Effects of Geometric Nonlinearities and Uniform Temperature Fields on the Detachment of Patched Beam-Plates under Pressure Loading

by

PAMELA M. CARABETTA

A thesis submitted to the

Graduate School – New Brunswick

Rutgers, the State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

written under the direction of

Professor William J. Bottega

and approved by

_____

_____

_____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE THESIS

Effects of Geometric Nonlinearities and Uniform Temperature Fields on the Detachment of Patched Beam-Plates under Pressure Loading

by Pamela M. Carabetta

Thesis Director:
Professor William J. Bottega

Patched structures have many engineering applications, ranging from aircraft repair to electronic devices and beyond. Delamination growth at the edge of the interface between the substructures may lead to catastrophic results. In the spirit of preventing this, edge delamination failure is studied such that behavior may be predicted as well as characterized for a patched beam-plate structure under different physical conditions. The loading scenario on such structures includes transverse pressure and an applied uniform temperature field. These conditions are intended to mimic possible conditions under which a patched beam-plate may be used. The patched beam-plate consists of two plates: the "patch," which is of shorter length, adhered to the longer "baseplate". This composite structure is examined under various support conditions on the endpoints: hinged-free, clamped-free, hinged-fixed, and clamped-fixed.

The problem is formulated analytically, from first principles, using linear and geometrically nonlinear models. Governing equations are derived using a local force balance on an element of the structure incorporating an appropriate geometrically nonlinear thin structure theory, and can be reduced accordingly to simplify to a corresponding linear model. A Griffith type criterion, is adopted to determine the onset of delamination growth. Exact analytical solutions are obtained for nonlinear as well as

linear problems and numerical simulations based on these solutions are performed using MATLAB.

Results are presented in the form of *delamination growth paths*, which may be interpreted physically to describe the onset, extent, and stability of the delamination growth resulting from each scenario. The effect of various physical parameters on delamination behavior is explored. The results show that a temperature difference has a profound effect on the delamination behavior of the patched plates when subject to transverse pressure, as it significantly alters the delamination growth thresholds. There are scenarios where delamination propagation begins in the absence of transverse pressure, due solely to the applied temperature difference. Conclusions are drawn and physical explanations are given for the behavior of each patched plate system. The nonlinear model reveals different delamination growth thresholds and stabilities than the linear model. It observed that the linear model does not adequately describe the salient behavior in comparison to the nonlinear model.

Acknowledgement

It is important to pay homage to those who have made it possible for me to continue my education. To get an advanced degree is a stepping stone to my future, and like any other learning process, is not without setbacks and hardship. There are several people I wish to thank.

First and foremost, I would like to thank my advisor, Dr. William Bottega. He contains a seemingly infinite well of knowledge when it comes to this material. But just as importantly (or perhaps more so), he has always been honest with me. He has also been very patient with me, and I truly appreciate this.

I am also very grateful to Dr. Haim Baruh, the Graduate Director of the Mechanical and Aerospace Engineering department, who has served on my thesis committee. He has offered valuable advice, and has always been helpful with other matters pertaining to my education.

In addition I would like to thank Dr. Haym Benaroya, for also serving on my thesis committee, and for inviting me to work in his Center for Structures in Extreme Environments. I have enjoyed working alongside such wonderful people. I'd also like to specifically thank Dr. Yuriy Gulak and Elan Borenstein of CSXE, for their computational help and expertise. And Trisha Mazucco, who is there, no matter what I need.

I appreciate the help I've received from the staff here at Rutgers, Aiesha Jenkins, Helene Press, and Virginia Dare. These women have been very helpful, and of course, kind.

I'd also like to thank my family. I know I have a tendency to keep my parents uninformed, but they never press me, and I appreciate that. They have been nothing but supportive my entire life. And finally, there is my sister Val. I imagine that I am not always easy to share an apartment with, but somehow she manages. Thanks, Val.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Composite structures play a vital role in many modern technologies. Such structures can be used to provide additional stiffness and strength to any system. They are already widely used in aircraft and ships as primary as well as secondary members, and for repair of components. Laminates also have widespread use as "smart" structures in structural, electronic and computer applications. While an assembly of materials possessing disparate properties provides many structural advantages, it may present undesirable effects as well. Having a material mismatch may adversely affect the structure's ability to perform, depending on usage. With such responsibility placed on composite assemblies, it is paramount to be able to predict and characterize failure. Delamination, or the separation of the layers, is a crucial aspect of any composite structure. What originally manifests as an interfacial flaw ultimately governs the useful life of such a structure. The conditions under which a structure is loaded will determine when and how much delamination will propagate. This knowledge is crucial to the effective design and operation of structures and loadings of the class considered.

