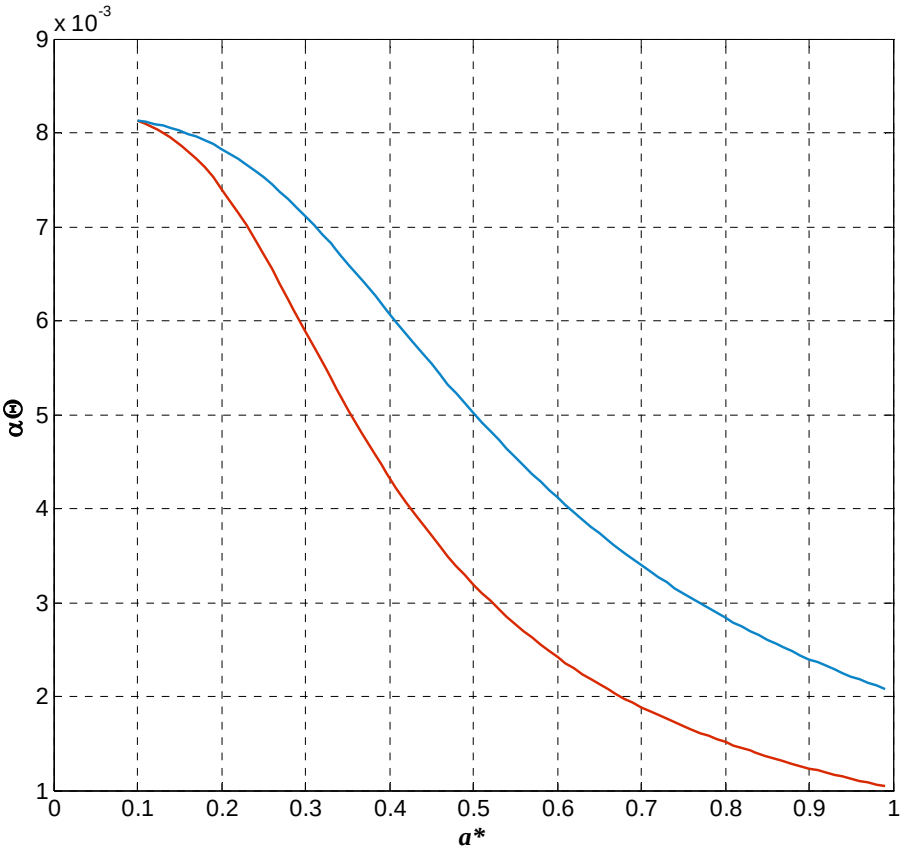

Figure 25. Critical buckling temperature differences, shown for conjugate bond zone sizes, hinged-fixed ends, L_p = 0.9. Red indicates the case of fully lifted flaps and blue indicates the case of full contact.

We consider a structure for which L_p = 0.9, $\alpha_p^0 = 0.5$, $h_0 = 1$, and E_0 = 1. The figure indicates the lowest buckling threshold for when the flaps are fully lifted (red), or when they are in full contact with the baseplate (blue). The latter is appropriately captured using the model for an equivalent perfectly intact structure of the same bond zone size. The threshold for the onset of sling-shot buckling is greater when a full contact zone is present, because it increases the overall stiffness in comparison to the case when the flaps have lifted off completely. The predicted buckling levels intersect at $a^* = 0.1$. This corresponds to $a = 0.9$, which equals the patch length for this example structure. As one

would expect, the critical level decreases significantly as the conjugate bond zone size increases, or as the patch and baseplate further separate. The increase in compliance lowers the critical threshold level.

For a partially debonded structure which is hinged at both of its ends, the flaps completely lift off the baseplate in pre-buckled configurations. Post-buckling, the flaps touch the baseplate, and hence a full contact zone is present. Figure 26 shows a plot of energy release rate vs. conjugate bond zone size, as described in Section 6.2. A few representative temperature contours are shown.

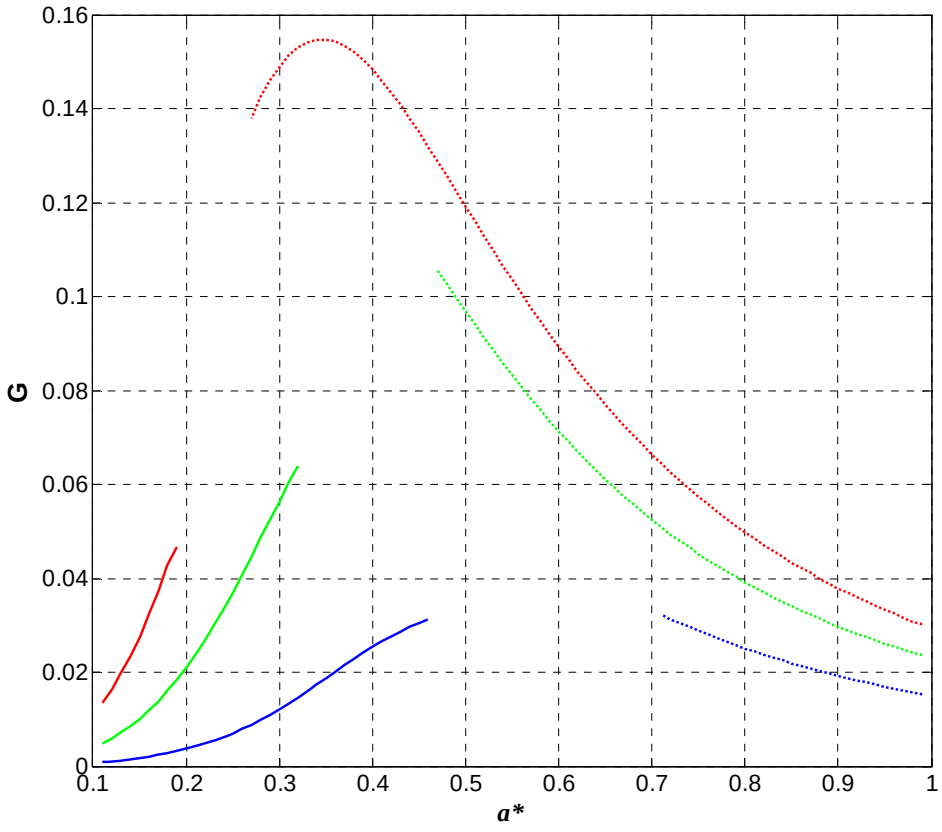

Figure 26. Energy release rate vs. conjugate bond zone size, hinged-fixed ends, $\alpha\Theta$ = 0.0036 (blue), $\alpha\Theta$ = 0.0055 (green), $\alpha\Theta$ = 0.0075 (red). Dotted lines represent post-buckled full contact zone configurations.

We can use the critical buckling levels given in Figure 25 to deduce if the points on Figure 26 represent pre- or post-buckled configurations. For hinged structures, because the flaps lift off initially upon loading, buckling occurs at the lower of the predicted thresholds. The solid lines depict the energy release rates of the structure when the flaps are fully lifted off. The dotted lines represent the energy release rates of the structure when it possesses a full contact zone. These paths exist where the presence of a full contact zone, for the given conditions, is physically appropriate. The apparent gap in domain is due to the different critical sling-shot buckling thresholds predicted for a structure with a contact zone, and for a structure where the flaps have fully lifted off. It is a manifestation of the buckle-trapping phenomenon, discussed in Section 4.4.

Based on the discussion from Section 6.2, it is evident from Figure 26 that for the pre-buckled structure, debonding occurs in an unstable fashion and leads to buckling. The extent and stability of the debonding depend on the critical energy release rate. For example, suppose a structure has a critical energy release rate of $G = 0.05$. If the initial conjugate bond zone size is $a_0^* = 0.28$ ($a_0 = 0.72$), then when $\tilde{\Theta} = 0.0055$ the current energy release rate is just enough to exceed the critical energy release rate. The structure debonds in an unstable manner along the (green) temperature contour until $a^* = 0.32$, at which point it buckles, and gets caught in the buckle-trap as it continues to debond. We cannot say conclusively the effect that buckle-trapping has on debonding, due to the effects of kinetic energy during the process. If the temperature is increased, say, to $\tilde{\Theta} = 0.0075$, the structure then follows the (red) dotted contour until $G = 0.05$, which corresponds to $a^* = 0.8$. At this point, the thermal load would have to be increased to

produce any further debonding. This is an example of unstable debonding followed by stable debonding.

6.3.2 Clamped Ends

Figure 27 shows the critical temperature differences for sling-shot buckling of a partially debonded clamped structure where the flaps are either fully lifted (red) or in full contact (blue).

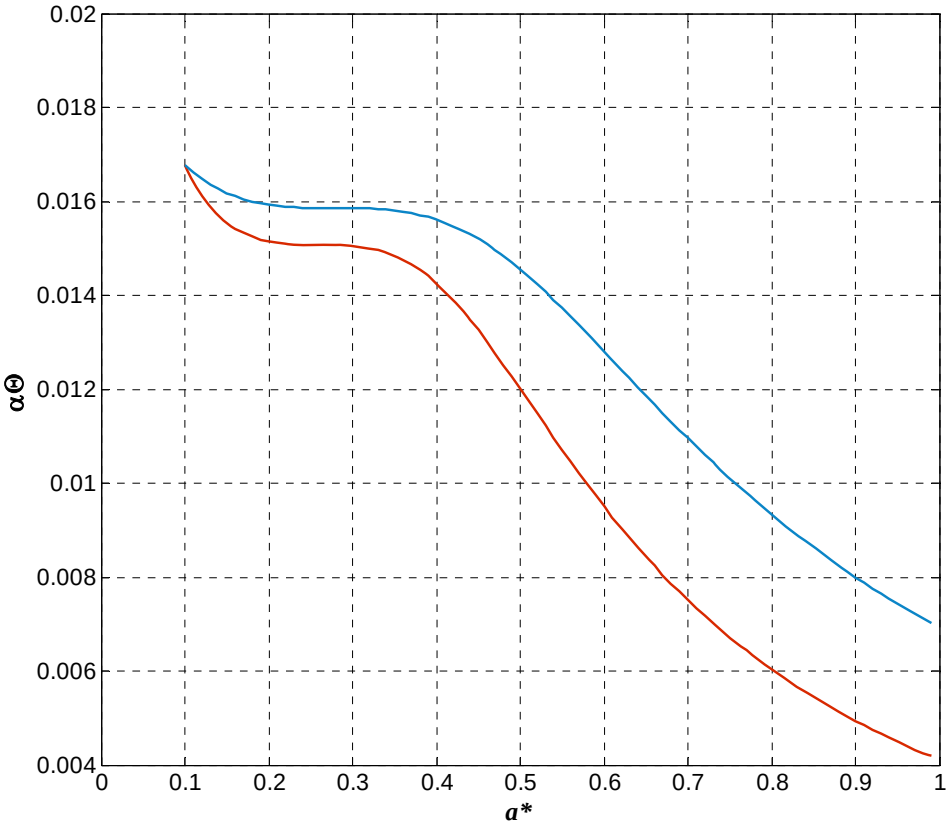

Figure 27. Critical buckling temperature differences, shown for conjugate bond zone sizes, clamped-fixed ends, L_p = 0.9. Red indicates the case of fully lifted flaps and blue indicates the case of full contact.

Like the corresponding figure for the hinged case, we see a significant decrease in the threshold for thermal buckling as the bond zone size becomes smaller, again showing that a less stiff structure will buckle at lower temperature differences. There is a noticeable

plateau when the conjugate bond zone size is small (or the bond zone size is longer than a = 0.7). When the bond zone size decreases, eventually, the pseudo inflection point moves outside of the bond zone. Hence, the unbonded portion of the baseplate (region S_3) becomes compliant enough such that more significant differences in critical buckling threshold exist.

Figure 28 shows the energy release rates vs. conjugate bond zone size for the clamped case for three representative temperature differences.

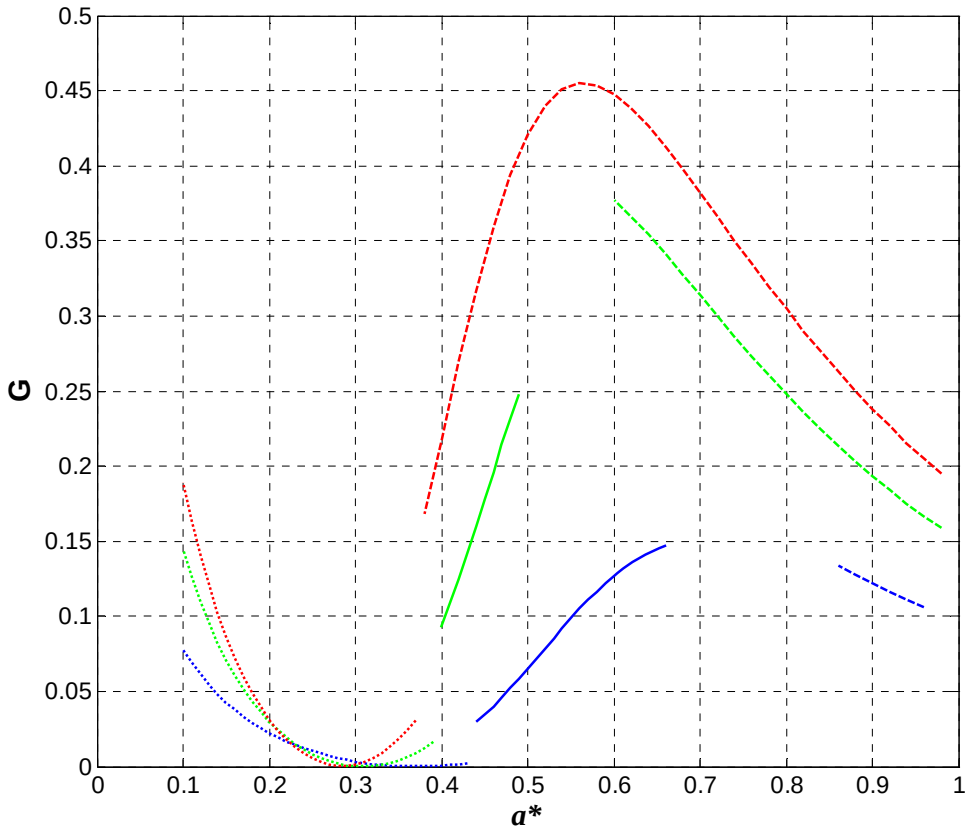

Figure 28. Energy release rate vs. conjugate bond zone size, clamped-fixed ends, $\alpha\Theta$ = 0.008 (blue), $\alpha\Theta$ = 0.012 (green), $\alpha\Theta$ = 0.015 (red). Dotted lines represent pre-buckled full contact zone configurations, and dashed lines represent post-buckled propagating contact configurations.

It may be determined via Figure 27 if a point on Figure 28 represents a pre- or post-buckled state. In Figure 28, the solid lines indicate where the flaps have lifted off the

baseplate, while dotted lines indicate where a full contact zone is present. Whenever present, the structure assumes the configurations represented by the dotted lines, rather than the solid. For pre-buckled configurations, at a certain temperature difference, when a dotted path is not present, the structure would follow the corresponding solid path until experiencing sling-shot buckling. The dashed lines represent post-buckled propagating contact zone configurations, where some (but not all) of the patch remains in contact with the baseplate. The gap (in a^*) that may exist at a certain temperature difference, between the last pre-buckled configuration and the first post-buckled (propagating contact) configuration, again manifests where the buckle-trapping phenomenon occurs.

Debonding is observed to be stable for a range of large patch sizes similar to where the plateau appears in Figure 27. As the structure debonds, the pseudo inflection moves toward the fixed edge and the size of region S_3 increases. The structure becomes more compliant near the edges and hence behaves more like a pre-buckled hinged-fixed structure. Many debonding scenarios are possible, depending on the given parameters. Suppose there is a structure that has an initial conjugate bond zone size of $a_0^* = 0.15$. If the critical energy release rate is $G = 0.1$, and the structure is loaded to $\tilde{\Theta} = 0.015$, it will not debond. The contact zone's presence essentially raises the temperature threshold for the onset of debonding.

For another example, suppose for some structure where $a_0^* = 0.44$, the critical energy release rate is $G = 0.3$. The contact zone has lifted off by the time the temperature difference passes through $\tilde{\Theta} = 0.008$ (blue) and $\tilde{\Theta} = 0.012$ (green). If the temperature difference is increased to $\tilde{\Theta} = 0.015$, the structure will have buckled to a configuration with an intermediate contact zone, and the energy release rate will become just enough to

exceed the critical level. The structure will debond and follow the dashed red contour in an unstable manner, until $a^* = 0.8$. Beyond this point, the debonding is stable.

With knowledge of the initial bond zone size, as well as the bond strength, we can predict the onset, extent, and stability of debonding, for a system in which the temperature difference is controlled. We can also predict the sequence of phenomena (thermal buckling and edge debonding) for the evolving structure. In Chapter 7, we present concluding remarks on the findings of this work.

Chapter 7

Concluding Remarks

7.1 Overview

The prevalence of the patched structure in modern engineering technology requires further understanding of failure phenomena in order to be able to determine effective and safe usage. The goal of this research has been to characterize thermal buckling and edge debonding phenomena, and their interaction, for a layered beam-plate structure subjected to a uniform temperature field. The analytical model for the problem has been developed by Bottega and Carabetta (2009), and was reproduced for this work. The problem is formulated from first principles, using a thin structure theory to model the constituent substructures, and begins with construction of a potential energy functional. The Theorem of Stationary Potential Energy states that equilibrium configurations occur at stationary points of the potential energy functional. That is, when the first variation of the potential energy functional vanishes. The formulation allows for the boundaries of each domain of the layered structure to vary, in addition to the transverse and in-plane displacements. The appropriate variations are taken, and the Lagrange multipliers are eliminated. From there, the governing differential equations, constitutive relations, boundary/matching conditions, and transversality equations can be extracted. The problem is restated in a mixed formulation, where the analytical closed-form solution is expressed in terms of the transverse deflection and a (numerically) to-be-determined membrane force, rather than the in-plane deflection. This allows for mathematical ease, as well as physical interpretation. Hence, it is necessary to establish compatibility

(enforce continuity) of the displacements. This is achieved by integrating the strain-displacement relations and adding the resulting expressions along each domain to generate, what is referred to as an *integrability condition*. To check stability of an equilibrium configuration, the second variation of the potential energy functional is tested for positive-definiteness. The mathematical model allows for the possibility of a contact zone, a region where the layers are not bonded, but remain in sliding contact. A check on the curvatures is used to determine the physical validity of this proposition. By performing numerical simulations based on the analytical model, pertinent data may be computed and interpreted.

7.2 On Thermal Buckling

A thermal buckling analysis for a perfectly intact structure was performed by Karlsson and Bottega (2000). In the present work, we consider a layered structure that is partially debonded at the interface between the edges of the patch and the baseplate. Two support cases are considered, both of which are fixed with respect to in-plane translation. The first is hinged (at both ends) with respect to rotation, and the second is clamped (at both ends) with respect to rotation.

For the hinged scenario it is seen that when the structure (for the given parameter values) initially deflects up, the unbonded portion of the patch, i.e. the flaps, will lift off of the baseplate. Energetically and structurally, this is equivalent to the intact model studied by Karlsson and Bottega (2000), for the same bond zone size. Hence that model is used to capture the salient behavior. The structure will then experience sling-shot buckling when the temperature difference reaches the critical value predicted by the equivalent model. This critical level is less than the corresponding predicted level if the

structure were to artificially possess full contact. When the hinged structure deflects downward, the flaps come into contact with the baseplate. The structure is essentially two different structures, depending on which way it deflects. When deflected down, the structure possesses a full contact zone, and when deflected up, the flaps lift off. This *dual nature* cannot be captured by either of the two cases (fully lifted flaps or full contact) alone, which are computed independently. Rather, we must infer the behavior at the critical buckling threshold. The structure buckles, and experiences a change in stiffness as it passes dynamically through the flat configuration. There is no stable equilibrium configuration for it to "land" in. For a structure which now has a full contact zone, at the particular load level, the predicted stable configuration is an upward deflection. So the structure's deflection reverses direction, and it dynamically moves to an upward configuration. However, upon passing through the flat position, the stiffness is once again altered as the flaps lift off. But the structure with lifted flaps "prefers" to be buckled to a downward deflected configuration. The result is an alternating process, where the structure deflects around the flat position, which is the only configuration where both cases can exist simultaneously. We call this "buckle-trapping," because rather than completely buckling, the structure becomes trapped between two unstable configurations. For an actual structure there will be some degree of structural damping and it is postulated that, ultimately, the structure would settle at the flat position. This would be due to an energy "cusp" at zero deflection for the piecewise defined system.

Thermal buckling phenomena for the clamped case depend largely on the bond zone size. For the largest bond zone sizes, a full contact zone is present while the structure deflects upward, all the way until the critical thermal load is reached. The

structure buckles at the threshold level predicted by the model for a partially debonded structure with a full contact zone. In this case, upon buckling to a downward deflection, the structure bends in a way that the flaps fully lift off. There is a dual nature; however, at the critical sling-shot buckling threshold, there is a physically valid stable equilibrium configuration available for the structure. Hence, buckle-trapping does not occur. Alternatively, for slightly smaller bond zone sizes, the full contact zone is valid pre-buckling, and upon buckling the structure assumes a configuration of propagating contact. This means that a portion of the flap is in contact with the baseplate, but not the entire flap. As the loading continues to increase, the contact zone becomes smaller, until the flaps fully lift off. Finally, for smaller bond zone sizes, the full contact zone actually lifts off before the structure buckles. Thus, the structure buckles at a lower predicted level. The post-buckled configuration is one of propagating contact. However, upon buckling, there is no physically valid stable intermediate contact zone equilibrium configuration. This means that we again observe the buckle-trapping phenomenon. A less stiff clamped specimen experiences similar behavior to the hinged specimen.

These findings have important implications on design as well as the possible use of patched structures.

7.3 On Debonding

To examine debonding behavior, we evaluate the energy release rates at each point on a load-deflection path of the structure. By plotting energy release rates for each conjugate bond zone size, at constant temperature values, we can study the evolution of the structure as we control the thermal load.

The strength of the bond greatly affects the potential buckling-debonding interaction scenarios that may play out. Debonding failure occurs when the energy release rate reaches a critical value. This means a stronger bond increases the debonding threshold load level, which is intuitive.

For a hinged structure, we observe unstable debonding in pre-buckled structures, and possible stable debonding in post-buckled structures. There is potential that upon loading to a certain level, the critical energy release rate is reached, which causes delamination. As the structure debonds, it becomes more compliant, which lowers the corresponding critical buckling temperature difference. Thus, while holding a structure at a constant temperature, when it begins to debond, it will reach a point where the current temperature exceeds the critical buckling temperature. Debonding precipitates buckling in this scenario. It is also seen that when a structure buckles, the energy release rate increases, which leads to possible debonding.

We can also predict the evolution of a clamped patched structure under given parameter values. The presence of a contact zone stiffens the structure and hence allows the structure to withstand a higher temperature difference before the onset of debonding. The existence of a propagating contact zone for post-buckled structures also affects the energy release rate.

Examination of the delamination mode mix will provide additional insight into the mechanism of debonding. When a contact zone is present, either intermediate or full, the debonding is pure mode-II (sliding), which corresponds to a higher threshold for the critical energy release rate from the debonding criterion. Thus, the current results, in the presence of any contact zone, offer a more conservative estimate for the onset of

debonding. When the contact zone is vanishing, the debonding mode is generally mixed. Increasing the temperature difference can cause a change in the ratio of mode-II to mode-I (opening). The fracture toughness at the interface, which governs the onset of debonding, is dependent on the mode mix. The mode mix can be varying as a^* or $\tilde{\Theta}$ vary. Thus, the critical threshold for the onset of debonding could be envisioned as a curve, rather than a flat line, across the plots of G vs a^* shown in Figures 23 and 24. This suggests that a structure may be able to withstand a higher temperature difference before debonding occurs, and hence the overall evolution and arrest may vary accordingly. For example, a hinged structure undergoes sling-shot buckling as a result of the debonding. This engages the patch flaps and thus a contact zone is present, which means that pure mode-II is the mechanism of separation. The critical energy release rate has thus increased, and hence debonding may cease, pending additional loading.

7.4 Future Considerations

The failure phenomena of a patched structure are of great importance from a design and analysis standpoint. This work elucidates the complex relationship between thermal buckling and edge debonding. It also reveals unique, unexpected behavior in terms of thermal buckling of a partially debonded structure. Such a finding warrants further pursuit.

The buckle-trapping phenomenon is a perceived dynamic effect. The structure alternates about an “equilibrium” flat configuration. In this work, a static analysis is performed. A dynamic analysis of the response of the structure to a thermal load would reveal more about the behavior. It would be especially informative to be able to “visualize” the potential energy “cusp” that exists for the piecewise defined functional.

Also helpful to better understanding of the observations noted in this work would be experimental evaluation.

Our formulation begins with use of a thin-structure theory, which is suitable for the needs of the dissertation. However, using a more sophisticated elastic theory containing a shear correction may further expand on this work. In addition the ramifications of the delamination mode mix can be considered. Another complication that may be explored would be consideration of thermally varying material properties, and their impact on the buckling/debonding behavior.

This overall work serves as a significant step toward the better fundamental understanding of an important engineering structural problem.

References

Alderliesten, R.C., Schijve, J, van der Zwaag, S. (2006) Application of the energy release rate approach for delamination growth in Glare, *Engineering Fracture Mechanics*, **73**, 697 – 709.

Aydogdu, M. (2007) Thermal buckling analysis of cross-ply laminated composite beams with general boundary conditions, *Composites Science and Technology*, **67**, 1096 – 1104.

Baker, A., Rajic, N., Davis, C. (2009) Towards a practical structural health monitoring technology for patched cracks in aircraft structure, *Composites: Part A*, **40**, 1340 – 1352.

Bigwood, D.A. and Crocombe, A.D. (1989), Elastic analysis and engineering design formulae for bonded joints, *International Journal of Adhesion and Adhesives*, **9** (4), 229 – 242.

Bottega, W.J. (1983) A growth law for propagation of arbitrary shaped delaminations in layered plates, *Int. J. Solids Structures*, **19** (11), 1009 – 1017.

Bottega, W.J. (1995) Separation failure in a class of bonded plates, *Composite Structures*, **30**, 253 – 269.

Bottega, W.J. (2003) Structural scale decomposition of energy release rates for delamination propagation, *Int. J. Fracture*, **122**, 89 – 100.

Bottega, W.J. (2006) Sling-shot buckling of composite structures under thermo-mechanical loading, *Int. J. Mechanical Structures*, **48**, 568 – 578.

Bottega, W.J, Carabetta, P.M. (2009) On the detachment of patched panels under thermomechanical loading, *JoMMS*, **4** (7-8), 1227 – 1250.

Byrd, L.W, Birman, V. (2006) Effect of temperature on stresses and delamination failure of z-pinned joints, *Int. J. Mechanical Sciences*, **48**, 938 – 949.

Carabetta, P.M., Bottega, W.J. (2008) Effects of geometric nonlinearities on damage propagation in patched beam-plates subjected to pressure loading, *Int. J. Fracture*, **152**, 51 – 62.

Carabetta, P.M. (2007) Effects of geometric nonlinearities and uniform temperature fields on the detachment of patched beam-plates under pressure loading. MS Thesis, Rutgers, the State University of New Jersey.

Comiez, J.M., Waas, A.M, Shawan, K.W. (1995) Delamination buckling; experiment and analysis, *Int. J. Solids and Structures*, **32** (6/7), 767 – 782.

Davies, G.A.O., Hitchings, D., Ankersen, J. (2006) Predicting delamination and debonding in modern aerospace composite structures, *Composites Science and Technology*, **66**, 846 – 854.

Delale, F. and Erdogan, F. (1981), Stresses in adhesively bonded joints: a closed-form solution, *Journal of Composite Material*, **15**, 249 – 271.

Gelfand, I.M, Fomin, S.V. (1963) *Calculus of variations*, Prentice Hall, Englewood Cliffs, New Jersey.

Goland, M. and Reissner, E. (1944), The stresses in cemented joints, *Journal of Applied Mechanics, Transactions of the ASME*, **66**, A17 – A27.

Greenberg, M.D. (1998), *Advanced engineering mathematics*, Prentice Hall, Englewood Cliffs, New Jersey.

Griffith, A.A. (1920) The phenomena of rupture and flow in solids, *Philosophical Transactions of the Royal Society of London. Series A*, **22B,** 163 – 198.

Hsueh, C., Lee, S., Lin, H. (2006) Analysis of mode I edge delamination by thermal stresses in multilayer systems, *Composites: Part B*, **37**, 1 – 9.

Johnson, W.S. (1987), Stress analysis of the cracked-lap shear specimen: an ASTM round robin, *Journal of Testing and Evaluation*, **6**, 303–324.

Karlsson, A.M. (1999) Aspects of the Failure of Patched Structures. PhD Dissertation, Rutgers, the State University of New Jersey.

Karlsson, A.M., Bottega, W.J. (1999) The Presence of Edge Contact and Its Influence on the Debonding of Patched Panels, *Int. J. Fracture*, **96**, 381 – 404.

Karlsson, A.M., Bottega, W.J. (2000) On Thermal Buckling of Patched Beam-Plates, *Int. J. Solids and Structures*, **37**, 4655 – 4690.

Lee, J.J., Oh, I., Lee, I., Yeom, C.H. (2002) Thermal post-buckling behavior of patched laminated panels under uniform and non-uniform temperature distributions, *Composite Structures*, **55**, 137 – 145.

Moore, T.D. (2005) Thermomechanical peeling in multilayer beams and plates – a solution from first principles, *Int. J. Solids and Structures*, **42**, 271 – 285.

Naboulsi, S., Mall, S. (1997) Thermal effects on adhesively bonded composite repair of cracked aluminum panels, *Theoretical and Applied Fracture Mechanics*, **26**, 1 – 12.

Özben, T. (2009) Analysis of critical buckling load of laminated composites plate with different boundary conditions using FEM and analytical methods, *Computational Materials Science*, **45**, 1006 – 1015.

Pi, Y., Bradford, M.A. (2008) Thermoelastic lateral-torsional buckling of fixed slender beams under linear temperature gradient, *Int. J. Mechanical Sciences*, **50**, 1183 – 1193.

Rabinovitch, O. (2010) Impact of thermal loads on interfacial debonding in FRP strengthened beams, *Int. J. Solids and Structures*, **47**, 3234 – 3244.

Reddy, J.N. (2000) Analysis of functionally graded plates, *Int. J. Numer. Methods. Eng.*, **47**, 663 – 684.

Rutgerson, S.E. (2001) Non-Linear Thermo-Mechanical Response of Bilaminate Composite Shell Structures. MS Thesis, Rutgers, the State University of New Jersey.

Rutgerson, S.E., Bottega, W.J. (2002) Thermo-Elastic Buckling of Layered Shell Segments, *Int. J. Solids and Structures*, **39**, 4867 – 4887.

Rutgerson, S.E., Bottega, W.J. (2004) Pre-Limit Point Buckling of Multilayer Cylindrical Panels Under Pressure, *AIAA Journal*, **42** (6), 1272 – 1275.

Song, X., Li, S. (2007) Thermal buckling and post-buckling of pinned-fixed Euler-Bernoulli beams on an elastic foundation, *Mechanics Research Communications*, **34**, 164 – 171.

Timoshenko, S. (1925) Analysis of bi-metal thermostats, *J. Opt. Soc. Am.*, **11** (3), 233 – 255.

Williams, J.G. (1988) On the calculation of energy release rates for cracker laminates, *Int. J. Fracture*, **36**, 101 – 119.

Wittrick, W.H. (1953) Stability of a bimetallic disk, Part I. *Quart. J. Mechs. And Appl. Math*, **6**, 15 – 26.

Wittrick, W.H., Myers, D.M., Blunden, W.R. (1953) Stability of a bimetallic disk, Part II. *Quart. J. Mechs. And Appl. Math*, **6**, 26 – 31.

Wu, L.C., Lo, C.Y., Nakamura, T., Kushner, A. (1998) Identifying failure mechanisms of composite structures under compressive load, *Int. J. Solids and Structures*, **35** (12), 1137 – 1161.

Yang, J., Liew, K.M., Wu, Y.F., Kitipornchai, S. (2006) Thermo-mechanical post-buckling of FGM cylindrical panels with temperature-dependent properties, *Int. J. Solids and Structures*, **43**, 307 – 324.

Yin, W. (1998) Thermoelastic postbuckling response of strip delamination models, *Int. J. Solids and Structures*, **35** (25), 3331 – 3346.

Zenkour, A.M., Sobhy, M. (2010) Thermal buckling of various types of FGM sandwich plates, *Composite Structures*, **93**, 93 – 102.

Appendix

The appendix presents a listing of the relevant MATLAB and Maple code created to generate the results.

A.1 MATLAB code

For model of intact structure:

```
% Separation of intact Patched Structure
% Patched Plate, thermal load, NO PRESSURE
% Non-Linear  COMPRESSION!!!  NO CZ
% This is the "theta" version, also the Buckling Version
%The code works for Clamped-Fixed ends or Hinged-fixed ends.
%All I need to do is add a _HF for the Hinged case to the Big Functions!
%There are N I N E places this needs to be done
clear
global h C D Dp Cp
h = 0.05; % height of baseplate
hp = 0.05; % height of patch
h0 = hp/h; % ratio of heights
E0 = 1; % elastic modulus
C = 12/(h^2); % membrane stiffness of baseplate
Cp = C*E0*h0; % membrane stiffness of patch
D = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rho Ds
% stiffnesses of composite structure
Astar = D+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rho = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rho*Bstar; %Dstar
global  Dc Cs Ce
Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar nstar etatilda alphaP
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 0.5; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rho*nstar;
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
%Set 2*gamma, try a few values
%gamma_vals = [0.01/2,0.05/2,0.1/2,0.2/2];
global gamma a astar
gamma = 0.1/2;
%Choose values for a and N
a_vals = (0.01:0.01:0.90);
a_vals_star = 1 - a_vals;
N = (0:0.01:40);
%for gin = 1:length(gamma_vals)
    %gamma = gamma_vals(gin);
    m=1; %initialize index value for result array
    n=1;
    nn = 1;
%for i = 1:length(a_vals);
%    a = a_vals(i);
%    astar = 1-a_vals(i);
%    for j = 1:length(N);
```

```
%        [Big(j,1),Big(j,2),Big(j,3),Big(j,4), Big(j,5)] = BigFun_B01(N(j)); %1 Or _HF
%        %[value of big F, Theta, H, PE]
%    end
   % for k = 1:length(N)-1;
   %     %This is for DEBONDING, look for BF=0
   %     if Big(k,1)*Big(k+1,1)<0;
   %         Nroot(k) = fzero(@BigFun_B01_root,N(k)); %2 Or _HF
   %         if Nroot(k)<max(N)+1 & Nroot(k)>min(N)-1;
   %         Roots4N(m) = Nroot(k); % The value of N that's a root
   %         m = m+1;
   %         else
   %         end
   %     else
   %     end
   % end
   %     %Okay, let's try something...all the results
   %     for ii = 1:length(Roots4N)
   %         TestResult(nn,1) = astar;
   %         TestResult(nn,2) = Roots4N(ii);
   %         [crud,TestResult(nn,3),crud2,TestResult(nn,4), TestResult(nn,5)] =
BigFun_B01(Roots4N(ii)); %3 Or _HF
   %         nn = nn+1;
   %     end
   %     %Now, pluck the lowest roots from the lot! Results for lowest N
   %     N_lowest = min(Roots4N);
   %     %PE_lowest = min(TestResult(:,5));
   %     m = 1; %Reset now that I'm done with it for this round!
   %     [blah,group(m,2),blah2,group(m,3),group(m,4)] = BigFun_B01(N_lowest); %4 Or _HF
   %     %[value of big F (should be zero), Theta, H,W0]
   %        %To assemble a matix of nice results...
   %         Results(n,1)= astar; %a*
   %         Results(n,2) = N_lowest; %N
   %         Results(n,3) = group(m,2); %Theta
   %         Results(n,4) = group(m,3); %W0
   %         Results(n,5) = group(m,4); %PE
   %         Results(n,6) = 2*gamma;
   %         n=n+1;

%end
%end
%Separate loops for Buckling
q=1;
for p = 1:length(a_vals);
    a = a_vals(p);
    astar = 1-a_vals(p);
    for j = 1:length(N);
        [Big(j,1),Big(j,2),Big(j,3),Big(j,4), Big(j,5)] = BigFun_B01(N(j)); %5 Or _HF
        %[value of big F, Theta, H]
    end
     for r = 1:length(N)-1;
        %This is for Critical BUCKLING, look for H=0
        if Big(r,3)*Big(r+1,3)<0;
            Ncrit(r) = fzero(@BigFun_B01_rootH,N(r));    %6 Or _HF     %The critical N
value
            Theta_crit(r) = -(rho+(1/2)*h)*Ncrit(r)/mstar; %Critical Theta
                BuckleResults(q,1)= astar;
                BuckleResults(q,2)= Ncrit(r);
                BuckleResults(q,3) = Theta_crit(r); %/.002; %to compare to Bottega's
result, Also change in BigFunPregrowth
                BuckleResults(q,4)= a;
                q=q+1;
        end
     end
end
%Pregrowth
Npg = (0.1:0.1:80);
%Allocate space for arrays
ResultsPG = zeros(length(Npg),7,length(a_vals));
StablePG = zeros(length(Npg),7,length(a_vals));
ResultsPG2 = zeros(length(Npg),7,length(a_vals));
StablePG2 = zeros(length(Npg),7,length(a_vals));
```

```
for nn=1:length(a_vals);
ain=a_vals(nn);
ainstar = 1-ain;
mm=1; %Reset, not really needed
for mm = 1:length(Npg);%so the format of Results PG is [N,theta,F,w0,ERR,PE, 3rd minor]
     ResultsPG(mm,1,nn)=Npg(mm); %N Root 1
     [ResultsPG(mm,2,nn),
ResultsPG(mm,3,nn),ResultsPG(mm,4,nn),ResultsPG(mm,5,nn),ResultsPG(mm,6,nn),ResultsPG(mm,
7,nn)] = BigFun_PreGrowth(ain,ainstar,Npg(mm)); %7 Or _HF

     if ResultsPG(mm,3,nn)>0;
         StablePG(mm,:,nn)=ResultsPG(mm,:,nn);
     end
     %New
     %if ResultsPG(mm,7,nn)>0;
     %    NStablePG(mm,:,nn)=ResultsPG(mm,:,nn);
     %end

     ResultsPG2(mm,1,nn)=Npg(mm); %N  Root 2
     [ResultsPG2(mm,2,nn),
ResultsPG2(mm,3,nn),ResultsPG2(mm,4,nn),ResultsPG2(mm,5,nn),ResultsPG2(mm,6,nn),ResultsPG
2(mm,7,nn)] = BigFun_PreGrowth2(ain,ainstar,Npg(mm)); %8 Or _HF

     if ResultsPG2(mm,3,nn)>0;
         StablePG2(mm,:,nn)=ResultsPG2(mm,:,nn);
     end
     %New
     %if ResultsPG2(mm,7,nn)>0;
     %    NStablePG2(mm,:,nn)=ResultsPG2(mm,:,nn);
     %end

end
end
%-----------------------------------------------------------
%Save the data with a new name for the COMBO program 4/6/10
%CHANGE to "_c" for clamped, "_h" for hinged!

%save NOCZRes_c.mat ResultsPG;
%save NOCZRes2_c.mat ResultsPG2;
%save NOCZStabRes_c.mat StablePG;
%save NOCZStabRes2_c.mat StablePG2;
%save NOCZCritBuck_c.mat BuckleResults;
%-----------------------------------------------------------
%Trying something new...
%load=50;
%for ugh=1:length(a_vals_star)
%    ERRplotPG(ugh)=ResultsPG(load,5,ugh);
%    ERRplotPG2(ugh)=ResultsPG(load,5,ugh);
%    ERRplotStab(ugh)=ResultsPG(load,5,ugh);
%    ERRplotStab2(ugh)=ResultsPG(load,5,ugh);
%end
%load2=100;
%for ugh=1:length(a_vals_star)
%    ERRplotPGl2(ugh)=ResultsPG(load2,5,ugh);
%    ERRplotPG2l2(ugh)=ResultsPG(load2,5,ugh);
%    ERRplotStabl2(ugh)=ResultsPG(load2,5,ugh);
%    ERRplotStab2l2(ugh)=ResultsPG(load2,5,ugh);
%end
%figure
%plot(a_vals_star,ERRplotPG,'k.','LineWidth',2)
%hold on
%plot(a_vals_star,ERRplotPG2,'k.','LineWidth',2)
%hold on
%plot(a_vals_star,ERRplotStab,'r.','LineWidth',2)
%hold on
%plot(a_vals_star,ERRplotStab2,'r.','LineWidth',2)
%hold on
%plot(a_vals_star,ERRplotPGl2,'b.','LineWidth',2)
%hold on
%plot(a_vals_star,ERRplotPG2l2,'b.','LineWidth',2)
%hold on
```

```
%plot(a_vals_star,ERRplotStabl2,'m.','LineWidth',2)
%hold on
%plot(a_vals_star,ERRplotStab2l2,'m.','LineWidth',2)
%xlabel('a*','fontsize',16,'fontweight','b')
%ylabel('ERR','fontsize',16,'fontweight','b')
%title('ERR vs. a*','fontsize',16,'fontweight','b')
%grid on
%____________________
%PLOTS
test = 80; % a =
test2 = 60;
test3 = 40;
sample = (10:10:90);
point = 340; %342; %Choose a data point to look at deflected structure
figure %Pregrowth load (Theta) deflection path plus ERR
plot3(ResultsPG(:,4,test),ResultsPG(:,2,test),ResultsPG(:,5,test),'r-','LineWidth',1)
hold on
plot3(ResultsPG2(:,4,test),ResultsPG2(:,2,test),ResultsPG2(:,5,test),'r-','LineWidth',1)
hold on
plot3(StablePG(:,4,test),StablePG(:,2,test),StablePG(:,5,test),'r.','LineWidth',2)
hold on
plot3(StablePG2(:,4,test),StablePG2(:,2,test),StablePG2(:,5,test),'r.','LineWidth',2)
hold on
plot3(ResultsPG(:,4,test2),ResultsPG(:,2,test2),ResultsPG(:,5,test2),'b-','LineWidth',1)
hold on
plot3(ResultsPG2(:,4,test2),ResultsPG2(:,2,test2),ResultsPG2(:,5,test2),'b-
','LineWidth',1)
hold on
plot3(StablePG(:,4,test2),StablePG(:,2,test2),StablePG(:,5,test2),'b.','LineWidth',2)
hold on
plot3(StablePG2(:,4,test2),StablePG2(:,2,test2),StablePG2(:,5,test2),'b.','LineWidth',2)
hold on
plot3(ResultsPG(:,4,test3),ResultsPG(:,2,test3),ResultsPG(:,5,test3),'g-','LineWidth',1)
hold on
plot3(ResultsPG2(:,4,test3),ResultsPG2(:,2,test3),ResultsPG2(:,5,test3),'g-
','LineWidth',1)
hold on
plot3(StablePG(:,4,test3),StablePG(:,2,test3),StablePG(:,5,test3),'g.','LineWidth',2)
hold on
plot3(StablePG2(:,4,test3),StablePG2(:,2,test3),StablePG2(:,5,test3),'g.','LineWidth',2)
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',18,'fontweight','b')
zlabel('ERR','fontsize',16,'fontweight','b')
title('PreGrowth w/ ERR','fontsize',16,'fontweight','b')
grid on
figure %Projection into theta-ERR space
%plot(ResultsPG(:,2,test),ResultsPG(:,5,test),'k-','LineWidth',2)
%hold on
%plot(ResultsPG2(:,2,test),ResultsPG2(:,5,test),'k-','LineWidth',2)
%hold on
plot(StablePG(:,2,test),StablePG(:,5,test),'r.','LineWidth',2)
hold on
plot(StablePG2(:,2,test),StablePG2(:,5,test),'r.','LineWidth',2)
hold on
%plot(ResultsPG(:,2,test2),ResultsPG(:,5,test2),'k-','LineWidth',2)
%hold on
%plot(ResultsPG2(:,2,test2),ResultsPG2(:,5,test2),'k-','LineWidth',2)
%hold on
plot(StablePG(:,2,test2),StablePG(:,5,test2),'b.','LineWidth',2)
hold on
plot(StablePG2(:,2,test2),StablePG2(:,5,test2),'b.','LineWidth',2)
hold on
%plot(ResultsPG(:,2,test3),ResultsPG(:,5,test3),'k-','LineWidth',2)
%hold on
%plot(ResultsPG2(:,2,test3),ResultsPG2(:,5,test3),'k-','LineWidth',2)
%hold on
plot(StablePG(:,2,test3),StablePG(:,5,test3),'g.','LineWidth',2)
hold on
plot(StablePG2(:,2,test3),StablePG2(:,5,test3),'g.','LineWidth',2)
hold on
```

```
%plot(Results(:,3),Results(:,6),'k.')
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('ERR','fontsize',16,'fontweight','b')
title('temperature vs ERR','fontsize',16,'fontweight','b')
axis([0 0.028 0 2])
grid on
figure %N vs theta
plot(ResultsPG(:,2,test),ResultsPG(:,1,test),'k-','LineWidth',2)
hold on
plot(ResultsPG2(:,2,test),ResultsPG2(:,1,test),'k-','LineWidth',2)
hold on
plot(StablePG(:,2,test),StablePG(:,1,test),'r.','MarkerSize',10)
hold on
plot(StablePG2(:,2,test),StablePG2(:,1,test),'r.','MarkerSize',10)
%hold on
%plot(TestResult(:,3),TestResult(:,2),'b.')
%hold on
%plot(Results(:,3),Results(:,2),'g.','Linewidth',1)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
grid on
figure %theta vs. PE
plot(ResultsPG(:,2,test),ResultsPG(:,6,test),'k-','LineWidth',2)
hold on
plot(ResultsPG2(:,2,test),ResultsPG2(:,6,test),'k-','LineWidth',2)
hold on
plot(StablePG(:,2,test),StablePG(:,6,test),'r.','MarkerSize',10)
hold on
plot(StablePG2(:,2,test),StablePG2(:,6,test),'r.','MarkerSize',10)
hold on
plot(ResultsPG2(point,2,test),ResultsPG2(point,6,test),'bo','Markersize',12) %adjust
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('P','fontsize',16,'fontweight','b')
title('Potential Energy','fontsize',16,'fontweight','b')
grid on
figure %theta vs. F
plot(ResultsPG(:,2,test),ResultsPG(:,3,test),'k-','LineWidth',2)
hold on
plot(ResultsPG2(:,2,test),ResultsPG2(:,3,test),'k-','LineWidth',2)
hold on
plot(StablePG(:,2,test),StablePG(:,3,test),'r.','LineWidth',2)
hold on
plot(StablePG2(:,2,test),StablePG2(:,3,test),'r.','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('F','fontsize',16,'fontweight','b')
grid on
figure %theta vs. 3rd Minor
plot(ResultsPG(:,2,test),ResultsPG(:,7,test),'k-','LineWidth',2)
hold on
plot(ResultsPG2(:,2,test),ResultsPG2(:,7,test),'k-','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('3rd Minor Det','fontsize',16,'fontweight','b')
grid on
figure %theta load vs deflection
plot(ResultsPG(:,4,test),ResultsPG(:,2,test),'k-','LineWidth',2)
hold on
plot(ResultsPG2(:,4,test),ResultsPG2(:,2,test),'k-','LineWidth',2)
hold on
plot(StablePG(:,4,test),StablePG(:,2,test),'r.','MarkerSize',10)
hold on
plot(StablePG2(:,4,test),StablePG2(:,2,test),'r.','MarkerSize',10)
hold on
%plot(NStablePG(:,4,test),NStablePG(:,2,test),'c.','MarkerSize',10)
%hold on
%plot(NStablePG2(:,4,test),NStablePG2(:,2,test),'c.','MarkerSize',10)
%hold on
plot(ResultsPG2(point,4,test),ResultsPG2(point,2,test),'bo','Markersize',12) %adjust
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',18,'fontweight','b')
grid on
```

```
x1plot = (0:0.01:a_vals(test));
x3plot = (a_vals(test):0.01:1);
xp3plot = (a_vals(test):0.01:0.9); % Say Lp = 0.9
[a1,b1,c1,d1,a3,b3,c3,d3,K1,K3]=
Constants(ResultsPG2(point,1,test),ResultsPG2(point,2,test),a_vals(test)); %9 Or _HF
            for i1 = 1:length(x1plot);
                w1starplot(i1) = -
(a1*cos(K1*x1plot(i1))+b1*sin(K1*x1plot(i1))+c1*x1plot(i1)+d1);
                kappa1pl(i1) = -a1*K1^2*cos(K1*x1plot(i1))-b1*K1^2*sin(K1*x1plot(i1));
            end
            slope_at_a = -a1*K1*sin(K1*a_vals(test))+b1*K1*cos(K1*a_vals(test))+c1;
            for i3 = 1:length(x3plot);
                w3plot(i3) = -
(a3*cos(K3*x3plot(i3))+b3*sin(K3*x3plot(i3))+c3*x3plot(i3)+d3);
                kappa3pl(i3) = -a3*K3^2*cos(K3*x3plot(i3))-b3*K3^2*sin(K3*x3plot(i3));
            end
            for ip3 = 1:length(xp3plot);
                testflap(ip3) = -(slope_at_a*(xp3plot(ip3)-a_vals(test))-w3plot(1));%mess
with signs to get nice plot
            end
figure
plot(x1plot,w1starplot,'k','LineWidth',2)
hold on
plot(x3plot,w3plot,'r','LineWidth',2)
hold on
plot(xp3plot,testflap,'y','LineWidth',2)
hold on
plot(x1plot,kappa1pl,'g','LineWidth',2)
hold on
plot(x3plot,kappa3pl,'m','LineWidth',2)
if ResultsPG2(point,3,test) > 0;   %adjust!
    title(['Stable Configuration, a='
num2str(a_vals(test))],'fontsize',16,'fontweight','b')
else
    title(['Unstable Configuration, a='
num2str(a_vals(test))],'fontsize',16,'fontweight','b')
end
grid on
%figure %delam path and critical buckle theta vs a*
%plot(Results(:,1),Results(:,3),'c*','Linewidth',1)
%hold on
%plot(BuckleResults(:,1),BuckleResults(:,3),'m.','Linewidth',1)
%xlabel('a*')
%ylabel('aQ')
%title('Theta vs a*')
%grid on
%astarconstant1 = a_vals_star(test3)*ones(length(Npg),1);
%astarconstant2 = a_vals_star(test2)*ones(length(Npg),1);
%astarconstant3 = a_vals_star(test)*ones(length(Npg),1);
%Delamination Growth Paths and critical buckling temps
figure
%plot3(Results(:,1),Results(:,3),Results(:,2),'k.-','Linewidth',1)
%hold on
plot3(BuckleResults(:,1),BuckleResults(:,3),BuckleResults(:,2),'c.-','Linewidth',1)
hold on
for index = 1:length(sample);

plot3(a_vals_star(sample(index))*ones(length(Npg),1),StablePG(:,2,sample(index)),StablePG
(:,1,sample(index)),'r.','Linewidth',1)
    hold on

plot3(a_vals_star(sample(index))*ones(length(Npg),1),StablePG2(:,2,sample(index)),StableP
G2(:,1,sample(index)),'r.','Linewidth',1)
    hold on
end
%plot3(astarconstant1,StablePG(:,2,test3),StablePG(:,1,test3),'g.','Linewidth',1)
%hold on
%plot3(astarconstant2,StablePG(:,2,test2),StablePG(:,1,test2),'b.','Linewidth',1)
%hold on
%plot3(astarconstant3,StablePG(:,2,test),StablePG(:,1,test),'r.','Linewidth',1)
%hold on
```

```
%plot3(astarconstant1,StablePG2(:,2,test3),StablePG2(:,1,test3),'g.','Linewidth',1)
%hold on
%plot3(astarconstant2,StablePG2(:,2,test2),StablePG2(:,1,test2),'b.','Linewidth',1)
%hold on
%plot3(astarconstant3,StablePG2(:,2,test),StablePG2(:,1,test),'r.','Linewidth',1)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',16,'fontweight','b')
zlabel('N','fontsize',16,'fontweight','b')
title('Pregrowth loads','fontsize',16,'fontweight','b')
grid on
%figure
%plot3(Results(:,1),Results(:,3),Results(:,4),'b*','Linewidth',1)
%hold on
%plot3(astarconstant1,ResultsPG(:,2,30),ResultsPG(:,4,30),'k.','Linewidth',1)
%hold on
%plot3(astarconstant2,ResultsPG(:,2,50),ResultsPG(:,4,50),'k.','Linewidth',1)
%hold on
%plot3(astarconstant3,ResultsPG(:,2,70),ResultsPG(:,4,70),'k.','Linewidth',1)
%xlabel('a*')
%ylabel('aQ')
%zlabel('w0')
%title('Thermal Load-Deflection Space')
%grid on
```