This issue motivates the study of the growth of interfacial separation as it applies to a patched beam-plate structure under several relevant loading configurations. Such plate assemblies have their own importance in the larger realm of general composites. The patched beam-plate is an assemblage of two layers, a shorter plate (in the span-wise direction) adhered to another. We examine and characterize the one-dimensional edge

delamination propagation of a patched plate subjected to thermo-mechanical loading, for various end-support conditions. More specifically, we study patched structures subject to a uniform transverse pressure as well as a uniform temperature field, which mimic possible conditions based on the use of such a structure. We also explore the possibility of a *contact zone*, where the debonded segments of the plates remain in sliding contact during delamination growth. The presence of contact may affect the response of the structure to the given loading.

The approach to the problem is to model the governing equations using thin-structure theory as it applies to a local force balance on an element of the composite structure.

A Griffith-type *delamination criterion* will be used. Griffith (1920) offers a model for fracture relating to the elastic strain energy of a structure that is used today to predict the onset, extent, and stability of delamination failure. The *energy release rate* is of special importance to this problem, as it is used in our *growth law*, which is a Griffith-type delamination criterion. This law tells us when delamination growth begins, and how long it continues. Elastic strain energy is stored in a bonded structure. As it delaminates, energy is released; hence the energy release rate represents the rate at which this occurs. Bottega (1983) develops a general growth law for propagation using variational principles. Detailed descriptions and formulations of energy release rates in various settings are further discussed in Bottega (2003), Giannakopoulos (1994), and Hwu and Hu (1992), to name but a few.

By combining the derived governing equations and growth criterion with appropriate relationships established by Bottega (1997), we arrive at a self-consistent

model for the structure. We conduct the pertinent analyses using a mixed formulation, i.e. by casting the problem in terms of the transverse deflection and membrane force, which renders our equations in appropriate forms. Such forms allow us to obtain *exact* analytical solutions which apply to problems of this class. Exact analytical solutions are achieved by a non-linear as well as a linear analysis, in an attempt to uncover the full behavior of the evolving structure.

Once all governing equations and boundary/matching conditions are established, numerical simulations may be performed. We can interpret the *delamination growth paths* to get a physical sense of the debonding phenomena. These growth paths are plots corresponding to equilibrium configurations associated with the critical energy release rate. They represent a threshold, at which delamination propagation occurs. Results indicate that various parameters such as elastic modulus and bond strength have an effect of the thresholds for delamination growth. It is also seen that applying a uniform temperature field can have a profound effect on those thresholds.

In the following section we explore the related work in the field of delamination of structures.

1.2 A Survey of Related Literature

1.2.1 Delamination of General Composites

The problem of delaminating plates is of interest to many in the engineering community. It was in the late 1970's and early 1980's that many rigorous works on the delamination of composites came to fruition. Such work continues to the present day.

Some works focus exclusively on the composite structure known as the bi-material strip in the presence of a thermal load, which has been studied extensively, dating back to the mid 1920's from the work of Timoshenko. Altus and Ishai (1992) establish a delamination buckling criterion for a bi-material strip using analytical methods, and then compare the predicted results to observed results from experimentation. They observe, as expected, that it is the energy release rates that govern a structure's tendency to delaminate. According to this study, using a linear analysis to predict delamination is sufficient so long as the deflections are small. Others employing a more state-of-the-art approach, Klingbeil and Bontha (2003) use Finite Element Analysis (FEA) to determine a permissible initial flaw size for bi-material composite layers subject to a uniform edge-load that are to be debond-resistant. Geometry and material properties play a major role in delamination, and alter the maximum allowable flaw size. It is also shown that certain simplifications (i.e. taking certain parameters to be zero) allow for a conservative, but realistic solution. Our focus here is more on the delamination of plates, rather than the bi-material strip.

1.2.2 Double Cantilever Beams, Lap Joints, and Patched Plates

Much of the present work centers on the already established work of Bottega (1995), who studies a variety of patched plate assemblies subject to in-plane and three-point loading, as well as subject to applied transverse pressure loading. By formulating the problem as a moving boundary problem in the calculus of variations Bottega is able to derive all the governing equations of the system. The energy release rates are derived in functional form, and are used to establish what are called the *delamination growth*

paths for each system. These paths may be interpreted to explain the behavior of the system. It is found that the delamination behavior depends heavily on the types of load and support conditions present. The behavior also strongly depends on the material properties of the plates, as well as the initial flaw size. Bottega (1997) also begins to examine the concept of an applied uniform temperature field in addition to an applied transverse pressure. It is from this work that we obtain our delamination growth law.

Bottega and Karlsson (1999) establish the formulation of a similar problem, extended to include plates and shells, as well as explore the effects of introducing an edge-taper on such composite structures. Karlsson and Bottega (1999b) then provide the findings of such a formulation as it pertains to applied transverse pressure loading of a step-tapered structure. What is shown is that the degree of the edge-taper has an affect on the onset and properties of delamination. Karlsson and Bottega (1999a) study edge-point contact, and how it affects the delamination behavior of patched cylindrical panels. Edge-point contact may occur in the process of debonding, when the edge of the delaminated portion of the patch remains in contact with the baseplate. The panels are loaded in circumferential tension in addition to internal pressure. It is shown how the delamination is affected by load or support conditions as well. Following this work Karlsson and Bottega (2000) study the effects of a temperature difference applied to a patched plate loaded in in-plane tension. Various buckling situations are observed, and the new phenomenon, “sling-shot buckling,” is first predicted and explained. Sling-shot buckling is the phenomenon when the structure is observed to dynamically “sling” from one equilibrium configuration to an opposite deflection of another, in an unstable manner. These findings are echoed in Karlsson (1999) as well. Karlsson and Bottega (2000) then

explore the general behavior of delamination of patched plates as subjected to an in-plane edge-force, as well as a uniform temperature field. It is shown how the varying thermal load and membrane force affect the behavior of the system, and includes various buckling scenarios. The analytical methods used in the aforementioned works are of special relevance to this current work.

Also employing an analytical approach to such delamination problems are Qingchun and Xing (1993), who use a variational method to solve for the extent of delaminations in composites, to satisfy a damage tolerant design. The composites in this study are orthogonally and symmetrically stacked plates, and are loaded in in-plane tension. While the solution presented only applies to the problem described, it is postulated that the method can work for variations of the given scenario. It is also shown with a computational example that this formulation is convenient to solve numerically. Wang and Qiao (2004) use analytical methods to model the energy release rate at the delaminating surface of two perfectly bonded plates containing an initial delamination at the mating surface. They use a first-order shear deformation plate theory, and obtain the energy release rate in a functional form, by using the J-integral method. The solution accounts for transverse shear deformation in both the cracked and uncracked parts of the plates. What they find is that the energy release rate governing the delamination for a generic loading function depends on two parameters; the membrane force and transverse shear force. A numerical example considering a single leg bending test is performed to validate the results. Terms accounting for transverse shear are added to energy release rates, which do reduce to classical theory upon simplification. These terms are necessary when the specimen is of moderate thickness.