Associated subroutines:

```
function[BF,Theta_BF,H,w0,PE] = BigFun_B01(No)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D rho h C Cstar Ce gamma mstar nstar etatilda astar
k1 = sqrt(No/Ds);
k3 = sqrt(No/D);
%First off, H.  Not Delamination!
H = sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar);
%F =
(1/(4*H^2))*((Ds/D)*(sin(k3*astar)^2)*k1*sin(2*k1*a)+(sin(k1*a)^2)*k3*sin(2*k3*astar));
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*sin(k3*astar)/No/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*sin(k3*astar)/H*k1*sin(k1*a);
Z_3 = -1/2*(-1+cos(k3*astar)^2)*Ds*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_4 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(8*mstar*No*rho+4*mstar*No*h)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_5 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar^2*sin(k3)*sin(k3*a)^2-
8*cos(k3)^2*mstar^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*k3+4*mstar^2*cos(k3*a)*sin(k3*a)+8*cos(k3)*mstar^2*sin(k3)^3+8*cos(k3)^3*mstar^
2*sin(k3)-4*cos(k3)*mstar^2*sin(k3)+4*mstar^2*k3*a)/No^2/H^2;
Z_7 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar*sin(k3)*rho*No-
4*cos(k3)*mstar*sin(k3)*h*No-
8*cos(k3)^2*mstar*h*No*cos(k3*a)*sin(k3*a)+4*mstar*No*h*cos(k3*a)*sin(k3*a)-
16*cos(k3)*mstar*sin(k3)*rho*No*sin(k3*a)^2-
8*cos(k3)*mstar*sin(k3)*h*No*sin(k3*a)^2+8*mstar*No*rho*k3*a+4*mstar*No*h*k3*a+8*mstar*No
*rho*cos(k3*a)*sin(k3*a)-
16*cos(k3)^2*mstar*rho*No*cos(k3*a)*sin(k3*a)+16*cos(k3)*mstar*sin(k3)^3*rho*No+8*cos(k3)
*mstar*sin(k3)^3*h*No+16*cos(k3)^3*mstar*rho*No*sin(k3)+8*cos(k3)^3*mstar*h*No*sin(k3)-
8*mstar*No*rho*k3-4*mstar*No*h*k3)/No^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(-cos(k3)*h^2*sin(k3)*No^2-
4*cos(k3)*rho*sin(k3)*h*No^2+4*No^2*rho*h*cos(k3*a)*sin(k3*a)-
8*cos(k3)*rho^2*sin(k3)*No^2*sin(k3*a)^2-
8*cos(k3)*rho*sin(k3)*h*No^2*sin(k3*a)^2+4*No^2*rho*h*k3*a-
8*cos(k3)^2*rho^2*No^2*cos(k3*a)*sin(k3*a)-8*cos(k3)^2*rho*h*No^2*cos(k3*a)*sin(k3*a)-
2*cos(k3)^2*h^2*No^2*cos(k3*a)*sin(k3*a)-2*cos(k3)*h^2*sin(k3)*No^2*sin(k3*a)^2-
4*No^2*rho^2*k3-
No^2*h^2*k3+4*No^2*rho^2*cos(k3*a)*sin(k3*a)+No^2*h^2*cos(k3*a)*sin(k3*a)+8*cos(k3)^3*rho
*h*No^2*sin(k3)+8*cos(k3)*rho^2*sin(k3)^3*No^2+8*cos(k3)*rho*sin(k3)^3*h*No^2+2*cos(k3)*h
^2*sin(k3)^3*No^2-4*cos(k3)*rho^2*sin(k3)*No^2-
```

```
4*No^2*rho*h*k3+8*cos(k3)^3*rho^2*No^2*sin(k3)+2*cos(k3)^3*h^2*No^2*sin(k3)+4*No^2*rho^2*
k3*a+No^2*h^2*k3*a)/No^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -No*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
%Now the Q's, such that the transversality condition is 2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*D*(-1+cos(k1*a)^2)*(-
4*cos(k3)^2*mstar^2+8*cos(k3)^2*cos(k3*a)^2*mstar^2+4*mstar^2+8*cos(k3)*cos(k3*a)*mstar^2
*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2)*k3^4/No^2/H^2+1/2*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*mstar^2/D/No^2/H^2+1/2*etatilda;
Q5 = -1/8*D*(-1+cos(k1*a)^2)*(8*mstar*No*rho+4*mstar*No*h-8*cos(k3)^2*mstar*rho*No-
4*cos(k3)^2*mstar*h*No+16*cos(k3)^2*cos(k3*a)^2*mstar*rho*No+8*cos(k3)^2*cos(k3*a)^2*msta
r*h*No+16*cos(k3)*cos(k3*a)*mstar*sin(k3)*sin(k3*a)*rho*No+8*cos(k3)*cos(k3*a)*mstar*sin(
k3)*sin(k3*a)*h*No-8*mstar*No*rho*cos(k3*a)^2-
4*mstar*No*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(8*mstar*No*rho+4*mstar*No*h)/D/No^2/H^2-No*(1-
nstar/Cstar);
Q6 = -1/8*D*(-1+cos(k1*a)^2)*(4*No^2*rho*h-cos(k3)^2*h^2*No^2-
4*cos(k3)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho*sin(k3)*sin(k3*a)*h*No^2+2*cos(k3)*cos(k3*
a)*h^2*sin(k3)*sin(k3*a)*No^2+4*No^2*rho^2+No^2*h^2+8*cos(k3)^2*cos(k3*a)^2*rho*h*No^2-
4*cos(k3)^2*rho*h*No^2+2*cos(k3)^2*cos(k3*a)^2*h^2*No^2-4*No^2*rho^2*cos(k3*a)^2-
No^2*h^2*cos(k3*a)^2+8*cos(k3)^2*cos(k3*a)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho^2*sin(k3)
*sin(k3*a)*No^2-4*No^2*rho*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)/D/No^2/H^2+1/2*N
o^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE
M_l = mstar*Theta_BF+(rho+1/2*h)*No;
w0 = M_l*(H-sin(k1*a)-sqrt(Ds/D)*sin(k3*astar))/(No*H);
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (clamped), only need A's  B's  and C's for now
A1 = -M_l*sqrt(Ds/D)*sin(k3*astar)/(No*H);
A3 = M_l*sin(k1*a)*cos(k3)/(No*H);
B1 = 0;
B3 = M_l*sin(k1*a)*sin(k3)/(No*H);
C1 = 0;
C3 = 0;
PE = 1/4*k1^3*Ds*(2*B1*A1+A1^2*cos(k1*a)*sin(k1*a)+A1^2*k1*a-
2*B1*A1*cos(k1*a)^2+B1^2*a*k1-
B1^2*cos(k1*a)*sin(k1*a))+Theta_BF*mstar*k1*B1+Theta_BF*mstar*k1*A1*sin(k1*a)-
Theta_BF*mstar*B1*k1*cos(k1*a)+1/2*No^2/Cstar*a+1/2*etatilda*Theta_BF^2*a-
1/4*k3^3*D*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*No^2/C*(1-a);

function[BF,Theta_BF,H,w0,PE] = BigFun_B01_HF(N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D rho h C Cstar Ce gamma mstar nstar etatilda astar
k1 = sqrt(N0/Ds);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
H = sqrt(Ds/D)*cos(k1*a)*cos(k3*astar)-sin(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*cos(k3*(1-a))/N0/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*cos(k3*(1-a))/H*k1*sin(k1*a);
Z_3 =
1/8*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3
*a)-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_4 =
1/8*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*co
s(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)
*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-
```

```
4*mstar*N0*h*cos(k3*a)^2-4*mstar*N0*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_5 = 1/8*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(4*sin(k3)*mstar^2*cos(k3)-4*mstar^2*k3+4*mstar^2*k3*a-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*a)^2+8*mstar^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*cos(k3*a)*sin(k3*a))/N0^2/H^2;
Z_7 = 1/8*(-
1+cos(k1*a)^2)*k3*(8*mstar*N0*rho*k3*a+4*mstar*N0*h*k3*a+8*sin(k3)*mstar*cos(k3)*rho*N0-
8*mstar*N0*rho*cos(k3*a)*sin(k3*a)+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)*sin(k3*a)+8*mstar*
N0*h*cos(k3)^2*cos(k3*a)*sin(k3*a)-4*mstar*N0*h*cos(k3*a)*sin(k3*a)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*a)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*a)^2-
4*mstar*N0*h*k3-8*mstar*N0*rho*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(sin(k3)*h^2*cos(k3)*N0^2+4*N0^2*rho*h*k3*a-
4*N0^2*rho*h*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-4*N0^2*rho^2*k3+4*N0^2*rho^2*k3*a-
N0^2*h^2*k3+N0^2*h^2*k3*a-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)*sin(k3*a)+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)*sin(k3*a)+2*N0^2*h^2*co
s(k3)^2*cos(k3*a)*sin(k3*a)-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*a)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
N0^2*h^2*cos(k3*a)*sin(k3*a)-
4*N0^2*rho^2*cos(k3*a)*sin(k3*a)+4*sin(k3)*rho*cos(k3)*h*N0^2)/N0^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -N0*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = 1/8*D*(-
1+cos(k1*a)^2)*(8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar^2*cos(k3)^2*cos(k3*
a)^2-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*
a)*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2-
4*mstar^2*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*etatilda;
Q5 = 1/8*D*(-
1+cos(k1*a)^2)*(16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*cos(k3)*cos(k3*a)*s
in(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)*cos(k3*a)*sin(k3
)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+1
6*mstar*N0*rho*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*m
star*N0*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-
8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k1^4/D/N0^2/H^2-N0*(1-nstar/Cstar);
Q6 = 1/8*D*(-
1+cos(k1*a)^2)*(2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*s
in(k3*a)+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)^2+2*N0^2*h^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-
N0^2*h^2*cos(k3)^2-N0^2*h^2*cos(k3*a)^2-4*N0^2*rho^2*cos(k3)^2-4*N0^2*rho^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*k3^4/N0^2/H^2-1/8*Ds^2*cos(k1*a)^2*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*N0^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE
M_l = mstar*Theta_BF+(rho+1/2*h)*N0;
w0 = M_l*(H-sqrt(Ds/D)*cos(k3*astar))/(N0*H);
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (hinged), only need A's  B's  and C's for now
```

```
A1 = -M_l*sqrt(Ds/D)*cos(k3*astar)/(N0*H);
A3 = -M_l*sin(k1*a)*sin(k3)/(N0*H);
B1 = 0;
B3 = M_l*sin(k1*a)*cos(k3)/(N0*H);
C1 = 0;
C3 = 0;
PE = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+mstar*Theta_BF*k1*B1+mstar*Theta_BF*A1*k1*sin(k1*a)-
mstar*Theta_BF*B1*k1*cos(k1*a)+1/2*N0^2/Cstar*a+1/2*etatilda*Theta_BF^2*a-
1/4*D*k3^3*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2/C*(1-a);

function[BF] = BigFun_B01_root(No)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D rho h C Cstar Ce gamma mstar nstar etatilda astar
k1 = sqrt(No/Ds);
k3 = sqrt(No/D);
%First off, the loading parameter and H
H = sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar);
%Now for the Z's
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*sin(k3*astar)/No/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*sin(k3*astar)/H*k1*sin(k1*a);
Z_3 = -1/2*(-1+cos(k3*astar)^2)*Ds*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_4 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(8*mstar*No*rho+4*mstar*No*h)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_5 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar^2*sin(k3)*sin(k3*a)^2-
8*cos(k3)^2*mstar^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*k3+4*mstar^2*cos(k3*a)*sin(k3*a)+8*cos(k3)*mstar^2*sin(k3)^3+8*cos(k3)^3*mstar^
2*sin(k3)-4*cos(k3)*mstar^2*sin(k3)+4*mstar^2*k3*a)/No^2/H^2;
Z_7 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar*sin(k3)*rho*No-
4*cos(k3)*mstar*sin(k3)*h*No-
8*cos(k3)^2*mstar*h*No*cos(k3*a)*sin(k3*a)+4*mstar*No*h*cos(k3*a)*sin(k3*a)-
16*cos(k3)*mstar*sin(k3)*rho*No*sin(k3*a)^2-
8*cos(k3)*mstar*sin(k3)*h*No*sin(k3*a)^2+8*mstar*No*rho*k3*a+4*mstar*No*h*k3*a+8*mstar*No
*rho*cos(k3*a)*sin(k3*a)-
16*cos(k3)^2*mstar*rho*No*cos(k3*a)*sin(k3*a)+16*cos(k3)*mstar*sin(k3)^3*rho*No+8*cos(k3)
*mstar*sin(k3)^3*h*No+16*cos(k3)^3*mstar*rho*No*sin(k3)+8*cos(k3)^3*mstar*h*No*sin(k3)-
8*mstar*No*rho*k3-4*mstar*No*h*k3)/No^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(-cos(k3)*h^2*sin(k3)*No^2-
4*cos(k3)*rho*sin(k3)*h*No^2+4*No^2*rho*h*cos(k3*a)*sin(k3*a)-
8*cos(k3)*rho^2*sin(k3)*No^2*sin(k3*a)^2-
8*cos(k3)*rho*sin(k3)*h*No^2*sin(k3*a)^2+4*No^2*rho*h*k3*a-
8*cos(k3)^2*rho^2*No^2*cos(k3*a)*sin(k3*a)-8*cos(k3)^2*rho*h*No^2*cos(k3*a)*sin(k3*a)-
2*cos(k3)^2*h^2*No^2*cos(k3*a)*sin(k3*a)-2*cos(k3)*h^2*sin(k3)*No^2*sin(k3*a)^2-
4*No^2*rho^2*k3-
No^2*h^2*k3+4*No^2*rho^2*cos(k3*a)*sin(k3*a)+No^2*h^2*cos(k3*a)*sin(k3*a)+8*cos(k3)^3*rho
*h*No^2*sin(k3)+8*cos(k3)*rho^2*sin(k3)^3*No^2+8*cos(k3)*rho*sin(k3)^3*h*No^2+2*cos(k3)*h
^2*sin(k3)^3*No^2-4*cos(k3)*rho^2*sin(k3)*No^2-
4*No^2*rho*h*k3+8*cos(k3)^3*rho^2*No^2*sin(k3)+2*cos(k3)^3*h^2*No^2*sin(k3)+4*No^2*rho^2*
k3*a+No^2*h^2*k3*a)/No^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -No*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
%Now the Q's, such that the transversality condition is 2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*D*(-1+cos(k1*a)^2)*(-
4*cos(k3)^2*mstar^2+8*cos(k3)^2*cos(k3*a)^2*mstar^2+4*mstar^2+8*cos(k3)*cos(k3*a)*mstar^2
*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2)*k3^4/No^2/H^2+1/2*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*mstar^2/D/No^2/H^2+1/2*etatilda;
Q5 = -1/8*D*(-1+cos(k1*a)^2)*(8*mstar*No*rho+4*mstar*No*h-8*cos(k3)^2*mstar*rho*No-
4*cos(k3)^2*mstar*h*No+16*cos(k3)^2*cos(k3*a)^2*mstar*rho*No+8*cos(k3)^2*cos(k3*a)^2*msta
r*h*No+16*cos(k3)*cos(k3*a)*mstar*sin(k3)*sin(k3*a)*rho*No+8*cos(k3)*cos(k3*a)*mstar*sin(
k3)*sin(k3*a)*h*No-8*mstar*No*rho*cos(k3*a)^2-
4*mstar*No*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
```

```
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(8*mstar*No*rho+4*mstar*No*h)/D/No^2/H^2-No*(1-
nstar/Cstar);
Q6 = -1/8*D*(-1+cos(k1*a)^2)*(4*No^2*rho*h-cos(k3)^2*h^2*No^2-
4*cos(k3)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho*sin(k3)*sin(k3*a)*h*No^2+2*cos(k3)*cos(k3*
a)*h^2*sin(k3)*sin(k3*a)*No^2+4*No^2*rho^2+No^2*h^2+8*cos(k3)^2*cos(k3*a)^2*rho*h*No^2-
4*cos(k3)^2*rho*h*No^2+2*cos(k3)^2*cos(k3*a)^2*h^2*No^2-4*No^2*rho^2*cos(k3*a)^2-
No^2*h^2*cos(k3*a)^2+8*cos(k3)^2*cos(k3*a)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho^2*sin(k3)
*sin(k3*a)*No^2-4*No^2*rho*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)/D/No^2/H^2+1/2*N
o^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE

function[BF] = BigFun_B01_root_HF(N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D rho h C Cstar Ce gamma mstar nstar etatilda astar
k1 = sqrt(N0/Ds);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
H = sqrt(Ds/D)*cos(k1*a)*cos(k3*astar)-sin(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*cos(k3*(1-a))/N0/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*cos(k3*(1-a))/H*k1*sin(k1*a);
Z_3 =
1/8*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3
*a)-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_4 =
1/8*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*co
s(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)
*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-
4*mstar*N0*h*cos(k3*a)^2-4*mstar*N0*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_5 = 1/8*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(4*sin(k3)*mstar^2*cos(k3)-4*mstar^2*k3+4*mstar^2*k3*a-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*a)^2+8*mstar^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*cos(k3*a)*sin(k3*a))/N0^2/H^2;
Z_7 = 1/8*(-
1+cos(k1*a)^2)*k3*(8*mstar*N0*rho*k3*a+4*mstar*N0*h*k3*a+8*sin(k3)*mstar*cos(k3)*rho*N0-
8*mstar*N0*rho*cos(k3*a)*sin(k3*a)+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)*sin(k3*a)+8*mstar*
N0*h*cos(k3)^2*cos(k3*a)*sin(k3*a)-4*mstar*N0*h*cos(k3*a)*sin(k3*a)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*a)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*a)^2-
4*mstar*N0*h*k3-8*mstar*N0*rho*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(sin(k3)*h^2*cos(k3)*N0^2+4*N0^2*rho*h*k3*a-
4*N0^2*rho*h*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-4*N0^2*rho^2*k3+4*N0^2*rho^2*k3*a-
N0^2*h^2*k3+N0^2*h^2*k3*a-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)*sin(k3*a)+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)*sin(k3*a)+2*N0^2*h^2*co
s(k3)^2*cos(k3*a)*sin(k3*a)-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*a)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
N0^2*h^2*cos(k3*a)*sin(k3*a)-
4*N0^2*rho^2*cos(k3*a)*sin(k3*a)+4*sin(k3)*rho*cos(k3)*h*N0^2)/N0^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -N0*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
```

```
Q4 = 1/8*D*(-
1+cos(k1*a)^2)*(8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar^2*cos(k3)^2*cos(k3*
a)^2-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*
a)*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2-
4*mstar^2*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*etatilda;
Q5 = 1/8*D*(-
1+cos(k1*a)^2)*(16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*cos(k3)*cos(k3*a)*s
in(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)*cos(k3*a)*sin(k3
)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+1
6*mstar*N0*rho*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*m
star*N0*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-
8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k1^4/D/N0^2/H^2-N0*(1-nstar/Cstar);
Q6 = 1/8*D*(-
1+cos(k1*a)^2)*(2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*s
in(k3*a)+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)^2+2*N0^2*h^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-
N0^2*h^2*cos(k3)^2-N0^2*h^2*cos(k3*a)^2-4*N0^2*rho^2*cos(k3)^2-4*N0^2*rho^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*k3^4/N0^2/H^2-1/8*Ds^2*cos(k1*a)^2*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*N0^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE

function[Hr] = BigFun_B01_rootH(No)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D astar %rho h theta mstar astar
k1 = sqrt(No/Ds);
k3 = sqrt(No/D);
%First off, the loading parameter and H
%M_l = mstar*theta+(rho+1/2*h)*No;
Hr = sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar);

function[Hr] = BigFun_B01_rootH_HF(N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
%This is for hinged fixed
global a Ds D astar %rho h theta mstar astar
k1 = sqrt(N0/Ds);
k3 = sqrt(N0/D);
Hr = sqrt(Ds/D)*cos(k1*a)*cos(k3*astar)-sin(k1*a)*sin(k3*astar);

function[Thetapg_out,F,w0,err,PotEn,thirdminordet] = BigFun_PreGrowth(a,astar,No)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global Ds D rho h C Cstar mstar nstar etatilda Ce
k1 = sqrt(No/Ds);
k3 = sqrt(No/D);
%First off, H.  Not Delamination!
H = sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*sin(k3*astar)/No/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*sin(k3*astar)/H*k1*sin(k1*a);
Z_3 = -1/2*(-1+cos(k3*astar)^2)*Ds*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_4 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(8*mstar*No*rho+4*mstar*No*h)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_5 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
```

```
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar^2*sin(k3)*sin(k3*a)^2-
8*cos(k3)^2*mstar^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*k3+4*mstar^2*cos(k3*a)*sin(k3*a)+8*cos(k3)*mstar^2*sin(k3)^3+8*cos(k3)^3*mstar^
2*sin(k3)-4*cos(k3)*mstar^2*sin(k3)+4*mstar^2*k3*a)/No^2/H^2;
Z_7 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar*sin(k3)*rho*No-
4*cos(k3)*mstar*sin(k3)*h*No-
8*cos(k3)^2*mstar*h*No*cos(k3*a)*sin(k3*a)+4*mstar*No*h*cos(k3*a)*sin(k3*a)-
16*cos(k3)*mstar*sin(k3)*rho*No*sin(k3*a)^2-
8*cos(k3)*mstar*sin(k3)*h*No*sin(k3*a)^2+8*mstar*No*rho*k3*a+4*mstar*No*h*k3*a+8*mstar*No
*rho*cos(k3*a)*sin(k3*a)-
16*cos(k3)^2*mstar*rho*No*cos(k3*a)*sin(k3*a)+16*cos(k3)*mstar*sin(k3)^3*rho*No+8*cos(k3)
*mstar*sin(k3)^3*h*No+16*cos(k3)^3*mstar*rho*No*sin(k3)+8*cos(k3)^3*mstar*h*No*sin(k3)-
8*mstar*No*rho*k3-4*mstar*No*h*k3)/No^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(-cos(k3)*h^2*sin(k3)*No^2-
4*cos(k3)*rho*sin(k3)*h*No^2+4*No^2*rho*h*cos(k3*a)*sin(k3*a)-
8*cos(k3)*rho^2*sin(k3)*No^2*sin(k3*a)^2-
8*cos(k3)*rho*sin(k3)*h*No^2*sin(k3*a)^2+4*No^2*rho*h*k3*a-
8*cos(k3)^2*rho^2*No^2*cos(k3*a)*sin(k3*a)-8*cos(k3)^2*rho*h*No^2*cos(k3*a)*sin(k3*a)-
2*cos(k3)^2*h^2*No^2*cos(k3*a)*sin(k3*a)-2*cos(k3)*h^2*sin(k3)*No^2*sin(k3*a)^2-
4*No^2*rho^2*k3-
No^2*h^2*k3+4*No^2*rho^2*cos(k3*a)*sin(k3*a)+No^2*h^2*cos(k3*a)*sin(k3*a)+8*cos(k3)^3*rho
*h*No^2*sin(k3)+8*cos(k3)*rho^2*sin(k3)^3*No^2+8*cos(k3)*rho*sin(k3)^3*h*No^2+2*cos(k3)*h
^2*sin(k3)^3*No^2-4*cos(k3)*rho^2*sin(k3)*No^2-
4*No^2*rho*h*k3+8*cos(k3)^3*rho^2*No^2*sin(k3)+2*cos(k3)^3*h^2*No^2*sin(k3)+4*No^2*rho^2*
k3*a+No^2*h^2*k3*a)/No^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -No*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
Thetapg = (-Q2+sqrt((Q2^2)-4*Q1*Q3))/(2*Q1);
Thetapg_out = Thetapg; %/.002; %for comparison
M_l = mstar*Thetapg+(rho+1/2*h)*No;
F =
(1/(4*H^2))*((Ds/D)*(sin(k3*astar)^2)*k1*sin(2*k1*a)+(sin(k1*a)^2)*k3*sin(2*k3*astar));
w0 = M_l*(H-sin(k1*a)-sqrt(Ds/D)*sin(k3*astar))/(No*H);
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*D*(-1+cos(k1*a)^2)*(-
4*cos(k3)^2*mstar^2+8*cos(k3)^2*cos(k3*a)^2*mstar^2+4*mstar^2+8*cos(k3)*cos(k3*a)*mstar^2
*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2)*k3^4/No^2/H^2+1/2*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*mstar^2/D/No^2/H^2+1/2*etatilda;
Q5 = -1/8*D*(-1+cos(k1*a)^2)*(8*mstar*No*rho+4*mstar*No*h-8*cos(k3)^2*mstar*rho*No-
4*cos(k3)^2*mstar*h*No+16*cos(k3)^2*cos(k3*a)^2*mstar*rho*No+8*cos(k3)^2*cos(k3*a)^2*msta
r*h*No+16*cos(k3)*cos(k3*a)*mstar*sin(k3)*sin(k3*a)*rho*No+8*cos(k3)*cos(k3*a)*mstar*sin(
k3)*sin(k3*a)*h*No-8*mstar*No*rho*cos(k3*a)^2-
4*mstar*No*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(8*mstar*No*rho+4*mstar*No*h)/D/No^2/H^2-No*(1-
nstar/Cstar);
Q6 = -1/8*D*(-1+cos(k1*a)^2)*(4*No^2*rho*h-cos(k3)^2*h^2*No^2-
4*cos(k3)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho*sin(k3)*sin(k3*a)*h*No^2+2*cos(k3)*cos(k3*
a)*h^2*sin(k3)*sin(k3*a)*No^2+4*No^2*rho^2+No^2*h^2+8*cos(k3)^2*cos(k3*a)^2*rho*h*No^2-
4*cos(k3)^2*rho*h*No^2+2*cos(k3)^2*cos(k3*a)^2*h^2*No^2-4*No^2*rho^2*cos(k3*a)^2-
No^2*h^2*cos(k3*a)^2+8*cos(k3)^2*cos(k3*a)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho^2*sin(k3)
*sin(k3*a)*No^2-4*No^2*rho*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)/D/No^2/H^2+1/2*N
o^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (clamped), only need A's  B's  and C's for now
A1 = -M_l*sqrt(Ds/D)*sin(k3*astar)/(No*H);
A3 = M_l*sin(k1*a)*cos(k3)/(No*H);
B1 = 0;
B3 = M_l*sin(k1*a)*sin(k3)/(No*H);
C1 = 0;
C3 = 0;
PotEn = 1/4*k1^3*Ds*(2*B1*A1+A1^2*cos(k1*a)*sin(k1*a)+A1^2*k1*a-
2*B1*A1*cos(k1*a)^2+B1^2*a*k1-
B1^2*cos(k1*a)*sin(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*k1*A1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)+1/2*No^2/Cstar*a+1/2*etatilda*Thetapg^2*a-
1/4*k3^3*D*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
```

```
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*No^2/C*(1-a);

function[Thetapg,F,w0,err,PotEn,thirdminordet] = BigFun_PreGrowth_HF(a,astar,N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global Ds D rho h C Cstar Ce mstar nstar etatilda
k1 = sqrt(N0/Ds);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
H = sqrt(Ds/D)*cos(k1*a)*cos(k3*astar)-sin(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*cos(k3*(1-a))/N0/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*cos(k3*(1-a))/H*k1*sin(k1*a);
Z_3 =
1/8*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3
*a)-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_4 =
1/8*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*co
s(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)
*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-
4*mstar*N0*h*cos(k3*a)^2-4*mstar*N0*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_5 = 1/8*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(4*sin(k3)*mstar^2*cos(k3)-4*mstar^2*k3+4*mstar^2*k3*a-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*a)^2+8*mstar^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*cos(k3*a)*sin(k3*a))/N0^2/H^2;
Z_7 = 1/8*(-
1+cos(k1*a)^2)*k3*(8*mstar*N0*rho*k3*a+4*mstar*N0*h*k3*a+8*sin(k3)*mstar*cos(k3)*rho*N0-
8*mstar*N0*rho*cos(k3*a)*sin(k3*a)+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)*sin(k3*a)+8*mstar*
N0*h*cos(k3)^2*cos(k3*a)*sin(k3*a)-4*mstar*N0*h*cos(k3*a)*sin(k3*a)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*a)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*a)^2-
4*mstar*N0*h*k3-8*mstar*N0*rho*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(sin(k3)*h^2*cos(k3)*N0^2+4*N0^2*rho*h*k3*a-
4*N0^2*rho*h*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-4*N0^2*rho^2*k3+4*N0^2*rho^2*k3*a-
N0^2*h^2*k3+N0^2*h^2*k3*a-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)*sin(k3*a)+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)*sin(k3*a)+2*N0^2*h^2*co
s(k3)^2*cos(k3*a)*sin(k3*a)-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*a)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
N0^2*h^2*cos(k3*a)*sin(k3*a)-
4*N0^2*rho^2*cos(k3*a)*sin(k3*a)+4*sin(k3)*rho*cos(k3)*h*N0^2)/N0^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -N0*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
Thetapg = (-Q2+sqrt((Q2^2)-4*Q1*Q3))/(2*Q1);
%Thetapg_out = Thetapg/.002; %for comparison
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
Fa = -sqrt(Ds/D)*cos(k3*astar)/H;
Fb = sin(k1*a)/H;
F = (1/4)*((Fa^2)*k1*sin(2*k1*a)-(Fb^2)*k3*sin(2*k3*astar));
w0 = M_l*(H-sqrt(Ds/D)*cos(k3*astar))/(N0*H);
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = 1/8*D*(-
1+cos(k1*a)^2)*(8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar^2*cos(k3)^2*cos(k3*
a)^2-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*
a)*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2-
4*mstar^2*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*etatilda;
Q5 = 1/8*D*(-
1+cos(k1*a)^2)*(16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*cos(k3)*cos(k3*a)*s
```

```
in(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)*cos(k3*a)*sin(k3
)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+1
6*mstar*N0*rho*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*m
star*N0*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-
8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k1^4/D/N0^2/H^2-N0*(1-nstar/Cstar);
Q6 = 1/8*D*(-
1+cos(k1*a)^2)*(2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*s
in(k3*a)+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)^2+2*N0^2*h^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-
N0^2*h^2*cos(k3)^2-N0^2*h^2*cos(k3*a)^2-4*N0^2*rho^2*cos(k3)^2-4*N0^2*rho^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*k3^4/N0^2/H^2-1/8*Ds^2*cos(k1*a)^2*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (hinged), only need A's  B's  and C's for now
A1 = -M_l*sqrt(Ds/D)*cos(k3*astar)/(N0*H);
A3 = -M_l*sin(k1*a)*sin(k3)/(N0*H);
B1 = 0;
B3 = M_l*sin(k1*a)*cos(k3)/(N0*H);
C1 = 0;
C3 = 0;
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+mstar*Thetapg*k1*B1+mstar*Thetapg*A1*k1*sin(k1*a)-
mstar*Thetapg*B1*k1*cos(k1*a)+1/2*N0^2/Cstar*a+1/2*etatilda*Thetapg^2*a-
1/4*D*k3^3*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2/C*(1-a);

function[Thetapg_out,F,w0,err,PotEn,thirdminordet] = BigFun_PreGrowth2(a,astar,No)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global Ds D rho h C Cstar mstar nstar etatilda Ce
k1 = sqrt(No/Ds);
k3 = sqrt(No/D);
%First off, H.  Not Delamination!
H = sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*sin(k3*astar)/No/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*sin(k3*astar)/H*k1*sin(k1*a);
Z_3 = -1/2*(-1+cos(k3*astar)^2)*Ds*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_4 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(8*mstar*No*rho+4*mstar*No*h)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_5 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar^2*sin(k3)*sin(k3*a)^2-
8*cos(k3)^2*mstar^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*k3+4*mstar^2*cos(k3*a)*sin(k3*a)+8*cos(k3)*mstar^2*sin(k3)^3+8*cos(k3)^3*mstar^
2*sin(k3)-4*cos(k3)*mstar^2*sin(k3)+4*mstar^2*k3*a)/No^2/H^2;
Z_7 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar*sin(k3)*rho*No-
4*cos(k3)*mstar*sin(k3)*h*No-
8*cos(k3)^2*mstar*h*No*cos(k3*a)*sin(k3*a)+4*mstar*No*h*cos(k3*a)*sin(k3*a)-
16*cos(k3)*mstar*sin(k3)*rho*No*sin(k3*a)^2-
8*cos(k3)*mstar*sin(k3)*h*No*sin(k3*a)^2+8*mstar*No*rho*k3*a+4*mstar*No*h*k3*a+8*mstar*No
*rho*cos(k3*a)*sin(k3*a)-
16*cos(k3)^2*mstar*rho*No*cos(k3*a)*sin(k3*a)+16*cos(k3)*mstar*sin(k3)^3*rho*No+8*cos(k3)
*mstar*sin(k3)^3*h*No+16*cos(k3)^3*mstar*rho*No*sin(k3)+8*cos(k3)^3*mstar*h*No*sin(k3)-
8*mstar*No*rho*k3-4*mstar*No*h*k3)/No^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(-cos(k3)*h^2*sin(k3)*No^2-
4*cos(k3)*rho*sin(k3)*h*No^2+4*No^2*rho*h*cos(k3*a)*sin(k3*a)-
8*cos(k3)*rho^2*sin(k3)*No^2*sin(k3*a)^2-
8*cos(k3)*rho*sin(k3)*h*No^2*sin(k3*a)^2+4*No^2*rho*h*k3*a-
```

```
8*cos(k3)^2*rho^2*No^2*cos(k3*a)*sin(k3*a)-8*cos(k3)^2*rho*h*No^2*cos(k3*a)*sin(k3*a)-
2*cos(k3)^2*h^2*No^2*cos(k3*a)*sin(k3*a)-2*cos(k3)*h^2*sin(k3)*No^2*sin(k3*a)^2-
4*No^2*rho^2*k3-
No^2*h^2*k3+4*No^2*rho^2*cos(k3*a)*sin(k3*a)+No^2*h^2*cos(k3*a)*sin(k3*a)+8*cos(k3)^3*rho
*h*No^2*sin(k3)+8*cos(k3)*rho^2*sin(k3)^3*No^2+8*cos(k3)*rho*sin(k3)^3*h*No^2+2*cos(k3)*h
^2*sin(k3)^3*No^2-4*cos(k3)*rho^2*sin(k3)*No^2-
4*No^2*rho*h*k3+8*cos(k3)^3*rho^2*No^2*sin(k3)+2*cos(k3)^3*h^2*No^2*sin(k3)+4*No^2*rho^2*
k3*a+No^2*h^2*k3*a)/No^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -No*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
Thetapg = (-Q2-sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %SECOND ROOT
Thetapg_out = Thetapg; %/.002; %for comparison
M_l = mstar*Thetapg+(rho+1/2*h)*No;
F =
(1/(4*H^2))*((Ds/D)*(sin(k3*astar)^2)*k1*sin(2*k1*a)+(sin(k1*a)^2)*k3*sin(2*k3*astar));
w0 = M_l*(H-sin(k1*a)-sqrt(Ds/D)*sin(k3*astar))/(No*H);
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*D*(-1+cos(k1*a)^2)*(-
4*cos(k3)^2*mstar^2+8*cos(k3)^2*cos(k3*a)^2*mstar^2+4*mstar^2+8*cos(k3)*cos(k3*a)*mstar^2
*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2)*k3^4/No^2/H^2+1/2*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*mstar^2/D/No^2/H^2+1/2*etatilda;
Q5 = -1/8*D*(-1+cos(k1*a)^2)*(8*mstar*No*rho+4*mstar*No*h-8*cos(k3)^2*mstar*rho*No-
4*cos(k3)^2*mstar*h*No+16*cos(k3)^2*cos(k3*a)^2*mstar*rho*No+8*cos(k3)^2*cos(k3*a)^2*msta
r*h*No+16*cos(k3)*cos(k3*a)*mstar*sin(k3)*sin(k3*a)*rho*No+8*cos(k3)*cos(k3*a)*mstar*sin(
k3)*sin(k3*a)*h*No-8*mstar*No*rho*cos(k3*a)^2-
4*mstar*No*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(8*mstar*No*rho+4*mstar*No*h)/D/No^2/H^2-No*(1-
nstar/Cstar);
Q6 = -1/8*D*(-1+cos(k1*a)^2)*(4*No^2*rho*h-cos(k3)^2*h^2*No^2-
4*cos(k3)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho*sin(k3)*sin(k3*a)*h*No^2+2*cos(k3)*cos(k3*
a)*h^2*sin(k3)*sin(k3*a)*No^2+4*No^2*rho^2+No^2*h^2+8*cos(k3)^2*cos(k3*a)^2*rho*h*No^2-
4*cos(k3)^2*rho*h*No^2+2*cos(k3)^2*cos(k3*a)^2*h^2*No^2-4*No^2*rho^2*cos(k3*a)^2-
No^2*h^2*cos(k3*a)^2+8*cos(k3)^2*cos(k3*a)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho^2*sin(k3)
*sin(k3*a)*No^2-4*No^2*rho*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)/D/No^2/H^2+1/2*N
o^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (clamped), only need A's  B's  and C's for now
A1 = -M_l*sqrt(Ds/D)*sin(k3*astar)/(No*H);
A3 = M_l*sin(k1*a)*cos(k3)/(No*H);
B1 = 0;
B3 = M_l*sin(k1*a)*sin(k3)/(No*H);
C1 = 0;
C3 = 0;
PotEn = 1/4*k1^3*Ds*(2*B1*A1+A1^2*cos(k1*a)*sin(k1*a)+A1^2*k1*a-
2*B1*A1*cos(k1*a)^2+B1^2*a*k1-
B1^2*cos(k1*a)*sin(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*k1*A1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)+1/2*No^2/Cstar*a+1/2*etatilda*Thetapg^2*a-
1/4*k3^3*D*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*No^2/C*(1-a);

function[Thetapg,F,w0,err,PotEn,thirdminordet] = BigFun_PreGrowth2_HF(a,astar,N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global Ds D rho h C Cstar Ce mstar nstar etatilda
k1 = sqrt(N0/Ds);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
H = sqrt(Ds/D)*cos(k1*a)*cos(k3*astar)-sin(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*cos(k3*(1-a))/N0/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*cos(k3*(1-a))/H*k1*sin(k1*a);
Z_3 =
1/8*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3
```

```
*a)-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_4 =
1/8*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*co
s(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)
*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-
4*mstar*N0*h*cos(k3*a)^2-4*mstar*N0*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_5 = 1/8*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(4*sin(k3)*mstar^2*cos(k3)-4*mstar^2*k3+4*mstar^2*k3*a-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*a)^2+8*mstar^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*cos(k3*a)*sin(k3*a))/N0^2/H^2;
Z_7 = 1/8*(-
1+cos(k1*a)^2)*k3*(8*mstar*N0*rho*k3*a+4*mstar*N0*h*k3*a+8*sin(k3)*mstar*cos(k3)*rho*N0-
8*mstar*N0*rho*cos(k3*a)*sin(k3*a)+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)*sin(k3*a)+8*mstar*
N0*h*cos(k3)^2*cos(k3*a)*sin(k3*a)-4*mstar*N0*h*cos(k3*a)*sin(k3*a)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*a)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*a)^2-
4*mstar*N0*h*k3-8*mstar*N0*rho*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(sin(k3)*h^2*cos(k3)*N0^2+4*N0^2*rho*h*k3*a-
4*N0^2*rho*h*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-4*N0^2*rho^2*k3+4*N0^2*rho^2*k3*a-
N0^2*h^2*k3+N0^2*h^2*k3*a-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)*sin(k3*a)+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)*sin(k3*a)+2*N0^2*h^2*co
s(k3)^2*cos(k3*a)*sin(k3*a)-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*a)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
N0^2*h^2*cos(k3*a)*sin(k3*a)-
4*N0^2*rho^2*cos(k3*a)*sin(k3*a)+4*sin(k3)*rho*cos(k3)*h*N0^2)/N0^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -N0*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
Thetapg = (-Q2-sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %THE OTHER ROOT!!!!!
%Thetapg_out = Thetapg/.002;
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
Fa = -sqrt(Ds/D)*cos(k3*astar)/H;
Fb = sin(k1*a)/H;
F = (1/4)*((Fa^2)*k1*sin(2*k1*a)-(Fb^2)*k3*sin(2*k3*astar));
w0 = M_l*(H-sqrt(Ds/D)*cos(k3*astar))/(N0*H);
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = 1/8*D*(-
1+cos(k1*a)^2)*(8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar^2*cos(k3)^2*cos(k3*
a)^2-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*
a)*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2-
4*mstar^2*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*etatilda;
Q5 = 1/8*D*(-
1+cos(k1*a)^2)*(16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*cos(k3)*cos(k3*a)*s
in(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)*cos(k3*a)*sin(k3
)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+1
6*mstar*N0*rho*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*m
star*N0*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-
8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k1^4/D/N0^2/H^2-N0*(1-nstar/Cstar);
Q6 = 1/8*D*(-
1+cos(k1*a)^2)*(2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*s
in(k3*a)+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)^2+2*N0^2*h^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-
N0^2*h^2*cos(k3)^2-N0^2*h^2*cos(k3*a)^2-4*N0^2*rho^2*cos(k3)^2-4*N0^2*rho^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*k3^4/N0^2/H^2-1/8*Ds^2*cos(k1*a)^2*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
```

```
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (hinged), only need A's  B's  and C's for now
A1 = -M_l*sqrt(Ds/D)*cos(k3*astar)/(N0*H);
A3 = -M_l*sin(k1*a)*sin(k3)/(N0*H);
B1 = 0;
B3 = M_l*sin(k1*a)*cos(k3)/(N0*H);
C1 = 0;
C3 = 0;
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+mstar*Thetapg*k1*B1+mstar*Thetapg*A1*k1*sin(k1*a)-
mstar*Thetapg*B1*k1*cos(k1*a)+1/2*N0^2/Cstar*a+1/2*etatilda*Thetapg^2*a-
1/4*D*k3^3*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2/C*(1-a);

function[a_1,b_1,c_1,d_1,a_3,b_3,c_3,d_3,k1,k3] = Constants(Nn,TH,aa) %No "P" for now...
% Uses the hand solution to find the const. of integ.
global Ds D mstar rho h
k1 = sqrt(Nn/Ds);
k3 = sqrt(Nn/D);
a_star = 1 - aa;
h_c = sin(k1*aa)*cos(k3*a_star)+sqrt(Ds/D)*cos(k1*aa)*sin(k3*a_star);
mlambda = mstar*TH+(rho+(h/2))*Nn;
a_1 = -(mlambda*sqrt(Ds/D)*sin(k3*a_star))/(Nn*h_c);
b_1 = 0;
c_1 = 0;
d_1 = (mlambda*(h_c-sin(k1*aa)))/(Nn*h_c);
a_3 = (mlambda*sin(k1*aa)*cos(k3))/(Nn*h_c);
b_3 = (mlambda*sin(k1*aa)*sin(k3))/(Nn*h_c);
c_3 = 0;
d_3 = -(mlambda*sin(k1*aa))/(Nn*h_c);

function[a_1,b_1,c_1,d_1,a_3,b_3,c_3,d_3,k1,k3] = Constants_HF(Nn,TH,aa) %No "P" for
now...
% Uses the hand solution to find the const. of integ.
global Ds D mstar rho h
k1 = sqrt(Nn/Ds);
k3 = sqrt(Nn/D);
a_star = 1 - aa;
h_h = sqrt(Ds/D)*cos(k1*aa)*cos(k3*a_star)-sin(k1*aa)*sin(k3*a_star);
mlambda = mstar*TH+(rho+(h/2))*Nn;
a_1 = -(mlambda*sqrt(Ds/D)*cos(k3*a_star))/(Nn*h_h);
b_1 = 0;
c_1 = 0;
d_1 = mlambda/Nn;
a_3 = -(mlambda*sin(k1*aa)*sin(k3))/(Nn*h_h);
b_3 = (mlambda*sin(k1*aa)*cos(k3))/(Nn*h_h);
c_3 = 0;
d_3 = 0;
```

Alternate version for model of intact structure:

```
% I shall call this one TempControl
% I am now prescribing temperature and solving for N from the IC, although
% this is more computationally intensive, it allows for me to hold the
% temperature value constant and look at the energy release rates vs a*
%We begin by defining the usual suspects
%The first version of this program is being written for HINGED and CLAMPED NO CZ
clear
global h C D Dp Cp
h = 0.05; % height of baseplate
hp = 0.05; % height of patch
h0 = hp/h; % ratio of heights
E0 = 1; % elastic modulus
C = 12/(h^2); % membrane stiffness of baseplate
```

```
Cp = C*E0*h0; % membrane stiffness of patch
D = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rho Ds
% stiffnesses of composite structure
Astar = D+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rho = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rho*Bstar; %Dstar
global  Dc Cs Ce
Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar nstar etatilda alphaP
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 0.5; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rho*nstar;
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
global a astar Theta_in
%Choose values for a and Theta
a_vals = (0.01:0.01:0.90);
a_vals_star = 1 - a_vals;
TH_vals = (0:0.0002:.018);%Limit for Hinge =0.009, Clamp=0.018, change step size to
0.0002
N_vals = (0:0.1:80);%Limit for hinge=60, clamp=80
%Find (a,TH,N) triples
%n=1; %Initialize for result array
p=0;
for i = 1:length(a_vals);
    p=p+1;
    n=1;
    a = a_vals(i);
    astar = 1-a_vals(i);
    for j = 1:length(TH_vals); %within this loop, find N roots
        Theta_in = TH_vals(j);
        for k = 1:length(N_vals)
            INTCON(k) = Integrability_clamp(N_vals(k)); %
HINGE/CLAMP
            %[value of integrability, we want it to be zero]
        end
        for m = 1:length(N_vals)-1;
            if INTCON(m)*INTCON(m+1)<0;%Look for a sign change
                Nroot(m)=fzero(@Integrability_clamp,N_vals(m));%Root of N
HINGE/CLAMP
                TEST_nosort(n,1,p)=Theta_in; %Unsorted Data
                TEST_nosort(n,2,p)=Nroot(m);
                n=n+1;
            else
            end
        end
    end
end
%Sort the data by order of the roots of N
for ii=1:length(a_vals);
    TEST(:,:,ii)=sortrows(TEST_nosort(:,:,ii));
end
%Find the other values associates w/ the triples
for r=1:length(a_vals);
    qq=1; %reset for new "a" value
    for q=1:length(TEST(:,1,1));
        DATA(q,1,r)=TEST(q,1,r);
        DATA(q,2,r)=TEST(q,2,r);%                          HINGE/CLAMP (right below)
[DATA(q,3,r),DATA(q,4,r),DATA(q,5,r),DATA(q,6,r),DATA(q,7,r)]=Evolving_clamp(TEST(q,2,r),
TEST(q,1,r),a_vals(r),a_vals_star(r));
        % Form of DATA [Theta,N,F,w0,ERR,PE,3rdMinor]
        if DATA(q,3,r)>0;
```

```
            StableDATA(qq,:,r)=DATA(q,:,r);
            qq=qq+1;
        else
        end
    end
end
%Clean Data sports one root of N (the lowest) for each value of theta
%For each value of a, the vectors are the exact same length
for tt=1:length(a_vals);
    rr=1;
    for pp = 1:length(TH_vals);
            nn=1;
        for jj=1:length(StableDATA(:,1,1))
            if StableDATA(jj,1,tt)==TH_vals(pp);
                test2(nn,1:7)=StableDATA(jj,1:7,tt);
                nn=nn+1;
            end
        end
        CleanDATA(rr,:,tt)=test2(1,:);%First one is lowest N root
        if test2(2,2)>test2(1,2)%if 2nd root > 1st, keep it
            CleanDATAroot2(rr,:,tt)=test2(2,:);
        else
            CleanDATAroot2(rr,:,tt)=zeros(1,7);
        end
        rr=rr+1;
    end
end
%------------------------
%save H_NOCZ.mat CleanDATA;
save C_NOCZ.mat CleanDATA;
% ~(((((((((^>
%PLOTS
test=80; %a value
figure %theta vs N
plot(DATA(:,1,test),DATA(:,2,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,1,test),StableDATA(:,2,test),'ro','LineWidth',2)
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
grid on
figure %PE vs theta
plot(DATA(:,1,test),DATA(:,6,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,1,test),StableDATA(:,6,test),'ro','LineWidth',2)
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('P','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
axis([0 0.03 0 1.5])
grid on
figure %F vs theta
plot(DATA(:,1,test),DATA(:,3,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,1,test),StableDATA(:,3,test),'ro','LineWidth',2)
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('F','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
axis([0 0.03 -100 100])
grid on
figure %3rd Minor Det vs theta
plot(DATA(:,1,test),DATA(:,7,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,1,test),StableDATA(:,7,test),'ro','LineWidth',2)
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('3rd Minor Det','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
grid on
figure %theta vs wo
plot(DATA(:,4,test),DATA(:,1,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,4,test),StableDATA(:,1,test),'ro','LineWidth',2)
```

```
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
axis([-0.2 0.25 0 0.03])
grid on
figure %theta vs wo
plot(DATA(:,4,test),DATA(:,2,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,4,test),StableDATA(:,2,test),'ro','LineWidth',2)
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
grid on
%-----------------------------------------------------------
%Trying something new...
%load=[2,4,6,8,10,12,14];  %HINGE
load=[5,10,15,20,25,30,35];  %CLAMP
for ugh2=1:length(load)
    for ugh=1:length(a_vals)
        ERRplot(ugh,ugh2)=CleanDATA(load(ugh2),5,ugh);
        %ERRplot2(ugh,ugh2)=CleanDATAroot2(load(ugh2),5,ugh);
    end
end
figure
%plot(a_vals_star',ERRplot(:,1),'k.','LineWidth',2)
%hold on
plot(a_vals_star',ERRplot(:,2),'b.','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,3),'r.','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,4),'g.','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,5),'c.','LineWidth',2)
hold on
plot(a_vals_star',Testing(:,5),'k.','LineWidth',2)%extra
hold on
%plot(a_vals_star',ERRplot2(:,5),'c*','LineWidth',2)%extra
%hold on
plot(a_vals_star',ERRplot(:,6),'m.','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,7),'y.','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('ERR','fontsize',16,'fontweight','b')
title('ERR vs. a*','fontsize',16,'fontweight','b')
grid on
%-----------------------------------------------------------
figure %theta vs N
plot(CleanDATA(:,1,test),CleanDATA(:,2,test),'r.','LineWidth',2)
hold on
plot(CleanDATAroot2(:,1,test),CleanDATAroot2(:,2,test),'b.','LineWidth',2)
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
grid on
```

Associated subroutines for alternate version:

```
function[ICvalue] = Integrability_clamp(No)
%Using my hand solution, this function expresses the IC
%I plug in values of theta and look for values of N that satisfy IC=0
global Ds D rho h C Cstar mstar nstar a astar Theta_in
k1 = sqrt(No/Ds);
k3 = sqrt(No/D);
%First off, H.  Not Delamination!
H = sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*sin(k3*astar)/No/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*sin(k3*astar)/H*k1*sin(k1*a);
```

```
Z_3 = -1/2*(-1+cos(k3*astar)^2)*Ds*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_4 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(8*mstar*No*rho+4*mstar*No*h)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_5 = -1/8*(-1+cos(k3*astar)^2)*Ds*k1*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/D/No^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar^2*sin(k3)*sin(k3*a)^2-
8*cos(k3)^2*mstar^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*k3+4*mstar^2*cos(k3*a)*sin(k3*a)+8*cos(k3)*mstar^2*sin(k3)^3+8*cos(k3)^3*mstar^
2*sin(k3)-4*cos(k3)*mstar^2*sin(k3)+4*mstar^2*k3*a)/No^2/H^2;
Z_7 = 1/8*(-1+cos(k1*a)^2)*k3*(-8*cos(k3)*mstar*sin(k3)*rho*No-
4*cos(k3)*mstar*sin(k3)*h*No-
8*cos(k3)^2*mstar*h*No*cos(k3*a)*sin(k3*a)+4*mstar*No*h*cos(k3*a)*sin(k3*a)-
16*cos(k3)*mstar*sin(k3)*rho*No*sin(k3*a)^2-
8*cos(k3)*mstar*sin(k3)*h*No*sin(k3*a)^2+8*mstar*No*rho*k3*a+4*mstar*No*h*k3*a+8*mstar*No
*rho*cos(k3*a)*sin(k3*a)-
16*cos(k3)^2*mstar*rho*No*cos(k3*a)*sin(k3*a)+16*cos(k3)*mstar*sin(k3)^3*rho*No+8*cos(k3)
*mstar*sin(k3)^3*h*No+16*cos(k3)^3*mstar*rho*No*sin(k3)+8*cos(k3)^3*mstar*h*No*sin(k3)-
8*mstar*No*rho*k3-4*mstar*No*h*k3)/No^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(-cos(k3)*h^2*sin(k3)*No^2-
4*cos(k3)*rho*sin(k3)*h*No^2+4*No^2*rho*h*cos(k3*a)*sin(k3*a)-
8*cos(k3)*rho^2*sin(k3)*No^2*sin(k3*a)^2-
8*cos(k3)*rho*sin(k3)*h*No^2*sin(k3*a)^2+4*No^2*rho*h*k3*a-
8*cos(k3)^2*rho^2*No^2*cos(k3*a)*sin(k3*a)-8*cos(k3)^2*rho*h*No^2*cos(k3*a)*sin(k3*a)-
2*cos(k3)^2*h^2*No^2*cos(k3*a)*sin(k3*a)-2*cos(k3)*h^2*sin(k3)*No^2*sin(k3*a)^2-
4*No^2*rho^2*k3-
No^2*h^2*k3+4*No^2*rho^2*cos(k3*a)*sin(k3*a)+No^2*h^2*cos(k3*a)*sin(k3*a)+8*cos(k3)^3*rho
*h*No^2*sin(k3)+8*cos(k3)*rho^2*sin(k3)^3*No^2+8*cos(k3)*rho*sin(k3)^3*h*No^2+2*cos(k3)*h
^2*sin(k3)^3*No^2-4*cos(k3)*rho^2*sin(k3)*No^2-
4*No^2*rho*h*k3+8*cos(k3)^3*rho^2*No^2*sin(k3)+2*cos(k3)^3*h^2*No^2*sin(k3)+4*No^2*rho^2*
k3*a+No^2*h^2*k3*a)/No^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -No*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
ICvalue = Q1*Theta_in^2+Q2*Theta_in+Q3;

function[ICvalue] = Integrability_hinge(N0)
%Using my hand solution, this function expresses the IC
%I plug in values of theta and look for values of N that satisfy IC=0
global Ds D rho h C Cstar mstar nstar a astar Theta_in
k1 = sqrt(N0/Ds);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
H = sqrt(Ds/D)*cos(k1*a)*cos(k3*astar)-sin(k1*a)*sin(k3*astar);
%Now for the Z's  start of Delamination stuff
Z_0 = astar+a*nstar/Cstar;
Z_1 = mstar*(Ds/D)^(1/2)*cos(k3*(1-a))/N0/H*k1*sin(k1*a);
Z_2 = (rho+1/2*h)*(Ds/D)^(1/2)*cos(k3*(1-a))/H*k1*sin(k1*a);
Z_3 =
1/8*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3
*a)-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_4 =
1/8*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*co
s(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)
*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-
4*mstar*N0*h*cos(k3*a)^2-4*mstar*N0*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_5 = 1/8*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*Ds*k1*(k1*a-
cos(k1*a)*sin(k1*a))/D/N0^2/H^2;
Z_6 = 1/8*(-1+cos(k1*a)^2)*k3*(4*sin(k3)*mstar^2*cos(k3)-4*mstar^2*k3+4*mstar^2*k3*a-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*a)^2+8*mstar^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
4*mstar^2*cos(k3*a)*sin(k3*a))/N0^2/H^2;
Z_7 = 1/8*(-
1+cos(k1*a)^2)*k3*(8*mstar*N0*rho*k3*a+4*mstar*N0*h*k3*a+8*sin(k3)*mstar*cos(k3)*rho*N0-
```

```
8*mstar*N0*rho*cos(k3*a)*sin(k3*a)+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)*sin(k3*a)+8*mstar*
N0*h*cos(k3)^2*cos(k3*a)*sin(k3*a)-4*mstar*N0*h*cos(k3*a)*sin(k3*a)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*a)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*a)^2-
4*mstar*N0*h*k3-8*mstar*N0*rho*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/H^2;
Z_8 = 1/8*(-1+cos(k1*a)^2)*k3*(sin(k3)*h^2*cos(k3)*N0^2+4*N0^2*rho*h*k3*a-
4*N0^2*rho*h*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-4*N0^2*rho^2*k3+4*N0^2*rho^2*k3*a-
N0^2*h^2*k3+N0^2*h^2*k3*a-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)*sin(k3*a)+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)*sin(k3*a)+2*N0^2*h^2*co
s(k3)^2*cos(k3*a)*sin(k3*a)-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*a)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)^2*cos(k3*a)*sin(k3*a)-
N0^2*h^2*cos(k3*a)*sin(k3*a)-
4*N0^2*rho^2*cos(k3*a)*sin(k3*a)+4*sin(k3)*rho*cos(k3)*h*N0^2)/N0^2/H^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*Z_3-1/2*Z_6;
Q2 = Z_0-(rho+1/2*h)*Z_1-1/2*Z_4-1/2*Z_7;
Q3 = -N0*Ca-(rho+1/2*h)*Z_2-1/2*Z_5-1/2*Z_8;
ICvalue = Q1*Theta_in^2+Q2*Theta_in+Q3;

function[F,w0,err,PotEn,Minor3] = Evolving_clamp(No,theta,a,astar)
%This function finds other values that i want,given a,N and TH
global Ds D rho h C Cstar Ce mstar nstar etatilda
k1 = sqrt(No/Ds);
k3 = sqrt(No/D);
%First off, H.  Not Delamination!
H = sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar);
M_l = mstar*theta+(rho+1/2*h)*No;
F =
(1/(4*H^2))*((Ds/D)*(sin(k3*astar)^2)*k1*sin(2*k1*a)+(sin(k1*a)^2)*k3*sin(2*k3*astar));
w0 = M_l*(H-sin(k1*a)-sqrt(Ds/D)*sin(k3*astar))/(No*H);
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*D*(-1+cos(k1*a)^2)*(-
4*cos(k3)^2*mstar^2+8*cos(k3)^2*cos(k3*a)^2*mstar^2+4*mstar^2+8*cos(k3)*cos(k3*a)*mstar^2
*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2)*k3^4/No^2/H^2+1/2*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*mstar^2/D/No^2/H^2+1/2*etatilda;
Q5 = -1/8*D*(-1+cos(k1*a)^2)*(8*mstar*No*rho+4*mstar*No*h-8*cos(k3)^2*mstar*rho*No-
4*cos(k3)^2*mstar*h*No+16*cos(k3)^2*cos(k3*a)^2*mstar*rho*No+8*cos(k3)^2*cos(k3*a)^2*msta
r*h*No+16*cos(k3)*cos(k3*a)*mstar*sin(k3)*sin(k3*a)*rho*No+8*cos(k3)*cos(k3*a)*mstar*sin(
k3)*sin(k3*a)*h*No-8*mstar*No*rho*cos(k3*a)^2-
4*mstar*No*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(8*mstar*No*rho+4*mstar*No*h)/D/No^2/H^2-No*(1-
nstar/Cstar);
Q6 = -1/8*D*(-1+cos(k1*a)^2)*(4*No^2*rho*h-cos(k3)^2*h^2*No^2-
4*cos(k3)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho*sin(k3)*sin(k3*a)*h*No^2+2*cos(k3)*cos(k3*
a)*h^2*sin(k3)*sin(k3*a)*No^2+4*No^2*rho^2+No^2*h^2+8*cos(k3)^2*cos(k3*a)^2*rho*h*No^2-
4*cos(k3)^2*rho*h*No^2+2*cos(k3)^2*cos(k3*a)^2*h^2*No^2-4*No^2*rho^2*cos(k3*a)^2-
No^2*h^2*cos(k3*a)^2+8*cos(k3)^2*cos(k3*a)^2*rho^2*No^2+8*cos(k3)*cos(k3*a)*rho^2*sin(k3)
*sin(k3*a)*No^2-4*No^2*rho*h*cos(k3*a)^2)*k3^4/No^2/H^2+1/8*Ds^2*(-
1+cos(k3*astar)^2)*cos(k1*a)^2*k1^4*(4*No^2*rho^2+4*No^2*rho*h+No^2*h^2)/D/No^2/H^2+1/2*N
o^2/Ce;
err = Q4*theta^2+Q5*theta+Q6;
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (clamped), only need A's  B's  and C's for now
A1 = -M_l*sqrt(Ds/D)*sin(k3*astar)/(No*H);
A3 = M_l*sin(k1*a)*cos(k3)/(No*H);
B1 = 0;
B3 = M_l*sin(k1*a)*sin(k3)/(No*H);
%C1 = 0;
%C3 = 0;
PotEn = 1/4*k1^3*Ds*(2*B1*A1+A1^2*cos(k1*a)*sin(k1*a)+A1^2*k1*a-
2*B1*A1*cos(k1*a)^2+B1^2*a*k1-
B1^2*cos(k1*a)*sin(k1*a))+theta*mstar*k1*B1+theta*mstar*k1*A1*sin(k1*a)-
theta*mstar*B1*k1*cos(k1*a)+1/2*No^2/Cstar*a+1/2*etatilda*theta^2*a-
1/4*k3^3*D*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*No^2/C*(1-a);

function[F,w0,err,PotEn,Minor3] = Evolving_hinge(N0,theta,a,astar)
%This function finds other values that i want,given a,N and TH
global Ds D rho h C Cstar Ce mstar nstar etatilda
```

```
k1 = sqrt(N0/Ds);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
H = sqrt(Ds/D)*cos(k1*a)*cos(k3*astar)-sin(k1*a)*sin(k3*astar);
M_l = mstar*theta+(rho+1/2*h)*N0;
Fa = -sqrt(Ds/D)*cos(k3*astar)/H;
Fb = sin(k1*a)/H;
F = (1/4)*((Fa^2)*k1*sin(2*k1*a)-(Fb^2)*k3*sin(2*k3*astar));
w0 = M_l*(H-sqrt(Ds/D)*cos(k3*astar))/(N0*H);
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = 1/8*D*(-
1+cos(k1*a)^2)*(8*mstar^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar^2*cos(k3)^2*cos(k3*
a)^2-4*mstar^2*cos(k3*a)^2-4*mstar^2*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(4*mstar^2+8*mstar^2*cos(k3)^2*cos(k3*a)^2+8*mstar^2*cos(k3)*cos(k3*
a)*sin(k3)*sin(k3*a)-4*mstar^2*cos(k3*a)^2-
4*mstar^2*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*etatilda;
Q5 = 1/8*D*(-
1+cos(k1*a)^2)*(16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+16*mstar*N0*rho*cos(k3)*cos(k3*a)*s
in(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*mstar*N0*h*cos(k3)*cos(k3*a)*sin(k3
)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k3^4/N0^2/H^2-
1/8*Ds^2*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h+16*mstar*N0*rho*cos(k3)^2*cos(k3*a)^2+1
6*mstar*N0*rho*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*mstar*N0*h*cos(k3)^2*cos(k3*a)^2+8*m
star*N0*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-8*mstar*N0*rho*cos(k3*a)^2-
8*mstar*N0*rho*cos(k3)^2-4*mstar*N0*h*cos(k3*a)^2-
4*mstar*N0*h*cos(k3)^2)*k1^4/D/N0^2/H^2-N0*(1-nstar/Cstar);
Q6 = 1/8*D*(-
1+cos(k1*a)^2)*(2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*s
in(k3*a)+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+8*N0^2*rho*h*cos(k3)^2*cos(k3*a)^2+2*N0^2*h^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-
N0^2*h^2*cos(k3)^2-N0^2*h^2*cos(k3*a)^2-4*N0^2*rho^2*cos(k3)^2-4*N0^2*rho^2*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3*a)^2-4*N0^2*rho*h*cos(k3)^2)*k3^4/N0^2/H^2-1/8*Ds^2*cos(k1*a)^2*(-
4*N0^2*rho^2*cos(k3)^2+4*N0^2*rho^2+N0^2*h^2+4*N0^2*rho*h+8*N0^2*rho^2*cos(k3)^2*cos(k3*a
)^2+2*N0^2*h^2*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+8*N
0^2*rho*h*cos(k3)^2*cos(k3*a)^2+8*N0^2*rho*h*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)+2*N0^2*h
^2*cos(k3)*cos(k3*a)*sin(k3)*sin(k3*a)-4*N0^2*rho^2*cos(k3*a)^2-N0^2*h^2*cos(k3*a)^2-
N0^2*h^2*cos(k3)^2-4*N0^2*rho*h*cos(k3*a)^2-
4*N0^2*rho*h*cos(k3)^2)*k1^4/D/N0^2/H^2+1/2*N0^2/Ce;
err = Q4*theta^2+Q5*theta+Q6;
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
%Constants (hinged), only need A's  B's  and C's for now
A1 = -M_l*sqrt(Ds/D)*cos(k3*astar)/(N0*H);
A3 = -M_l*sin(k1*a)*sin(k3)/(N0*H);
B1 = 0;
B3 = M_l*sin(k1*a)*cos(k3)/(N0*H);
%C1 = 0;
%C3 = 0;
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+mstar*theta*k1*B1+mstar*theta*A1*k1*sin(k1*a)-
mstar*theta*B1*k1*cos(k1*a)+1/2*N0^2/Cstar*a+1/2*etatilda*theta^2*a-
1/4*D*k3^3*(A3^2*cos(k3*a)*sin(k3*a)+A3^2*k3*a-2*A3*B3*cos(k3*a)^2+B3^2*a*k3-
B3^2*cos(k3*a)*sin(k3*a)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2/C*(1-a);
```

For model of partially debonded structure:

```
% Separation of intact Patched Structure
% Patched Plate, thermal load, NO PRESSURE
% Non-Linear  COMPRESSION!!!
%  CONTACT ZONE CASE
% This is the "theta" version, also the Buckling Version
%The code works for Clamped-Fixed ends or Hinged-fixed ends.
%All I need to do is change to clamp or hinge functions
%This change occurs in N I N E locations
clear
global h C D Dp
h = 0.05; % height of baseplate
```

```
hp = 0.05; % height of patch
h0 = hp/h; % ratio of heights
E0 = 1; % elastic modulus
C = 12/(h^2); % membrane stiffness of baseplate
Cp = C*E0*h0; % membrane stiffness of patch
D = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rho Ds
% stiffnesses of composite structure
Astar = D+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rho = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rho*Bstar; %Dstar
global  Dc Cs Ce
Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar nstar etatilda
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 0.5; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rho*nstar;
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
%Set 2*gamma, try a few values
%gamma_vals = [0.01/2,0.05/2,0.1/2,0.2/2];
global gamma a b astar
gamma = 0.1/2;
%Choose values for a, b and N
b = 0.9;
a_vals = (0.01:0.01:b);
a_vals_star = 1 - a_vals;
N = (0:0.01:45);
%for gin = 1:length(gamma_vals)
    %gamma = gamma_vals(gin);
m=1; %initialize index value for result array
n=1;
nn = 1;
%for i = 1:length(a_vals);
%    a = a_vals(i);
%    astar = 1-a_vals(i);
%    for j = 1:length(N);
%        [Big(j,1),Big(j,2),Big(j,3)] = BigFunCZ_clamp(N(j)); %HINGE/CLAMP 1
%        %[value of big F, Theta, w0]
%    end
 %   for k = 1:length(N)-1;
 %       %This is for DEBONDING, look for BF=0
 %       if Big(k,1)*Big(k+1,1)<0;
 %           Nroot(k) = fzero(@BigFunRootCZ_clamp,N(k)); %HINGE/CLAMP 2
 %           if Nroot(k)<max(N)+1 & Nroot(k)>min(N)-1;
 %           Roots4N(m) = Nroot(k); % The value of N that's a root
 %           m = m+1;
 %           else
 %           end
 %       else
 %       end
 %   end
 %   %Okay, let's try something...
 %       for ii = 1:length(Roots4N)
 %           TestResult(nn,1) = astar;
 %           TestResult(nn,2) = Roots4N(ii);
 %           [crud,TestResult(nn,3),TestResult(nn,4)] = BigFunCZ_clamp(Roots4N(ii));
%HINGE/CLAMP 3
 %           nn = nn+1;
 %           %[a*,N,Q,w0]
 %       end

 %   %Now, pluck the lowest root from the lot!
```

```
 %       N_lowest = min(Roots4N);
 %       m = 1; %Reset now that I'm done with it for this round!
 %       [blah,group(m,1),group(m,2)] = BigFunCZ_clamp(N_lowest); %HINGE/CLAMP 4
 %       %[value of big F (should be zero), Theta, W0]
 %
%end
%end
%Separate loop for Buckling
q=1;
for p = 1:length(a_vals);
    a = a_vals(p);
    astar = 1-a_vals(p);
    for j = 1:length(N);
        Big(j,1) = HCZ_clamp(N(j)); %HINGE/CLAMP 5
        %H
    end
     for r = 1:length(N)-1;
        %This is for Critical BUCKLING, look for H=0
        if Big(r,1)*Big(r+1,1)<0;
            Ncrit(r) = fzero(@HCZ_clamp,N(r)); %HINGE/CLAMP 6  %The critical N value
            Theta_crit(r) = -(rho+h/2)*Ncrit(r)/mstar; %Critical Theta
            %F_buckle = StableF(a,astar,Ncrit(k));
                CZBuckleResults(q,1)= astar;
                CZBuckleResults(q,2)= Ncrit(r);
                CZBuckleResults(q,3) = Theta_crit(r);
                q=q+1;
        end
     end
end
%Pregrowth
Npg = (0:0.1:80);
%Allocate space for result arrays
CZResultsPG = zeros(length(Npg),7,length(a_vals));
CZStablePG = zeros(length(Npg),7,length(a_vals));
CZValidPG = zeros(length(Npg),7,length(a_vals));
CZPropPG = zeros(length(Npg),7,length(a_vals));
CZResultsPG2 = zeros(length(Npg),7,length(a_vals));
CZStablePG2 = zeros(length(Npg),7,length(a_vals));
CZValidPG2 = zeros(length(Npg),7,length(a_vals));
CZPropPG2 = zeros(length(Npg),7,length(a_vals));
for nn=1:length(a_vals);
ain=a_vals(nn);
ainstar = 1-ain;
mm=1; %Reset, not really needed
for mm = 1:length(Npg); %Results 2 is the 2nd root!
     CZResultsPG(mm,1,nn)=Npg(mm); %N
     [CZResultsPG(mm,2,nn),
CZResultsPG(mm,3,nn),CZResultsPG(mm,4,nn),CZResultsPG(mm,5,nn),CZResultsPG(mm,6,nn),CZRes
ultsPG(mm,7,nn)] = PreGrowthCZ_clamp(ain,ainstar,Npg(mm)); %HINGE/CLAMP 7
     if CZResultsPG(mm,5,nn)>0;
         CZStablePG(mm,:,nn)=CZResultsPG(mm,:,nn);
     end
     if CZResultsPG(mm,7,nn)>0 && CZResultsPG(mm,7,nn)<2;
         CZValidPG(mm,:,nn)=CZResultsPG(mm,:,nn);
     end
     if CZStablePG(mm,7,nn)>1;
         CZPropPG(mm,:,nn)=CZResultsPG(mm,:,nn);
     End
CZResultsPG2(mm,1,nn)=Npg(mm); %N
     [CZResultsPG2(mm,2,nn),
CZResultsPG2(mm,3,nn),CZResultsPG2(mm,4,nn),CZResultsPG2(mm,5,nn),CZResultsPG2(mm,6,nn),C
ZResultsPG2(mm,7,nn)] = PreGrowthCZ_clamp2(ain,ainstar,Npg(mm)); %HINGE/CLAMP 8
     %[N0,theta,w0,ERR,F,PotentialE,ValidCZ (1 yes, 0 no)]
     if CZResultsPG2(mm,5,nn)>0;
         CZStablePG2(mm,:,nn)=CZResultsPG2(mm,:,nn);
     end
     if CZResultsPG2(mm,7,nn)>0&& CZResultsPG2(mm,7,nn)<2;
         CZValidPG2(mm,:,nn)=CZResultsPG2(mm,:,nn);
     end
     if CZStablePG2(mm,7,nn)>1;
         CZPropPG2(mm,:,nn)=CZResultsPG2(mm,:,nn);
```

```
    end

end
end
%=========================================================================
%Save the data with a new name for the COMBO program 4/6/10
%CHANGE to "_c" for clamped, "_h" for hinge!
%save CZRes_c.mat CZResultsPG;
%save CZRes2_c.mat CZResultsPG2;
%save CZStabRes_c.mat CZStablePG;
%save CZStabRes2_c.mat CZStablePG2;
%save CZValRes_c.mat CZValidPG;
%save CZValRes2_c.mat CZValidPG2;
%save CZPropRes_c.mat CZPropPG;
%save CZPropRes2_c.mat CZPropPG2;
%save CZCritBuck_c.mat CZBuckleResults;
%=========================================================================
test = 72; %Choose a value of a.  a = a_vals(test)
point = 370; %Choose a data point to look at deflected structure
test2 = 60;
test3 = 40;
sample = (10:10:b*100);
% PLOT 1 ---------------------------------------------------------
figure
plot3(CZResultsPG(:,3,test),CZResultsPG(:,2,test),CZResultsPG(:,4,test),'k-
','LineWidth',2)
hold on
plot3(CZResultsPG2(:,3,test),CZResultsPG2(:,2,test),CZResultsPG2(:,4,test),'k-
','LineWidth',2)
hold on
plot3(CZStablePG(:,3,test),CZStablePG(:,2,test),CZStablePG(:,4,test),'r.','LineWidth',2)
hold on
plot3(CZStablePG2(:,3,test),CZStablePG2(:,2,test),CZStablePG2(:,4,test),'r.','LineWidth',
2)
hold on
plot3(CZValidPG(:,3,test),CZValidPG(:,2,test),CZValidPG(:,4,test),'g*','LineWidth',2)
hold on
plot3(CZValidPG2(:,3,test),CZValidPG2(:,2,test),CZValidPG2(:,4,test),'g*','LineWidth',2)
hold on
plot3(CZPropPG(:,3,test),CZPropPG(:,2,test),CZPropPG(:,4,test),'y*','LineWidth',2)
hold on
plot3(CZPropPG2(:,3,test),CZPropPG2(:,2,test),CZPropPG2(:,4,test),'y*','LineWidth',2)
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',18,'fontweight','b')
zlabel('ERR','fontsize',16,'fontweight','b')
title('PreGrowth w/ ERR','fontsize',16,'fontweight','b')
grid on
% PLOT 2 ---------------------------------------------------------
figure
plot(CZResultsPG(:,2,test),CZResultsPG(:,4,test),'k-','LineWidth',2)
hold on
plot(CZResultsPG2(:,2,test),CZResultsPG2(:,4,test),'k-','LineWidth',2)
hold on
plot(CZStablePG(:,2,test),CZStablePG(:,4,test),'r.','LineWidth',2)
hold on
plot(CZStablePG2(:,2,test),CZStablePG2(:,4,test),'r.','LineWidth',2)
hold on
plot(CZResultsPG(:,2,test2),CZResultsPG(:,4,test2),'k-','LineWidth',2)
hold on
plot(CZResultsPG2(:,2,test2),CZResultsPG2(:,4,test2),'k-','LineWidth',2)
hold on
plot(CZStablePG(:,2,test2),CZStablePG(:,4,test2),'b.','LineWidth',2)
hold on
plot(CZStablePG2(:,2,test2),CZStablePG2(:,4,test2),'b.','LineWidth',2)
hold on
plot(CZResultsPG(:,2,test3),CZResultsPG(:,4,test3),'k-','LineWidth',2)
hold on
plot(CZResultsPG2(:,2,test3),CZResultsPG2(:,4,test3),'k-','LineWidth',2)
hold on
plot(CZStablePG(:,2,test3),CZStablePG(:,4,test3),'g.','LineWidth',2)
hold on
```

```
plot(CZStablePG2(:,2,test3),CZStablePG2(:,4,test3),'g.','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('ERR','fontsize',16,'fontweight','b')
title('temperature vs ERR','fontsize',16,'fontweight','b')
grid on
% PLOT 3 ------------------------------------------------------
figure
plot(CZResultsPG(:,2,test),CZResultsPG(:,1,test),'k-','LineWidth',2)
hold on
plot(CZResultsPG2(:,2,test),CZResultsPG2(:,1,test),'k-','LineWidth',2)
hold on
plot(CZStablePG(:,2,test),CZStablePG(:,1,test),'r.','MarkerSize',10)
hold on
plot(CZStablePG2(:,2,test),CZStablePG2(:,1,test),'r.','MarkerSize',10)
hold on
plot(CZValidPG(:,2,test),CZValidPG(:,1,test),'g*','MarkerSize',10)
hold on
plot(CZValidPG2(:,2,test),CZValidPG2(:,1,test),'g*','Markersize',10)
hold on
plot(CZPropPG(:,2,test),CZPropPG(:,1,test),'y*','MarkerSize',10)
hold on
plot(CZPropPG2(:,2,test),CZPropPG2(:,1,test),'y*','Markersize',10)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
grid on
% PLOT 4 ------------------------------------------------------
figure
plot(CZResultsPG(:,2,test),CZResultsPG(:,5,test),'k-','LineWidth',2)
hold on
plot(CZResultsPG2(:,2,test),CZResultsPG2(:,5,test),'k-','LineWidth',2)
hold on
plot(CZStablePG(:,2,test),CZStablePG(:,5,test),'r.','LineWidth',2)
hold on
plot(CZStablePG2(:,2,test),CZStablePG2(:,5,test),'r.','LineWidth',2)
hold on
plot(CZValidPG(:,2,test),CZValidPG(:,5,test),'g*','LineWidth',2)
hold on
plot(CZValidPG2(:,2,test),CZValidPG2(:,5,test),'g*','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('F','fontsize',16,'fontweight','b')
grid on
% PLOT 5 ------------------------------------------------------
%Remember to specify CZResultsPG or CZResultsPG2!  four PLACES
figure
plot(CZResultsPG(:,2,test),CZResultsPG(:,6,test),'k-','LineWidth',2)
hold on
plot(CZResultsPG2(:,2,test),CZResultsPG2(:,6,test),'k-','LineWidth',2)
hold on
plot(CZStablePG(:,2,test),CZStablePG(:,6,test),'r.','MarkerSize',10)
hold on
plot(CZStablePG2(:,2,test),CZStablePG2(:,6,test),'r.','MarkerSize',10)
hold on
plot(CZValidPG(:,2,test),CZValidPG(:,6,test),'g*','MarkerSize',10)
hold on
plot(CZValidPG2(:,2,test),CZValidPG2(:,6,test),'g*','Markersize',10)
hold on
plot(CZPropPG(:,2,test),CZPropPG(:,6,test),'y*','MarkerSize',10)
hold on
plot(CZPropPG2(:,2,test),CZPropPG2(:,6,test),'y*','Markersize',10)
%hold on
%plot(CZResultsPG2(point,2,test),CZResultsPG2(point,6,test),'bo','Markersize',12) %adjust
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('P','fontsize',16,'fontweight','b')
axis([0 0.028 0 1.4])
grid on
% PLOT 6 ------------------------------------------------------
figure
plot(CZResultsPG(:,3,test),CZResultsPG(:,2,test),'k-','LineWidth',2)
hold on
plot(CZResultsPG2(:,3,test),CZResultsPG2(:,2,test),'k-','LineWidth',2)
hold on
```

```
plot(CZStablePG(:,3,test),CZStablePG(:,2,test),'r.','MarkerSize',10)
hold on
plot(CZStablePG2(:,3,test),CZStablePG2(:,2,test),'r.','MarkerSize',10)
hold on
plot(CZValidPG(:,3,test),CZValidPG(:,2,test),'g*','MarkerSize',10)
hold on
plot(CZValidPG2(:,3,test),CZValidPG2(:,2,test),'g*','Markersize',10)
hold on
plot(CZPropPG(:,3,test),CZPropPG(:,2,test),'y*','MarkerSize',10)
hold on
plot(CZPropPG2(:,3,test),CZPropPG2(:,2,test),'y*','Markersize',10)
hold on
plot(CZResultsPG(point,3,test),CZResultsPG(point,2,test),'bo','Markersize',12) %adjust
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',18,'fontweight','b')
grid on
% PLOT 7 ------------------------------------------------------------
x1plot = (0:0.01:a_vals(test));
x2plot = (a_vals(test):0.01:b);
x3plot = (b:0.01:1);
                                    %adjust!
[a1,b1,c1,d1,a2,b2,c2,d2,a3,b3,c3,d3,K1,K2,K3]=
ConstantsCZ_clamp(CZResultsPG(point,1,test),CZResultsPG(point,2,test),a_vals(test));
%HINGE/CLAMP 9
            for i1 = 1:length(x1plot);
                w1starplot(i1) = -
(a1*cos(K1*x1plot(i1))+b1*sin(K1*x1plot(i1))+c1*x1plot(i1)+d1);
                kappa1pl(i1) = -a1*K1^2*cos(K1*x1plot(i1))-b1*K1^2*sin(K1*x1plot(i1));
            end
            slope_at_a = -a1*K1*sin(K1*a_vals(test))+b1*K1*cos(K1*a_vals(test))+c1;
            for i2 = 1:length(x2plot);
                w2starplot(i2) = -
(a2*cos(K2*x2plot(i2))+b2*sin(K2*x2plot(i2))+c2*x2plot(i2)+d2);
                testflap(i2) = -(slope_at_a*(x2plot(i2)-a_vals(test))-
w2starplot(1));%mess with signs to get nice plot
                kappa2pl(i2) = -a2*K2^2*cos(K2*x2plot(i2))-b2*K2^2*sin(K2*x2plot(i2));
            end
            for i3 = 1:length(x3plot);
                w3plot(i3) = -
(a3*cos(K3*x3plot(i3))+b3*sin(K3*x3plot(i3))+c3*x3plot(i3)+d3);
                kappa3pl(i3) = -a3*K3^2*cos(K3*x3plot(i3))-b3*K3^2*sin(K3*x3plot(i3));
            end
figure
plot(x1plot,w1starplot,'k','LineWidth',2)
hold on
plot(x2plot,w2starplot,'b','LineWidth',2)
hold on
plot(x2plot,testflap,'m','LineWidth',2)
hold on
plot(x3plot,w3plot,'r','LineWidth',2)
hold on
plot(x1plot,kappa1pl,'g','LineWidth',2)
hold on
plot(x2plot,kappa2pl,'c','LineWidth',2)
hold on
plot(x3plot,kappa3pl,'y','LineWidth',2)
if CZResultsPG(point,7,test) > 0; %adjust
    title('Valid CZ Configuration','fontsize',16,'fontweight','b')
else
    title('Invalid CZ Configuration','fontsize',16,'fontweight','b')
end
xlabel('x','fontsize',16,'fontweight','b')
ylabel('w','fontsize',16,'fontweight','b')
grid on
% PLOT 8 ------------------------------------------------------------
%astarconstant1 = a_vals_star(30)*ones(length(Npg),1);
%astarconstant2 = a_vals_star(50)*ones(length(Npg),1);
%astarconstant3 = a_vals_star(70)*ones(length(Npg),1);
figure
%plot3(Results1(:,1),Results1(:,3),Results1(:,2),'k.-','Linewidth',1)
%hold on
```

```
plot3(CZBuckleResults(:,1),CZBuckleResults(:,3),CZBuckleResults(:,2),'c.-','Linewidth',1)
hold on
for index = 1:length(sample);
plot3(a_vals_star(sample(index))*ones(length(Npg),1),CZStablePG(:,2,sample(index)),CZStab
lePG(:,1,sample(index)),'r.','Linewidth',1)
    hold on
plot3(a_vals_star(sample(index))*ones(length(Npg),1),CZStablePG2(:,2,sample(index)),CZSta
blePG2(:,1,sample(index)),'r.','Linewidth',1)
    hold on
end
%plot3(astarconstant1,CZStablePG(:,2,test3),CZStablePG(:,1,test3),'g.','Linewidth',1)
%hold on
%plot3(astarconstant2,CZStablePG(:,2,test2),CZStablePG(:,1,test2),'b.','Linewidth',1)
%hold on
%plot3(astarconstant3,CZStablePG(:,2,test),CZStablePG(:,1,test),'r.','Linewidth',1)
%hold on
%plot3(astarconstant1,CZStablePG2(:,2,test3),CZStablePG2(:,1,test3),'g.','Linewidth',1)
%hold on
%plot3(astarconstant2,CZStablePG2(:,2,test2),CZStablePG2(:,1,test2),'b.','Linewidth',1)
%hold on
%plot3(astarconstant3,CZStablePG2(:,2,test),CZStablePG2(:,1,test),'r.','Linewidth',1)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',18,'fontweight','b')
zlabel('N','fontsize',16,'fontweight','b')
title('Pregrowth loads','fontsize',16,'fontweight','b')
grid on
```