Duong and Yu (2002) use linear elasticity theory to study the problem of composite bonded repair subject to thermal loading. Using a generalized plane stress analysis, they are able to estimate the stress intensity factors paired with the thermal stresses using different analytical methods for their two-step procedure. The patch is treated as subject to an initial strain field, with the structure being subject to two different kinds of thermal loading: a thermal cycle and a uniform low operating temperature. The results are compared with a FEA simulation for an octagonal patch. Although neglecting out-of-plane bending, the model works well, according to the authors. Kim (2006) also uses a closed form solution to model delamination growth for post buckled composite flanged joints. The model is used to calculate the distribution of the energy release rate across the crack in the composite. The distribution of energy release rates is calculated as a partial derivative of the total potential energy. Using compressive loading, the results are validated with a simulation based on FEA. Moore (2005) derives a model to describe the peeling of a composite laminate using first principles. This model is used to calculate the peeling moment, which arises when the structure delaminates due to an applied uniform temperature difference. He studies a bonded structure featuring materials with mismatched thermal properties, starting with two layers, and then advancing to beyond four layers. The solution is a general one, applicable to any multi-layer beam of this class. The peeling moment expression can be split up such that the contributions of reaction moments and of thermal moments can be separated. Using FEA, the model is shown to be accurate, and it is also shown that the model is physically interpretable.

An overabundance of work in this particular subject does lie in the realm of FEA. This is a popular method (but is not without inherent limitations), and is used for a

variety of problems within the scope of delamination of composite structures. Tarn and Shek (1991), for instance, study a cracked plate repaired with a patch subject to in-plane loading. They use FEA for the patch and the boundary element method for the cracked plate. They examine the effect of adding the patch on the stress intensity factor at the crack tip on the original plate. Panigrahi and Pradhan (2007) use FEA to analyze delamination propagation in three dimensions of a bonded single lap joint, subject to an in-plane load. Others, such as Gregory and Spearing (2006) use FEA to model the contribution of inelastic effects on the delamination of a double cantilever beam (DCB) specimen. Tamzus et al. (2001) also study DCB specimens, and offer a numerical procedure to model crack propagation. The FEA considers linear and non-linear bridging laws, and compare results for propagation with an approximate energy release rate to establish the appropriate relationship to simulate actual crack growth. La Saponara et al. (2002) compare FEA model for crack propagation with experimental results for the delamination of a DCB. They explore different composite schemes, as well as notched and un-notched specimens. It is shown that the FEA results are confirmed by the experimental results. Also predicting delamination failure in a DCB is Song and Waas (1994). They examine “opening” loads applied at the endpoints of the laminated structure, and also have FEA results matching closely to observed experimental results. Müller (2007) studies the delamination of sandwich plates due to repeated forced oscillations. The proposed model and numerical simulation based on FEA are shown to compare favorably with experimental results.

1.2.3 Shells

Related work also focuses on the study of shell structures. A plate may be seen as a special case of a shell problem. Rutgerson (2001) as well as Rutgerson and Bottega (2002) extend the work Karlsson and Bottega, by examining the thermo-elastic buckling of shell segments. The shells are layered and subject to an applied transverse pressure in addition to a uniform temperature field. Performing a non-linear analysis they are also able to observe thermally induced "sling-shot" buckling, along with the other behaviors of such a structure. Rutgerson and Bottega (2004) go on to study pre-limit point buckling of cylindrical panels loaded under the same conditions. It is observed that snap-through buckling occurs well in advance of reaching a critical limit. The findings on sling-shot buckling have since been unified by Bottega (2006). It is concluded that this type of buckling occurs due to competing mechanical and thermal components of the loading. This concept is of importance to our present study.

Liu and Yu (2003) use FEA to model the delamination of various configurations of a layered shell, using an assumed displacement field. Borg et al. (2004) present a modified delamination model for FEA using shell elements, involving the adhesive forces. Tafreshi (2006) uses FEA to study the buckling effects of combining axial compression and applied transverse pressure on a composite shell assembly. Storåkers et al. (2004) use a theoretical analysis as well as FEA to examine the growth of a circular delamination imbedded in a shallow shell subject to compressive loading. They use a non-linear shell theory combined with a linear fracture mechanics theory to develop a model for delamination. The energy release rates are found by using FEA. What they found is that the energy release rate had a dependency on one geometric parameter

relating to the curvature. Tsamasphyros et al. (2003) consider a metallic bonded composite cylindrical panel, a structure commonly used in aircraft repair. Using three dimensional FEA they examine the delamination effects due to an applied pressure as well as an applied temperature difference.