Associated subroutines:

```
function[BF,Theta_BF,w0] = BigFunCZ_clamp(N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D Dc rho h C Cstar Ce mstar nstar etatilda b astar gamma
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+(Dc/D)*sin(k3-k3*b)*cos(k2*b);
Z4 = k3*sin(k3-k3*b);
Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = mstar*Z1/N0/Hc*k1*sin(k1*a);
y2 = (rho+1/2*h)*Z1/Hc*k1*sin(k1*a);
y3 = 1/2*Z1^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y4 = 1/8*Z1^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y5 = 1/8*Z1^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(-
4*Z2^2*mstar^2*b*k2+4*Z2^2*mstar^2*k2*a+4*Z3^2*mstar^2*k2*a-
4*Z2^2*mstar^2*cos(k2*a)*sin(k2*a)-4*Z3^2*mstar^2*k2*b-
8*Z2*mstar^2*Z3*cos(k2*b)^2+4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar^2*cos(k2*b)*sin(k2*b)+4*Z3^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z2*mstar^2*Z3*cos
(k2*a)^2)/N0^2/Hc^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(-8*Z2^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*b*k2+8*Z2^2*mstar*rho*N0*k2*a+4*Z2^2*mstar*h*N0*k2*a+8*Z3^2*mstar*rho*N
0*k2*a+4*Z3^2*mstar*h*N0*k2*a-8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z2^2*mstar*h*N0*cos(k2*a)*sin(k2*a)-8*Z3^2*mstar*rho*N0*k2*b-4*Z3^2*mstar*h*N0*k2*b-
16*Z2*mstar*Z3*rho*N0*cos(k2*b)^2-
8*Z2*mstar*Z3*h*N0*cos(k2*b)^2+8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)+4*Z2^2*mstar*h*N0*
cos(k2*b)*sin(k2*b)-8*Z3^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar*h*N0*cos(k2*b)*sin(k2*b)+8*Z3^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z3^2*msta
```

```
r*h*N0*cos(k2*a)*sin(k2*a)+8*Z2*mstar*Z3*h*N0*cos(k2*a)^2+16*Z2*mstar*Z3*rho*N0*cos(k2*a)
^2)/N0^2/Hc^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho^2*N0^2*b*k2-4*Z2^2*rho*h*N0^2*b*k2-
Z2^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*k2*a+4*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+4*Z3^
2*rho^2*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho*Z3*h*N0^2*cos(k2*a)^2-
8*Z2*rho^2*Z3*N0^2*cos(k2*b)^2-
4*Z3^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)+4*Z3^2*rho*h*N0^2*k2*a+Z3^2*h^2*N0^2*k2*a-
4*Z3^2*rho^2*N0^2*k2*b-4*Z3^2*rho*h*N0^2*k2*b-Z3^2*h^2*N0^2*k2*b-
8*Z2*rho*Z3*h*N0^2*cos(k2*b)^2-
2*Z2*h^2*Z3*N0^2*cos(k2*b)^2+4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+4*Z2^2*rho^2*N0^2*cos(
k2*b)*sin(k2*b)+Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-4*Z3^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
Z3^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-
Z2^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z3^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho^2*Z3*N0^
2*cos(k2*a)^2+2*Z2*h^2*Z3*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z3^2*h^2
*N0^2*cos(k2*a)*sin(k2*a))/N0^2/Hc^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*cos(k3)*mstar^2*sin(k3)^3-4*mstar^2*k3-
4*cos(k3)*mstar^2*sin(k3)-8*cos(k3)^2*mstar^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*mstar^2*sin(k3)*sin(k3*b)^2+4*mstar^2*cos(k3*b)*sin(k3*b)+8*cos(k3)^3*mstar^2*s
in(k3)+4*mstar^2*k3*b)/N0^2/Hc^2;
y10 = 1/8*(-
1+cos(k1*a)^2)*k3*(16*cos(k3)^3*mstar*rho*N0*sin(k3)+8*cos(k3)^3*mstar*h*N0*sin(k3)+16*co
s(k3)*mstar*sin(k3)^3*rho*N0+8*cos(k3)*mstar*sin(k3)^3*h*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3-8*cos(k3)*mstar*sin(k3)*rho*N0-
4*cos(k3)*mstar*sin(k3)*h*N0+8*mstar*N0*rho*k3*b+4*mstar*N0*h*k3*b-
16*cos(k3)^2*mstar*rho*N0*cos(k3*b)*sin(k3*b)-8*cos(k3)^2*mstar*h*N0*cos(k3*b)*sin(k3*b)-
16*cos(k3)*mstar*sin(k3)*rho*N0*sin(k3*b)^2-
8*cos(k3)*mstar*sin(k3)*h*N0*sin(k3*b)^2+8*mstar*N0*rho*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
cos(k3*b)*sin(k3*b))/N0^2/Hc^2;
y11 = 1/8*(-1+cos(k1*a)^2)*k3*(-
4*cos(k3)*rho*sin(k3)*h*N0^2+8*cos(k3)*rho^2*sin(k3)^3*N0^2+8*cos(k3)*rho*sin(k3)^3*h*N0^
2-4*N0^2*rho^2*k3-
N0^2*h^2*k3+8*cos(k3)^3*rho*h*N0^2*sin(k3)+4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+4*N0^2*rho*h*
k3*b-8*cos(k3)^2*rho^2*N0^2*cos(k3*b)*sin(k3*b)-2*cos(k3)^2*h^2*N0^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*rho^2*sin(k3)*N0^2*sin(k3*b)^2-8*cos(k3)*rho*sin(k3)*h*N0^2*sin(k3*b)^2-
2*cos(k3)*h^2*sin(k3)*N0^2*sin(k3*b)^2-
8*cos(k3)^2*rho*h*N0^2*cos(k3*b)*sin(k3*b)+4*N0^2*rho^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*cos(
k3*b)*sin(k3*b)-
4*N0^2*rho*h*k3+8*cos(k3)^3*rho^2*N0^2*sin(k3)+2*cos(k3)^3*h^2*N0^2*sin(k3)-
cos(k3)*h^2*sin(k3)*N0^2+2*cos(k3)*h^2*sin(k3)^3*N0^2+4*N0^2*rho^2*k3*b+N0^2*h^2*k3*b-
4*cos(k3)*rho^2*sin(k3)*N0^2)/N0^2/Hc^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z2^2*cos(k2*a)^2*mstar^2+8*Z2*cos(k2*a)*mstar^2*Z3*sin(k2*a)+4*Z3^2*mst
ar^2-4*Z3^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/2*Ds*Z1^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hc^2+1/2*etatilda;
Q5 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*mstar*h*N0-
4*Z3^2*mstar*h*N0*cos(k2*a)^2+8*Z2^2*cos(k2*a)^2*mstar*rho*N0+4*Z2^2*cos(k2*a)^2*mstar*h*
N0+16*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*rho*N0+8*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*h*N0+8*Z3^
2*mstar*rho*N0-8*Z3^2*mstar*rho*N0*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hc^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*rho^2*N0^2-
4*Z3^2*rho^2*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2-
4*Z3^2*rho*h*N0^2*cos(k2*a)^2+Z3^2*h^2*N0^2-
Z3^2*h^2*N0^2*cos(k2*a)^2+4*Z2^2*cos(k2*a)^2*rho^2*N0^2+4*Z2^2*cos(k2*a)^2*rho*h*N0^2+8*Z
2*cos(k2*a)*rho^2*Z3*sin(k2*a)*N0^2+8*Z2*cos(k2*a)*rho*Z3*sin(k2*a)*h*N0^2+Z2^2*cos(k2*a)
^2*h^2*N0^2+2*Z2*cos(k2*a)*h^2*Z3*sin(k2*a)*N0^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hc^2+1/2*N0^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE
M_l = mstar*Theta_BF+(rho+1/2*h)*N0;
```

```
w0 = M_l*(Hc-Z1-sin(k1*a))/(N0*Hc);

function[BF,Theta_BF,w0] = BigFunCZ_hinge(N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D Dc rho h C Cstar Ce mstar nstar etatilda b astar gamma
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hh = sin(k1*a)*sin(k3*b-k3)*cos(k2*a-k2*b)+sqrt(Dc/D)*sin(k1*a)*cos(k3*b-k3)*sin(k2*a-
k2*b)-sqrt(Ds/Dc)*cos(k1*a)*sin(k3*b-k3)*sin(k2*a-k2*b)+sqrt(Ds/D)*cos(k1*a)*cos(k3*b-
k3)*cos(k2*a-k2*b);
%Now the Z's from my hand solutions
Z1 = cos(k2*b)*sin(k3*b-k3)-sqrt(Dc/D)*sin(k2*b)*cos(k3*b-k3);
Z2 = sin(k2*b)*sin(k3*b-k3)+sqrt(Dc/D)*cos(k2*b)*cos(k3*b-k3);
Z3 = sqrt(Ds/Dc)*sin(k3*b-k3)*sin(k2*a-k2*b)-sqrt(Ds/D)*cos(k3*b-k3)*cos(k2*a-k2*b);
Z4 = k3*cos(k3*b-k3);
Z5 = sin(k3*b-k3)-Z4*b; %for completeness, we don't need it!
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = -mstar*Z3/N0/Hh*k1*sin(k1*a);
y2 = -(rho+1/2*h)*Z3/Hh*k1*sin(k1*a);
y3 = 1/2*Z3^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y4 = 1/8*Z3^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y5 = 1/8*Z3^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(4*Z1^2*mstar^2*cos(k2*b)*sin(k2*b)-
8*Z1*mstar^2*Z2*cos(k2*b)^2-4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-4*Z2^2*mstar^2*k2*b-
4*Z1^2*mstar^2*b*k2+4*Z1^2*mstar^2*k2*a+4*Z2^2*mstar^2*k2*a-
4*Z1^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z1*mstar^2*Z2*cos(k2*a)^2+4*Z2^2*mstar^2*cos(k2*a)*s
in(k2*a))/N0^2/Hh^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(8*Z1^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
8*Z1*mstar*Z2*h*N0*cos(k2*b)^2+4*Z1^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-
16*Z1*mstar*Z2*rho*N0*cos(k2*b)^2-8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z2^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-8*Z2^2*mstar*rho*N0*k2*b-4*Z1^2*mstar*h*N0*b*k2-
8*Z1^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*k2*b+8*Z1^2*mstar*rho*N0*k2*a+4*Z1^2*mstar*h*N0*k2*a+8*Z2^2*mstar*rho*N
0*k2*a+4*Z2^2*mstar*h*N0*k2*a+16*Z1*mstar*Z2*rho*N0*cos(k2*a)^2+8*Z1*mstar*Z2*h*N0*cos(k2
*a)^2-8*Z1^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z1^2*mstar*h*N0*cos(k2*a)*sin(k2*a)+8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z2^2*msta
r*h*N0*cos(k2*a)*sin(k2*a))/N0^2/Hh^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho*h*N0^2*k2*b-Z2^2*h^2*N0^2*k2*b-
Z1^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-8*Z1*rho*Z2*h*N0^2*cos(k2*b)^2-
2*Z1*h^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
8*Z1*rho^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+Z1^2*h^2*N0^2*cos(k2
*b)*sin(k2*b)+4*Z1^2*rho*h*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho*h*N0^2*b*k2-4*Z1^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho^2*N0^2*b*k2+4*Z1^2*rho^2*N0^2*k2*a+Z1^2*h^2*N0^2*k2*a+4*Z2^2*rho^2*N0^2*k2*a+4
*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+8*Z1*rho^2*Z2*N0^2*cos(k2*a)^2+8*Z1*rho*Z2*h*N0^
2*cos(k2*a)^2+2*Z1*h^2*Z2*N0^2*cos(k2*a)^2-4*Z1^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)-
Z1^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z2^2*h^2*N0^2*cos
(k2*a)*sin(k2*a)-4*Z2^2*rho^2*N0^2*k2*b)/N0^2/Hh^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*mstar^2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*b)^2+4*mstar^2*b*k3+4*sin(k3)*mstar^2*cos(k3)-
4*mstar^2*k3-4*mstar^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
y10 = 1/8*(-1+cos(k1*a)^2)*k3*(16*mstar*N0*rho*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*b)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*b)^2-
8*mstar*N0*rho*cos(k3*b)*sin(k3*b)-
4*mstar*N0*h*cos(k3*b)*sin(k3*b)+8*mstar*N0*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
b*k3+8*mstar*N0*rho*b*k3+8*sin(k3)*mstar*cos(k3)*rho*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/Hh^2;
y11 = 1/8*(-
1+cos(k1*a)^2)*k3*(4*N0^2*rho^2*b*k3+8*N0^2*rho^2*cos(k3)^2*cos(k3*b)*sin(k3*b)+2*N0^2*h^
2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+8*N0^2*rho*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*b*k3
-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*b)^2-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*b)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*b)^2+4*N0^2*rho*h*b*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-
```

```
4*N0^2*rho*h*k3+4*sin(k3)*rho*cos(k3)*h*N0^2+sin(k3)*h^2*cos(k3)*N0^2-N0^2*h^2*k3-
4*N0^2*rho^2*k3-N0^2*h^2*cos(k3*b)*sin(k3*b)-4*N0^2*rho^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*mstar^2+8*Z1*cos(k2*a)*mstar^2*Z2*sin(k2*a)+4*Z2^2*mst
ar^2-4*Z2^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/2*Ds*Z3^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hh^2+1/2*etatilda;
Q5 = -1/8*Dc*(-
1+cos(k1*a)^2)*(8*Z1^2*cos(k2*a)^2*mstar*rho*N0+4*Z1^2*cos(k2*a)^2*mstar*h*N0+16*Z1*cos(k
2*a)*mstar*Z2*sin(k2*a)*rho*N0+8*Z1*cos(k2*a)*mstar*Z2*sin(k2*a)*h*N0+8*Z2^2*mstar*rho*N0
-8*Z2^2*mstar*rho*N0*cos(k2*a)^2+4*Z2^2*mstar*h*N0-
4*Z2^2*mstar*h*N0*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hh^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*rho^2*N0^2+4*Z1^2*cos(k2*a)^2*rho*h*N0^2+8*Z1*cos(k2*a
)*rho^2*Z2*sin(k2*a)*N0^2+8*Z1*cos(k2*a)*rho*Z2*sin(k2*a)*h*N0^2+Z1^2*cos(k2*a)^2*h^2*N0^
2+2*Z1*cos(k2*a)*h^2*Z2*sin(k2*a)*N0^2+4*Z2^2*rho^2*N0^2-
4*Z2^2*rho^2*N0^2*cos(k2*a)^2+4*Z2^2*rho*h*N0^2-
4*Z2^2*rho*h*N0^2*cos(k2*a)^2+Z2^2*h^2*N0^2-Z2^2*h^2*N0^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hh^2+1/2*N0^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE
M_l = mstar*Theta_BF+(rho+1/2*h)*N0;
w0 = (M_l/(N0*Hh))*(Z3+Hh);

function[BF] = BigFunRootCZ_clamp(N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D Dc rho h C Cstar Ce mstar nstar etatilda b astar gamma
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+(Dc/D)*sin(k3-k3*b)*cos(k2*b);
Z4 = k3*sin(k3-k3*b);
Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = mstar*Z1/N0/Hc*k1*sin(k1*a);
y2 = (rho+1/2*h)*Z1/Hc*k1*sin(k1*a);
y3 = 1/2*Z1^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y4 = 1/8*Z1^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y5 = 1/8*Z1^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(-
4*Z2^2*mstar^2*b*k2+4*Z2^2*mstar^2*k2*a+4*Z3^2*mstar^2*k2*a-
4*Z2^2*mstar^2*cos(k2*a)*sin(k2*a)-4*Z3^2*mstar^2*k2*b-
8*Z2*mstar^2*Z3*cos(k2*b)^2+4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar^2*cos(k2*b)*sin(k2*b)+4*Z3^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z2*mstar^2*Z3*cos
(k2*a)^2)/N0^2/Hc^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(-8*Z2^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*b*k2+8*Z2^2*mstar*rho*N0*k2*a+4*Z2^2*mstar*h*N0*k2*a+8*Z3^2*mstar*rho*N
0*k2*a+4*Z3^2*mstar*h*N0*k2*a-8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z2^2*mstar*h*N0*cos(k2*a)*sin(k2*a)-8*Z3^2*mstar*rho*N0*k2*b-4*Z3^2*mstar*h*N0*k2*b-
16*Z2*mstar*Z3*rho*N0*cos(k2*b)^2-
```

```
8*Z2*mstar*Z3*h*N0*cos(k2*b)^2+8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)+4*Z2^2*mstar*h*N0*
cos(k2*b)*sin(k2*b)-8*Z3^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar*h*N0*cos(k2*b)*sin(k2*b)+8*Z3^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z3^2*msta
r*h*N0*cos(k2*a)*sin(k2*a)+8*Z2*mstar*Z3*h*N0*cos(k2*a)^2+16*Z2*mstar*Z3*rho*N0*cos(k2*a)
^2)/N0^2/Hc^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho^2*N0^2*b*k2-4*Z2^2*rho*h*N0^2*b*k2-
Z2^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*k2*a+4*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+4*Z3^
2*rho^2*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho*Z3*h*N0^2*cos(k2*a)^2-
8*Z2*rho^2*Z3*N0^2*cos(k2*b)^2-
4*Z3^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)+4*Z3^2*rho*h*N0^2*k2*a+Z3^2*h^2*N0^2*k2*a-
4*Z3^2*rho^2*N0^2*k2*b-4*Z3^2*rho*h*N0^2*k2*b-Z3^2*h^2*N0^2*k2*b-
8*Z2*rho*Z3*h*N0^2*cos(k2*b)^2-
2*Z2*h^2*Z3*N0^2*cos(k2*b)^2+4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+4*Z2^2*rho^2*N0^2*cos(
k2*b)*sin(k2*b)+Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-4*Z3^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
Z3^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-
Z2^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z3^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho^2*Z3*N0^
2*cos(k2*a)^2+2*Z2*h^2*Z3*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z3^2*h^2
*N0^2*cos(k2*a)*sin(k2*a))/N0^2/Hc^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*cos(k3)*mstar^2*sin(k3)^3-4*mstar^2*k3-
4*cos(k3)*mstar^2*sin(k3)-8*cos(k3)^2*mstar^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*mstar^2*sin(k3)*sin(k3*b)^2+4*mstar^2*cos(k3*b)*sin(k3*b)+8*cos(k3)^3*mstar^2*s
in(k3)+4*mstar^2*k3*b)/N0^2/Hc^2;
y10 = 1/8*(-
1+cos(k1*a)^2)*k3*(16*cos(k3)^3*mstar*rho*N0*sin(k3)+8*cos(k3)^3*mstar*h*N0*sin(k3)+16*co
s(k3)*mstar*sin(k3)^3*rho*N0+8*cos(k3)*mstar*sin(k3)^3*h*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3-8*cos(k3)*mstar*sin(k3)*rho*N0-
4*cos(k3)*mstar*sin(k3)*h*N0+8*mstar*N0*rho*k3*b+4*mstar*N0*h*k3*b-
16*cos(k3)^2*mstar*rho*N0*cos(k3*b)*sin(k3*b)-8*cos(k3)^2*mstar*h*N0*cos(k3*b)*sin(k3*b)-
16*cos(k3)*mstar*sin(k3)*rho*N0*sin(k3*b)^2-
8*cos(k3)*mstar*sin(k3)*h*N0*sin(k3*b)^2+8*mstar*N0*rho*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
cos(k3*b)*sin(k3*b))/N0^2/Hc^2;
y11 = 1/8*(-1+cos(k1*a)^2)*k3*(-
4*cos(k3)*rho*sin(k3)*h*N0^2+8*cos(k3)*rho^2*sin(k3)^3*N0^2+8*cos(k3)*rho*sin(k3)^3*h*N0^
2-4*N0^2*rho^2*k3-
N0^2*h^2*k3+8*cos(k3)^3*rho*h*N0^2*sin(k3)+4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+4*N0^2*rho*h*
k3*b-8*cos(k3)^2*rho^2*N0^2*cos(k3*b)*sin(k3*b)-2*cos(k3)^2*h^2*N0^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*rho^2*sin(k3)*N0^2*sin(k3*b)^2-8*cos(k3)*rho*sin(k3)*h*N0^2*sin(k3*b)^2-
2*cos(k3)*h^2*sin(k3)*N0^2*sin(k3*b)^2-
8*cos(k3)^2*rho*h*N0^2*cos(k3*b)*sin(k3*b)+4*N0^2*rho^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*cos(
k3*b)*sin(k3*b)-
4*N0^2*rho*h*k3+8*cos(k3)^3*rho^2*N0^2*sin(k3)+2*cos(k3)^3*h^2*N0^2*sin(k3)-
cos(k3)*h^2*sin(k3)*N0^2+2*cos(k3)*h^2*sin(k3)^3*N0^2+4*N0^2*rho^2*k3*b+N0^2*h^2*k3*b-
4*cos(k3)*rho^2*sin(k3)*N0^2)/N0^2/Hc^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z2^2*cos(k2*a)^2*mstar^2+8*Z2*cos(k2*a)*mstar^2*Z3*sin(k2*a)+4*Z3^2*mst
ar^2-4*Z3^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/2*Ds*Z1^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hc^2+1/2*etatilda;
Q5 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*mstar*h*N0-
4*Z3^2*mstar*h*N0*cos(k2*a)^2+8*Z2^2*cos(k2*a)^2*mstar*rho*N0+4*Z2^2*cos(k2*a)^2*mstar*h*
N0+16*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*rho*N0+8*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*h*N0+8*Z3^
2*mstar*rho*N0-8*Z3^2*mstar*rho*N0*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hc^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*rho^2*N0^2-
4*Z3^2*rho^2*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2-
4*Z3^2*rho*h*N0^2*cos(k2*a)^2+Z3^2*h^2*N0^2-
Z3^2*h^2*N0^2*cos(k2*a)^2+4*Z2^2*cos(k2*a)^2*rho^2*N0^2+4*Z2^2*cos(k2*a)^2*rho*h*N0^2+8*Z
2*cos(k2*a)*rho^2*Z3*sin(k2*a)*N0^2+8*Z2*cos(k2*a)*rho*Z3*sin(k2*a)*h*N0^2+Z2^2*cos(k2*a)
^2*h^2*N0^2+2*Z2*cos(k2*a)*h^2*Z3*sin(k2*a)*N0^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hc^2+1/2*N0^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
```

```
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE

function[BF] = BigFunRootCZ_hinge(N0)
%Using my hand solution, this function expresses the IC and TC as
%quadratics in theta, then solves for one Big Function of N, a, and theta
global a Ds D Dc rho h C Cstar Ce mstar nstar etatilda b astar gamma
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hh = sin(k1*a)*sin(k3*b-k3)*cos(k2*a-k2*b)+sqrt(Dc/D)*sin(k1*a)*cos(k3*b-k3)*sin(k2*a-
k2*b)-sqrt(Ds/Dc)*cos(k1*a)*sin(k3*b-k3)*sin(k2*a-k2*b)+sqrt(Ds/D)*cos(k1*a)*cos(k3*b-
k3)*cos(k2*a-k2*b);
%Now the Z's from my hand solutions
Z1 = cos(k2*b)*sin(k3*b-k3)-sqrt(Dc/D)*sin(k2*b)*cos(k3*b-k3);
Z2 = sin(k2*b)*sin(k3*b-k3)+sqrt(Dc/D)*cos(k2*b)*cos(k3*b-k3);
Z3 = sqrt(Ds/Dc)*sin(k3*b-k3)*sin(k2*a-k2*b)-sqrt(Ds/D)*cos(k3*b-k3)*cos(k2*a-k2*b);
Z4 = k3*cos(k3*b-k3);
Z5 = sin(k3*b-k3)-Z4*b; %for completeness, we don't need it!
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = -mstar*Z3/N0/Hh*k1*sin(k1*a);
y2 = -(rho+1/2*h)*Z3/Hh*k1*sin(k1*a);
y3 = 1/2*Z3^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y4 = 1/8*Z3^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y5 = 1/8*Z3^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(4*Z1^2*mstar^2*cos(k2*b)*sin(k2*b)-
8*Z1*mstar^2*Z2*cos(k2*b)^2-4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-4*Z2^2*mstar^2*k2*b-
4*Z1^2*mstar^2*b*k2+4*Z1^2*mstar^2*k2*a+4*Z2^2*mstar^2*k2*a-
4*Z1^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z1*mstar^2*Z2*cos(k2*a)^2+4*Z2^2*mstar^2*cos(k2*a)*s
in(k2*a))/N0^2/Hh^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(8*Z1^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
8*Z1*mstar*Z2*h*N0*cos(k2*b)^2+4*Z1^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-
16*Z1*mstar*Z2*rho*N0*cos(k2*b)^2-8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z2^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-8*Z2^2*mstar*rho*N0*k2*b-4*Z1^2*mstar*h*N0*b*k2-
8*Z1^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*k2*b+8*Z1^2*mstar*rho*N0*k2*a+4*Z1^2*mstar*h*N0*k2*a+8*Z2^2*mstar*rho*N
0*k2*a+4*Z2^2*mstar*h*N0*k2*a+16*Z1*mstar*Z2*rho*N0*cos(k2*a)^2+8*Z1*mstar*Z2*h*N0*cos(k2
*a)^2-8*Z1^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z1^2*mstar*h*N0*cos(k2*a)*sin(k2*a)+8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z2^2*msta
r*h*N0*cos(k2*a)*sin(k2*a))/N0^2/Hh^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho*h*N0^2*k2*b-Z2^2*h^2*N0^2*k2*b-
Z1^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-8*Z1*rho*Z2*h*N0^2*cos(k2*b)^2-
2*Z1*h^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
8*Z1*rho^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+Z1^2*h^2*N0^2*cos(k2
*b)*sin(k2*b)+4*Z1^2*rho*h*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho*h*N0^2*b*k2-4*Z1^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho^2*N0^2*b*k2+4*Z1^2*rho^2*N0^2*k2*a+Z1^2*h^2*N0^2*k2*a+4*Z2^2*rho^2*N0^2*k2*a+4
*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+8*Z1*rho^2*Z2*N0^2*cos(k2*a)^2+8*Z1*rho*Z2*h*N0^
2*cos(k2*a)^2+2*Z1*h^2*Z2*N0^2*cos(k2*a)^2-4*Z1^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)-
Z1^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z2^2*h^2*N0^2*cos
(k2*a)*sin(k2*a)-4*Z2^2*rho^2*N0^2*k2*b)/N0^2/Hh^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*mstar^2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*b)^2+4*mstar^2*b*k3+4*sin(k3)*mstar^2*cos(k3)-
4*mstar^2*k3-4*mstar^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
y10 = 1/8*(-1+cos(k1*a)^2)*k3*(16*mstar*N0*rho*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*b)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*b)^2-
8*mstar*N0*rho*cos(k3*b)*sin(k3*b)-
4*mstar*N0*h*cos(k3*b)*sin(k3*b)+8*mstar*N0*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
b*k3+8*mstar*N0*rho*b*k3+8*sin(k3)*mstar*cos(k3)*rho*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/Hh^2;
y11 = 1/8*(-
1+cos(k1*a)^2)*k3*(4*N0^2*rho^2*b*k3+8*N0^2*rho^2*cos(k3)^2*cos(k3*b)*sin(k3*b)+2*N0^2*h^
2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+8*N0^2*rho*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*b*k3
-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*b)^2-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*b)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*b)^2+4*N0^2*rho*h*b*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-
```

```
4*N0^2*rho*h*k3+4*sin(k3)*rho*cos(k3)*h*N0^2+sin(k3)*h^2*cos(k3)*N0^2-N0^2*h^2*k3-
4*N0^2*rho^2*k3-N0^2*h^2*cos(k3*b)*sin(k3*b)-4*N0^2*rho^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*mstar^2+8*Z1*cos(k2*a)*mstar^2*Z2*sin(k2*a)+4*Z2^2*mst
ar^2-4*Z2^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/2*Ds*Z3^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hh^2+1/2*etatilda;
Q5 = -1/8*Dc*(-
1+cos(k1*a)^2)*(8*Z1^2*cos(k2*a)^2*mstar*rho*N0+4*Z1^2*cos(k2*a)^2*mstar*h*N0+16*Z1*cos(k
2*a)*mstar*Z2*sin(k2*a)*rho*N0+8*Z1*cos(k2*a)*mstar*Z2*sin(k2*a)*h*N0+8*Z2^2*mstar*rho*N0
-8*Z2^2*mstar*rho*N0*cos(k2*a)^2+4*Z2^2*mstar*h*N0-
4*Z2^2*mstar*h*N0*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hh^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*rho^2*N0^2+4*Z1^2*cos(k2*a)^2*rho*h*N0^2+8*Z1*cos(k2*a
)*rho^2*Z2*sin(k2*a)*N0^2+8*Z1*cos(k2*a)*rho*Z2*sin(k2*a)*h*N0^2+Z1^2*cos(k2*a)^2*h^2*N0^
2+2*Z1*cos(k2*a)*h^2*Z2*sin(k2*a)*N0^2+4*Z2^2*rho^2*N0^2-
4*Z2^2*rho^2*N0^2*cos(k2*a)^2+4*Z2^2*rho*h*N0^2-
4*Z2^2*rho*h*N0^2*cos(k2*a)^2+Z2^2*h^2*N0^2-Z2^2*h^2*N0^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hh^2+1/2*N0^2/Ce;
Q6_bar = Q6-2*gamma; %Absorb the 2y into the constant Q6_bar
%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6_bar*Q1-Q3*Q4;
Theta_BF = -Q8/Q7;
BF = Q4*(Theta_BF^2)+Q5*Theta_BF+Q6_bar; %Subbing solution into TE

function[Hout] = HCZ_clamp(N0)
%Clamped Fixed CZ H function
global a b Ds D Dc
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
Hout = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);

function[Hout] = HCZ_hinge(N0)
%Hinged fixed CZ H function
global a b Ds D Dc
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
Hout = sin(k1*a)*sin(k3*b-k3)*cos(k2*a-k2*b)+sqrt(Dc/D)*sin(k1*a)*cos(k3*b-k3)*sin(k2*a-
k2*b)-sqrt(Ds/Dc)*cos(k1*a)*sin(k3*b-k3)*sin(k2*a-k2*b)+sqrt(Ds/D)*cos(k1*a)*cos(k3*b-
k3)*cos(k2*a-k2*b);

function[Thetapg,w0,err,F,PotEn,ValidCZ] = PreGrowthCZ_clamp(a,astar,N0)
%Pregrowth CFIX CX 1
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda b
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-sqrt(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+sqrt(Dc/D)*sin(k3-k3*b)*cos(k2*b);
Z4 = k3*sin(k3-k3*b);
Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
```

```
y1 = mstar*Z1/N0/Hc*k1*sin(k1*a);
y2 = (rho+1/2*h)*Z1/Hc*k1*sin(k1*a);
y3 = 1/2*Z1^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y4 = 1/8*Z1^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y5 = 1/8*Z1^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(-
4*Z2^2*mstar^2*b*k2+4*Z2^2*mstar^2*k2*a+4*Z3^2*mstar^2*k2*a-
4*Z2^2*mstar^2*cos(k2*a)*sin(k2*a)-4*Z3^2*mstar^2*k2*b-
8*Z2*mstar^2*Z3*cos(k2*b)^2+4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar^2*cos(k2*b)*sin(k2*b)+4*Z3^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z2*mstar^2*Z3*cos
(k2*a)^2)/N0^2/Hc^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(-8*Z2^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*b*k2+8*Z2^2*mstar*rho*N0*k2*a+4*Z2^2*mstar*h*N0*k2*a+8*Z3^2*mstar*rho*N
0*k2*a+4*Z3^2*mstar*h*N0*k2*a-8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z2^2*mstar*h*N0*cos(k2*a)*sin(k2*a)-8*Z3^2*mstar*rho*N0*k2*b-4*Z3^2*mstar*h*N0*k2*b-
16*Z2*mstar*Z3*rho*N0*cos(k2*b)^2-
8*Z2*mstar*Z3*h*N0*cos(k2*b)^2+8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)+4*Z2^2*mstar*h*N0*
cos(k2*b)*sin(k2*b)-8*Z3^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar*h*N0*cos(k2*b)*sin(k2*b)+8*Z3^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z3^2*msta
r*h*N0*cos(k2*a)*sin(k2*a)+8*Z2*mstar*Z3*h*N0*cos(k2*a)^2+16*Z2*mstar*Z3*rho*N0*cos(k2*a)
^2)/N0^2/Hc^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho^2*N0^2*b*k2-4*Z2^2*rho*h*N0^2*b*k2-
Z2^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*k2*a+4*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+4*Z3^
2*rho^2*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho*Z3*h*N0^2*cos(k2*a)^2-
8*Z2*rho^2*Z3*N0^2*cos(k2*b)^2-
4*Z3^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)+4*Z3^2*rho*h*N0^2*k2*a+Z3^2*h^2*N0^2*k2*a-
4*Z3^2*rho^2*N0^2*k2*b-4*Z3^2*rho*h*N0^2*k2*b-Z3^2*h^2*N0^2*k2*b-
8*Z2*rho*Z3*h*N0^2*cos(k2*b)^2-
2*Z2*h^2*Z3*N0^2*cos(k2*b)^2+4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+4*Z2^2*rho^2*N0^2*cos(
k2*b)*sin(k2*b)+Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-4*Z3^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
Z3^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-
Z2^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z3^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho^2*Z3*N0^
2*cos(k2*a)^2+2*Z2*h^2*Z3*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z3^2*h^2
*N0^2*cos(k2*a)*sin(k2*a))/N0^2/Hc^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*cos(k3)*mstar^2*sin(k3)^3-4*mstar^2*k3-
4*cos(k3)*mstar^2*sin(k3)-8*cos(k3)^2*mstar^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*mstar^2*sin(k3)*sin(k3*b)^2+4*mstar^2*cos(k3*b)*sin(k3*b)+8*cos(k3)^3*mstar^2*s
in(k3)+4*mstar^2*k3*b)/N0^2/Hc^2;
y10 = 1/8*(-
1+cos(k1*a)^2)*k3*(16*cos(k3)^3*mstar*rho*N0*sin(k3)+8*cos(k3)^3*mstar*h*N0*sin(k3)+16*co
s(k3)*mstar*sin(k3)^3*rho*N0+8*cos(k3)*mstar*sin(k3)^3*h*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3-8*cos(k3)*mstar*sin(k3)*rho*N0-
4*cos(k3)*mstar*sin(k3)*h*N0+8*mstar*N0*rho*k3*b+4*mstar*N0*h*k3*b-
16*cos(k3)^2*mstar*rho*N0*cos(k3*b)*sin(k3*b)-8*cos(k3)^2*mstar*h*N0*cos(k3*b)*sin(k3*b)-
16*cos(k3)*mstar*sin(k3)*rho*N0*sin(k3*b)^2-
8*cos(k3)*mstar*sin(k3)*h*N0*sin(k3*b)^2+8*mstar*N0*rho*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
cos(k3*b)*sin(k3*b))/N0^2/Hc^2;
y11 = 1/8*(-1+cos(k1*a)^2)*k3*(-
4*cos(k3)*rho*sin(k3)*h*N0^2+8*cos(k3)*rho^2*sin(k3)^3*N0^2+8*cos(k3)*rho*sin(k3)^3*h*N0^
2-4*N0^2*rho^2*k3-
N0^2*h^2*k3+8*cos(k3)^3*rho*h*N0^2*sin(k3)+4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+4*N0^2*rho*h*
k3*b-8*cos(k3)^2*rho^2*N0^2*cos(k3*b)*sin(k3*b)-2*cos(k3)^2*h^2*N0^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*rho^2*sin(k3)*N0^2*sin(k3*b)^2-8*cos(k3)*rho*sin(k3)*h*N0^2*sin(k3*b)^2-
2*cos(k3)*h^2*sin(k3)*N0^2*sin(k3*b)^2-
8*cos(k3)^2*rho*h*N0^2*cos(k3*b)*sin(k3*b)+4*N0^2*rho^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*cos(
k3*b)*sin(k3*b)-
4*N0^2*rho*h*k3+8*cos(k3)^3*rho^2*N0^2*sin(k3)+2*cos(k3)^3*h^2*N0^2*sin(k3)-
cos(k3)*h^2*sin(k3)*N0^2+2*cos(k3)*h^2*sin(k3)^3*N0^2+4*N0^2*rho^2*k3*b+N0^2*h^2*k3*b-
4*cos(k3)*rho^2*sin(k3)*N0^2)/N0^2/Hc^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
Thetapg = (-Q2+sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %1st root!
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = cos(k3*(1-b));
B2 = sqrt(Dc/D)*sin(k3*(1-b));
```

```
F = (1/4)*(k1*sin(2*k1*a)*(Z1/Hc)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hc)^2+2*k2*B1*B2*(cos(2*k2*(b-a))-1)*(sin(k1*a)/Hc)^2+k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hc)^2);
w0 = M_l*(Hc-Z1-sin(k1*a))/(N0*Hc);
upordown = sign(w0); % + means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z2^2*cos(k2*a)^2*mstar^2+8*Z2*cos(k2*a)*mstar^2*Z3*sin(k2*a)+4*Z3^2*mst
ar^2-4*Z3^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/2*Ds*Z1^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hc^2+1/2*etatilda;
Q5 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*mstar*h*N0-
4*Z3^2*mstar*h*N0*cos(k2*a)^2+8*Z2^2*cos(k2*a)^2*mstar*rho*N0+4*Z2^2*cos(k2*a)^2*mstar*h*
N0+16*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*rho*N0+8*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*h*N0+8*Z3^
2*mstar*rho*N0-8*Z3^2*mstar*rho*N0*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hc^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*rho^2*N0^2-
4*Z3^2*rho^2*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2-
4*Z3^2*rho*h*N0^2*cos(k2*a)^2+Z3^2*h^2*N0^2-
Z3^2*h^2*N0^2*cos(k2*a)^2+4*Z2^2*cos(k2*a)^2*rho^2*N0^2+4*Z2^2*cos(k2*a)^2*rho*h*N0^2+8*Z
2*cos(k2*a)*rho^2*Z3*sin(k2*a)*N0^2+8*Z2*cos(k2*a)*rho*Z3*sin(k2*a)*h*N0^2+Z2^2*cos(k2*a)
^2*h^2*N0^2+2*Z2*cos(k2*a)*h^2*Z3*sin(k2*a)*N0^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hc^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants (clamped)
A1 = -(M_l*Z1)/(N0*Hc);
B1 = 0;
C1 = 0;
D1 = M_l*(Hc-sin(k1*a))/(N0*Hc);
A2 = (M_l*Z2*sin(k1*a))/(N0*Hc);
B2 = (M_l*Z3*sin(k1*a))/(N0*Hc);
C2 = 0;
D2 = -(M_l*sin(k1*a))/(N0*Hc);
A3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hc);
B3 = (M_l*sin(k3)*sin(k1*a))/(N0*Hc);
C3 = 0;
D3 = -(M_l*sin(k1*a))/(N0*Hc);
%Ap3 = 0;
%Bp3 = 0;
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hc);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hc);
% CZ Validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.01:b);
%x3 = (b:0.01:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
slopeata = -A1*k1*sin(k1*a)+B1*k1*cos(k1*a)+C1; %New
for j2 = 1:length(x2);
    w2star(j2) = -(A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2);
    flap(j2) = -(slopeata*(x2(j2)-a)-w2star(1)); %New, signs messed w/
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
end
%for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
%    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
%end
%--------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0;
      ValidCZa(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZa(ja) = 1;
    end
  end

  if find(ValidCZa,1)>0;%1 valid, 0 not valid again
      for jp=1:length(x2); %Check for penetration
```

```
          if flap(jp)<w2star(jp);
              flaptest(jp)=1; %Valid!
          else
              flaptest(jp)=0; %not valid
          end
      end

      if find(flaptest)>0
          ValidCZ = 1;
      else
          ValidCZ = 0;
      end
  else
    ValidCZ = 1;
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
        ValidCZ = 0;
%---------------------
elseif upordown > 0;    %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0; %was <=
      ValidCZb(jb) = 0; %0 valid, 1 not valid
    else
      ValidCZb(jb) = 1;
     end
  end

  if find(ValidCZb,1)>0; %1 valid, 0 not valid again
    if sum(ValidCZb) == length(ValidCZb);
      ValidCZ = 0;
    else
      ValidCZ = 2; %Propagating Contact
    end
  else
    ValidCZ = 1;
  end
%----------------------
else
    ValidCZ = 1;
end
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;

function[Thetapg,w0,err,F,PotEn,ValidCZ] = PreGrowthCZ_hinge(a,astar,N0)
%Pregrowth HFIX CZ 1
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda b
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hh = sin(k1*a)*sin(k3*b-k3)*cos(k2*a-k2*b)+sqrt(Dc/D)*sin(k1*a)*cos(k3*b-k3)*sin(k2*a-
k2*b)-sqrt(Ds/Dc)*cos(k1*a)*sin(k3*b-k3)*sin(k2*a-k2*b)+sqrt(Ds/D)*cos(k1*a)*cos(k3*b-
k3)*cos(k2*a-k2*b);
%Now the Z's from my hand solutions
Z1 = cos(k2*b)*sin(k3*b-k3)-sqrt(Dc/D)*sin(k2*b)*cos(k3*b-k3);
Z2 = sin(k2*b)*sin(k3*b-k3)+sqrt(Dc/D)*cos(k2*b)*cos(k3*b-k3);
Z3 = sqrt(Ds/Dc)*sin(k3*b-k3)*sin(k2*a-k2*b)-sqrt(Ds/D)*cos(k3*b-k3)*cos(k2*a-k2*b);
Z4 = k3*cos(k3*b-k3);
Z5 = sin(k3*b-k3)-Z4*b; %for completeness, we don't need it!
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
```

```
y1 = -mstar*Z3/N0/Hh*k1*sin(k1*a);
y2 = -(rho+1/2*h)*Z3/Hh*k1*sin(k1*a);
y3 = 1/2*Z3^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y4 = 1/8*Z3^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y5 = 1/8*Z3^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(4*Z1^2*mstar^2*cos(k2*b)*sin(k2*b)-
8*Z1*mstar^2*Z2*cos(k2*b)^2-4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-4*Z2^2*mstar^2*k2*b-
4*Z1^2*mstar^2*b*k2+4*Z1^2*mstar^2*k2*a+4*Z2^2*mstar^2*k2*a-
4*Z1^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z1*mstar^2*Z2*cos(k2*a)^2+4*Z2^2*mstar^2*cos(k2*a)*s
in(k2*a))/N0^2/Hh^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(8*Z1^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
8*Z1*mstar*Z2*h*N0*cos(k2*b)^2+4*Z1^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-
16*Z1*mstar*Z2*rho*N0*cos(k2*b)^2-8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z2^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-8*Z2^2*mstar*rho*N0*k2*b-4*Z1^2*mstar*h*N0*b*k2-
8*Z1^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*k2*b+8*Z1^2*mstar*rho*N0*k2*a+4*Z1^2*mstar*h*N0*k2*a+8*Z2^2*mstar*rho*N
0*k2*a+4*Z2^2*mstar*h*N0*k2*a+16*Z1*mstar*Z2*rho*N0*cos(k2*a)^2+8*Z1*mstar*Z2*h*N0*cos(k2
*a)^2-8*Z1^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z1^2*mstar*h*N0*cos(k2*a)*sin(k2*a)+8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z2^2*msta
r*h*N0*cos(k2*a)*sin(k2*a))/N0^2/Hh^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho*h*N0^2*k2*b-Z2^2*h^2*N0^2*k2*b-
Z1^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-8*Z1*rho*Z2*h*N0^2*cos(k2*b)^2-
2*Z1*h^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
8*Z1*rho^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+Z1^2*h^2*N0^2*cos(k2
*b)*sin(k2*b)+4*Z1^2*rho*h*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho*h*N0^2*b*k2-4*Z1^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho^2*N0^2*b*k2+4*Z1^2*rho^2*N0^2*k2*a+Z1^2*h^2*N0^2*k2*a+4*Z2^2*rho^2*N0^2*k2*a+4
*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+8*Z1*rho^2*Z2*N0^2*cos(k2*a)^2+8*Z1*rho*Z2*h*N0^
2*cos(k2*a)^2+2*Z1*h^2*Z2*N0^2*cos(k2*a)^2-4*Z1^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)-
Z1^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z2^2*h^2*N0^2*cos
(k2*a)*sin(k2*a)-4*Z2^2*rho^2*N0^2*k2*b)/N0^2/Hh^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*mstar^2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*b)^2+4*mstar^2*b*k3+4*sin(k3)*mstar^2*cos(k3)-
4*mstar^2*k3-4*mstar^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
y10 = 1/8*(-1+cos(k1*a)^2)*k3*(16*mstar*N0*rho*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*b)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*b)^2-
8*mstar*N0*rho*cos(k3*b)*sin(k3*b)-
4*mstar*N0*h*cos(k3*b)*sin(k3*b)+8*mstar*N0*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
b*k3+8*mstar*N0*rho*b*k3+8*sin(k3)*mstar*cos(k3)*rho*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/Hh^2;
y11 = 1/8*(-
1+cos(k1*a)^2)*k3*(4*N0^2*rho^2*b*k3+8*N0^2*rho^2*cos(k3)^2*cos(k3*b)*sin(k3*b)+2*N0^2*h^
2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+8*N0^2*rho*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*b*k3
-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*b)^2-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*b)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*b)^2+4*N0^2*rho*h*b*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-
4*N0^2*rho*h*k3+4*sin(k3)*rho*cos(k3)*h*N0^2+sin(k3)*h^2*cos(k3)*N0^2-N0^2*h^2*k3-
4*N0^2*rho^2*k3-N0^2*h^2*cos(k3*b)*sin(k3*b)-4*N0^2*rho^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
Thetapg = (-Q2+sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %1st root!
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = sin(k3*(1-b));
B2 = sqrt(Dc/D)*cos(k3*(1-b));
F = (1/4)*(k1*sin(2*k1*a)*(Z3/Hh)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hh)^2+2*k2*B1*B2*(1-cos(2*k2*(b-a)))*(sin(k1*a)/Hh)^2-k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hh)^2);
w0 = (M_l/(N0*Hh))*(Z3+Hh);
upordown = sign(w0); % + means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*mstar^2+8*Z1*cos(k2*a)*mstar^2*Z2*sin(k2*a)+4*Z2^2*mst
ar^2-4*Z2^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/2*Ds*Z3^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hh^2+1/2*etatilda;
```

```
Q5 = -1/8*Dc*(-
1+cos(k1*a)^2)*(8*Z1^2*cos(k2*a)^2*mstar*rho*N0+4*Z1^2*cos(k2*a)^2*mstar*h*N0+16*Z1*cos(k
2*a)*mstar*Z2*sin(k2*a)*rho*N0+8*Z1*cos(k2*a)*mstar*Z2*sin(k2*a)*h*N0+8*Z2^2*mstar*rho*N0
-8*Z2^2*mstar*rho*N0*cos(k2*a)^2+4*Z2^2*mstar*h*N0-
4*Z2^2*mstar*h*N0*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hh^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*rho^2*N0^2+4*Z1^2*cos(k2*a)^2*rho*h*N0^2+8*Z1*cos(k2*a
)*rho^2*Z2*sin(k2*a)*N0^2+8*Z1*cos(k2*a)*rho*Z2*sin(k2*a)*h*N0^2+Z1^2*cos(k2*a)^2*h^2*N0^
2+2*Z1*cos(k2*a)*h^2*Z2*sin(k2*a)*N0^2+4*Z2^2*rho^2*N0^2-
4*Z2^2*rho^2*N0^2*cos(k2*a)^2+4*Z2^2*rho*h*N0^2-
4*Z2^2*rho*h*N0^2*cos(k2*a)^2+Z2^2*h^2*N0^2-Z2^2*h^2*N0^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hh^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants of integration for deflections(hinged)
A1 = (M_l*Z3)/(N0*Hh);
B1 = 0;
C1 = 0;
D1 = M_l/N0;
A2 = (M_l*Z1*sin(k1*a))/(N0*Hh);
B2 = (M_l*Z2*sin(k1*a))/(N0*Hh);
C2 = 0;
D2 = 0;
A3 = -(M_l*sin(k3)*sin(k1*a))/(N0*Hh);
B3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hh);
C3 = 0;
D3 = 0;
%Ap3 = 0;
%Bp3 = 0;
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hh);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hh);
%CZ validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.01:b);
%x3 = (b:0.01:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
for j2 = 1:length(x2);
    %w2star(j2) = A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2;
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
end
%for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
%    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
%end
%--------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0;
      ValidCZa(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZa(ja) = 1;
    end
  end

  if find(ValidCZa,1)>0;%1 valid, 0 not valid again
    ValidCZ = 0;
  else
    ValidCZ = 1;
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
       ValidCZ = 0;
%---------------------
elseif upordown > 0;    %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0; %was <=
      ValidCZb(jb) = 0; %0 valid, 1 not valid
    else
```

```
      ValidCZb(jb) = 1;
     end
  end

  if find(ValidCZb,1)>0; %1 valid, 0 not valid again
    ValidCZ = 0;
  else
    ValidCZ = 1;
  end
%----------------------
else
    ValidCZ = 0;
end
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;

function[Thetapg,w0,err,F,PotEn,ValidCZ2] = PreGrowthCZ_clamp2(a,astar,N0)
%Pregrowth CFIX CX 2
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda b
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-sqrt(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+sqrt(Dc/D)*sin(k3-k3*b)*cos(k2*b);
Z4 = k3*sin(k3-k3*b);
Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = mstar*Z1/N0/Hc*k1*sin(k1*a);
y2 = (rho+1/2*h)*Z1/Hc*k1*sin(k1*a);
y3 = 1/2*Z1^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y4 = 1/8*Z1^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y5 = 1/8*Z1^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(-
4*Z2^2*mstar^2*b*k2+4*Z2^2*mstar^2*k2*a+4*Z3^2*mstar^2*k2*a-
4*Z2^2*mstar^2*cos(k2*a)*sin(k2*a)-4*Z3^2*mstar^2*k2*b-
8*Z2*mstar^2*Z3*cos(k2*b)^2+4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar^2*cos(k2*b)*sin(k2*b)+4*Z3^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z2*mstar^2*Z3*cos
(k2*a)^2)/N0^2/Hc^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(-8*Z2^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*b*k2+8*Z2^2*mstar*rho*N0*k2*a+4*Z2^2*mstar*h*N0*k2*a+8*Z3^2*mstar*rho*N
0*k2*a+4*Z3^2*mstar*h*N0*k2*a-8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z2^2*mstar*h*N0*cos(k2*a)*sin(k2*a)-8*Z3^2*mstar*rho*N0*k2*b-4*Z3^2*mstar*h*N0*k2*b-
16*Z2*mstar*Z3*rho*N0*cos(k2*b)^2-
8*Z2*mstar*Z3*h*N0*cos(k2*b)^2+8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)+4*Z2^2*mstar*h*N0*
cos(k2*b)*sin(k2*b)-8*Z3^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar*h*N0*cos(k2*b)*sin(k2*b)+8*Z3^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z3^2*msta
r*h*N0*cos(k2*a)*sin(k2*a)+8*Z2*mstar*Z3*h*N0*cos(k2*a)^2+16*Z2*mstar*Z3*rho*N0*cos(k2*a)
^2)/N0^2/Hc^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho^2*N0^2*b*k2-4*Z2^2*rho*h*N0^2*b*k2-
Z2^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*k2*a+4*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+4*Z3^
2*rho^2*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho*Z3*h*N0^2*cos(k2*a)^2-
8*Z2*rho^2*Z3*N0^2*cos(k2*b)^2-
4*Z3^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)+4*Z3^2*rho*h*N0^2*k2*a+Z3^2*h^2*N0^2*k2*a-
```

```
4*Z3^2*rho^2*N0^2*k2*b-4*Z3^2*rho*h*N0^2*k2*b-Z3^2*h^2*N0^2*k2*b-
8*Z2*rho*Z3*h*N0^2*cos(k2*b)^2-
2*Z2*h^2*Z3*N0^2*cos(k2*b)^2+4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+4*Z2^2*rho^2*N0^2*cos(
k2*b)*sin(k2*b)+Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-4*Z3^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
Z3^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-
Z2^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z3^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho^2*Z3*N0^
2*cos(k2*a)^2+2*Z2*h^2*Z3*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z3^2*h^2
*N0^2*cos(k2*a)*sin(k2*a))/N0^2/Hc^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*cos(k3)*mstar^2*sin(k3)^3-4*mstar^2*k3-
4*cos(k3)*mstar^2*sin(k3)-8*cos(k3)^2*mstar^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*mstar^2*sin(k3)*sin(k3*b)^2+4*mstar^2*cos(k3*b)*sin(k3*b)+8*cos(k3)^3*mstar^2*s
in(k3)+4*mstar^2*k3*b)/N0^2/Hc^2;
y10 = 1/8*(-
1+cos(k1*a)^2)*k3*(16*cos(k3)^3*mstar*rho*N0*sin(k3)+8*cos(k3)^3*mstar*h*N0*sin(k3)+16*co
s(k3)*mstar*sin(k3)^3*rho*N0+8*cos(k3)*mstar*sin(k3)^3*h*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3-8*cos(k3)*mstar*sin(k3)*rho*N0-
4*cos(k3)*mstar*sin(k3)*h*N0+8*mstar*N0*rho*k3*b+4*mstar*N0*h*k3*b-
16*cos(k3)^2*mstar*rho*N0*cos(k3*b)*sin(k3*b)-8*cos(k3)^2*mstar*h*N0*cos(k3*b)*sin(k3*b)-
16*cos(k3)*mstar*sin(k3)*rho*N0*sin(k3*b)^2-
8*cos(k3)*mstar*sin(k3)*h*N0*sin(k3*b)^2+8*mstar*N0*rho*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
cos(k3*b)*sin(k3*b))/N0^2/Hc^2;
y11 = 1/8*(-1+cos(k1*a)^2)*k3*(-
4*cos(k3)*rho*sin(k3)*h*N0^2+8*cos(k3)*rho^2*sin(k3)^3*N0^2+8*cos(k3)*rho*sin(k3)^3*h*N0^
2-4*N0^2*rho^2*k3-
N0^2*h^2*k3+8*cos(k3)^3*rho*h*N0^2*sin(k3)+4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+4*N0^2*rho*h*
k3*b-8*cos(k3)^2*rho^2*N0^2*cos(k3*b)*sin(k3*b)-2*cos(k3)^2*h^2*N0^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*rho^2*sin(k3)*N0^2*sin(k3*b)^2-8*cos(k3)*rho*sin(k3)*h*N0^2*sin(k3*b)^2-
2*cos(k3)*h^2*sin(k3)*N0^2*sin(k3*b)^2-
8*cos(k3)^2*rho*h*N0^2*cos(k3*b)*sin(k3*b)+4*N0^2*rho^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*cos(
k3*b)*sin(k3*b)-
4*N0^2*rho*h*k3+8*cos(k3)^3*rho^2*N0^2*sin(k3)+2*cos(k3)^3*h^2*N0^2*sin(k3)-
cos(k3)*h^2*sin(k3)*N0^2+2*cos(k3)*h^2*sin(k3)^3*N0^2+4*N0^2*rho^2*k3*b+N0^2*h^2*k3*b-
4*cos(k3)*rho^2*sin(k3)*N0^2)/N0^2/Hc^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
Thetapg = (-Q2-sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %2nd root!
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = cos(k3*(1-b));
B2 = sqrt(Dc/D)*sin(k3*(1-b));
F = (1/4)*(k1*sin(2*k1*a)*(Z1/Hc)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hc)^2+2*k2*B1*B2*(cos(2*k2*(b-a))-1)*(sin(k1*a)/Hc)^2+k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hc)^2);
w0 = M_l*(Hc-Z1-sin(k1*a))/(N0*Hc);
upordown = sign(w0); % + means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z2^2*cos(k2*a)^2*mstar^2+8*Z2*cos(k2*a)*mstar^2*Z3*sin(k2*a)+4*Z3^2*mst
ar^2-4*Z3^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/2*Ds*Z1^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hc^2+1/2*etatilda;
Q5 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*mstar*h*N0-
4*Z3^2*mstar*h*N0*cos(k2*a)^2+8*Z2^2*cos(k2*a)^2*mstar*rho*N0+4*Z2^2*cos(k2*a)^2*mstar*h*
N0+16*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*rho*N0+8*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*h*N0+8*Z3^
2*mstar*rho*N0-8*Z3^2*mstar*rho*N0*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hc^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*rho^2*N0^2-
4*Z3^2*rho^2*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2-
4*Z3^2*rho*h*N0^2*cos(k2*a)^2+Z3^2*h^2*N0^2-
Z3^2*h^2*N0^2*cos(k2*a)^2+4*Z2^2*cos(k2*a)^2*rho^2*N0^2+4*Z2^2*cos(k2*a)^2*rho*h*N0^2+8*Z
2*cos(k2*a)*rho^2*Z3*sin(k2*a)*N0^2+8*Z2*cos(k2*a)*rho*Z3*sin(k2*a)*h*N0^2+Z2^2*cos(k2*a)
^2*h^2*N0^2+2*Z2*cos(k2*a)*h^2*Z3*sin(k2*a)*N0^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hc^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants (clamped)
A1 = -(M_l*Z1)/(N0*Hc);
B1 = 0;
C1 = 0;
D1 = M_l*(Hc-sin(k1*a))/(N0*Hc);
```

```
A2 = (M_l*Z2*sin(k1*a))/(N0*Hc);
B2 = (M_l*Z3*sin(k1*a))/(N0*Hc);
C2 = 0;
D2 = -(M_l*sin(k1*a))/(N0*Hc);
A3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hc);
B3 = (M_l*sin(k3)*sin(k1*a))/(N0*Hc);
C3 = 0;
D3 = -(M_l*sin(k1*a))/(N0*Hc);
%Ap3 = 0;
%Bp3 = 0;
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hc);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hc);
% CZ Validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.01:b);
%x3 = (b:0.01:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
slopeata = -A1*k1*sin(k1*a)+B1*k1*cos(k1*a)+C1; %New
for j2 = 1:length(x2);
    w2star(j2) = -(A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2);
    flap(j2) = -(slopeata*(x2(j2)-a)-w2star(1)); %New, signs messed w/
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
end
%for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
%    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
%end
%--------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0;
      ValidCZ2a(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZ2a(ja) = 1;
    end
  end

  if find(ValidCZ2a,1)>0;%1 valid, 0 not valid again
    for jp=1:length(x2); %Check for penetration
          if flap(jp)<w2star(jp);
              flaptest2(jp)=1; %Valid!
              else
              flaptest2(jp)=0; %not valid
          end
      end

      if find(flaptest2)>0
          ValidCZ2 = 1;
      else
          ValidCZ2 = 0;
      end
  else
    ValidCZ2 = 1;
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
        ValidCZ2 = 0;
%--------------------
elseif upordown > 0;    %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0;
      ValidCZ2b(jb) = 0; %0 valid, 1 not valid
    else
      ValidCZ2b(jb) = 1;
     end
  end

  if find(ValidCZ2b,1)>0; %1 valid, 0 not valid again
```

```
    if sum(ValidCZ2b) == length(ValidCZ2b);
      ValidCZ2 = 0;
    else
      ValidCZ2 = 2; %Propagating Contact
    end
  else
    ValidCZ2 = 1;
  end
%----------------------
else                          %should be no "else"
    ValidCZ2 = 1;
end
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;

function[Thetapg,w0,err,F,PotEn,ValidCZ2] = PreGrowthCZ_hinge2(a,astar,N0)
% pregrowth HFIX CZ 2
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda b
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hh = sin(k1*a)*sin(k3*b-k3)*cos(k2*a-k2*b)+sqrt(Dc/D)*sin(k1*a)*cos(k3*b-k3)*sin(k2*a-
k2*b)-sqrt(Ds/Dc)*cos(k1*a)*sin(k3*b-k3)*sin(k2*a-k2*b)+sqrt(Ds/D)*cos(k1*a)*cos(k3*b-
k3)*cos(k2*a-k2*b);
%Now the Z's from my hand solutions
Z1 = cos(k2*b)*sin(k3*b-k3)-sqrt(Dc/D)*sin(k2*b)*cos(k3*b-k3);
Z2 = sin(k2*b)*sin(k3*b-k3)+sqrt(Dc/D)*cos(k2*b)*cos(k3*b-k3);
Z3 = sqrt(Ds/Dc)*sin(k3*b-k3)*sin(k2*a-k2*b)-sqrt(Ds/D)*cos(k3*b-k3)*cos(k2*a-k2*b);
Z4 = k3*cos(k3*b-k3);
Z5 = sin(k3*b-k3)-Z4*b; %for completeness, we don't need it!
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = -mstar*Z3/N0/Hh*k1*sin(k1*a);
y2 = -(rho+1/2*h)*Z3/Hh*k1*sin(k1*a);
y3 = 1/2*Z3^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y4 = 1/8*Z3^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y5 = 1/8*Z3^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(4*Z1^2*mstar^2*cos(k2*b)*sin(k2*b)-
8*Z1*mstar^2*Z2*cos(k2*b)^2-4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-4*Z2^2*mstar^2*k2*b-
4*Z1^2*mstar^2*b*k2+4*Z1^2*mstar^2*k2*a+4*Z2^2*mstar^2*k2*a-
4*Z1^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z1*mstar^2*Z2*cos(k2*a)^2+4*Z2^2*mstar^2*cos(k2*a)*s
in(k2*a))/N0^2/Hh^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(8*Z1^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
8*Z1*mstar*Z2*h*N0*cos(k2*b)^2+4*Z1^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-
16*Z1*mstar*Z2*rho*N0*cos(k2*b)^2-8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z2^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-8*Z2^2*mstar*rho*N0*k2*b-4*Z1^2*mstar*h*N0*b*k2-
8*Z1^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*k2*b+8*Z1^2*mstar*rho*N0*k2*a+4*Z1^2*mstar*h*N0*k2*a+8*Z2^2*mstar*rho*N
0*k2*a+4*Z2^2*mstar*h*N0*k2*a+16*Z1*mstar*Z2*rho*N0*cos(k2*a)^2+8*Z1*mstar*Z2*h*N0*cos(k2
*a)^2-8*Z1^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z1^2*mstar*h*N0*cos(k2*a)*sin(k2*a)+8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z2^2*msta
r*h*N0*cos(k2*a)*sin(k2*a))/N0^2/Hh^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho*h*N0^2*k2*b-Z2^2*h^2*N0^2*k2*b-
Z1^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-8*Z1*rho*Z2*h*N0^2*cos(k2*b)^2-
2*Z1*h^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
8*Z1*rho^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+Z1^2*h^2*N0^2*cos(k2
*b)*sin(k2*b)+4*Z1^2*rho*h*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho*h*N0^2*b*k2-4*Z1^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
```

```
4*Z1^2*rho^2*N0^2*b*k2+4*Z1^2*rho^2*N0^2*k2*a+Z1^2*h^2*N0^2*k2*a+4*Z2^2*rho^2*N0^2*k2*a+4
*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+8*Z1*rho^2*Z2*N0^2*cos(k2*a)^2+8*Z1*rho*Z2*h*N0^
2*cos(k2*a)^2+2*Z1*h^2*Z2*N0^2*cos(k2*a)^2-4*Z1^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)-
Z1^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z2^2*h^2*N0^2*cos
(k2*a)*sin(k2*a)-4*Z2^2*rho^2*N0^2*k2*b)/N0^2/Hh^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*mstar^2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*b)^2+4*mstar^2*b*k3+4*sin(k3)*mstar^2*cos(k3)-
4*mstar^2*k3-4*mstar^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
y10 = 1/8*(-1+cos(k1*a)^2)*k3*(16*mstar*N0*rho*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*b)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*b)^2-
8*mstar*N0*rho*cos(k3*b)*sin(k3*b)-
4*mstar*N0*h*cos(k3*b)*sin(k3*b)+8*mstar*N0*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
b*k3+8*mstar*N0*rho*b*k3+8*sin(k3)*mstar*cos(k3)*rho*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/Hh^2;
y11 = 1/8*(-
1+cos(k1*a)^2)*k3*(4*N0^2*rho^2*b*k3+8*N0^2*rho^2*cos(k3)^2*cos(k3*b)*sin(k3*b)+2*N0^2*h^
2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+8*N0^2*rho*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*b*k3
-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*b)^2-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*b)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*b)^2+4*N0^2*rho*h*b*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-
4*N0^2*rho*h*k3+4*sin(k3)*rho*cos(k3)*h*N0^2+sin(k3)*h^2*cos(k3)*N0^2-N0^2*h^2*k3-
4*N0^2*rho^2*k3-N0^2*h^2*cos(k3*b)*sin(k3*b)-4*N0^2*rho^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
Thetapg = (-Q2-sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %2nd root!
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = sin(k3*(1-b));
B2 = sqrt(Dc/D)*cos(k3*(1-b));
F = (1/4)*(k1*sin(2*k1*a)*(Z3/Hh)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hh)^2+2*k2*B1*B2*(1-cos(2*k2*(b-a)))*(sin(k1*a)/Hh)^2-k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hh)^2);
w0 = (M_l/(N0*Hh))*(Z3+Hh);
upordown = sign(w0); %+ means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*mstar^2+8*Z1*cos(k2*a)*mstar^2*Z2*sin(k2*a)+4*Z2^2*mst
ar^2-4*Z2^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/2*Ds*Z3^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hh^2+1/2*etatilda;
Q5 = -1/8*Dc*(-
1+cos(k1*a)^2)*(8*Z1^2*cos(k2*a)^2*mstar*rho*N0+4*Z1^2*cos(k2*a)^2*mstar*h*N0+16*Z1*cos(k
2*a)*mstar*Z2*sin(k2*a)*rho*N0+8*Z1*cos(k2*a)*mstar*Z2*sin(k2*a)*h*N0+8*Z2^2*mstar*rho*N0
-8*Z2^2*mstar*rho*N0*cos(k2*a)^2+4*Z2^2*mstar*h*N0-
4*Z2^2*mstar*h*N0*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hh^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*rho^2*N0^2+4*Z1^2*cos(k2*a)^2*rho*h*N0^2+8*Z1*cos(k2*a
)*rho^2*Z2*sin(k2*a)*N0^2+8*Z1*cos(k2*a)*rho*Z2*sin(k2*a)*h*N0^2+Z1^2*cos(k2*a)^2*h^2*N0^
2+2*Z1*cos(k2*a)*h^2*Z2*sin(k2*a)*N0^2+4*Z2^2*rho^2*N0^2-
4*Z2^2*rho^2*N0^2*cos(k2*a)^2+4*Z2^2*rho*h*N0^2-
4*Z2^2*rho*h*N0^2*cos(k2*a)^2+Z2^2*h^2*N0^2-Z2^2*h^2*N0^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hh^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants of integration for deflections(hinged)
A1 = (M_l*Z3)/(N0*Hh);
B1 = 0;
C1 = 0;
D1 = M_l/N0;
A2 = (M_l*Z1*sin(k1*a))/(N0*Hh);
B2 = (M_l*Z2*sin(k1*a))/(N0*Hh);
C2 = 0;
D2 = 0;
A3 = -(M_l*sin(k3)*sin(k1*a))/(N0*Hh);
B3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hh);
C3 = 0;
D3 = 0;
%Ap3 = 0;
%Bp3 = 0;
```

```
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hh);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hh);
%CZ validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.01:b);
%x3 = (b:0.01:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
for j2 = 1:length(x2);
    %w2star(j2) = A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2;
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
    %K2sign(j2) = sign(kappa2(j2
end
%for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
%    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
%end
%-------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0;
      ValidCZ2a(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZ2a(ja) = 1;
    end
  end

  if find(ValidCZ2a,1)>0;%1 valid, 0 not valid again
    ValidCZ2 = 0;
  else
    ValidCZ2 = 1;
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
        ValidCZ2 = 0;
%---------------------
elseif upordown > 0;    %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0;
      ValidCZ2b(jb) = 0; %0 valid, 1 not valid
    else
      ValidCZ2b(jb) = 1;
     end
  end

  if find(ValidCZ2b,1)>0; %1 valid, 0 not valid again
    ValidCZ2 = 0;
  else
    ValidCZ2 = 1;
  end
%----------------------
else                          %should be no "else"
    ValidCZ2 = 0;
End
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;

function[a_1,b_1,c_1,d_1,a_2,b_2,c_2,d_2,a_3,b_3,c_3,d_3,k1,k2,k3] =
ConstantsCZ_clamp(Nn,TH,aa) %No "P" for now...
% Uses the hand solution to find the const. of integ.
```

```
global Ds D Dc b mstar rho h
k1 = sqrt(Nn/Ds);
k2 = sqrt(Nn/Dc);
k3 = sqrt(Nn/D);
%First, the Z's:
z_1 = sqrt(Ds/Dc)*sin(k2*b-k2*aa)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*aa)*sin(k3-k3*b);
z_2 = cos(k3-k3*b)*cos(k2*b)-sqrt(Dc/D)*sin(k3-k3*b)*sin(k2*b);
z_3 = cos(k3-k3*b)*sin(k2*b)+sqrt(Dc/D)*sin(k3-k3*b)*cos(k2*b);
%z_4 = k3*sin(k3-k3*b);
%z_5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
hc = sin(k1*aa)*cos(k2*b-k2*aa)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*aa)*sin(k2*b-
k2*aa)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k1*aa)*cos(k2*b-k2*aa)*sin(k3-k3*b)-
sqrt(Dc/D)*sin(k1*aa)*sin(k2*b-k2*aa)*sin(k3-k3*b);
mlambda = mstar*TH+(rho+(h/2))*Nn;
a_1 = -(mlambda*z_1)/(Nn*hc);
b_1 = 0;
c_1 = 0;
d_1 = (mlambda/(Nn*hc))*(hc-sin(k1*aa));
a_2 = (mlambda*z_2*sin(k1*aa))/(Nn*hc);
b_2 = (mlambda*z_3*sin(k1*aa))/(Nn*hc);
c_2 = 0;
d_2 = -(mlambda*sin(k1*aa))/(Nn*hc);
a_3 = (mlambda*sin(k1*aa)*cos(k3))/(Nn*hc);
b_3 = (mlambda*sin(k1*aa)*sin(k3))/(Nn*hc);
c_3 = 0;
d_3 = -(mlambda*sin(k1*aa))/(Nn*hc);
%For completeness, but not needed
%a_p3 = 0;
%b_p3 = 0;
%c_p3 = (mlambda*z_4*sin(k1*aa))/(Nn*hc);
%d_p3 = (mlambda*z_5*sin(k1*aa))/(Nn*hc);

function[a_1,b_1,c_1,d_1,a_2,b_2,c_2,d_2,a_3,b_3,c_3,d_3,k1,k2,k3] =
ConstantsCZ_hinge(Nn,TH,aa) %No "P" for now...
% Uses the hand solution to find the const. of integ.
global Ds D Dc b mstar rho h
k1 = sqrt(Nn/Ds);
k2 = sqrt(Nn/Dc);
k3 = sqrt(Nn/D);
%First, the Z's:
z_1 = cos(k2*b)*sin(k3*(b-1))-sqrt(Dc/D)*sin(k2*b)*cos(k3*(b-1));
z_2 = sin(k2*b)*sin(k3*(b-1))+sqrt(Dc/D)*cos(k2*b)*cos(k3*(b-1));
z_3 = sqrt(Ds/Dc)*sin(k3*(b-1))*sin(k2*(aa-b))-sqrt(Ds/D)*cos(k3*(b-1))*cos(k2*(aa-b));
%z_4 = k3*cos(k3*(b-1));
%z_5 = sin(k3*(b-1))-z_4*b;
hh = sin(k1*aa)*sin(k3*(b-1))*cos(k2*(aa-b))+sqrt(Dc/D)*sin(k1*aa)*cos(k3*(b-
1))*sin(k2*(aa-b))-sqrt(Ds/Dc)*cos(k1*aa)*sin(k3*(b-1))*sin(k2*(aa-
b))+sqrt(Ds/D)*cos(k1*aa)*cos(k3*(b-1))*cos(k2*(aa-b));
mlambda = mstar*TH+(rho+(h/2))*Nn;
a_1 = (mlambda*z_3)/(Nn*hh);
b_1 = 0;
c_1 = 0;
d_1 = mlambda/Nn;
a_2 = (mlambda*z_1*sin(k1*aa))/(Nn*hh);
b_2 = (mlambda*z_2*sin(k1*aa))/(Nn*hh);
c_2 = 0;
d_2 = 0;
a_3 = -(mlambda*sin(k1*aa)*sin(k3))/(Nn*hh);
b_3 = (mlambda*sin(k1*aa)*cos(k3))/(Nn*hh);
c_3 = 0;
d_3 = 0;
%For completeness, but not needed
%a_p3 = 0;
%b_p3 = 0;
%c_p3 = (mlambda*z_4*sin(k1*aa))/(Nn*hh);
%d_p3 = (mlambda*z_5*sin(k1*aa))/(Nn*hh);
```