1.2.4 Other Areas of Interest

Fatigue is also a subject of interest to delamination problems. Carpinteri et al. (2004) use a fracture mechanics model to fully describe the behavior of a composite beam subject to a cyclic external bending moment. It is shown how the crack experiences fatigue propagation, and how material properties such as brittleness affect the behavior. Numerical simulations are presented to confirm the effectiveness of the proposed model. Blanco et al. (2004) also study fatigue delamination growth, as it pertains to a carbon-fiber composite subject to cyclic transverse end loads. They develop a model to describe the non-constant delamination propagation. After comparing the models with experimental results, it is shown that an improved model is necessary to take into account Paris-law parameters and delamination modes.

Experimental studies have been performed in addition to the many theoretical/numerical works (see, for example, Bois et al. (2007)). Our focus is on the analytical formulation, followed by interpretations of the numerical simulations of derived governing equations. What is unique regarding this approach on delamination propagation is that the mathematical models are designed such that governing behavior of the systems is derived in physically interpretable *exact* functional form. Numerical

simulations are then based on these exact functions, such that the results offered represent an *exact* solution for the problems of this class.

1.3 Outline of the Thesis

The thesis is presented in six chapters. Chapter 2 describes the problem at the outset, and then presents a derivation of the governing equations. It begins with a description of the geometry, followed by a statement of the normalized parameters used. The governing equations of motion are then derived along with an *integrability condition*. The delamination criterion is quoted as well. Chapter 3 presents the method of solution to this problem. First the analytical solutions to the governing equations are given. Relevant system parameters are then defined, and the numerical procedure is discussed for both the linear and non-linear solutions. Chapter 4 is a summary of the results of the numerical simulations, for the linear model. The simplest case of hinged-free ends is presented first, followed by the results for the clamped-free ends. After that the solutions for hinged-fixed and clamped-fixed end conditions are given. The nonlinear results will be detailed in Chapter 5. Conclusions and discussions based on the results are presented in Chapter 6.

Chapter 2

Formulation

2.1 Geometry

The formulation presented here follows the development in Bottega (1995) and Bottega (1997), where the full formulation is achieved via a variational approach. However, in this work we use a differential element approach, a local force balance. We examine in this study a structure composed of one plate, the "patch," adhered to another, the "baseplate." We assume symmetric debonding, which means that the deflected shape is symmetric with respect to the center point of the structure. Figure 1 depicts the geometry of the composite structure, noting that the half-span has been normalized to unity.

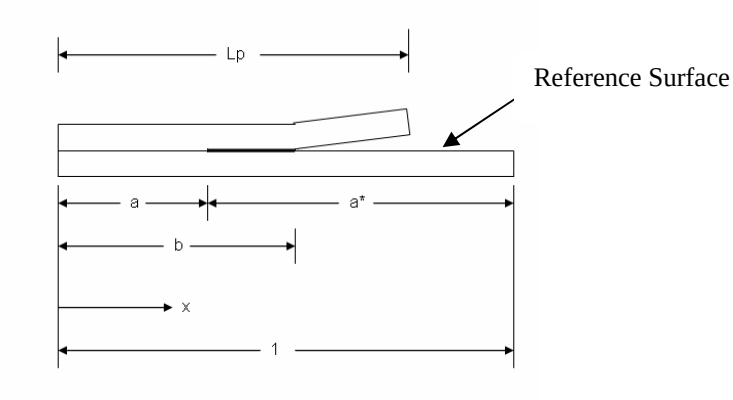


Figure 1. Half-span of the structure

To represent the structure mathematically, the following domains are designated: Region S_1: $0 \le x \le a$ represents the "bond zone," where the patch and baseplate are perfectly adhered; S_2: $a \le x \le b$ represents the "contact zone," where the plates are no longer adhered, yet remain in sliding contact (the debonded surfaces are taken as

frictionless); and S_3: $b \le x \le 1$ represents the "region of separation," where the plates are no longer touching. For completeness, the domain of the lifted portion of the patch is denoted as S_{p3}: $b \le x \le L_p$. It is seen from the figure that a is the length of the bonded region, whereas a^* is the length of the debonded region, otherwise known as the "conjugate bond length." In addition, the parameter b defines the "contact zone/region of separation" boundary, and L_p defines the length of the patch. (Throughout the formulation, all quantities containing a subscript "p" refer to the patch.) The x-coordinate is measured outward from the center of the beam. The transverse thicknesses of the plates are designated as h and h_p for the baseplate and patch, respectively. All length scales will be normalized with respect to the half-length of the baseplate. Finally, the reference surface is taken as the uppermost surface of the baseplate.