Alternate version for model of partially debonded structure:

```
% I shall call this one TempControlCZ
```

```
% I am now prescribing temperature and solving for N from the IC, although
% this is more computationally intensive, it allows for me to hold the
% temperature value constant and look at the energy release rates vs a*
%We begin by defining the usual suspects
%The first version of this program is being written for HINGED and CLAMPED
%   CONTACT ZONE
clear
global h C D Dp Cp
h = 0.05; % height of baseplate
hp = 0.05; % height of patch
h0 = hp/h; % ratio of heights
E0 = 1; % elastic modulus
C = 12/(h^2); % membrane stiffness of baseplate
Cp = C*E0*h0; % membrane stiffness of patch
D = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rho Ds
% stiffnesses of composite structure
Astar = D+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rho = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rho*Bstar; %Dstar
global  Dc Cs Ce
Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar nstar etatilda alphaP
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 0.5; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rho*nstar;
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
global a astar Theta_in b
%Choose values for a and Theta
a_vals = (0.01:0.01:0.90);
a_vals_star = 1 - a_vals;
TH_vals = (0:0.0002:.018);%Limit for Hinge =0.009, Clamp=0.018
N_vals = (0:0.1:80);%Limit for hinge=60, clamp=80
b = 0.9;
%Find (a,TH,N) triples
%n=1; %Initialize for result array
p=0;
for i = 1:length(a_vals);
    p=p+1;
    n=1;
    a = a_vals(i);
    astar = 1-a_vals(i);
    for j = 1:length(TH_vals); %within this loop, find N roots
        Theta_in = TH_vals(j);
        for k = 1:length(N_vals)
            INTCON(k) = Integrability_clamp_CZ(N_vals(k)); %
HINGE/CLAMP
            %[value of integrability, we want it to be zero]
        end
        for m = 1:length(N_vals)-1;
            if INTCON(m)*INTCON(m+1)<0;%Look for a sign change
                Nroot(m)=fzero(@Integrability_clamp_CZ,N_vals(m));%Root of N
HINGE/CLAMP
                TEST_nosort(n,1,p)=Theta_in; %Unsorted Data
                TEST_nosort(n,2,p)=Nroot(m);
                n=n+1;
            else
            end
        end
    end
end
%Sort the data by order of the roots of N
```

```
for ii=1:length(a_vals);
    TEST(:,:,ii)=sortrows(TEST_nosort(:,:,ii));
end
%Find the other values associates w/ the triples
for r=1:length(a_vals);
    qq=1; %reset for new "a" value
    for q=1:length(TEST(:,1,1));
        DATA(q,1,r)=TEST(q,1,r);
        DATA(q,2,r)=TEST(q,2,r);%                          HINGE/CLAMP (right below)

[DATA(q,3,r),DATA(q,4,r),DATA(q,5,r),DATA(q,6,r),DATA(q,7,r)]=Evolving_clamp_CZ(TEST(q,2,
r),TEST(q,1,r),a_vals(r));
        % Form of DATA [Theta,N,F,w0,ERR,PE,Valid]
        if DATA(q,3,r)>0;
            StableDATA(qq,:,r)=DATA(q,:,r);
            qq=qq+1;
        else
        end
    end
end
%Clean Data sports one root of N (the lowest) for each value of theta
%For each value of a, the vectors are the exact same length
for tt=1:length(a_vals);
    rr=1;
    for pp = 2:length(TH_vals);
            nn=1;
        for jj=1:length(StableDATA(:,1,1))
            if StableDATA(jj,1,tt)==TH_vals(pp);
                test2(nn,1:7)=StableDATA(jj,1:7,tt);
                nn=nn+1;
            end
        end
        CleanDATA(rr,:,tt)=test2(1,:);%First one is lowest N root
        rr=rr+1;
    end
end
%Now pluck out of the Clean Data, those points which are Valid!
for iii=1:length(a_vals)
    kkk=1;
    for jjj=1:length(CleanDATA(:,1,1))
        if CleanDATA(jjj,7,iii)>0 && CleanDATA(jjj,7,iii) <2;
            ValidDATA(kkk,:,iii)=CleanDATA(jjj,:,iii);
            kkk=kkk+1;
        else
            ValidDATA(kkk,:,iii)=zeros(1,7);
            kkk=kkk+1;
        end
    end
end
%------------------------
%save H_CZ.mat CleanDATA ValidDATA;
save C_CZ.mat CleanDATA ValidDATA;
% ~(((((((((^>
%PLOTS
test=80; %a value
figure %theta vs N
plot(DATA(:,1,test),DATA(:,2,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,1,test),StableDATA(:,2,test),'ro','LineWidth',2)
hold on
plot(ValidDATA(:,1,test),ValidDATA(:,2,test),'g+','LineWidth',2)
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
grid on
figure %PE vs theta
plot(DATA(:,1,test),DATA(:,6,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,1,test),StableDATA(:,6,test),'ro','LineWidth',2)
hold on
plot(ValidDATA(:,1,test),ValidDATA(:,6,test),'g+','LineWidth',2)
```

```
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('P','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
axis([0 0.03 0 1.5])
grid on
figure %F vs theta
plot(DATA(:,1,test),DATA(:,3,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,1,test),StableDATA(:,3,test),'ro','LineWidth',2)
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('F','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
axis([0 0.03 -100 100])
grid on
figure %theta vs wo
plot(DATA(:,4,test),DATA(:,1,test),'k.','LineWidth',2)
hold on
plot(StableDATA(:,4,test),StableDATA(:,1,test),'ro','LineWidth',2)
hold on
plot(ValidDATA(:,4,test),ValidDATA(:,1,test),'g+','LineWidth',2)
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',16,'fontweight','b')
title(['a=' num2str(a_vals(test))],'fontsize',16,'fontweight','b')
axis([-0.2 0.25 0 0.03])
grid on
%-------------------------------------------------------------
%Trying something new...
%load=[2,4,6,8,10,12,14];  %HINGE
load=[5,10,15,20,25,30,35];  %CLAMP
for ugh2=1:length(load)
    for ugh=1:length(a_vals)
        ERRplot(ugh,ugh2)=CleanDATA(load(ugh2),5,ugh);
        ValidERRplot(ugh,ugh2)=ValidDATA(load(ugh2),5,ugh);
    end
end
figure
plot(a_vals_star',ERRplot(:,1),'k.','LineWidth',2)
hold on
plot(a_vals_star',ValidERRplot(:,1),'ko','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,2),'b.','LineWidth',2)
hold on
plot(a_vals_star',ValidERRplot(:,2),'bo','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,3),'r.','LineWidth',2)
hold on
plot(a_vals_star',ValidERRplot(:,3),'ro','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,4),'g.','LineWidth',2)
hold on
plot(a_vals_star',ValidERRplot(:,4),'go','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,5),'c.','LineWidth',2)
hold on
plot(a_vals_star',ValidERRplot(:,5),'co','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,6),'m.','LineWidth',2)
hold on
plot(a_vals_star',ValidERRplot(:,6),'mo','LineWidth',2)
hold on
plot(a_vals_star',ERRplot(:,7),'y.','LineWidth',2)
hold on
plot(a_vals_star',ValidERRplot(:,7),'yo','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('ERR','fontsize',16,'fontweight','b')
title('ERR vs. a*','fontsize',16,'fontweight','b')
grid on
```

Associated subroutines for alternate version:

```
function[ICvalue] = Integrability_clamp_CZ(N0)
%Using my hand solution, this function expresses the IC
%I plug in values of theta and look for values of N that satisfy IC=0
% CZ solution!!!!
global Ds D Dc rho h C Cstar mstar nstar a astar b Theta_in
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-sqrt(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+sqrt(Dc/D)*sin(k3-k3*b)*cos(k2*b);
%Z4 = k3*sin(k3-k3*b);
%Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = mstar*Z1/N0/Hc*k1*sin(k1*a);
y2 = (rho+1/2*h)*Z1/Hc*k1*sin(k1*a);
y3 = 1/2*Z1^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y4 = 1/8*Z1^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y5 = 1/8*Z1^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(-
4*Z2^2*mstar^2*b*k2+4*Z2^2*mstar^2*k2*a+4*Z3^2*mstar^2*k2*a-
4*Z2^2*mstar^2*cos(k2*a)*sin(k2*a)-4*Z3^2*mstar^2*k2*b-
8*Z2*mstar^2*Z3*cos(k2*b)^2+4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar^2*cos(k2*b)*sin(k2*b)+4*Z3^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z2*mstar^2*Z3*cos
(k2*a)^2)/N0^2/Hc^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(-8*Z2^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*b*k2+8*Z2^2*mstar*rho*N0*k2*a+4*Z2^2*mstar*h*N0*k2*a+8*Z3^2*mstar*rho*N
0*k2*a+4*Z3^2*mstar*h*N0*k2*a-8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z2^2*mstar*h*N0*cos(k2*a)*sin(k2*a)-8*Z3^2*mstar*rho*N0*k2*b-4*Z3^2*mstar*h*N0*k2*b-
16*Z2*mstar*Z3*rho*N0*cos(k2*b)^2-
8*Z2*mstar*Z3*h*N0*cos(k2*b)^2+8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)+4*Z2^2*mstar*h*N0*
cos(k2*b)*sin(k2*b)-8*Z3^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar*h*N0*cos(k2*b)*sin(k2*b)+8*Z3^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z3^2*msta
r*h*N0*cos(k2*a)*sin(k2*a)+8*Z2*mstar*Z3*h*N0*cos(k2*a)^2+16*Z2*mstar*Z3*rho*N0*cos(k2*a)
^2)/N0^2/Hc^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho^2*N0^2*b*k2-4*Z2^2*rho*h*N0^2*b*k2-
Z2^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*k2*a+4*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+4*Z3^
2*rho^2*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho*Z3*h*N0^2*cos(k2*a)^2-
8*Z2*rho^2*Z3*N0^2*cos(k2*b)^2-
4*Z3^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)+4*Z3^2*rho*h*N0^2*k2*a+Z3^2*h^2*N0^2*k2*a-
4*Z3^2*rho^2*N0^2*k2*b-4*Z3^2*rho*h*N0^2*k2*b-Z3^2*h^2*N0^2*k2*b-
8*Z2*rho*Z3*h*N0^2*cos(k2*b)^2-
2*Z2*h^2*Z3*N0^2*cos(k2*b)^2+4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+4*Z2^2*rho^2*N0^2*cos(
k2*b)*sin(k2*b)+Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-4*Z3^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
Z3^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-
Z2^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z3^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho^2*Z3*N0^
2*cos(k2*a)^2+2*Z2*h^2*Z3*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z3^2*h^2
*N0^2*cos(k2*a)*sin(k2*a))/N0^2/Hc^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*cos(k3)*mstar^2*sin(k3)^3-4*mstar^2*k3-
4*cos(k3)*mstar^2*sin(k3)-8*cos(k3)^2*mstar^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*mstar^2*sin(k3)*sin(k3*b)^2+4*mstar^2*cos(k3*b)*sin(k3*b)+8*cos(k3)^3*mstar^2*s
in(k3)+4*mstar^2*k3*b)/N0^2/Hc^2;
y10 = 1/8*(-
1+cos(k1*a)^2)*k3*(16*cos(k3)^3*mstar*rho*N0*sin(k3)+8*cos(k3)^3*mstar*h*N0*sin(k3)+16*co
s(k3)*mstar*sin(k3)^3*rho*N0+8*cos(k3)*mstar*sin(k3)^3*h*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3-8*cos(k3)*mstar*sin(k3)*rho*N0-
4*cos(k3)*mstar*sin(k3)*h*N0+8*mstar*N0*rho*k3*b+4*mstar*N0*h*k3*b-
16*cos(k3)^2*mstar*rho*N0*cos(k3*b)*sin(k3*b)-8*cos(k3)^2*mstar*h*N0*cos(k3*b)*sin(k3*b)-
16*cos(k3)*mstar*sin(k3)*rho*N0*sin(k3*b)^2-
8*cos(k3)*mstar*sin(k3)*h*N0*sin(k3*b)^2+8*mstar*N0*rho*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
cos(k3*b)*sin(k3*b))/N0^2/Hc^2;
```

```
y11 = 1/8*(-1+cos(k1*a)^2)*k3*(-
4*cos(k3)*rho*sin(k3)*h*N0^2+8*cos(k3)*rho^2*sin(k3)^3*N0^2+8*cos(k3)*rho*sin(k3)^3*h*N0^
2-4*N0^2*rho^2*k3-
N0^2*h^2*k3+8*cos(k3)^3*rho*h*N0^2*sin(k3)+4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+4*N0^2*rho*h*
k3*b-8*cos(k3)^2*rho^2*N0^2*cos(k3*b)*sin(k3*b)-2*cos(k3)^2*h^2*N0^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*rho^2*sin(k3)*N0^2*sin(k3*b)^2-8*cos(k3)*rho*sin(k3)*h*N0^2*sin(k3*b)^2-
2*cos(k3)*h^2*sin(k3)*N0^2*sin(k3*b)^2-
8*cos(k3)^2*rho*h*N0^2*cos(k3*b)*sin(k3*b)+4*N0^2*rho^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*cos(
k3*b)*sin(k3*b)-
4*N0^2*rho*h*k3+8*cos(k3)^3*rho^2*N0^2*sin(k3)+2*cos(k3)^3*h^2*N0^2*sin(k3)-
cos(k3)*h^2*sin(k3)*N0^2+2*cos(k3)*h^2*sin(k3)^3*N0^2+4*N0^2*rho^2*k3*b+N0^2*h^2*k3*b-
4*cos(k3)*rho^2*sin(k3)*N0^2)/N0^2/Hc^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
ICvalue = Q1*Theta_in^2+Q2*Theta_in+Q3;

function[ICvalue] = Integrability_hinge_CZ(N0)
%Using my hand solution, this function expresses the IC
%I plug in values of theta and look for values of N that satisfy IC=0
% CZ solution!!!!
global Ds D Dc rho h C Cstar mstar nstar a astar b Theta_in
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hh = sin(k1*a)*sin(k3*b-k3)*cos(k2*a-k2*b)+sqrt(Dc/D)*sin(k1*a)*cos(k3*b-k3)*sin(k2*a-
k2*b)-sqrt(Ds/Dc)*cos(k1*a)*sin(k3*b-k3)*sin(k2*a-k2*b)+sqrt(Ds/D)*cos(k1*a)*cos(k3*b-
k3)*cos(k2*a-k2*b);
%Now the Z's from my hand solutions
Z1 = cos(k2*b)*sin(k3*b-k3)-sqrt(Dc/D)*sin(k2*b)*cos(k3*b-k3);
Z2 = sin(k2*b)*sin(k3*b-k3)+sqrt(Dc/D)*cos(k2*b)*cos(k3*b-k3);
Z3 = sqrt(Ds/Dc)*sin(k3*b-k3)*sin(k2*a-k2*b)-sqrt(Ds/D)*cos(k3*b-k3)*cos(k2*a-k2*b);
Z4 = k3*cos(k3*b-k3);
%Z5 = sin(k3*b-k3)-Z4*b; %for completeness, we don't need it!
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = -mstar*Z3/N0/Hh*k1*sin(k1*a);
y2 = -(rho+1/2*h)*Z3/Hh*k1*sin(k1*a);
y3 = 1/2*Z3^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y4 = 1/8*Z3^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y5 = 1/8*Z3^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hh^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(4*Z1^2*mstar^2*cos(k2*b)*sin(k2*b)-
8*Z1*mstar^2*Z2*cos(k2*b)^2-4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-4*Z2^2*mstar^2*k2*b-
4*Z1^2*mstar^2*b*k2+4*Z1^2*mstar^2*k2*a+4*Z2^2*mstar^2*k2*a-
4*Z1^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z1*mstar^2*Z2*cos(k2*a)^2+4*Z2^2*mstar^2*cos(k2*a)*s
in(k2*a))/N0^2/Hh^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(8*Z1^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
8*Z1*mstar*Z2*h*N0*cos(k2*b)^2+4*Z1^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-
16*Z1*mstar*Z2*rho*N0*cos(k2*b)^2-8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z2^2*mstar*h*N0*cos(k2*b)*sin(k2*b)-8*Z2^2*mstar*rho*N0*k2*b-4*Z1^2*mstar*h*N0*b*k2-
8*Z1^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*k2*b+8*Z1^2*mstar*rho*N0*k2*a+4*Z1^2*mstar*h*N0*k2*a+8*Z2^2*mstar*rho*N
0*k2*a+4*Z2^2*mstar*h*N0*k2*a+16*Z1*mstar*Z2*rho*N0*cos(k2*a)^2+8*Z1*mstar*Z2*h*N0*cos(k2
*a)^2-8*Z1^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z1^2*mstar*h*N0*cos(k2*a)*sin(k2*a)+8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z2^2*msta
r*h*N0*cos(k2*a)*sin(k2*a))/N0^2/Hh^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho*h*N0^2*k2*b-Z2^2*h^2*N0^2*k2*b-
Z1^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-8*Z1*rho*Z2*h*N0^2*cos(k2*b)^2-
2*Z1*h^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
8*Z1*rho^2*Z2*N0^2*cos(k2*b)^2+4*Z1^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+Z1^2*h^2*N0^2*cos(k2
*b)*sin(k2*b)+4*Z1^2*rho*h*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho*h*N0^2*b*k2-4*Z1^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
4*Z1^2*rho^2*N0^2*b*k2+4*Z1^2*rho^2*N0^2*k2*a+Z1^2*h^2*N0^2*k2*a+4*Z2^2*rho^2*N0^2*k2*a+4
*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+8*Z1*rho^2*Z2*N0^2*cos(k2*a)^2+8*Z1*rho*Z2*h*N0^
2*cos(k2*a)^2+2*Z1*h^2*Z2*N0^2*cos(k2*a)^2-4*Z1^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)-
```

```
Z1^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z2^2*h^2*N0^2*cos
(k2*a)*sin(k2*a)-4*Z2^2*rho^2*N0^2*k2*b)/N0^2/Hh^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*mstar^2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar^2*cos(k3)*cos(k3*b)^2+4*mstar^2*b*k3+4*sin(k3)*mstar^2*cos(k3)-
4*mstar^2*k3-4*mstar^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
y10 = 1/8*(-1+cos(k1*a)^2)*k3*(16*mstar*N0*rho*cos(k3)^2*cos(k3*b)*sin(k3*b)-
8*sin(k3)*mstar*cos(k3)*h*N0*cos(k3*b)^2-16*sin(k3)*mstar*cos(k3)*rho*N0*cos(k3*b)^2-
8*mstar*N0*rho*cos(k3*b)*sin(k3*b)-
4*mstar*N0*h*cos(k3*b)*sin(k3*b)+8*mstar*N0*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
b*k3+8*mstar*N0*rho*b*k3+8*sin(k3)*mstar*cos(k3)*rho*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3+4*sin(k3)*mstar*cos(k3)*h*N0)/N0^2/Hh^2;
y11 = 1/8*(-
1+cos(k1*a)^2)*k3*(4*N0^2*rho^2*b*k3+8*N0^2*rho^2*cos(k3)^2*cos(k3*b)*sin(k3*b)+2*N0^2*h^
2*cos(k3)^2*cos(k3*b)*sin(k3*b)-
4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+8*N0^2*rho*h*cos(k3)^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*b*k3
-8*sin(k3)*rho*cos(k3)*h*N0^2*cos(k3*b)^2-2*sin(k3)*h^2*cos(k3)*N0^2*cos(k3*b)^2-
8*sin(k3)*rho^2*cos(k3)*N0^2*cos(k3*b)^2+4*N0^2*rho*h*b*k3+4*sin(k3)*rho^2*cos(k3)*N0^2-
4*N0^2*rho*h*k3+4*sin(k3)*rho*cos(k3)*h*N0^2+sin(k3)*h^2*cos(k3)*N0^2-N0^2*h^2*k3-
4*N0^2*rho^2*k3-N0^2*h^2*cos(k3*b)*sin(k3*b)-4*N0^2*rho^2*cos(k3*b)*sin(k3*b))/N0^2/Hh^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
ICvalue = Q1*Theta_in^2+Q2*Theta_in+Q3;

function[F,w0,err,PotEn,ValidCZ] = Evolving_clamp_CZ(N0,Thetapg,a)
%This function finds other values that i want,given a,N and TH
% CZ
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda b
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-sqrt(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+sqrt(Dc/D)*sin(k3-k3*b)*cos(k2*b);
%Z4 = k3*sin(k3-k3*b);
%Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = cos(k3*(1-b));
B2 = sqrt(Dc/D)*sin(k3*(1-b));
F = (1/4)*(k1*sin(2*k1*a)*(Z1/Hc)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hc)^2+2*k2*B1*B2*(cos(2*k2*(b-a))-1)*(sin(k1*a)/Hc)^2+k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hc)^2);
w0 = M_l*(Hc-Z1-sin(k1*a))/(N0*Hc);
upordown = sign(w0); % + means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z2^2*cos(k2*a)^2*mstar^2+8*Z2*cos(k2*a)*mstar^2*Z3*sin(k2*a)+4*Z3^2*mst
ar^2-4*Z3^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/2*Ds*Z1^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hc^2+1/2*etatilda;
Q5 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*mstar*h*N0-
4*Z3^2*mstar*h*N0*cos(k2*a)^2+8*Z2^2*cos(k2*a)^2*mstar*rho*N0+4*Z2^2*cos(k2*a)^2*mstar*h*
N0+16*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*rho*N0+8*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*h*N0+8*Z3^
2*mstar*rho*N0-8*Z3^2*mstar*rho*N0*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hc^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*rho^2*N0^2-
4*Z3^2*rho^2*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2-
4*Z3^2*rho*h*N0^2*cos(k2*a)^2+Z3^2*h^2*N0^2-
Z3^2*h^2*N0^2*cos(k2*a)^2+4*Z2^2*cos(k2*a)^2*rho^2*N0^2+4*Z2^2*cos(k2*a)^2*rho*h*N0^2+8*Z
2*cos(k2*a)*rho^2*Z3*sin(k2*a)*N0^2+8*Z2*cos(k2*a)*rho*Z3*sin(k2*a)*h*N0^2+Z2^2*cos(k2*a)
^2*h^2*N0^2+2*Z2*cos(k2*a)*h^2*Z3*sin(k2*a)*N0^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hc^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants (clamped)
```

```
A1 = -(M_l*Z1)/(N0*Hc);
B1 = 0;
C1 = 0;
D1 = M_l*(Hc-sin(k1*a))/(N0*Hc);
A2 = (M_l*Z2*sin(k1*a))/(N0*Hc);
B2 = (M_l*Z3*sin(k1*a))/(N0*Hc);
C2 = 0;
D2 = -(M_l*sin(k1*a))/(N0*Hc);
A3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hc);
B3 = (M_l*sin(k3)*sin(k1*a))/(N0*Hc);
C3 = 0;
D3 = -(M_l*sin(k1*a))/(N0*Hc);
%Ap3 = 0;
%Bp3 = 0;
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hc);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hc);
% CZ Validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.01:b);
%x3 = (b:0.01:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
slopeata = -A1*k1*sin(k1*a)+B1*k1*cos(k1*a)+C1; %New
for j2 = 1:length(x2);
    w2star(j2) = -(A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2);
    flap(j2) = -(slopeata*(x2(j2)-a)-w2star(1)); %New, signs messed w/
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
end
%for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
%    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
%end
%--------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0;
      ValidCZa(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZa(ja) = 1;
    end
  end

  if find(ValidCZa,1)>0;%1 valid, 0 not valid again
      for jp=1:length(x2); %Check for penetration
          if flap(jp)<w2star(jp);
              flaptest(jp)=1; %Valid!
          else
              flaptest(jp)=0; %not valid
          end
      end

      if find(flaptest)>0
          ValidCZ = 1;
      else
          ValidCZ = 0;
      end
  else
    ValidCZ = 1;
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
        ValidCZ = 0;
%---------------------
elseif upordown > 0;    %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0; %was <=
      ValidCZb(jb) = 0; %0 valid, 1 not valid
    else
      ValidCZb(jb) = 1;
```

```
    end
  end

  if find(ValidCZb,1)>0; %1 valid, 0 not valid again
    if sum(ValidCZb) == length(ValidCZb);
      ValidCZ = 0;
    else
      ValidCZ = 2; %Propagating Contact
    end
  else
    ValidCZ = 1;
  end
%----------------------
else
    ValidCZ = 1;
end
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;

function[F,w0,err,PotEn,ValidCZ] = Evolving_hinge_CZ(N0,Thetapg,a)
%This function finds other values that i want,given a,N and TH
% CZ
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda b
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hh = sin(k1*a)*sin(k3*b-k3)*cos(k2*a-k2*b)+sqrt(Dc/D)*sin(k1*a)*cos(k3*b-k3)*sin(k2*a-
k2*b)-sqrt(Ds/Dc)*cos(k1*a)*sin(k3*b-k3)*sin(k2*a-k2*b)+sqrt(Ds/D)*cos(k1*a)*cos(k3*b-
k3)*cos(k2*a-k2*b);
%Now the Z's from my hand solutions
Z1 = cos(k2*b)*sin(k3*b-k3)-sqrt(Dc/D)*sin(k2*b)*cos(k3*b-k3);
Z2 = sin(k2*b)*sin(k3*b-k3)+sqrt(Dc/D)*cos(k2*b)*cos(k3*b-k3);
Z3 = sqrt(Ds/Dc)*sin(k3*b-k3)*sin(k2*a-k2*b)-sqrt(Ds/D)*cos(k3*b-k3)*cos(k2*a-k2*b);
%Z4 = k3*cos(k3*b-k3);
%Z5 = sin(k3*b-k3)-Z4*b; %for completeness, we don't need it!
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = sin(k3*(1-b));
B2 = sqrt(Dc/D)*cos(k3*(1-b));
F = (1/4)*(k1*sin(2*k1*a)*(Z3/Hh)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hh)^2+2*k2*B1*B2*(1-cos(2*k2*(b-a)))*(sin(k1*a)/Hh)^2-k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hh)^2);
w0 = (M_l/(N0*Hh))*(Z3+Hh);
upordown = sign(w0); % + means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*mstar^2+8*Z1*cos(k2*a)*mstar^2*Z2*sin(k2*a)+4*Z2^2*mst
ar^2-4*Z2^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/2*Ds*Z3^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hh^2+1/2*etatilda;
Q5 = -1/8*Dc*(-
1+cos(k1*a)^2)*(8*Z1^2*cos(k2*a)^2*mstar*rho*N0+4*Z1^2*cos(k2*a)^2*mstar*h*N0+16*Z1*cos(k
2*a)*mstar*Z2*sin(k2*a)*rho*N0+8*Z1*cos(k2*a)*mstar*Z2*sin(k2*a)*h*N0+8*Z2^2*mstar*rho*N0
-8*Z2^2*mstar*rho*N0*cos(k2*a)^2+4*Z2^2*mstar*h*N0-
4*Z2^2*mstar*h*N0*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hh^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z1^2*cos(k2*a)^2*rho^2*N0^2+4*Z1^2*cos(k2*a)^2*rho*h*N0^2+8*Z1*cos(k2*a
)*rho^2*Z2*sin(k2*a)*N0^2+8*Z1*cos(k2*a)*rho*Z2*sin(k2*a)*h*N0^2+Z1^2*cos(k2*a)^2*h^2*N0^
2+2*Z1*cos(k2*a)*h^2*Z2*sin(k2*a)*N0^2+4*Z2^2*rho^2*N0^2-
4*Z2^2*rho^2*N0^2*cos(k2*a)^2+4*Z2^2*rho*h*N0^2-
```

```
4*Z2^2*rho*h*N0^2*cos(k2*a)^2+Z2^2*h^2*N0^2-Z2^2*h^2*N0^2*cos(k2*a)^2)*k2^4/N0^2/Hh^2-
1/8*Ds*Z3^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hh^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants of integration for deflections(hinged)
A1 = (M_l*Z3)/(N0*Hh);
B1 = 0;
%C1 = 0;
%D1 = M_l/N0;
A2 = (M_l*Z1*sin(k1*a))/(N0*Hh);
B2 = (M_l*Z2*sin(k1*a))/(N0*Hh);
%C2 = 0;
%D2 = 0;
A3 = -(M_l*sin(k3)*sin(k1*a))/(N0*Hh);
B3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hh);
%C3 = 0;
%D3 = 0;
%Ap3 = 0;
%Bp3 = 0;
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hh);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hh);
%CZ validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.01:b);
%x3 = (b:0.01:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
for j2 = 1:length(x2);
    %w2star(j2) = A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2;
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
end
%for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
%    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
%end
%--------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0;
      ValidCZa(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZa(ja) = 1;
    end
  end

  if find(ValidCZa,1)>0;%1 valid, 0 not valid again
    ValidCZ = 0;
  else
    ValidCZ = 1;
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
       ValidCZ = 0;
%---------------------
elseif upordown > 0;    %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0; %was <=
      ValidCZb(jb) = 0; %0 valid, 1 not valid
    else
      ValidCZb(jb) = 1;
     end
  end

  if find(ValidCZb,1)>0; %1 valid, 0 not valid again
    ValidCZ = 0;
  else
    ValidCZ = 1;
  end
%----------------------
else
```

```
    ValidCZ = 0;
end
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;
```

To find intermediate contact zone configurations:

```
%Looking for Legit cases of propgating contact
%I will assign a value of "a" and look for valid cases...
%This is for when the flap is really Lp = 0.9
%This is for clamped ends, we don't see this with Hinged.
clear
%Define the usual stuff
global h C D Dp
h = 0.05; % height of baseplate
hp = 0.05; % height of patch
h0 = hp/h; % ratio of heights
E0 = 1; % elastic modulus
C = 12/(h^2); % membrane stiffness of baseplate
Cp = C*E0*h0; % membrane stiffness of patch
D = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rho Ds
% stiffnesses of composite structure
Astar = D+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rho = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rho*Bstar; %Dstar
global  Dc Cs Ce
Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar nstar etatilda
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 0.5; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rho*nstar;
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
global a astar b
%Choose values for a, b and N
a = 0.60;
astar = 1 - a;
Lp = 0.9; %patch length
b_vals = (a:0.001:Lp);
Npg = (0:0.1:50); %Same values used in Buckle01_CZ
%Separate loop for Buckling
q=1;
for p = 1:length(b_vals);
    b = b_vals(p);
    for j = 1:length(Npg);
        Big(j,1) = HCZ_clamp(Npg(j)); %HINGE/CLAMP 5
        %H
    end
     for r = 1:length(Npg)-1;
        %This is for Critical BUCKLING, look for H=0
        if Big(r,1)*Big(r+1,1)<0;
```

```
            Ncrit(r) = fzero(@HCZ_clamp,Npg(r)); %HINGE/CLAMP 6  %The critical N value
            Theta_crit(r) = -(rho+h/2)*Ncrit(r)/mstar; %Critical Theta
            %F_buckle = StableF(a,astar,Ncrit(k));
                CZBuckleResults(q,1)= b;
                CZBuckleResults(q,2)= Ncrit(r);
                CZBuckleResults(q,3) = Theta_crit(r);
                q=q+1;
        end
    end
end
%Pregrowth
%Allocate space for result arrays
CZResultsPGB = zeros(length(Npg),7,length(b_vals));
CZResultsPG2B = zeros(length(Npg),7,length(b_vals));
CZStablePGB = zeros(length(Npg),7,length(b_vals));
CZStablePG2B = zeros(length(Npg),7,length(b_vals));
CZValidPGB = zeros(length(Npg),7,length(b_vals));
CZValidPG2B = zeros(length(Npg),7,length(b_vals));
CZPropPGB = zeros(length(Npg),7,length(b_vals));
CZPropPG2B = zeros(length(Npg),7,length(b_vals));
for nn=1:length(b_vals);
bin=b_vals(nn);
mm=1; %Reset, not really needed
for mm = 1:length(Npg); %Results 2 is the 2nd root!
    CZResultsPGB(mm,1,nn)=Npg(mm); %N
    [CZResultsPGB(mm,2,nn),
CZResultsPGB(mm,3,nn),CZResultsPGB(mm,4,nn),CZResultsPGB(mm,5,nn),CZResultsPGB(mm,6,nn),C
ZResultsPGB(mm,7,nn)] = PreGrowthCZ_clamp_B(a,astar,Npg(mm),bin);
    if CZResultsPGB(mm,5,nn)>0;
        CZStablePGB(mm,:,nn)=CZResultsPGB(mm,:,nn);
    end
    if CZStablePGB(mm,7,nn)>0 && CZStablePGB(mm,7,nn)<2; %Is Full CZ valid?
        CZValidPGB(mm,:,nn)=CZResultsPGB(mm,:,nn);
    end
    if CZResultsPGB(mm,7,nn)>1;  %Is Prop CZ valid? (i.e., CZ when b<Lp)
        CZPropPGB(mm,:,nn)=CZResultsPGB(mm,:,nn);
    End
CZResultsPG2B(mm,1,nn)=Npg(mm); %N
    [CZResultsPG2B(mm,2,nn),
CZResultsPG2B(mm,3,nn),CZResultsPG2B(mm,4,nn),CZResultsPG2B(mm,5,nn),CZResultsPG2B(mm,6,n
n),CZResultsPG2B(mm,7,nn)] = PreGrowthCZ_clamp2_B(a,astar,Npg(mm),bin);
    %[N0,theta,w0,ERR,F,PotentialE,ValidCZ (1 yes, 0 no)]
    if CZResultsPG2B(mm,5,nn)>0;
        CZStablePG2B(mm,:,nn)=CZResultsPG2B(mm,:,nn);
    end
    if CZStablePG2B(mm,7,nn)>0&& CZStablePG2B(mm,7,nn)<2;%Is Full CZ valid?
        CZValidPG2B(mm,:,nn)=CZResultsPG2B(mm,:,nn);
    end
    if CZStablePG2B(mm,7,nn)>1;  %Is Prop CZ valid? (i.e., CZ when b<Lp)
        CZPropPG2B(mm,:,nn)=CZResultsPG2B(mm,:,nn);
    end

end
end
%=====================================================================
%Save the data with a new name for the COMBO program!
%save CZProp_b_60.mat CZPropPGB;
%save CZProp_b2_60.mat CZPropPG2B;
%=====================================================================
test = 40; %Choose a value of b.  b = b_vals(test)
point = 375; %Choose a data point to look at deflected structure
% PLOT 1 ------------------------------------------------------------
figure
plot(CZResultsPGB(:,3,test),CZResultsPGB(:,2,test),'k-','LineWidth',2)%regular load-def
path
hold on
plot(CZResultsPG2B(:,3,test),CZResultsPG2B(:,2,test),'k-','LineWidth',2)
hold on
plot(CZStablePGB(:,3,test),CZStablePGB(:,2,test),'r.','MarkerSize',10)%stable portion
hold on
plot(CZStablePG2B(:,3,test),CZStablePG2B(:,2,test),'r.','MarkerSize',10)
```

```
hold on
plot(CZValidPGB(:,3,test),CZValidPGB(:,2,test),'g*','MarkerSize',10)%Valid Full CZ
hold on
plot(CZValidPG2B(:,3,test),CZValidPG2B(:,2,test),'g*','Markersize',10)
hold on
plot(CZPropPGB(:,3,test),CZPropPGB(:,2,test),'y*','MarkerSize',10)%Valid Prop CZ
hold on
plot(CZPropPG2B(:,3,test),CZPropPG2B(:,2,test),'y*','Markersize',10)
hold on
plot(CZResultsPGB(point,3,test),CZResultsPGB(point,2,test),'bo','Markersize',12) %adjust
%for index = 1:length(b_vals)
%hold on
%plot(CZStablePG2B(:,3,index),CZStablePG2B(:,2,index),'b.','MarkerSize',10)
%hold on
%plot(CZValidPG2B(:,3,index),CZValidPG2B(:,2,index),'m*','Markersize',10)
%hold on
%plot(CZPropPG2B(:,3,index),CZPropPG2B(:,2,index),'y*','Markersize',10)
%end
xlabel('w0','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',16,'fontweight','b')
title(['b = ',num2str(b_vals(test))],'fontsize',16,'fontweight','b')
grid on
% PLOT 2 -------------------------------------------------------
figure
plot(CZResultsPGB(:,2,test),CZResultsPGB(:,6,test),'k-','LineWidth',2)%regular load-def
path
hold on
plot(CZResultsPG2B(:,2,test),CZResultsPG2B(:,6,test),'k-','LineWidth',2)
hold on
plot(CZStablePGB(:,2,test),CZStablePGB(:,6,test),'r.','MarkerSize',10)%stable portion
hold on
plot(CZStablePG2B(:,2,test),CZStablePG2B(:,6,test),'r.','MarkerSize',10)
hold on
plot(CZValidPGB(:,2,test),CZValidPGB(:,6,test),'g*','MarkerSize',10)%Valid Full CZ
hold on
plot(CZValidPG2B(:,2,test),CZValidPG2B(:,6,test),'g*','Markersize',10)
hold on
plot(CZPropPGB(:,2,test),CZPropPGB(:,6,test),'y*','MarkerSize',10)%Valid Prop CZ
hold on
plot(CZPropPG2B(:,2,test),CZPropPG2B(:,6,test),'y*','Markersize',10)
hold on
plot(CZResultsPGB(point,2,test),CZResultsPGB(point,6,test),'bo','Markersize',12) %adjust
%for index = 1:length(b_vals)
%hold on
%plot(CZStablePG2B(:,2,index),CZStablePG2B(:,6,index),'b.','MarkerSize',10)
%hold on
%plot(CZValidPG2B(:,2,index),CZValidPG2B(:,6,index),'m*','Markersize',10)
%hold on
%plot(CZPropPG2B(:,2,index),CZPropPG2B(:,6,index),'y*','Markersize',10)
%end
axis([0 0.04 0 2])
xlabel('aQ','fontsize',16,'fontweight','b')
ylabel('P','fontsize',16,'fontweight','b')
title(['b = ',num2str(b_vals(test))],'fontsize',16,'fontweight','b')
grid on
% PLOT 3 -------------------------------------------------------
x1plot = (0:0.005:a);
x2plot = (a:0.001:b_vals(test));
x3plot = (b_vals(test):0.001:1);
xp3plot = (b_vals(test):0.001:0.9); %Lp = 0.9
                                    %adjust!
[a1,b1,c1,d1,a2,b2,c2,d2,a3,b3,c3,d3,K1,K2,K3]=
ConstantsCZ_clampB(CZResultsPGB(point,1,test),CZResultsPGB(point,2,test),a,b_vals(test));
            for i1 = 1:length(x1plot);
                w1starplot(i1) = -
(a1*cos(K1*x1plot(i1))+b1*sin(K1*x1plot(i1))+c1*x1plot(i1)+d1);
                kappa1pl(i1) = -a1*K1^2*cos(K1*x1plot(i1))-b1*K1^2*sin(K1*x1plot(i1));
            end
            for i2 = 1:length(x2plot);
                w2starplot(i2) = -
(a2*cos(K2*x2plot(i2))+b2*sin(K2*x2plot(i2))+c2*x2plot(i2)+d2);
```

```
                kappa2pl(i2) = -a2*K2^2*cos(K2*x2plot(i2))-b2*K2^2*sin(K2*x2plot(i2));
            end
            slope_at_b = -a2*K2*sin(K2*b_vals(test))+b2*K2*cos(K2*b_vals(test))+c2;
            for i3 = 1:length(x3plot);
                w3plot(i3) = -
(a3*cos(K3*x3plot(i3))+b3*sin(K3*x3plot(i3))+c3*x3plot(i3)+d3);
                kappa3pl(i3) = -a3*K3^2*cos(K3*x3plot(i3))-b3*K3^2*sin(K3*x3plot(i3));
                %testflap(i3) = -(slope_at_b*(x3plot(i3)-b_vals(test))-w3plot(1));%mess
with signs to get nice plot
            end
            for ip3 = 1:length(xp3plot);
                testflap(ip3) = -(slope_at_b*(x3plot(ip3)-b_vals(test))-w3plot(1));%mess
with signs to get nice plot
            end
figure
plot(x1plot,w1starplot,'k','LineWidth',2)
hold on
plot(x2plot,w2starplot,'b','LineWidth',2)
hold on
plot(x3plot,w3plot,'r','LineWidth',2)
hold on
plot(x1plot,kappa1pl,'g','LineWidth',2)
hold on
plot(x2plot,kappa2pl,'c','LineWidth',2)
hold on
plot(x3plot,kappa3pl,'m','LineWidth',2)
hold on
plot(xp3plot,testflap,'y','LineWidth',2)
if CZResultsPGB(point,7,test) == 2; %adjust
    title('Propagating CZ','fontsize',16,'fontweight','b')
elseif CZResultsPGB(point,7,test) == 1; %adjust
    title('Valid CZ Configuration','fontsize',16,'fontweight','b')
else
    title('Invalid CZ Configuration','fontsize',16,'fontweight','b')
end
xlabel('x','fontsize',16,'fontweight','b')
ylabel('w','fontsize',16,'fontweight','b')
grid on
```