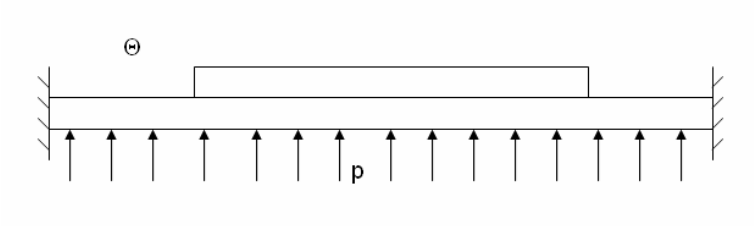


Figure 2. Structure depicted with applied transverse pressure and uniform temperature field (Shown with clamped-fixed ends)

Figure 2 shows the full-span of the structure, consisting of the baseplate with an adhered patch centered on top. The loading scenario shown consists of a transverse pressure applied along the baseplate, as well as an applied uniform temperature field (denoted by Θ). We consider end supports which are either hinged or clamped, the latter not allowing for rotation at the ends. The ends are also either free or fixed with respect to in-plane translation. Shown in Figure 2 is the clamped-fixed end support condition. In

addition to clamped-fixed, we will also examine ends which are hinged-free, clamped-free, and hinged-fixed.

2.1 Normalization of Parameters

It is convenient to normalize relevant parameters of the system, and this normalization is performed as in Bottega (1995). Parameters having units of length will be normalized with respect to the dimensional half-span length, $\overline{L}$. (All quantities designated with an over-bar represent dimensional values.) We begin with the non-dimensional half-span length L, which is defined as

$$L = \frac{\overline{L}}{\overline{L}} = 1 \tag{1a}$$

Our x-coordinate is thus

$$x = \frac{\overline{x}}{\overline{L}} \tag{1b}$$

so our displacements are

$$w = \frac{\overline{w}}{\overline{L}} \tag{1c}$$

in the transverse direction, and

$$u = \frac{\overline{u}}{\overline{L}} \tag{1d}$$

in the plane of the plate.

Further, the transverse thicknesses of the baseplate and patch are,

$$h = \frac{\overline{h}}{\overline{L}} \tag{1e}$$

$$h_p = \frac{\bar{h}_p}{\bar{L}} \tag{1f}$$

where $\bar{h}$ and $\bar{h}_p$ are the dimensional transverse thicknesses of the baseplate and patch.

We also define a thickness ratio

$$h_0 = \frac{h_p}{h} \tag{2}$$

Another important parameter is the Ratio of Elastic Moduli, which is the ratio of the elastic properties of the patch to that of the baseplate. Hence, for plane stress it is the ratio of Elastic Moduli

$$E_0 = \frac{\bar{E}_p}{\bar{E}} \tag{3a}$$

In contrast, for plane strain it is the ratio of augmented Elastic Moduli

$$E_0 = \frac{\bar{E}_p / \left(1 - \nu_p^2\right)}{\bar{E} / \left(1 - \nu^2\right)} \tag{3b}$$

where ν and ν_p are the Poisson's ratios associated with the baseplate and the patch, respectively.

The non-dimensional bending stiffnesses of the baseplate and patch will be denoted by D and D_p, respectively. Similarly, the membrane stiffnesses of the baseplate and patch will be denoted by C and C_p. The normalization of these parameters is with respect to the dimensional bending stiffness of the baseplate, and hence,

$$D = 1 \tag{4a}$$

$$C = \frac{12}{h^2} \tag{5a}$$

$$D_p = E_0 h_0^3 \tag{4b}$$

$$C_p = CE_0h_0 \tag{5b}$$

Also consistent with Bottega (1995) is our normalization of membrane force, applied transverse pressure, and bond strength between the plates. We have

$$N = \frac{\overline{N}\overline{L}^2}{\overline{D}} \tag{6a}$$

$$p = \frac{\overline{p}\overline{L}^3}{\overline{D}} \tag{6b}$$

$$\gamma = \frac{\overline{\gamma}\overline{L}^2}{\overline{D}} \tag{6c}$$

where $\overline{N}$, $\overline{p}$ and $\overline{\gamma}$ represent the dimensional values of membrane force, pressure, and bond strength, respectively. It is noted that these quantities and hence our results correspond to a state of plane stress or a state of plain strain, depending on the interpretation of the parameter E_0.