Associated subroutines:

```
function[Thetapg,w0,err,F,PotEn,ValidCZ] = PreGrowthCZ_clamp_B(a,astar,N0,b)
%Pregrowth CFIX CX 1
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-sqrt(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+sqrt(Dc/D)*sin(k3-k3*b)*cos(k2*b);
Z4 = k3*sin(k3-k3*b);
Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = mstar*Z1/N0/Hc*k1*sin(k1*a);
y2 = (rho+1/2*h)*Z1/Hc*k1*sin(k1*a);
y3 = 1/2*Z1^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y4 = 1/8*Z1^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y5 = 1/8*Z1^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(-
4*Z2^2*mstar^2*b*k2+4*Z2^2*mstar^2*k2*a+4*Z3^2*mstar^2*k2*a-
4*Z2^2*mstar^2*cos(k2*a)*sin(k2*a)-4*Z3^2*mstar^2*k2*b-
8*Z2*mstar^2*Z3*cos(k2*b)^2+4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-
```

```
4*Z3^2*mstar^2*cos(k2*b)*sin(k2*b)+4*Z3^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z2*mstar^2*Z3*cos
(k2*a)^2)/N0^2/Hc^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(-8*Z2^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*b*k2+8*Z2^2*mstar*rho*N0*k2*a+4*Z2^2*mstar*h*N0*k2*a+8*Z3^2*mstar*rho*N
0*k2*a+4*Z3^2*mstar*h*N0*k2*a-8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z2^2*mstar*h*N0*cos(k2*a)*sin(k2*a)-8*Z3^2*mstar*rho*N0*k2*b-4*Z3^2*mstar*h*N0*k2*b-
16*Z2*mstar*Z3*rho*N0*cos(k2*b)^2-
8*Z2*mstar*Z3*h*N0*cos(k2*b)^2+8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)+4*Z2^2*mstar*h*N0*
cos(k2*b)*sin(k2*b)-8*Z3^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar*h*N0*cos(k2*b)*sin(k2*b)+8*Z3^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z3^2*msta
r*h*N0*cos(k2*a)*sin(k2*a)+8*Z2*mstar*Z3*h*N0*cos(k2*a)^2+16*Z2*mstar*Z3*rho*N0*cos(k2*a)
^2)/N0^2/Hc^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho^2*N0^2*b*k2-4*Z2^2*rho*h*N0^2*b*k2-
Z2^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*k2*a+4*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+4*Z3^
2*rho^2*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho*Z3*h*N0^2*cos(k2*a)^2-
8*Z2*rho^2*Z3*N0^2*cos(k2*b)^2-
4*Z3^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)+4*Z3^2*rho*h*N0^2*k2*a+Z3^2*h^2*N0^2*k2*a-
4*Z3^2*rho^2*N0^2*k2*b-4*Z3^2*rho*h*N0^2*k2*b-Z3^2*h^2*N0^2*k2*b-
8*Z2*rho*Z3*h*N0^2*cos(k2*b)^2-
2*Z2*h^2*Z3*N0^2*cos(k2*b)^2+4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+4*Z2^2*rho^2*N0^2*cos(
k2*b)*sin(k2*b)+Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-4*Z3^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
Z3^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-
Z2^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z3^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho^2*Z3*N0^
2*cos(k2*a)^2+2*Z2*h^2*Z3*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z3^2*h^2
*N0^2*cos(k2*a)*sin(k2*a))/N0^2/Hc^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*cos(k3)*mstar^2*sin(k3)^3-4*mstar^2*k3-
4*cos(k3)*mstar^2*sin(k3)-8*cos(k3)^2*mstar^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*mstar^2*sin(k3)*sin(k3*b)^2+4*mstar^2*cos(k3*b)*sin(k3*b)+8*cos(k3)^3*mstar^2*s
in(k3)+4*mstar^2*k3*b)/N0^2/Hc^2;
y10 = 1/8*(-
1+cos(k1*a)^2)*k3*(16*cos(k3)^3*mstar*rho*N0*sin(k3)+8*cos(k3)^3*mstar*h*N0*sin(k3)+16*co
s(k3)*mstar*sin(k3)^3*rho*N0+8*cos(k3)*mstar*sin(k3)^3*h*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3-8*cos(k3)*mstar*sin(k3)*rho*N0-
4*cos(k3)*mstar*sin(k3)*h*N0+8*mstar*N0*rho*k3*b+4*mstar*N0*h*k3*b-
16*cos(k3)^2*mstar*rho*N0*cos(k3*b)*sin(k3*b)-8*cos(k3)^2*mstar*h*N0*cos(k3*b)*sin(k3*b)-
16*cos(k3)*mstar*sin(k3)*rho*N0*sin(k3*b)^2-
8*cos(k3)*mstar*sin(k3)*h*N0*sin(k3*b)^2+8*mstar*N0*rho*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
cos(k3*b)*sin(k3*b))/N0^2/Hc^2;
y11 = 1/8*(-1+cos(k1*a)^2)*k3*(-
4*cos(k3)*rho*sin(k3)*h*N0^2+8*cos(k3)*rho^2*sin(k3)^3*N0^2+8*cos(k3)*rho*sin(k3)^3*h*N0^
2-4*N0^2*rho^2*k3-
N0^2*h^2*k3+8*cos(k3)^3*rho*h*N0^2*sin(k3)+4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+4*N0^2*rho*h*
k3*b-8*cos(k3)^2*rho^2*N0^2*cos(k3*b)*sin(k3*b)-2*cos(k3)^2*h^2*N0^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*rho^2*sin(k3)*N0^2*sin(k3*b)^2-8*cos(k3)*rho*sin(k3)*h*N0^2*sin(k3*b)^2-
2*cos(k3)*h^2*sin(k3)*N0^2*sin(k3*b)^2-
8*cos(k3)^2*rho*h*N0^2*cos(k3*b)*sin(k3*b)+4*N0^2*rho^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*cos(
k3*b)*sin(k3*b)-
4*N0^2*rho*h*k3+8*cos(k3)^3*rho^2*N0^2*sin(k3)+2*cos(k3)^3*h^2*N0^2*sin(k3)-
cos(k3)*h^2*sin(k3)*N0^2+2*cos(k3)*h^2*sin(k3)^3*N0^2+4*N0^2*rho^2*k3*b+N0^2*h^2*k3*b-
4*cos(k3)*rho^2*sin(k3)*N0^2)/N0^2/Hc^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
Thetapg = (-Q2+sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %1st root!
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = cos(k3*(1-b));
B2 = sqrt(Dc/D)*sin(k3*(1-b));
F = (1/4)*(k1*sin(2*k1*a)*(Z1/Hc)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hc)^2+2*k2*B1*B2*(cos(2*k2*(b-a))-1)*(sin(k1*a)/Hc)^2+k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hc)^2);
w0 = M_l*(Hc-Z1-sin(k1*a))/(N0*Hc);
upordown = sign(w0); % + means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z2^2*cos(k2*a)^2*mstar^2+8*Z2*cos(k2*a)*mstar^2*Z3*sin(k2*a)+4*Z3^2*mst
ar^2-4*Z3^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/2*Ds*Z1^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hc^2+1/2*etatilda;
```

```
Q5 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*mstar*h*N0-
4*Z3^2*mstar*h*N0*cos(k2*a)^2+8*Z2^2*cos(k2*a)^2*mstar*rho*N0+4*Z2^2*cos(k2*a)^2*mstar*h*
N0+16*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*rho*N0+8*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*h*N0+8*Z3^
2*mstar*rho*N0-8*Z3^2*mstar*rho*N0*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hc^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*rho^2*N0^2-
4*Z3^2*rho^2*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2-
4*Z3^2*rho*h*N0^2*cos(k2*a)^2+Z3^2*h^2*N0^2-
Z3^2*h^2*N0^2*cos(k2*a)^2+4*Z2^2*cos(k2*a)^2*rho^2*N0^2+4*Z2^2*cos(k2*a)^2*rho*h*N0^2+8*Z
2*cos(k2*a)*rho^2*Z3*sin(k2*a)*N0^2+8*Z2*cos(k2*a)*rho*Z3*sin(k2*a)*h*N0^2+Z2^2*cos(k2*a)
^2*h^2*N0^2+2*Z2*cos(k2*a)*h^2*Z3*sin(k2*a)*N0^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hc^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants (clamped)
A1 = -(M_l*Z1)/(N0*Hc);
B1 = 0;
C1 = 0;
D1 = M_l*(Hc-sin(k1*a))/(N0*Hc);
A2 = (M_l*Z2*sin(k1*a))/(N0*Hc);
B2 = (M_l*Z3*sin(k1*a))/(N0*Hc);
C2 = 0;
D2 = -(M_l*sin(k1*a))/(N0*Hc);
A3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hc);
B3 = (M_l*sin(k3)*sin(k1*a))/(N0*Hc);
C3 = 0;
D3 = -(M_l*sin(k1*a))/(N0*Hc);
%Ap3 = 0;
%Bp3 = 0;
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hc);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hc);
% CZ Validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.005:b);  %Need a finer grid size!
x3 = (b:0.005:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
slopeata = -A1*k1*sin(k1*a)+B1*k1*cos(k1*a)+C1; %New
for j2 = 1:length(x2);
    w2star(j2) = -(A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2);
    flap(j2) = -(slopeata*(x2(j2)-a)-w2star(1)); %New, signs messed w/
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
end
for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
end

if length(kappa2)>1;
    index_in =length(kappa2)-1;
else
    index_in = length(kappa2);
end
%--------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0; %Check the sign of kappa 2
      ValidCZa(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZa(ja) = 1;
    end
  end
                        %1 valid, 0 not valid again
  if find(ValidCZa,1)>0; %Look for kappa2 positive
    for jp=1:length(x2); %Check for penetration
          if flap(jp)<w2star(jp);
              flaptest(jp)=1; %penetration
              else
              flaptest(jp)=0; %no penetration
          end
```

```
    end

      if find(flaptest)>0
          ValidCZ = 1; %Penetration, hence Valid full CZ
      else
          ValidCZ = 0; %NO Penetration, no CZ
      end
  else
    ValidCZ = 1; %Kappa 2 is negative, full CZ valid!
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
       ValidCZ = 0;
%--------------------
elseif upordown > 0; %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0; %Check the sign of kappa 2 again,
      ValidCZb(jb) = 0; %Negative
    else
      ValidCZb(jb) = 1; % Positive
     end
  end
                           %So the last point of k2 doesnt count...used
                            %to be > sign
  if find(ValidCZb,1)<length(ValidCZb); %0; %1 valid, 0 not valid again
    if sum(ValidCZb) == length(ValidCZb);
      ValidCZ = 0;
    else
      ValidCZ = 0; %Not a real structure...formerly prop. CZ
    end
  else %If kappa2 is all negative...is prop CZ valid?
    for jc=1:length(x3);
        if kappa3(jc)>0;
            ValidCZc(jc) = 1;%positive
        else
            ValidCZc(jc)=0;%negative
        end
    end
    if kappa2(index_in)*kappa3(2)<0; %Check condition that kappa2(b-)*kappa3(b+)<0,
kappa3(b+)>0,
      ValidCZ = 2; %Propagating Contact
    else
      ValidCZ = 1;
    end
  end

%----------------------
else                          %should be no "else"
    ValidCZ = 1;
end
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;

function[Thetapg,w0,err,F,PotEn,ValidCZ2] = PreGrowthCZ_clamp2_B(a,astar,N0,b)
%Pregrowth CFIX CX 2
global Ds D Dc rho h C Cstar Ce mstar nstar etatilda
k1 = sqrt(N0/Ds);
k2 = sqrt(N0/Dc);
k3 = sqrt(N0/D);
%First off, H.  Not Delamination!
```

```
Hc = sin(k1*a)*cos(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/Dc)*cos(k1*a)*sin(k2*b-k2*a)*cos(k3-
k3*b)+sqrt(Ds/D)*cos(k1*a)*cos(k2*b-k2*a)*sin(k3-k3*b)-sqrt(Dc/D)*sin(k1*a)*sin(k2*b-
k2*a)*sin(k3-k3*b);
%Now the Z's from my hand solutions
Z1 = sqrt(Ds/Dc)*sin(k2*b-k2*a)*cos(k3-k3*b)+sqrt(Ds/D)*cos(k2*b-k2*a)*sin(k3-k3*b);
Z2 = cos(k3-k3*b)*cos(k2*b)-sqrt(Dc/D)*sin(k3-k3*b)*sin(k2*b);
Z3 = cos(k3-k3*b)*sin(k2*b)+sqrt(Dc/D)*sin(k3-k3*b)*cos(k2*b);
Z4 = k3*sin(k3-k3*b);
Z5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
%Now for the Y's  start of Delamination stuff
y0 = astar+a*nstar/Cstar;
y1 = mstar*Z1/N0/Hc*k1*sin(k1*a);
y2 = (rho+1/2*h)*Z1/Hc*k1*sin(k1*a);
y3 = 1/2*Z1^2*k1*mstar^2*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y4 = 1/8*Z1^2*k1*(8*mstar*N0*rho+4*mstar*N0*h)*(k1*a-cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y5 = 1/8*Z1^2*k1*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)*(k1*a-
cos(k1*a)*sin(k1*a))/N0^2/Hc^2;
y6 = 1/8*(-1+cos(k1*a)^2)*k2*(-
4*Z2^2*mstar^2*b*k2+4*Z2^2*mstar^2*k2*a+4*Z3^2*mstar^2*k2*a-
4*Z2^2*mstar^2*cos(k2*a)*sin(k2*a)-4*Z3^2*mstar^2*k2*b-
8*Z2*mstar^2*Z3*cos(k2*b)^2+4*Z2^2*mstar^2*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar^2*cos(k2*b)*sin(k2*b)+4*Z3^2*mstar^2*cos(k2*a)*sin(k2*a)+8*Z2*mstar^2*Z3*cos
(k2*a)^2)/N0^2/Hc^2;
y7 = 1/8*(-1+cos(k1*a)^2)*k2*(-8*Z2^2*mstar*rho*N0*b*k2-
4*Z2^2*mstar*h*N0*b*k2+8*Z2^2*mstar*rho*N0*k2*a+4*Z2^2*mstar*h*N0*k2*a+8*Z3^2*mstar*rho*N
0*k2*a+4*Z3^2*mstar*h*N0*k2*a-8*Z2^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)-
4*Z2^2*mstar*h*N0*cos(k2*a)*sin(k2*a)-8*Z3^2*mstar*rho*N0*k2*b-4*Z3^2*mstar*h*N0*k2*b-
16*Z2*mstar*Z3*rho*N0*cos(k2*b)^2-
8*Z2*mstar*Z3*h*N0*cos(k2*b)^2+8*Z2^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)+4*Z2^2*mstar*h*N0*
cos(k2*b)*sin(k2*b)-8*Z3^2*mstar*rho*N0*cos(k2*b)*sin(k2*b)-
4*Z3^2*mstar*h*N0*cos(k2*b)*sin(k2*b)+8*Z3^2*mstar*rho*N0*cos(k2*a)*sin(k2*a)+4*Z3^2*msta
r*h*N0*cos(k2*a)*sin(k2*a)+8*Z2*mstar*Z3*h*N0*cos(k2*a)^2+16*Z2*mstar*Z3*rho*N0*cos(k2*a)
^2)/N0^2/Hc^2;
y8 = 1/8*(-1+cos(k1*a)^2)*k2*(-4*Z2^2*rho^2*N0^2*b*k2-4*Z2^2*rho*h*N0^2*b*k2-
Z2^2*h^2*N0^2*b*k2+4*Z2^2*rho^2*N0^2*k2*a+4*Z2^2*rho*h*N0^2*k2*a+Z2^2*h^2*N0^2*k2*a+4*Z3^
2*rho^2*N0^2*k2*a-4*Z2^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)-
4*Z2^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho*Z3*h*N0^2*cos(k2*a)^2-
8*Z2*rho^2*Z3*N0^2*cos(k2*b)^2-
4*Z3^2*rho^2*N0^2*cos(k2*b)*sin(k2*b)+4*Z3^2*rho*h*N0^2*k2*a+Z3^2*h^2*N0^2*k2*a-
4*Z3^2*rho^2*N0^2*k2*b-4*Z3^2*rho*h*N0^2*k2*b-Z3^2*h^2*N0^2*k2*b-
8*Z2*rho*Z3*h*N0^2*cos(k2*b)^2-
2*Z2*h^2*Z3*N0^2*cos(k2*b)^2+4*Z2^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)+4*Z2^2*rho^2*N0^2*cos(
k2*b)*sin(k2*b)+Z2^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-4*Z3^2*rho*h*N0^2*cos(k2*b)*sin(k2*b)-
Z3^2*h^2*N0^2*cos(k2*b)*sin(k2*b)-
Z2^2*h^2*N0^2*cos(k2*a)*sin(k2*a)+4*Z3^2*rho^2*N0^2*cos(k2*a)*sin(k2*a)+8*Z2*rho^2*Z3*N0^
2*cos(k2*a)^2+2*Z2*h^2*Z3*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2*cos(k2*a)*sin(k2*a)+Z3^2*h^2
*N0^2*cos(k2*a)*sin(k2*a))/N0^2/Hc^2;
y9 = 1/8*(-1+cos(k1*a)^2)*k3*(8*cos(k3)*mstar^2*sin(k3)^3-4*mstar^2*k3-
4*cos(k3)*mstar^2*sin(k3)-8*cos(k3)^2*mstar^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*mstar^2*sin(k3)*sin(k3*b)^2+4*mstar^2*cos(k3*b)*sin(k3*b)+8*cos(k3)^3*mstar^2*s
in(k3)+4*mstar^2*k3*b)/N0^2/Hc^2;
y10 = 1/8*(-
1+cos(k1*a)^2)*k3*(16*cos(k3)^3*mstar*rho*N0*sin(k3)+8*cos(k3)^3*mstar*h*N0*sin(k3)+16*co
s(k3)*mstar*sin(k3)^3*rho*N0+8*cos(k3)*mstar*sin(k3)^3*h*N0-8*mstar*N0*rho*k3-
4*mstar*N0*h*k3-8*cos(k3)*mstar*sin(k3)*rho*N0-
4*cos(k3)*mstar*sin(k3)*h*N0+8*mstar*N0*rho*k3*b+4*mstar*N0*h*k3*b-
16*cos(k3)^2*mstar*rho*N0*cos(k3*b)*sin(k3*b)-8*cos(k3)^2*mstar*h*N0*cos(k3*b)*sin(k3*b)-
16*cos(k3)*mstar*sin(k3)*rho*N0*sin(k3*b)^2-
8*cos(k3)*mstar*sin(k3)*h*N0*sin(k3*b)^2+8*mstar*N0*rho*cos(k3*b)*sin(k3*b)+4*mstar*N0*h*
cos(k3*b)*sin(k3*b))/N0^2/Hc^2;
y11 = 1/8*(-1+cos(k1*a)^2)*k3*(-
4*cos(k3)*rho*sin(k3)*h*N0^2+8*cos(k3)*rho^2*sin(k3)^3*N0^2+8*cos(k3)*rho*sin(k3)^3*h*N0^
2-4*N0^2*rho^2*k3-
N0^2*h^2*k3+8*cos(k3)^3*rho*h*N0^2*sin(k3)+4*N0^2*rho*h*cos(k3*b)*sin(k3*b)+4*N0^2*rho*h*
k3*b-8*cos(k3)^2*rho^2*N0^2*cos(k3*b)*sin(k3*b)-2*cos(k3)^2*h^2*N0^2*cos(k3*b)*sin(k3*b)-
8*cos(k3)*rho^2*sin(k3)*N0^2*sin(k3*b)^2-8*cos(k3)*rho*sin(k3)*h*N0^2*sin(k3*b)^2-
2*cos(k3)*h^2*sin(k3)*N0^2*sin(k3*b)^2-
8*cos(k3)^2*rho*h*N0^2*cos(k3*b)*sin(k3*b)+4*N0^2*rho^2*cos(k3*b)*sin(k3*b)+N0^2*h^2*cos(
k3*b)*sin(k3*b)-
4*N0^2*rho*h*k3+8*cos(k3)^3*rho^2*N0^2*sin(k3)+2*cos(k3)^3*h^2*N0^2*sin(k3)-
```

```
cos(k3)*h^2*sin(k3)*N0^2+2*cos(k3)*h^2*sin(k3)^3*N0^2+4*N0^2*rho^2*k3*b+N0^2*h^2*k3*b-
4*cos(k3)*rho^2*sin(k3)*N0^2)/N0^2/Hc^2;
Ca = astar/C+a/Cstar;
%Now the Q's, such that the integrability condition is 0=Q1th^2+Q2th+Q3
Q1 = -1/2*y3-1/2*y6-1/2*y9;
Q2 = y0-(rho+1/2*h)*y1-1/2*y4-1/2*y7-1/2*y10;
Q3 = -N0*Ca-(rho+1/2*h)*y2-1/2*y5-1/2*y8-1/2*y11;
Thetapg = (-Q2-sqrt((Q2^2)-4*Q1*Q3))/(2*Q1); %2nd root!
M_l = mstar*Thetapg+(rho+1/2*h)*N0;
B1 = cos(k3*(1-b));
B2 = sqrt(Dc/D)*sin(k3*(1-b));
F = (1/4)*(k1*sin(2*k1*a)*(Z1/Hc)^2+k2*(B1^2-B2^2)*sin(2*k2*(b-
a))*(sin(k1*a)/Hc)^2+2*k2*B1*B2*(cos(2*k2*(b-a))-1)*(sin(k1*a)/Hc)^2+k3*sin(2*k3*(1-
b))*(sin(k1*a)/Hc)^2);
w0 = M_l*(Hc-Z1-sin(k1*a))/(N0*Hc);
upordown = sign(w0); % + means deflection is down, - means deflection is up!
%Now the Q's, such that the transversality condition is
%2gamma=Q4th^2+Q5th+Q6
Q4 = -1/8*Dc*(-
1+cos(k1*a)^2)*(4*Z2^2*cos(k2*a)^2*mstar^2+8*Z2*cos(k2*a)*mstar^2*Z3*sin(k2*a)+4*Z3^2*mst
ar^2-4*Z3^2*mstar^2*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/2*Ds*Z1^2*k1^4*cos(k1*a)^2*mstar^2/N0^2/Hc^2+1/2*etatilda;
Q5 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*mstar*h*N0-
4*Z3^2*mstar*h*N0*cos(k2*a)^2+8*Z2^2*cos(k2*a)^2*mstar*rho*N0+4*Z2^2*cos(k2*a)^2*mstar*h*
N0+16*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*rho*N0+8*Z2*cos(k2*a)*mstar*Z3*sin(k2*a)*h*N0+8*Z3^
2*mstar*rho*N0-8*Z3^2*mstar*rho*N0*cos(k2*a)^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(8*mstar*N0*rho+4*mstar*N0*h)/N0^2/Hc^2-N0*(1-nstar/Cstar);
Q6 = -1/8*Dc*(-1+cos(k1*a)^2)*(4*Z3^2*rho^2*N0^2-
4*Z3^2*rho^2*N0^2*cos(k2*a)^2+4*Z3^2*rho*h*N0^2-
4*Z3^2*rho*h*N0^2*cos(k2*a)^2+Z3^2*h^2*N0^2-
Z3^2*h^2*N0^2*cos(k2*a)^2+4*Z2^2*cos(k2*a)^2*rho^2*N0^2+4*Z2^2*cos(k2*a)^2*rho*h*N0^2+8*Z
2*cos(k2*a)*rho^2*Z3*sin(k2*a)*N0^2+8*Z2*cos(k2*a)*rho*Z3*sin(k2*a)*h*N0^2+Z2^2*cos(k2*a)
^2*h^2*N0^2+2*Z2*cos(k2*a)*h^2*Z3*sin(k2*a)*N0^2)*k2^4/N0^2/Hc^2-
1/8*Ds*Z1^2*k1^4*cos(k1*a)^2*(4*N0^2*rho^2+4*N0^2*rho*h+N0^2*h^2)/N0^2/Hc^2+1/2*N0^2/Ce;
err = Q4*Thetapg^2+Q5*Thetapg+Q6;
%Constants (clamped)
A1 = -(M_l*Z1)/(N0*Hc);
B1 = 0;
C1 = 0;
D1 = M_l*(Hc-sin(k1*a))/(N0*Hc);
A2 = (M_l*Z2*sin(k1*a))/(N0*Hc);
B2 = (M_l*Z3*sin(k1*a))/(N0*Hc);
C2 = 0;
D2 = -(M_l*sin(k1*a))/(N0*Hc);
A3 = (M_l*cos(k3)*sin(k1*a))/(N0*Hc);
B3 = (M_l*sin(k3)*sin(k1*a))/(N0*Hc);
C3 = 0;
D3 = -(M_l*sin(k1*a))/(N0*Hc);
%Ap3 = 0;
%Bp3 = 0;
%Cp3 = (M_l*Z4*sin(k1*a))/(N0*Hc);
%Dp3 = (M_l*Z5*sin(k1*a))/(N0*Hc);
% CZ Validity Check!
%x1 = (0:0.01:a);
x2 = (a:0.005:b);  %Need a finer grid size!
x3 = (b:0.005:1);
%for j1 = 1:length(x1);
%    w1star(j1) = A1*cos(k1*x1(j1))+B1*sin(k1*x1(j1))+C1*x1(j1)+D1;
%    kappa1(j1) = -A1*k1^2*cos(k1*x1(j1))-B1*k1^2*sin(k1*x1(j1));
%end
slopeata = -A1*k1*sin(k1*a)+B1*k1*cos(k1*a)+C1; %New
for j2 = 1:length(x2);
    w2star(j2) = -(A2*cos(k2*x2(j2))+B2*sin(k2*x2(j2))+C2*x2(j2)+D2);
    flap(j2) = -(slopeata*(x2(j2)-a)-w2star(1)); %New, signs messed w/
    kappa2(j2) = -A2*k2^2*cos(k2*x2(j2))-B2*k2^2*sin(k2*x2(j2));
end
for j3 = 1:length(x3);
%    w3(j3) = A3*cos(k3*x3(j3))+B3*sin(k3*x3(j3))+C3*x3(j3)+D3;
    kappa3(j3) = -A3*k3^2*cos(k3*x3(j3))-B3*k3^2*sin(k3*x3(j3));
end
```

```
if length(kappa2)>1;
    index_in =length(kappa2)-1;
else
    index_in = length(kappa2);
end
%-------------------
if upordown < 0;  %structure bends up
  for ja=1:length(x2);
    if kappa2(ja) < 0; %Check the sign of kappa 2
      ValidCZ2a(ja) = 0; %0 valid, 1 not valid
    else
      ValidCZ2a(ja) = 1;
    end
  end
                         %1 valid, 0 not valid again
  if find(ValidCZ2a,1)>0; %Look for kappa2 positive
    for jp=1:length(x2); %Check for penetration
          if flap(jp)<w2star(jp);
              flaptest2(jp)=1; %penetration
              else
              flaptest2(jp)=0; %no penetration
          end
    end

      if find(flaptest2)>0
          ValidCZ2 = 1; %Penetration, hence Valid full CZ
      else
          ValidCZ2 = 0; %NO Penetration, no CZ
      end
  else
    ValidCZ2 = 1; %Kappa 2 is negative, full CZ valid!
  end
%--------------------
elseif upordown == 0; %structure doesn't bend
        ValidCZ2 = 0;
%---------------------
elseif upordown > 0; %structure bends down
  for jb=1:length(x2);
    if kappa2(jb) < 0; %Check the sign of kappa 2 again,
      ValidCZ2b(jb) = 0; %Negative
    else
      ValidCZ2b(jb) = 1; % Positive
     end
  end
                            %So the last point of k2 doesnt count...used
                             %to be > sign
  if find(ValidCZ2b,1)<length(ValidCZ2b); %0; %1 valid, 0 not valid again
    if sum(ValidCZ2b) == length(ValidCZ2b);
      ValidCZ2 = 0;
    else
      ValidCZ2 = 0; %Not a real structure...formerly prop. CZ
    end
  else %If kappa2 is all negative...is prop CZ valid?
    for jc=1:length(x3);
        if kappa3(jc)>0;
            ValidCZ2c(jc) = 1;%positive
        else
            ValidCZ2c(jc)=0;%negative
        end
    end
    if kappa2(index_in)*kappa3(2)<0; %Check condition that kappa2(b-)*kappa3(b+)<0,
kappa3(b+)>0,
      ValidCZ2 = 2; %Propagating Contact
    else
      ValidCZ2 = 1;
    end
  end
%----------------------
else                           %should be no "else"
    ValidCZ2 = 1;
end
```

```
%POTENTIAL ENERGY OF AN EQUILIBRIUM POSITION
PotEn = 1/4*Ds*k1^3*(2*A1*B1+A1^2*sin(k1*a)*cos(k1*a)+A1^2*k1*a-
2*A1*B1*cos(k1*a)^2+B1^2*a*k1-
B1^2*sin(k1*a)*cos(k1*a))+Thetapg*mstar*k1*B1+Thetapg*mstar*A1*k1*sin(k1*a)-
Thetapg*mstar*B1*k1*cos(k1*a)-1/4*Dc*k2^3*(A2^2*cos(k2*a)*sin(k2*a)+A2^2*k2*a-
2*A2*B2*cos(k2*a)^2+B2^2*a*k2-B2^2*cos(k2*a)*sin(k2*a)-A2^2*cos(k2*b)*sin(k2*b)-
A2^2*k2*b+2*A2*B2*cos(k2*b)^2-B2^2*b*k2+B2^2*cos(k2*b)*sin(k2*b))-
1/4*D*k3^3*(A3^2*cos(k3*b)*sin(k3*b)+A3^2*k3*b-2*A3*B3*cos(k3*b)^2+B3^2*b*k3-
B3^2*cos(k3*b)*sin(k3*b)-A3^2*cos(k3)*sin(k3)-A3^2*k3+2*A3*B3*cos(k3)^2-
B3^2*k3+B3^2*cos(k3)*sin(k3))+1/2*N0^2*a/Cstar+1/2*etatilda*a*Thetapg^2+1/2*N0^2*(b-
a)/C+1/2*N0^2*(1-b)/C;

function[a_1,b_1,c_1,d_1,a_2,b_2,c_2,d_2,a_3,b_3,c_3,d_3,k1,k2,k3] =
ConstantsCZ_clampB(Nn,TH,aa,bin) %No "P" for now...
% Uses the hand solution to find the const. of integ.
global Ds D Dc mstar rho h
k1 = sqrt(Nn/Ds);
k2 = sqrt(Nn/Dc);
k3 = sqrt(Nn/D);
%First, the Z's:
z_1 = sqrt(Ds/Dc)*sin(k2*bin-k2*aa)*cos(k3-k3*bin)+sqrt(Ds/D)*cos(k2*bin-k2*aa)*sin(k3-
k3*bin);
z_2 = cos(k3-k3*bin)*cos(k2*bin)-sqrt(Dc/D)*sin(k3-k3*bin)*sin(k2*bin);
z_3 = cos(k3-k3*bin)*sin(k2*bin)+sqrt(Dc/D)*sin(k3-k3*bin)*cos(k2*bin);
%z_4 = k3*sin(k3-k3*b);
%z_5 = cos(k3-k3*b)-(1+k3*sin(k3-k3*b));
hc = sin(k1*aa)*cos(k2*bin-k2*aa)*cos(k3-k3*bin)+sqrt(Ds/Dc)*cos(k1*aa)*sin(k2*bin-
k2*aa)*cos(k3-k3*bin)+sqrt(Ds/D)*cos(k1*aa)*cos(k2*bin-k2*aa)*sin(k3-k3*bin)-
sqrt(Dc/D)*sin(k1*aa)*sin(k2*bin-k2*aa)*sin(k3-k3*bin);
mlambda = mstar*TH+(rho+(h/2))*Nn;
a_1 = -(mlambda*z_1)/(Nn*hc);
b_1 = 0;
c_1 = 0;
d_1 = (mlambda/(Nn*hc))*(hc-sin(k1*aa));
a_2 = (mlambda*z_2*sin(k1*aa))/(Nn*hc);
b_2 = (mlambda*z_3*sin(k1*aa))/(Nn*hc);
c_2 = 0;
d_2 = -(mlambda*sin(k1*aa))/(Nn*hc);
a_3 = (mlambda*sin(k1*aa)*cos(k3))/(Nn*hc);
b_3 = (mlambda*sin(k1*aa)*sin(k3))/(Nn*hc);
c_3 = 0;
d_3 = -(mlambda*sin(k1*aa))/(Nn*hc);
```

To plot solutions together:

```
%The purpose of this function is to plot the results for the no contact
%zone as well as the contact zone solution on the same axes
%  H I N G E D
clear
bondsize = (0.01:0.01:0.90);
%First, call the the data saved from the No CZ and CZ solutions
load NOCZRes_h.mat;
load NOCZRes2_h.mat;
load NOCZStabRes_h.mat;
load NOCZStabRes2_h.mat;
%[N0,Q,F,w0,ERR,PE]
load CZRes_h.mat;
load CZRes2_h.mat;
load CZStabRes_h.mat;
load CZStabRes2_h.mat;
load CZValRes_h.mat;
load CZValRes2_h.mat;
%[N0,Q,w0,ERR,F,PE,Val] (1 yes, 0 no)]
load NOCZCritBuck_h.mat;
load CZCritBuck_h.mat;
%[a*,Ncr,Qcr]
%load EPRes2_h.mat; %Edge Point Solution (2nd Root)
%[N0,Q,w0,ERR,PE,Val] (1 yes, 0 no)]
%-----------------------------------------------
test = 1; % a = size(test)
```

```
size = [80,70,60,50,40,30,20];
%Find the critical buckling temperature
CritVal = zeros(length(size),3);
CZCritVal = zeros(length(size),3);
for i = 1:length(bondsize)
    for ii = 1:length(size)
    if BuckleResults(i,1) == (1- bondsize(size(ii)));  %matching value of a*
        CritVal(ii,1) = bondsize(size(ii)); % a
        CritVal(ii,2) = BuckleResults(i,2); % N
        CritVal(ii,3) = BuckleResults(i,3); % Q
    end
    if CZBuckleResults(i,1) == (1- bondsize(size(ii)));  %matching value of a*
        CZCritVal(ii,1) = bondsize(size(ii)); % a
        CZCritVal(ii,2) = CZBuckleResults(i,2); % N
        CZCritVal(ii,3) = CZBuckleResults(i,3); % Q
    end
    end
end
%We must sort the data!
%1st separate the full NOCZ data
%ResultsPG and ResultsPG2 are the same length.
dex11=1;dex12=1;dex21=1;dex22=1;
for p=1:length(ResultsPG(:,1,size(test)))
    if ResultsPG(p,1,size(test))<CritVal(test,2) &&
ResultsPG(p,2,size(test))<CritVal(test,3) %N<Ncr, Q<Qcr
        full11(dex11,:)=ResultsPG(p,:,size(test));
        dex11=dex11+1;
    else
        full12(dex12,:)=ResultsPG(p,:,size(test));
        dex12=dex12+1;
    end
    if ResultsPG2(p,1,size(test))<CritVal(test,2) &&
ResultsPG2(p,2,size(test))>CritVal(test,3) %N<Ncr, Q>Qcr
        full21(dex21,:)=ResultsPG2(p,:,size(test));
        dex21=dex21+1;
    else
        full22(dex22,:)=ResultsPG2(p,:,size(test));
        dex22=dex22+1;
    end
end
%2nd separate the full NOCZ data
%CZResultsPG and CZResultsPG2 are the same length.
czdex11=1;czdex12=1;czdex21=1;czdex22=1;
for q=1:length(CZResultsPG(:,1,size(test)))
    if CZResultsPG(q,1,size(test))<CZCritVal(test,2) &&
CZResultsPG(q,2,size(test))<CZCritVal(test,3) %N<Ncr, Q<Qcr
        CZfull11(czdex11,:)=CZResultsPG(q,:,size(test));
        czdex11=czdex11+1;
    else
        CZfull12(czdex12,:)=CZResultsPG(q,:,size(test));
        czdex12=czdex12+1;
    end
    if CZResultsPG2(q,1,size(test))<CZCritVal(test,2) &&
CZResultsPG2(q,2,size(test))>CZCritVal(test,3) %N<Ncr, Q>Qcr
        CZfull21(czdex21,:)=CZResultsPG2(q,:,size(test));
        czdex21=czdex21+1;
    else
        CZfull22(czdex22,:)=CZResultsPG2(q,:,size(test));
        czdex22=czdex22+1;
    end
end
%3rd separate the NOCZ stable branches
%StablePG and StablePG2 are the same length.
ind11=1; ind12=1; ind21=1; ind22=1;
for j=1:length(StablePG(:,1,size(test)))
    if StablePG(j,1,size(test))<CritVal(test,2) && StablePG(j,1,size(test))>0 %N<Ncr
        root11s(ind11,:)=StablePG(j,:,size(test));
        ind11=ind11+1;
    elseif StablePG(j,1,size(test))>CritVal(test,2)&& StablePG(j,1,size(test))>0%N>Ncr
        root12s(ind12,:)=StablePG(j,:,size(test));
        ind12=ind12+1;
```

```
    end
    if StablePG2(j,1,size(test))<CritVal(test,2) && StablePG2(j,1,size(test))>0%N<Ncr
        root21s(ind21,:)=StablePG2(j,:,size(test));
        ind21=ind21+1;
    elseif StablePG2(j,1,size(test))>CritVal(test,2)&& StablePG2(j,1,size(test))>0%N>Ncr
        root22s(ind22,:)=StablePG2(j,:,size(test));
        ind22=ind22+1;
    end
end
%4th separate the CZ stable branches
%CZStablePG and CZStablePG2 are the same length.
czin11=1; czin12=1; czin21=1; czin22=1;
for v=1:length(CZStablePG(:,1,size(test)))
    if CZStablePG(v,1,size(test))<CZCritVal(test,2) && CZStablePG(v,1,size(test))>0
%N<Ncr
        CZroot11s(czin11,:)=CZStablePG(v,:,size(test));
        czin11=czin11+1;
    %elseif CZStablePG(v,1,size(test))>CZCritVal(test,2)&&
CZStablePG(v,1,size(test))>0%N>Ncr
    %    CZroot12s(czin12,:)=CZStablePG(v,:,size(test));
    %    czin12=czin12+1;
    end
    if CZStablePG2(v,1,size(test))<CZCritVal(test,2) &&
CZStablePG2(v,1,size(test))>0%N<Ncr
        CZroot21s(czin21,:)=CZStablePG2(v,:,size(test));
        czin21=czin21+1;
    %elseif CZStablePG2(v,1,size(test))>CZCritVal(test,2)&&
CZStablePG2(v,1,size(test))>0%N>Ncr
    %    CZroot22s(czin22,:)=CZStablePG2(v,:,size(test));
    %    czin22=czin22+1;
    end
end
%5th, deal with valid CZ for hinged ends...
%CZvalidPG and CZvalidPG2 are the same length
v1=1;v2=1;
for count=1:length(CZValidPG(:,1,size(test)))
    if CZValidPG(count,5,size(test))<0 && CZValidPG(count,1,size(test))>0 %F<0, and N>0,
1st root unstable
        valid1(v1,:)=CZValidPG(count,:,size(test));
        v1=v1+1;
    end
    if CZValidPG2(count,5,size(test))>0 && CZValidPG2(count,1,size(test))>0 %F>0, and
N>0, 2nd root stable
        valid2(v2,:)=CZValidPG2(count,:,size(test));
        v2=v2+1;
    end
end
Sample = (-0.3:0.05:0.3); %w vals To plot flat line for crit buckle temp
Sample2 = (0:2:80); %w vals To plot flat line for crit buckle temp
%PLOT 1 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,2),full11(:,6),'k--','LineWidth',2)
hold on
plot(full12(:,2),full12(:,6),'k--','LineWidth',2)
hold on
plot(full21(:,2),full21(:,6),'k--','LineWidth',2)
hold on
plot(full22(:,2),full22(:,6),'k--','LineWidth',2)
hold on
plot(root11s(:,2),root11s(:,6),'k','LineWidth',2)
hold on
plot(root12s(:,2),root12s(:,6),'k','LineWidth',2)
hold on
plot(root21s(:,2),root21s(:,6),'k','LineWidth',2)
hold on
plot(root22s(:,2),root22s(:,6),'k','LineWidth',2)
hold on
plot(CZfull11(:,2),CZfull11(:,6),'b--','LineWidth',2)
hold on
plot(CZfull12(:,2),CZfull12(:,6),'b--','LineWidth',2)
hold on
```

```
plot(CZfull21(:,2),CZfull21(:,6),'b--','LineWidth',2)
hold on
plot(CZfull22(:,2),CZfull22(:,6),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,2),CZroot11s(:,6),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,2),CZroot12s(:,6),'b','LineWidth',2)
hold on
plot(CZroot21s(:,2),CZroot21s(:,6),'b','LineWidth',2)
%hold on
%plot(CZroot22s(:,2),CZroot22s(:,6),'b','LineWidth',2)
hold on
plot(valid1(:,2),valid1(:,6),'r--','LineWidth',2)
hold on
plot(valid2(:,2),valid2(:,6),'r','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('P','fontsize',16,'fontweight','b')
title(['Potential Energy, a='
num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([0 0.028 0 1])
%grid on
%PLOT 2 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,4),full11(:,2),'k--','LineWidth',2)
hold on
plot(full12(:,4),full12(:,2),'k--','LineWidth',2)
hold on
plot(full21(:,4),full21(:,2),'k--','LineWidth',2)
hold on
plot(full22(:,4),full22(:,2),'k--','LineWidth',2)
hold on
plot(root11s(:,4),root11s(:,2),'k','LineWidth',2)
hold on
plot(root12s(:,4),root12s(:,2),'k','LineWidth',2)
hold on
plot(root21s(:,4),root21s(:,2),'k','LineWidth',2)
hold on
plot(root22s(:,4),root22s(:,2),'k','LineWidth',2)
hold on
plot(CZfull11(:,3),CZfull11(:,2),'b--','LineWidth',2)
hold on
plot(CZfull12(:,3),CZfull12(:,2),'b--','LineWidth',2)
hold on
plot(CZfull21(:,3),CZfull21(:,2),'b--','LineWidth',2)
hold on
plot(CZfull22(:,3),CZfull22(:,2),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,3),CZroot11s(:,2),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,3),CZroot12s(:,2),'b','LineWidth',2)
hold on
plot(CZroot21s(:,3),CZroot21s(:,2),'b','LineWidth',2)
%hold on
%plot(CZroot22s(:,3),CZroot22s(:,2),'b','LineWidth',2)
hold on
plot(valid1(:,3),valid1(:,2),'r--','LineWidth',2)
hold on
plot(valid2(:,3),valid2(:,2),'r','LineWidth',2)
hold on
plot(Sample,CritVal(test,3)*ones(1,length(Sample)))
hold on
plot(Sample,CZCritVal(test,3)*ones(1,length(Sample)))
xlabel('w(0)','fontsize',16,'fontweight','b')
ylabel('aQ','fontsize',18,'fontweight','b')
title(['Deflection, a=' num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([-0.2 0.25 0 0.028])
%grid on
%PLOT 3 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,2),full11(:,1),'k--','LineWidth',2)
hold on
```

```
plot(full12(:,2),full12(:,1),'k--','LineWidth',2)
hold on
plot(full21(:,2),full21(:,1),'k--','LineWidth',2)
hold on
plot(full22(:,2),full22(:,1),'k--','LineWidth',2)
hold on
plot(root11s(:,2),root11s(:,1),'k','LineWidth',2)
hold on
plot(root12s(:,2),root12s(:,1),'k','LineWidth',2)
hold on
plot(root21s(:,2),root21s(:,1),'k','LineWidth',2)
hold on
plot(root22s(:,2),root22s(:,1),'k','LineWidth',2)
hold on
plot(CZfull11(:,2),CZfull11(:,1),'b--','LineWidth',2)
hold on
plot(CZfull12(:,2),CZfull12(:,1),'b--','LineWidth',2)
hold on
plot(CZfull21(:,2),CZfull21(:,1),'b--','LineWidth',2)
hold on
plot(CZfull22(:,2),CZfull22(:,1),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,2),CZroot11s(:,1),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,2),CZroot12s(:,1),'b','LineWidth',2)
hold on
plot(CZroot21s(:,2),CZroot21s(:,1),'b','LineWidth',2)
%hold on
%plot(CZroot22s(:,2),CZroot22s(:,1),'b','LineWidth',2)
hold on
plot(valid1(:,2),valid1(:,1),'r--','LineWidth',2)
hold on
plot(valid2(:,2),valid2(:,1),'r','LineWidth',2)
hold on
plot(CritVal(test,3)*ones(1,length(Sample2)),Sample2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
title(['N vs theta, a=' num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([0 0.028 0 80])
%grid on
%PLOT 4 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,2),full11(:,3),'k--','LineWidth',2)
hold on
plot(full12(:,2),full12(:,3),'k--','LineWidth',2)
hold on
plot(full21(:,2),full21(:,3),'k--','LineWidth',2)
hold on
plot(full22(:,2),full22(:,3),'k--','LineWidth',2)
hold on
plot(root11s(:,2),root11s(:,3),'k','LineWidth',2)
hold on
plot(root12s(:,2),root12s(:,3),'k','LineWidth',2)
hold on
plot(root21s(:,2),root21s(:,3),'k','LineWidth',2)
hold on
plot(root22s(:,2),root22s(:,3),'k','LineWidth',2)
hold on
plot(CZfull11(:,2),CZfull11(:,5),'b--','LineWidth',2)
hold on
plot(CZfull12(:,2),CZfull12(:,5),'b--','LineWidth',2)
hold on
plot(CZfull21(:,2),CZfull21(:,5),'b--','LineWidth',2)
hold on
plot(CZfull22(:,2),CZfull22(:,5),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,2),CZroot11s(:,5),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,2),CZroot12s(:,5),'b','LineWidth',2)
hold on
plot(CZroot21s(:,2),CZroot21s(:,5),'b','LineWidth',2)
```

```
%hold on
%plot(CZroot22s(:,2),CZroot22s(:,5),'b','LineWidth',2)
hold on
plot(valid1(:,2),valid1(:,5),'r--','LineWidth',2)
hold on
plot(valid2(:,2),valid2(:,5),'r','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('F','fontsize',16,'fontweight','b')
title(['F vs theta, a=' num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([0 0.028 -60 80])
%grid on
%-------------------------------------------------------------
%Plot of critical buckling results
%Get rid of higher buckling modes first
index2=1; index4=1;
for index = 1:length(BuckleResults(:,1))
if BuckleResults(index,3)<0.009% && BuckleResults(index,3)~=0
    CritTempPlot(index2,:)=BuckleResults(index,:);
    index2=index2+1;
end
end
for index3 = 1:length(CZBuckleResults(:,1))
if CZBuckleResults(index3,3)<0.009% && CZBuckleResults(index3,3)~=0
    CZCritTempPlot(index4,:)=CZBuckleResults(index3,:);
    index4=index4+1;
end
end
figure
plot(CritTempPlot(:,1),CritTempPlot(:,3),'b','LineWidth',2)
hold on
plot(CZCritTempPlot(:,1),CZCritTempPlot(:,3),'r','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('aQcr','fontsize',18,'fontweight','b')
grid on
figure
plot(CritTempPlot(:,1),CritTempPlot(:,2),'b','LineWidth',2)
hold on
plot(CZCritTempPlot(:,1),CZCritTempPlot(:,2),'r','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('Ncr','fontsize',18,'fontweight','b')
grid on

%The purpose of this function is to plot the results for the no contact
%zone as well as the contact zone solution on the same axes
% C L A M P E D
clear
bondsize = (0.01:0.01:0.90);
%First, call the the data saved from the No CZ and CZ solutions
load NOCZRes_c.mat;
load NOCZRes2_c.mat;
load NOCZStabRes_c.mat;
load NOCZStabRes2_c.mat;
%[N0,Q,F,w0,ERR,PE]
load CZRes_c.mat;
load CZRes2_c.mat;
load CZStabRes_c.mat;
load CZStabRes2_c.mat;
load CZValRes_c.mat;
load CZValRes2_c.mat;
load CZPropRes_c.mat;
load CZPropRes2_c.mat;
%[N0,Q,w0,ERR,F,PE,Val] (1 yes, 0 no)]
load NOCZCritBuck_c.mat;
load CZCritBuck_c.mat;
%[a*,Ncr,Qcr]
%---------------------------------------------
% introduce the prop CZ results to superpose...
%Choose the appropriate data for what case I'm looking at!
load CZProp_b_60.mat;
load CZProp_b2_60.mat;
%[N0,Q,w0,ERR,F,PE,Val=2]
```

```
%must make one nice array for this
pcount=1;
for p_in = 1:length(CZPropPG2B(1,1,:))%Every moving b value  %PROP CZ!!!
    for p_in2 = 1:length(CZPropPG2B(:,1,1))
        if CZPropPG2B(p_in2,5,p_in)>0 && CZPropPG2B(p_in2,1,p_in)>0
            CZ_prop(pcount,:)=CZPropPG2B(p_in2,:,p_in);
            pcount=pcount+1;
        end
    end
end
CZ_propsort=sortrows(CZ_prop);
pcountnew=1;
for p_innew = 1:length(CZPropPGB(1,1,:))%Every moving b value  %PROP CZ!!!
    for p_in2new = 1:length(CZPropPGB(:,1,1))
        if CZPropPGB(p_in2new,7,p_innew)>1 && CZPropPGB(p_in2new,1,p_innew)>0
            CZ_propnew(pcountnew,:)=CZPropPGB(p_in2new,:,p_innew);
            pcountnew=pcountnew+1;
        end
    end
end
CZ_propsortnew=sortrows(CZ_propnew);
%-----------------------------------------------
test = 3; % a = size(test) %When test >=3, there is no "valid2"
size = [80,70,60,50,40,30,20];
%Find the critical buckling temperature
CritVal = zeros(length(size),3);
CZCritVal = zeros(length(size),3);
for i = 1:length(BuckleResults(:,1))
    for ii = 1:length(size)
    if BuckleResults(i,1) == (1- bondsize(size(ii)));  %matching value of a*
        CritVal(ii,1) = bondsize(size(ii)); % a
        CritVal(ii,2) = BuckleResults(i,2); % N
        CritVal(ii,3) = BuckleResults(i,3); % Q
    end
    end
end
for iii = 1:length(bondsize)
    for iv = 1:length(size)
    if CZBuckleResults(iii,1) == (1- bondsize(size(iv)));  %matching value of a*
        CZCritVal(iv,1) = bondsize(size(iv)); % a
        CZCritVal(iv,2) = CZBuckleResults(iii,2); % N
        CZCritVal(iv,3) = CZBuckleResults(iii,3); % Q
    end
    end
end
%We must sort the data!
%1st separate the full NOCZ data
%ResultsPG and ResultsPG2 are the same length.
dex11=1;dex12=1;dex21=1;dex22=1;
for p=1:length(ResultsPG(:,1,size(test)))
    if ResultsPG(p,1,size(test))<CritVal(test,2) &&
ResultsPG(p,2,size(test))<CritVal(test,3) %N<Ncr, Q<Qcr
        full11(dex11,:)=ResultsPG(p,:,size(test));
        dex11=dex11+1;
    else
        full12(dex12,:)=ResultsPG(p,:,size(test));
        dex12=dex12+1;
    end
    if ResultsPG2(p,1,size(test))<CritVal(test,2) &&
ResultsPG2(p,2,size(test))>CritVal(test,3) %N<Ncr, Q>Qcr
        full21(dex21,:)=ResultsPG2(p,:,size(test));
        dex21=dex21+1;
    else
        full22(dex22,:)=ResultsPG2(p,:,size(test));
        dex22=dex22+1;
    end
end
%2nd separate the full NOCZ data
%CZResultsPG and CZResultsPG2 are the same length.
czdex11=1;czdex12=1;czdex21=1;czdex22=1;
for q=1:length(CZResultsPG(:,1,size(test)))
```

```
    if CZResultsPG(q,1,size(test))<CZCritVal(test,2) &&
CZResultsPG(q,2,size(test))<CZCritVal(test,3) %N<Ncr, Q<Qcr
        CZfull11(czdex11,:)=CZResultsPG(q,:,size(test));
        czdex11=czdex11+1;
    else
        CZfull12(czdex12,:)=CZResultsPG(q,:,size(test));
        czdex12=czdex12+1;
    end
    if CZResultsPG2(q,1,size(test))<CZCritVal(test,2) &&
CZResultsPG2(q,2,size(test))>CZCritVal(test,3) %N<Ncr, Q>Qcr
        CZfull21(czdex21,:)=CZResultsPG2(q,:,size(test));
        czdex21=czdex21+1;
    else
        CZfull22(czdex22,:)=CZResultsPG2(q,:,size(test));
        czdex22=czdex22+1;
    end
end
%3rd separate the NOCZ stable branches
%StablePG and StablePG2 are the same length.
ind11=1; ind12=1; ind21=1; ind22=1;
for j=1:length(StablePG(:,1,size(test)))
    if StablePG(j,1,size(test))<CritVal(test,2) && StablePG(j,1,size(test))>0 %N<Ncr
        root11s(ind11,:)=StablePG(j,:,size(test));
        ind11=ind11+1;
    elseif StablePG(j,1,size(test))>CritVal(test,2)&& StablePG(j,1,size(test))>0%N>Ncr
        root12s(ind12,:)=StablePG(j,:,size(test));
        ind12=ind12+1;
    end
    if StablePG2(j,1,size(test))<CritVal(test,2) && StablePG2(j,1,size(test))>0%N<Ncr
        root21s(ind21,:)=StablePG2(j,:,size(test));
        ind21=ind21+1;
    elseif StablePG2(j,1,size(test))>CritVal(test,2)&& StablePG2(j,1,size(test))>0%N>Ncr
        root22s(ind22,:)=StablePG2(j,:,size(test));
        ind22=ind22+1;
    end
end
%4th separate the CZ stable branches
%CZStablePG and CZStablePG2 are the same length.
czin11=1; czin12=1; czin21=1; czin22=1;
for v=1:length(CZStablePG(:,1,size(test)))
    if CZStablePG(v,1,size(test))<CZCritVal(test,2) && CZStablePG(v,1,size(test))>0
%N<Ncr
        CZroot11s(czin11,:)=CZStablePG(v,:,size(test));
        czin11=czin11+1;
    %elseif CZStablePG(v,1,size(test))>CZCritVal(test,2)&&
CZStablePG(v,1,size(test))>0%N>Ncr
    %    CZroot12s(czin12,:)=CZStablePG(v,:,size(test));
    %    czin12=czin12+1;
    end
    if CZStablePG2(v,1,size(test))<CZCritVal(test,2) &&
CZStablePG2(v,1,size(test))>0%N<Ncr
        CZroot21s(czin21,:)=CZStablePG2(v,:,size(test));
        czin21=czin21+1;
    %elseif CZStablePG2(v,1,size(test))>CZCritVal(test,2)&&
CZStablePG2(v,1,size(test))>0%N>Ncr
    %    CZroot22s(czin22,:)=CZStablePG2(v,:,size(test));
    %    czin22=czin22+1;
    end
end
%5th, deal with valid CZ for hinged ends...
%CZvalidPG and CZvalidPG2 are the same length
v1=1;v2=1;
for count=1:length(CZValidPG(:,1,size(test)))
    if CZValidPG(count,5,size(test))>0 && CZValidPG(count,1,size(test))>0 %F>0, and N>0,
1st root stable
        valid1(v1,:)=CZValidPG(count,:,size(test));
        v1=v1+1;
    end
    if CZValidPG2(count,5,size(test))<0 && CZValidPG2(count,1,size(test))>0 %F<0, and
N>0, 2nd root unstable
        valid2(v2,:)=CZValidPG2(count,:,size(test));
```

```
        v2=v2+1;
    end
end
Sample = (-0.3:0.05:0.3); %w vals To plot flat line for crit buckle temp
Sample2 = (0:2:80); %w vals To plot flat line for crit buckle temp
%PLOT 1 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,2),full11(:,6),'k--','LineWidth',2)
hold on
plot(full12(:,2),full12(:,6),'k--','LineWidth',2)
hold on
plot(full21(:,2),full21(:,6),'k--','LineWidth',2)
hold on
plot(full22(:,2),full22(:,6),'k--','LineWidth',2)
hold on
plot(root11s(:,2),root11s(:,6),'k','LineWidth',2)
%hold on
%plot(root12s(:,2),root12s(:,6),'k','LineWidth',2)
hold on
plot(root21s(:,2),root21s(:,6),'k','LineWidth',2)
%hold on
%plot(root22s(:,2),root22s(:,6),'k','LineWidth',2)
hold on
plot(CZfull11(:,2),CZfull11(:,6),'b--','LineWidth',2)
hold on
plot(CZfull12(:,2),CZfull12(:,6),'b--','LineWidth',2)
hold on
plot(CZfull21(:,2),CZfull21(:,6),'b--','LineWidth',2)
hold on
plot(CZfull22(:,2),CZfull22(:,6),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,2),CZroot11s(:,6),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,2),CZroot12s(:,6),'b','LineWidth',2)
hold on
plot(CZroot21s(:,2),CZroot21s(:,6),'b','LineWidth',2)
%hold on
%plot(CZroot22s(:,2),CZroot22s(:,6),'b','LineWidth',2)
hold on
plot(valid1(:,2),valid1(:,6),'r','LineWidth',2)
%hold on
%plot(valid2(:,2),valid2(:,6),'r--','LineWidth',2)
hold on
plot(CZ_propsort(:,2),CZ_propsort(:,6),'g','LineWidth',2)
hold on
plot(CZ_propsortnew(:,2),CZ_propsortnew(:,6),'g--','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('P','fontsize',16,'fontweight','b')
title(['Potential Energy, a='
num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([0 0.028 0 1])
%grid on
%PLOT 2 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,4),full11(:,2),'k--','LineWidth',2)
hold on
plot(full12(:,4),full12(:,2),'k--','LineWidth',2)
hold on
plot(full21(:,4),full21(:,2),'k--','LineWidth',2)
hold on
plot(full22(:,4),full22(:,2),'k--','LineWidth',2)
hold on
plot(root11s(:,4),root11s(:,2),'k','LineWidth',2)
%hold on
%plot(root12s(:,4),root12s(:,2),'k','LineWidth',2)
hold on
plot(root21s(:,4),root21s(:,2),'k','LineWidth',2)
%hold on
%plot(root22s(:,4),root22s(:,2),'k','LineWidth',2)
hold on
plot(CZfull11(:,3),CZfull11(:,2),'b--','LineWidth',2)
```

```
hold on
plot(CZfull12(:,3),CZfull12(:,2),'b--','LineWidth',2)
hold on
plot(CZfull21(:,3),CZfull21(:,2),'b--','LineWidth',2)
hold on
plot(CZfull22(:,3),CZfull22(:,2),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,3),CZroot11s(:,2),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,3),CZroot12s(:,2),'b','LineWidth',2)
hold on
plot(CZroot21s(:,3),CZroot21s(:,2),'b','LineWidth',2)
%hold on
%plot(CZroot22s(:,3),CZroot22s(:,2),'b','LineWidth',2)
hold on
plot(valid1(:,3),valid1(:,2),'r','LineWidth',2)
%hold on
%plot(valid2(:,3),valid2(:,2),'r--','LineWidth',2)
hold on
plot(CZ_propsort(:,3),CZ_propsort(:,2),'g','LineWidth',2)
hold on
plot(CZ_propsortnew(:,3),CZ_propsortnew(:,2),'g--','LineWidth',2)
hold on
plot(Sample,CritVal(test,3)*ones(1,length(Sample)),'k--','LineWidth',1)
hold on
plot(Sample,CZCritVal(test,3)*ones(1,length(Sample)),'b--','LineWidth',1)
xlabel('w(0)','fontsize',24,'fontweight','b')
ylabel('aQ','fontsize',24,'fontweight','b')
title(['Deflection, a=' num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([-0.2 0.25 0 0.028])
%grid on
%PLOT 3 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,2),full11(:,1),'k--','LineWidth',2)
hold on
plot(full12(:,2),full12(:,1),'k--','LineWidth',2)
hold on
plot(full21(:,2),full21(:,1),'k--','LineWidth',2)
hold on
plot(full22(:,2),full22(:,1),'k--','LineWidth',2)
hold on
plot(root11s(:,2),root11s(:,1),'k','LineWidth',2)
%hold on
%plot(root12s(:,2),root12s(:,1),'k','LineWidth',2)
hold on
plot(root21s(:,2),root21s(:,1),'k','LineWidth',2)
%hold on
%plot(root22s(:,2),root22s(:,1),'k','LineWidth',2)
hold on
plot(CZfull11(:,2),CZfull11(:,1),'b--','LineWidth',2)
hold on
plot(CZfull12(:,2),CZfull12(:,1),'b--','LineWidth',2)
hold on
plot(CZfull21(:,2),CZfull21(:,1),'b--','LineWidth',2)
hold on
plot(CZfull22(:,2),CZfull22(:,1),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,2),CZroot11s(:,1),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,2),CZroot12s(:,1),'b','LineWidth',2)
hold on
plot(CZroot21s(:,2),CZroot21s(:,1),'b','LineWidth',2)
%hold on
%plot(CZroot22s(:,2),CZroot22s(:,1),'b','LineWidth',2)
hold on
plot(valid1(:,2),valid1(:,1),'r','LineWidth',2)
%hold on
%plot(valid2(:,2),valid2(:,1),'r--','LineWidth',2)
hold on
plot(CZ_propsort(:,2),CZ_propsort(:,1),'g','LineWidth',2)
hold on
```

```
plot(CritVal(test,3)*ones(1,length(Sample2)),Sample2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
title(['N vs theta, a=' num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([0 0.028 0 80])
%grid on
%PLOT 4 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure
plot(full11(:,2),full11(:,3),'k--','LineWidth',2)
hold on
plot(full12(:,2),full12(:,3),'k--','LineWidth',2)
hold on
plot(full21(:,2),full21(:,3),'k--','LineWidth',2)
hold on
plot(full22(:,2),full22(:,3),'k--','LineWidth',2)
hold on
plot(root11s(:,2),root11s(:,3),'k','LineWidth',2)
%hold on
%plot(root12s(:,2),root12s(:,3),'k','LineWidth',2)
hold on
plot(root21s(:,2),root21s(:,3),'k','LineWidth',2)
%hold on
%plot(root22s(:,2),root22s(:,3),'k','LineWidth',2)
hold on
plot(CZfull11(:,2),CZfull11(:,5),'b--','LineWidth',2)
hold on
plot(CZfull12(:,2),CZfull12(:,5),'b--','LineWidth',2)
hold on
plot(CZfull21(:,2),CZfull21(:,5),'b--','LineWidth',2)
hold on
plot(CZfull22(:,2),CZfull22(:,5),'b--','LineWidth',2)
hold on
plot(CZroot11s(:,2),CZroot11s(:,5),'b','LineWidth',2)
%hold on
%plot(CZroot12s(:,2),CZroot12s(:,5),'b','LineWidth',2)
hold on
plot(CZroot21s(:,2),CZroot21s(:,5),'b','LineWidth',2)
%hold on
%plot(CZroot22s(:,2),CZroot22s(:,5),'b','LineWidth',2)
hold on
plot(valid1(:,2),valid1(:,5),'r','LineWidth',2)
%hold on
%plot(valid2(:,2),valid2(:,5),'r--','LineWidth',2)
hold on
plot(CZ_propsort(:,2),CZ_propsort(:,5),'g','LineWidth',2)
xlabel('aQ','fontsize',18,'fontweight','b')
ylabel('F','fontsize',16,'fontweight','b')
title(['F vs theta, a=' num2str(bondsize(size(test)))],'fontsize',16,'fontweight','b')
axis([0 0.028 -60 80])
%grid on
%=====================================================
%Plot of critical buckling results
%Get rid of higher buckling modes first
index2=1; index4=1;
for index = 1:length(BuckleResults(:,1))
if BuckleResults(index,3)<0.0166
    CritTempPlot(index2,:)=BuckleResults(index,:);
    index2=index2+1;
elseif BuckleResults(index,3)<0.017 && BuckleResults(index,1)<0.5
    CritTempPlot(index2,:)=BuckleResults(index,:);
    index2=index2+1;
end
end
for index3 = 1:length(CZBuckleResults(:,1))
if CZBuckleResults(index3,3)<0.0168
    CZCritTempPlot(index4,:)=CZBuckleResults(index3,:);
    index4=index4+1;
end
end
figure
plot(CritTempPlot(:,1),CritTempPlot(:,3),'b','LineWidth',2)
```

```
hold on
plot(CZCritTempPlot(:,1),CZCritTempPlot(:,3),'r','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('aQcr','fontsize',18,'fontweight','b')
grid on
```

To observe debonding behavior:

```
%Looking for Legit cases of propgating contact
%I will assign a value of "a" and look for valid cases...
%This is for when the flap is really Lp = 0.9
% CLAMPED ENDS
%will be performed for 3 specific temperatures, to go along with the
%debonding plot
clear
%Define the usual stuff
global h C D Dp
h = 0.05; % height of baseplate
hp = 0.05; % height of patch
h0 = hp/h; % ratio of heights
E0 = 1; % elastic modulus
C = 12/(h^2); % membrane stiffness of baseplate
Cp = C*E0*h0; % membrane stiffness of patch
D = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rho Ds
% stiffnesses of composite structure
Astar = D+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rho = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rho*Bstar; %Dstar
global  Dc Cs Ce
Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar nstar etatilda
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 0.5; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rho*nstar;
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
global Theta_in a astar b
%Choose values for a, b and N
a_range = (0.02:0.02:0.35);%adjust range depending on Theta
%.015 use .76, .012 use 0.62, 0.008 use 0.35
N_range = (0:0.1:25); %Same values used in Buckle01_CZ
%.015 use 40, .012 use 35, 0.008 use 25
Theta_in = 0.008; %chosen value to plot
Lp = 0.9; %patch length
p=0;
for i=1:length(a_range) %Cycle a/a* values
    p=p+1;
    n=1;
    a=a_range(i); %set present value of a
    astar=1-a; %set present value of a*
    b_range=(a_range(i):0.005:Lp);
    for ii=1:length(b_range)%Cycle possible b values, based on a
        b=b_range(ii); %set present value of b
        for iii=1:length(N_range)%Cycle possible N's
            INTCON(iii) = Integrability_clamp_CZ(N_range(iii));
            %[value of integrability, we want it to be zero]
        end
        for iii2 = 1:length(N_range)-1;%Look for a sign change, find root, save it
            if INTCON(iii2)*INTCON(iii2+1)<0;
                Nroot(iii2)=fzero(@Integrability_clamp_CZ,N_range(iii2));%Root of N
```

```
                TEST(n,1,p)=b; %Unsorted Data at this point
                TEST(n,2,p)=Nroot(iii2);

[TEST(n,3,p),TEST(n,4,p),TEST(n,5,p),TEST(n,6,p),TEST(n,7,p)]=B_info_CCZ(Nroot(iii2));
                n=n+1;
                %[b,N,w0,ERR,F,PE,ValidCZ] 0-no, 1-full, 2-prop
            end
        end
    end
end
for j=1:length(a_range)
    for jj=1:length(TEST(:,1,1))
        if TEST(jj,5,j)>0 && TEST(jj,7,j)>1
            StableTEST(jj,1:7,j)=TEST(jj,1:7,j);
            StableTEST(jj,8,j)=1-a_range(j);%a*
        end
    end
end
k=1;
for m=1:length(StableTEST(1,1,:))
    for mm=1:length(StableTEST(:,1,1))
        if StableTEST(mm,8,m)>0 && StableTEST(mm,2,m)<N_range(length(N_range))
            propDATA(k,1:8)=StableTEST(mm,1:8,m);
            k=k+1;
        end
    end
end
%save the Data for the debond plot program
%Change the name depending on what theta you use
save Prop_1.mat propDATA;
figure
plot(propDATA(:,8),propDATA(:,4),'LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('ERR','fontsize',16,'fontweight','b')
grid on
figure
plot(propDATA(:,8),propDATA(:,2),'LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
grid on

%This is to make the Debonding Plot..
%Clamped-Fixed
clear
a_vals = (0.01:0.01:0.90);
a_vals_star=1-a_vals;
load C_NOCZ.mat
IntactData = CleanDATA;
 % IntactDATA [Theta,N,F,w0,ERR,PE,3rdMinor], value of a is 3rd dimension
load C_CZ.mat
PDData = CleanDATA;
PDValidData = ValidDATA;
% Partially Debonded DATA [Theta,N,F,w0,ERR,PE,Valid]
load NOCZCritBuck_c.mat;
load CZCritBuck_c.mat;
%[a*,Ncr,Qcr]
%Important note, Intact data contains one more row of theta values.  So to
%make the first index, subtract one from intact to get corresponding PD
TheLoad = [41,61,76];%For intact, -1 for PD
last=length(TheLoad)+4;
for p1=1:length(TheLoad)
    for p2=1:length(a_vals)
        IntactPlot(p2,p1)=IntactData(TheLoad(p1),5,p2); %ERR for each a val
        if p1==1
            IntactPlot(p2,last-3)=IntactData(TheLoad(p1),2,p2);
        elseif p1==2
            IntactPlot(p2,last-2)=IntactData(TheLoad(p1),2,p2);
        else
            IntactPlot(p2,last-1)=IntactData(TheLoad(p1),2,p2);
        end
        IntactPlot(p2,last)=a_vals_star(p2);
```

```
        PDplottemp(p2,p1)=PDValidData(TheLoad(p1)-1,5,p2);
        if p1==1
            PDplottemp(p2,last-3)=PDValidData(TheLoad(p1)-1,2,p2);
        elseif p1==2
            PDplottemp(p2,last-2)=PDValidData(TheLoad(p1)-1,2,p2);
        else
            PDplottemp(p2,last-1)=PDValidData(TheLoad(p1)-1,2,p2);
        end
        PDplottemp(p2,last)=a_vals_star(p2);
    end
end
n1=1; n2=1; n3=1;
for v1=1:length(PDplottemp(:,1))
    if PDplottemp(v1,1)>0
        PDplot1(n1,1)=PDplottemp(v1,1); %ERR
        PDplot1(n1,2)=PDplottemp(v1,4); %N
        PDplot1(n1,3)=PDplottemp(v1,last); %last column is a*
        n1=n1+1;
    end
end
for v2=2:length(PDplottemp(:,2))%Two to discard 1st wild point
    if PDplottemp(v2,2)>0
        PDplot2(n2,1)=PDplottemp(v2,2); %ERR
        PDplot2(n2,2)=PDplottemp(v2,5); %N
        PDplot2(n2,3)=PDplottemp(v2,last);%a*
        n2=n2+1;
    end
end
for v3=1:length(PDplottemp(:,3))
    if PDplottemp(v3,3)>0
        PDplot3(n3,1)=PDplottemp(v3,3);%ERR
        PDplot3(n3,2)=PDplottemp(v3,6); %N
        PDplot3(n3,3)=PDplottemp(v3,last);%a*
        n3=n3+1;
    end
end
%Now some smoothing is required.
%IntactPlot = [ERR1,ERR2,ERR3,N1,N2,N3,a*]
val=2; val2=2; val3=2; %2 because 1st point is specified
start=length(IntactPlot(:,last));
for pam=1:start-1
    INplot1(1,:)=[IntactPlot(start,1),IntactPlot(start,4),IntactPlot(start,last)];%1st
point
    if IntactPlot(start-pam,4)<IntactPlot(start-pam+1,4)%Each N should be less than
previous N
        INplot1(val,:)=[IntactPlot(start-pam,1),IntactPlot(start-pam,4),IntactPlot(start-
pam,last)];
        val=val+1;
    end
    INplot2(1,:)=[IntactPlot(start,2),IntactPlot(start,5),IntactPlot(start,last)];%1st
point
    if IntactPlot(start-pam,5)<IntactPlot(start-pam+1,5)%Each N should be less than
previous N
        INplot2(val2,:)=[IntactPlot(start-pam,2),IntactPlot(start-
pam,5),IntactPlot(start-pam,last)];
        val2=val2+1;
    end
    INplot3(1,:)=[IntactPlot(start,3),IntactPlot(start,6),IntactPlot(start,last)];%1st
point
    if IntactPlot(start-pam,6)<IntactPlot(start-pam+1,6)%Each N should be less than
previous N
        INplot3(val3,:)=[IntactPlot(start-pam,3),IntactPlot(start-
pam,6),IntactPlot(start-pam,last)];
        val3=val3+1;
    end
end
%One more thing, to smooth...to break up pre and postbuckling
trav=1; trav2=1; trav3=1; ted=1; ted2=1; ted3=1;
for zach=1:length(INplot1(:,3))
    if INplot1(zach,3)>0.67 %adjust as necessary
        postINplot1(trav,:)=INplot1(zach,:);
```

```
        trav=trav+1;
    elseif INplot1(zach,3)>0.43%adjust as necessary, cz lifted off here
        preINplot1(ted,:)=INplot1(zach,:);
        ted=ted+1;
    end
end
for zach2=1:length(INplot2(:,3))
    if INplot2(zach2,3)>0.5 %adjust as necessary
        postINplot2(trav2,:)=INplot2(zach2,:);
        trav2=trav2+1;
    elseif INplot2(zach2,3)>0.39 %adjust as necessary, cz lifted off here
        preINplot2(ted2,:)=INplot2(zach2,:);
        ted2=ted2+1;
    end
end
for zach3=1:length(INplot3(:,3))
    if INplot3(zach3,3)>0.325 %adjust as necessary
        postINplot3(trav3,:)=INplot3(zach3,:);
        trav3=trav3+1;
    else
        preINplot3(ted3,:)=INplot3(zach3,:);
        ted3=ted3+1;
    end
end
%Load the Data associated w/ prop contact for each load level
load Prop_1.mat;
prop1=propDATA;
load Prop_2.mat;
prop2=propDATA;
load Prop_3.mat;
prop3=propDATA;
%[b,N,w0,ERR,F,PE,ValidCZ] 0-no, 1-full, 2-prop
figure
plot(preINplot1(:,3),preINplot1(:,1),'b','LineWidth',4)
hold on
plot(preINplot2(:,3),preINplot2(:,1),'g','LineWidth',4)
hold on
%plot(preINplot3(:,3),preINplot3(:,1),'r','LineWidth',4)
%hold on
%plot(postINplot1(:,3),postINplot1(:,1),'b','LineWidth',4)
%hold on
%plot(postINplot2(:,3),postINplot2(:,1),'g','LineWidth',4)
%hold on
%plot(postINplot3(:,3),postINplot3(:,1),'r','LineWidth',4)
%hold on
plot(PDplot1(:,3),PDplot1(:,1),'b:','LineWidth',4)
hold on
plot(PDplot2(:,3),PDplot2(:,1),'g:','LineWidth',4)
hold on
plot(PDplot3(:,3),PDplot3(:,1),'r:','LineWidth',4)
hold on
plot(prop1(:,8),prop1(:,4),'b--','LineWidth',4)
hold on
plot(prop2(:,8),prop2(:,4),'g--','LineWidth',4)
hold on
plot(prop3(:,8),prop3(:,4),'r--','LineWidth',4)
xlabel('a*','fontsize',24,'fontweight','b')
ylabel('G','fontsize',24,'fontweight','b')
grid on
figure
plot(INplot1(:,3),INplot1(:,2),'b.','Markersize',8)
hold on
plot(INplot2(:,3),INplot2(:,2),'g.','Markersize',8)
hold on
plot(INplot3(:,3),INplot3(:,2),'r.','Markersize',8)
hold on
plot(PDplot1(:,3),PDplot1(:,2),'b:','LineWidth',2)
hold on
plot(PDplot2(:,3),PDplot2(:,2),'g:','LineWidth',2)
hold on
plot(PDplot3(:,3),PDplot3(:,2),'r:','LineWidth',2)
```

```
hold on
plot(prop1(:,8),prop1(:,2),'b-.','LineWidth',2)
hold on
plot(prop2(:,8),prop2(:,2),'g-.','LineWidth',2)
hold on
plot(prop3(:,8),prop3(:,2),'r-.','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
grid on

%This is to make the Debonding Plot..
%Hinged-Fixed
clear
a_vals = (0.01:0.01:0.90);
a_vals_star=1-a_vals;
load H_NOCZ.mat
IntactData = CleanDATA;
 % IntactDATA [Theta,N,F,w0,ERR,PE,3rdMinor], value of a is 3rd dimension
load H_CZ.mat
PDData = CleanDATA;
PDValidData = ValidDATA;
% Partially Debonded DATA [Theta,N,F,w0,ERR,PE,Valid]
load NOCZCritBuck_h.mat;
load CZCritBuck_h.mat;
%[a*,Ncr,Qcr]
%Important note, Intact data contains one more row of theta values.  So to
%make the first index, subtract one from intact to get corresponding PD
TheLoad = [36,56,76];%For intact, -1 for PD
last=length(TheLoad)+4;  %was +1 w/o N vals included in results
for p1=1:length(TheLoad)
    for p2=1:length(a_vals)
        IntactPlot(p2,p1)=IntactData(TheLoad(p1),5,p2); %ERR for each a val
        if p1==1
            IntactPlot(p2,last-3)=IntactData(TheLoad(p1),2,p2);
        elseif p1==2
            IntactPlot(p2,last-2)=IntactData(TheLoad(p1),2,p2);
        else
            IntactPlot(p2,last-1)=IntactData(TheLoad(p1),2,p2);
        end
        IntactPlot(p2,last)=a_vals_star(p2);
        PDplottemp(p2,p1)=PDValidData(TheLoad(p1)-1,5,p2);
        if p1==1
            PDplottemp(p2,last-3)=PDValidData(TheLoad(p1),2,p2);
        elseif p1==2
            PDplottemp(p2,last-2)=PDValidData(TheLoad(p1),2,p2);
        else
            PDplottemp(p2,last-1)=PDValidData(TheLoad(p1),2,p2);
        end
        PDplottemp(p2,last)=a_vals_star(p2);
    end
end
n1=1; n2=1; n3=1;
for v1=1:length(PDplottemp(:,1))
    if PDplottemp(v1,1)>0
        PDplot1t(n1,1)=PDplottemp(v1,1); %ERR
        PDplot1t(n1,2)=PDplottemp(v1,4); % N
        PDplot1t(n1,3)=PDplottemp(v1,last); %last column is a*
        n1=n1+1;
    end
end
for v2=1:length(PDplottemp(:,2))
    if PDplottemp(v2,2)>0
        PDplot2t(n2,1)=PDplottemp(v2,2);
        PDplot2t(n2,2)=PDplottemp(v2,5);
        PDplot2t(n2,3)=PDplottemp(v2,last);
        n2=n2+1;
    end
end
for v3=1:length(PDplottemp(:,3))
    if PDplottemp(v3,3)>0
        PDplot3t(n3,1)=PDplottemp(v3,3);
```

```
        PDplot3t(n3,2)=PDplottemp(v3,6);
        PDplot3t(n3,3)=PDplottemp(v3,last);
        n3=n3+1;
    end
end
%Now some smoothing is required.
%IntactPlot = [ERR1,ERR2,ERR3,N1,N2,N3,a*]
val=1; val2=1; val3=1;
for pam=1:length(IntactPlot(:,last))
    if IntactPlot(pam,4)<IntactPlot(length(IntactPlot(:,last)),4)
        INplot1(val,:)=[IntactPlot(pam,1),IntactPlot(pam,4),IntactPlot(pam,last)];
        val=val+1;
    end
    if IntactPlot(pam,5)<IntactPlot(length(IntactPlot(:,last)),5)
        INplot2(val2,:)=[IntactPlot(pam,2),IntactPlot(pam,5),IntactPlot(pam,last)];
        val2=val2+1;
    end
    if IntactPlot(pam,6)<IntactPlot(length(IntactPlot(:,last)),6)
        INplot3(val3,:)=[IntactPlot(pam,3),IntactPlot(pam,6),IntactPlot(pam,last)];
        val3=val3+1;
    end
end
len=1; len2=1; len3=1;
for paul=1:length(PDplot1t(:,3))
    if PDplot1t(paul,2)<PDplot1t(length(PDplot1t(:,3))-1,2)%1st one is weird...check if
necessary
        PDplot1(len,:)=PDplot1t(paul,:);
        len=len+1;
    end
end
for paul2=1:length(PDplot2t(:,3))
    if PDplot2t(paul2,2)<PDplot2t(length(PDplot2t(:,3)),2)
        PDplot2(len2,:)=PDplot2t(paul2,:);
        len2=len2+1;
    end
end
for paul3=1:length(PDplot3t(:,3))
    if PDplot3t(paul3,2)<PDplot3t(length(PDplot3t(:,3)),2)
        PDplot3(len3,:)=PDplot3t(paul3,:);
        len3=len3+1;
    end
end
%One more thing, to smooth...to break up pre and postbuckling
trav=1; trav2=1; trav3=1; ted=1; ted2=1; ted3=1;
for zach=1:length(INplot1(:,3))
    if INplot1(zach,3)>0.47 %adjust as necessary
        postINplot1(trav,:)=INplot1(zach,:);
        trav=trav+1;
    else
        preINplot1(ted,:)=INplot1(zach,:);
        ted=ted+1;
    end
end
for zach2=1:length(INplot2(:,3))
    if INplot2(zach2,3)>0.325 %adjust as necessary
        postINplot2(trav2,:)=INplot2(zach2,:);
        trav2=trav2+1;
    else
        preINplot2(ted2,:)=INplot2(zach2,:);
        ted2=ted2+1;
    end
end
for zach3=1:length(INplot3(:,3))
    if INplot3(zach3,3)>0.195 %adjust as necessary
        postINplot3(trav3,:)=INplot3(zach3,:);
        trav3=trav3+1;
    else
        preINplot3(ted3,:)=INplot3(zach3,:);
        ted3=ted3+1;
    end
end
```

```
figure
plot(preINplot1(:,3),preINplot1(:,1),'b','LineWidth',2)
hold on
plot(preINplot2(:,3),preINplot2(:,1),'g','LineWidth',2)
hold on
plot(preINplot3(:,3),preINplot3(:,1),'r','LineWidth',2)
hold on
plot(postINplot1(:,3),postINplot1(:,1),'b','LineWidth',2)
hold on
plot(postINplot2(:,3),postINplot2(:,1),'g','LineWidth',2)
hold on
plot(postINplot3(:,3),postINplot3(:,1),'r','LineWidth',2)
hold on
plot(PDplot1(:,3),PDplot1(:,1),'b:','LineWidth',2)
hold on
plot(PDplot2(:,3),PDplot2(:,1),'g:','LineWidth',2)
hold on
plot(PDplot3(:,3),PDplot3(:,1),'r:','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('ERR','fontsize',16,'fontweight','b')
grid on
figure
plot(INplot1(:,3),INplot1(:,2),'b.','Markersize',8)
hold on
plot(INplot2(:,3),INplot2(:,2),'g.','Markersize',8)
hold on
plot(INplot3(:,3),INplot3(:,2),'r.','Markersize',8)
hold on
plot(PDplot1(:,3),PDplot1(:,2),'b:','LineWidth',2)
hold on
plot(PDplot2(:,3),PDplot2(:,2),'g:','LineWidth',2)
hold on
plot(PDplot3(:,3),PDplot3(:,2),'r:','LineWidth',2)
xlabel('a*','fontsize',16,'fontweight','b')
ylabel('N','fontsize',16,'fontweight','b')
grid on
```

A.2 Maple code

For model of intact structure:

```
restart:
#No contact zone, clamped fixed
M_l := mstar*theta+(rho+h/2)*No:
#H := sin(k1*a)*cos(k3*astar)+sqrt(Ds/D)*cos(k1*a)*sin(k3*astar); A1 := -
M_l*sqrt(Ds/D)*sin(k3*astar)/(No*H):
B1 := 0: #for sym deflection
C1 := 0: #for sym deflection
D1 := (M_l/(No*H))*(H-sin(k1*a)):
A3 := M_l*sin(k1*a)*cos(k3)/(No*H):
B3 := M_l*sin(k1*a)*sin(k3)/(No*H):
C3 := 0:
D3 := -M_l*sin(k1*a)/(No*H):
w1 := A1*cos(k1*x)+B1*sin(K1*x)+C1*x+D1:
w3 := A3*cos(k3*x)+B3*sin(K3*x)+C3*x+D3:
w1_prime := -A1*k1*sin(k1*x)+B1*k1*cos(k1*x)+C1:
w1prime_sq := simplify(expand(w1_prime^2)):
type(w1prime_sq,polynom(anything,theta)): degree(w1prime_sq,theta):
w3_prime := -A3*k3*sin(k3*x)+B3*k3*cos(k3*x)+C3:
w3prime_sq := simplify(expand(w3_prime^2)):
type(w3prime_sq,polynom(anything,theta)):
degree(w3prime_sq,theta):
w1_prime_at_a := subs(x=a,w1_prime):
type(w1_prime_at_a,polynom(anything,theta)):
degree(w1_prime_at_a,theta):
IC1 := int(w1prime_sq, x = 0..a):
IC3 := int(w3prime_sq, x = a..1):
type(IC1,polynom(anything,theta)):
```

```
degree(IC1,theta):
type(IC3,polynom(anything,theta)):
degree(IC3,theta):
#Now let's define the "Z's"
Z0 := astar+a*nstar/Cstar:
Z1 := coeff(w1_prime_at_a,theta,1):
Z2 := coeff(w1_prime_at_a,theta,0):
Z3 := coeff(IC1,theta,2):
Z4 := coeff(IC1,theta,1):
Z5 := coeff(IC1,theta,0):
Z6 := coeff(IC3,theta,2):
Z7 := coeff(IC3,theta,1):
Z8 := coeff(IC3,theta,0):
IntCon := -No*Ca+Z_0*theta-(h/2+rho)*(Z_1*theta+Z_2)-
(1/2)*(Z_3*theta^2+Z_4*theta+Z_5+Z_6*theta^2+Z_7*theta+Z_8):
type(IntCon,polynom(anything,theta)):
degree(IntCon,theta):
Q1 := coeff(IntCon,theta,2):
Q2 := coeff(IntCon,theta,1):
Q3 := coeff(IntCon,theta,0):
#Transversality Condition
kappa1 := -A1*k1^2*cos(k1*a)-B1*k1^2*sin(k1*a): #at a
kappa3 := -A3*k3^2*cos(k3*a)-B3*k3^2*sin(k3*a): #at a
kappa1_sq := simplify(expand(kappa1^2)):
kappa3_sq := simplify(expand(kappa3^2)):
TransCon := (1/2)*(D*kappa3_sq-Ds*kappa1_sq+(No^2)/Ce-2*No*(1-
nstar/Cstar)*theta+etatilda*theta^2):
type(TransCon,polynom(anything,theta)):
degree(TransCon,theta):
Q4 := coeff(TransCon,theta,2):
Q5 := coeff(TransCon,theta,1):
Q6 := coeff(TransCon,theta,0):

restart:
#No contact zone, hinged fixed
M_l := mstar*theta+(rho+h/2)*N0:
A1 := -M_l*(Ds/D)^(1/2)*cos(k3*(1-a))/N0/H:
B1 := 0: #for sym deflection
C1 := 0: #for sym deflection
D1 := M_l/N0:
A3 := -M_l*sin(k1*a)*sin(k3)/(N0*H):
B3 := M_l*sin(k1*a)*cos(k3)/(N0*H):
C3 := 0:
D3 := 0:
w1 := A1*cos(k1*x)+B1*sin(K1*x)+C1*x+D1:
w3 := A3*cos(k3*x)+B3*sin(K3*x)+C3*x+D3:
w1_prime := -A1*k1*sin(k1*x)+B1*k1*cos(k1*x)+C1:
w1prime_sq := simplify(expand(w1_prime^2)):
type(w1prime_sq,polynom(anything,theta)):
degree(w1prime_sq,theta):
w3_prime := -A3*k3*sin(k3*x)+B3*k3*cos(k3*x)+C3:
w3prime_sq := simplify(expand(w3_prime^2)):
type(w3prime_sq,polynom(anything,theta)):
degree(w3prime_sq,theta):
w1_prime_at_a := subs(x=a,w1_prime):
type(w1_prime_at_a,polynom(anything,theta)):
degree(w1_prime_at_a,theta):
IC1 := int(w1prime_sq, x = 0..a):
IC3 := int(w3prime_sq, x = a..1):
type(IC1,polynom(anything,theta)):
degree(IC1,theta):
type(IC3,polynom(anything,theta)):
degree(IC3,theta):
#Now let's define the "Z's"
Z0 := astar+a*nstar/Cstar:
Z1 := coeff(w1_prime_at_a,theta,1):
Z2 := coeff(w1_prime_at_a,theta,0):
Z3 := coeff(IC1,theta,2):
Z4 := coeff(IC1,theta,1):
Z5 := coeff(IC1,theta,0):
Z6 := coeff(IC3,theta,2):
```

```
Z7 := coeff(IC3,theta,1):
Z8 := coeff(IC3,theta,0):
IntCon := -N0*Ca+Z_0*theta-(h/2+rho)*(Z_1*theta+Z_2)-
(1/2)*(Z_3*theta^2+Z_4*theta+Z_5+Z_6*theta^2+Z_7*theta+Z_8):
type(IntCon,polynom(anything,theta)):
degree(IntCon,theta):
Q1 := coeff(IntCon,theta,2):
Q2 := coeff(IntCon,theta,1):
Q3 := coeff(IntCon,theta,0):
#Transversality Condition
kappa1 := -A1*k1^2*cos(k1*a)-B1*k1^2*sin(k1*a): #at a
kappa3 := -A3*k3^2*cos(k3*a)-B3*k3^2*sin(k3*a): #at a
kappa1_sq := simplify(expand(kappa1^2)):
kappa3_sq := simplify(expand(kappa3^2)):
TransCon := (1/2)*(D*kappa3_sq-Ds*kappa1_sq+(N0^2)/Ce-2*N0*(1-
nstar/Cstar)*theta+etatilda*theta^2):
type(TransCon,polynom(anything,theta)):
degree(TransCon,theta):
Q4 := coeff(TransCon,theta,2):
Q5 := coeff(TransCon,theta,1):
Q6 := coeff(TransCon,theta,0):

restart:
#POTENTIAL ENERGY
#No contact zone, clamped fixed (or hinged - it's general)
w1prime := -A1*k1*sin(k1*x)+B1*k1*cos(k1*x)+C1:
w3prime := -A3*k3*sin(k3*x)+B3*k3*cos(k3*x)+C3:
kappa1 := -A1*(k1^2)*cos(k1*x)-B1*(k1^2)*sin(k1*x):
kappa3 := -A3*(k3^2)*cos(k3*x)-B3*(k3^2)*sin(k3*x):
w1prime_sq := simplify(expand(w1prime^2)):
w3prime_sq := simplify(expand(w3prime^2)):
kappa1_sq := simplify(expand(kappa1^2)):
kappa3_sq := simplify(expand(kappa3^2)):
expression1 := Ds*kappa1_sq:
IC1 := int(expression1,x=0..a):
expression2 := -2*mstar*kappa1*Thetapg:
IC2 := int(expression2,x=0..a):
expression3 := (N0^2)/Cstar:
IC3 := int(expression3,x=0..a):
expression4 := etatilda*Thetapg^2:
IC4 := int(expression4,x=0..a):
expression5 := D*kappa3_sq:
IC5 := int(expression5,x=a..1):
expression6 := (N0^2)/C:
IC6 := int(expression6,x=a..1):
PE := (1/2)*(IC1+IC2+IC3+IC4+IC5+IC6):
```

For model of partially debonded structure:

```
restart:
# CONTACT ZONE, Clamped Fixed
M_l := mstar*theta+(rho+h/2)*N0:
A1 := -M_l*Z1/(N0*Hc):
B1 := 0:
C1 := 0:
D1 := (M_l*(Hc-sin(k1*a)))/(N0*Hc):
A2 := sin(k1*a)*M_l*Z2/(N0*Hc):
B2 := sin(k1*a)*M_l*Z3/(N0*Hc):
C2 := 0:
D2 := -sin(k1*a)*M_l/(N0*Hc):
A3 := sin(k1*a)*cos(k3)*M_l/(N0*Hc):
B3 := sin(k1*a)*sin(k3)*M_l/(N0*Hc):
C3 := 0:
D3 := -sin(k1*a)*M_l/(N0*Hc):
Ap3 := 0:
Bp3 := 0:
Cp3 := sin(k1*a)*M_l*Z4/(N0*Hc):
Dp3 := sin(k1*a)*M_l*Z5/(N0*Hc):
w1_prime := -A1*k1*sin(k1*x)+B1*k1*cos(k1*x)+C1:
```

```
w2_prime := -A2*k2*sin(k2*x)+B2*k2*cos(k2*x)+C2:
w3_prime := -A3*k3*sin(k3*x)+B3*k3*cos(k3*x)+C3:
w1prime_sq := simplify(expand(w1_prime^2)):
w2prime_sq := simplify(expand(w2_prime^2)):
w3prime_sq := simplify(expand(w3_prime^2)):
type(w1prime_sq,polynom(anything,theta)):
degree(w1prime_sq,theta):
type(w2prime_sq,polynom(anything,theta)):
degree(w2prime_sq,theta):
type(w3prime_sq,polynom(anything,theta)):
degree(w3prime_sq,theta):
w1_prime_at_a := subs(x=a,w1_prime):
type(w1_prime_at_a,polynom(anything,theta)):
degree(w1_prime_at_a,theta):
Y_1 := coeff(w1_prime_at_a,theta,1):
Y_2 := coeff(w1_prime_at_a,theta,0):
IC1 := int(w1prime_sq, x = 0..a):
IC2 := int(w2prime_sq, x = a..b):
IC3 := int(w3prime_sq, x = b..1):
type(IC1,polynom(anything,theta)):
degree(IC1,theta):
type(IC2,polynom(anything,theta)):
degree(IC2,theta):
type(IC3,polynom(anything,theta)):
degree(IC3,theta):
Y_3 := coeff(IC1,theta,2):
Y_4 := coeff(IC1,theta,1):
Y_5 := coeff(IC1,theta,0):
Y_6 := coeff(IC2,theta,2):
Y_7 := coeff(IC2,theta,1):
Y_8 := coeff(IC2,theta,0):
Y_9 := coeff(IC3,theta,2):
Y_10 := coeff(IC3,theta,1):
Y_11 := coeff(IC3,theta,0):
Y_0 := astar+a*nstar/Cstar: #also
IntCon := -N0*Ca+y0*theta-(h/2+rho)*(y1*theta+y2)-
(1/2)*(y3*theta^2+y4*theta+y5+y6*theta^2+y7*theta+y8+y9*theta^2+y10*theta+y11):
type(IntCon,polynom(anything,theta)):
degree(IntCon,theta):
Q1 := coeff(IntCon,theta,2):
Q2 := coeff(IntCon,theta,1):
Q3 := coeff(IntCon,theta,0):
#Transversality Condition
kappa1 := -A1*k1^2*cos(k1*a)-B1*k1^2*sin(k1*a): #at a
kappa2 := -A2*k2^2*cos(k2*a)-B2*k2^2*sin(k2*a): #at a
kappa1_sq := simplify(expand(kappa1^2)):
kappa2_sq := simplify(expand(kappa2^2)):
TransCon := (1/2)*(Dc*kappa2_sq-Ds*kappa1_sq+(N0^2)/Ce-2*N0*(1-
nstar/Cstar)*theta+etatilda*theta^2):
type(TransCon,polynom(anything,theta)):
degree(TransCon,theta):
Q4 := coeff(TransCon,theta,2):
Q5 := coeff(TransCon,theta,1):
Q6 := coeff(TransCon,theta,0):

restart:
# CONTACT ZONE, Hinged Fixed
M_l := mstar*theta+(rho+h/2)*N0:
A1 := M_l*Z3/(N0*Hh):
B1 := 0:
C1 := 0:
D1 := M_l/N0:
A2 := sin(k1*a)*M_l*Z1/(N0*Hh):
B2 := sin(k1*a)*M_l*Z2/(N0*Hh):
C2 := 0:
D2 := 0:
A3 := -sin(k1*a)*sin(k3)*M_l/(N0*Hh):
B3 := sin(k1*a)*cos(k3)*M_l/(N0*Hh):
C3 := 0:
D3 := 0:
Ap3 := 0:
```

```
Bp3 := 0:
Cp3 := sin(k1*a)*M_l*Z4/(N0*Hh):
Dp3 := sin(k1*a)*M_l*Z5/(N0*Hh):
w1_prime := -A1*k1*sin(k1*x)+B1*k1*cos(k1*x)+C1:
w2_prime := -A2*k2*sin(k2*x)+B2*k2*cos(k2*x)+C2:
w3_prime := -A3*k3*sin(k3*x)+B3*k3*cos(k3*x)+C3:
w1prime_sq := simplify(expand(w1_prime^2)):
w2prime_sq := simplify(expand(w2_prime^2)):
w3prime_sq := simplify(expand(w3_prime^2)):
type(w1prime_sq,polynom(anything,theta)):
degree(w1prime_sq,theta):
type(w2prime_sq,polynom(anything,theta)):
degree(w2prime_sq,theta):
type(w3prime_sq,polynom(anything,theta)):
degree(w3prime_sq,theta):
w1_prime_at_a := subs(x=a,w1_prime):
type(w1_prime_at_a,polynom(anything,theta)):
degree(w1_prime_at_a,theta):
Y_1 := coeff(w1_prime_at_a,theta,1):
Y_2 := coeff(w1_prime_at_a,theta,0):
IC1 := int(w1prime_sq, x = 0..a):
IC2 := int(w2prime_sq, x = a..b):
IC3 := int(w3prime_sq, x = b..1):
type(IC1,polynom(anything,theta)):
degree(IC1,theta):
type(IC2,polynom(anything,theta)):
degree(IC2,theta):
type(IC3,polynom(anything,theta)):
degree(IC3,theta):
Y_3 := coeff(IC1,theta,2):
Y_4 := coeff(IC1,theta,1):
Y_5 := coeff(IC1,theta,0):
Y_6 := coeff(IC2,theta,2):
Y_7 := coeff(IC2,theta,1):
Y_8 := coeff(IC2,theta,0):
Y_9 := coeff(IC3,theta,2):
Y_10 := coeff(IC3,theta,1):
Y_11 := coeff(IC3,theta,0):
Y_0 := astar+a*nstar/Cstar: #also
IntCon := -N0*Ca+y0*theta-(h/2+rho)*(y1*theta+y2)-
(1/2)*(y3*theta^2+y4*theta+y5+y6*theta^2+y7*theta+y8+y9*theta^2+y10*theta+y11):
type(IntCon,polynom(anything,theta)):
degree(IntCon,theta):
Q1 := coeff(IntCon,theta,2):
Q2 := coeff(IntCon,theta,1):
Q3 := coeff(IntCon,theta,0):
#Transversality Condition
kappa1 := -A1*k1^2*cos(k1*a)-B1*k1^2*sin(k1*a): #at a
kappa2 := -A2*k2^2*cos(k2*a)-B2*k2^2*sin(k2*a): #at a
kappa1_sq := simplify(expand(kappa1^2)):
kappa2_sq := simplify(expand(kappa2^2)):
TransCon := (1/2)*(Dc*kappa2_sq-Ds*kappa1_sq+(N0^2)/Ce-2*N0*(1-
nstar/Cstar)*theta+etatilda*theta^2):
type(TransCon,polynom(anything,theta)):
degree(TransCon,theta):
Q4 := coeff(TransCon,theta,2):
Q5 := coeff(TransCon,theta,1):
Q6 := coeff(TransCon,theta,0):

restart:
#POTENTIAL ENERGY
#CONTACT ZONE, clamped fixed (or hinged - it's general)
kappa1 := -A1*(k1^2)*cos(k1*x)-B1*(k1^2)*sin(k1*x):
kappa2 := -A2*(k2^2)*cos(k2*x)-B2*(k2^2)*sin(k2*x):
kappa3 := -A3*(k3^2)*cos(k3*x)-B3*(k3^2)*sin(k3*x):
kappa1_sq := simplify(expand(kappa1^2)):
kappa2_sq := simplify(expand(kappa2^2)):
kappa3_sq := simplify(expand(kappa3^2)):
# integrals 1st
expression1 := Ds*kappa1_sq:
I1 := int(expression1,x=0..a):
```

```
expression2 := -2*mstar*kappa1*Thetapg:
I2 := int(expression2,x=0..a):
expression3 := Dc*kappa2_sq:
I3 := int(expression3,x=a..b):
expression4 := D*kappa3_sq:
I4 := int(expression4,x=b..1):
#Add the rest
T1 := ((N0^2)*a)/Cstar:
T2 := etatilda*a*Thetapg^2:
T3 := ((N0^2)*(b-a))/C:
T4 := ((N0^2)*(1-b))/C:
PE := (1/2)*(I1+I2+I3+I4+T1+T2+T3+T4):
```

Curriculum Vitae

Pamela Marie Carabetta

Academic:

Ph.D. Mechanical and Aerospace Engineering
Rutgers University: Graduate School – New Brunswick, New Brunswick, NJ
May 2011

M.S. Mechanical and Aerospace Engineering
Rutgers University: Graduate School – New Brunswick, New Brunswick, NJ
October 2007

B.S., Aerospace Certificate. Mechanical and Aerospace Engineering,
Rutgers University: School of Engineering, Piscataway, NJ
May 2004

Occupations:

Summer Instructor, Rutgers University, Piscataway, NJ
July 2010 – August 2010

Teaching Assistant, Rutgers University, Piscataway, NJ
September 2005 – May 2010

Publications:

Carabetta, P.M., Bottega, W.J. (2011) Edge debonding and its coupling with thermal buckling of a partially debonded patched structure. (In preparation)

Carabetta, P.M., Bottega, W.J. (2011) Thermal buckling phenomena of a partially debonded patched structure. (In preparation)

Bottega, W.J, Carabetta, P.M. (2009) On the detachment of patched panels under thermomechanical loading, *JoMMS*, **4** (7-8), 1227 – 1250.

Carabetta, P.M., Bottega, W.J. (2008) Effects of geometric nonlinearities on damage propagation in patched beam-plates subjected to pressure loading, *Int. J. Fracture*, **152**, 51 – 62.