We now establish a temperature scale. We will use the scale designated by Rutgerson and Bottega (2002), such that the non-dimensional temperature change is defined by

$$\Theta = \alpha \frac{\overline{\Theta} - \overline{\Theta}_0}{\overline{\Theta}_0} \tag{7}$$

where $\overline{\Theta}_0$ represents a reference temperature (at which the system experiences no thermal deflections or strains), $\overline{\Theta}$ represents the current temperature of the system, and α represents the coefficient of thermal expansion of the baseplate. For plane stress,

$$\alpha = \overline{\alpha}\overline{\Theta}_0 \tag{8a}$$

and, for plane strain,

$$\alpha = \overline{\alpha}\overline{\Theta}_0(1+\nu) \tag{8b}$$

In Eqs. (8a) and (8b) $\overline{\alpha}$ represents the dimensional thermal expansion coefficient of the baseplate.

The normalized thermal expansion coefficient of the patch will be taken as a ratio of its dimensional value to that of the baseplate

$$\alpha_p = \frac{\overline{\alpha}_p}{\overline{\alpha}} \tag{9}$$

2.3 Development of the Governing Equations

To obtain the equations of motion, we begin by showing equivalent forces and moments for each region of the structure, as shown in Figure 3. The superscript "*" corresponds to values that describe the composite system, as opposed to those for a single constituent.

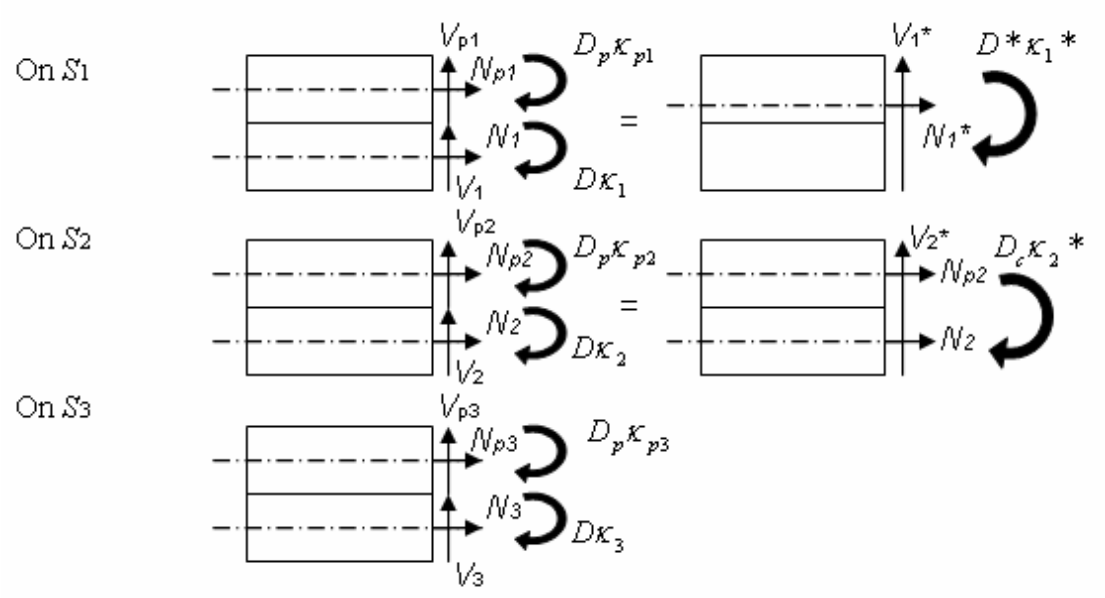


Figure 3. Equivalent forces and moments in each domain

2.3.1 Constitutive Relations

We now state a sign convention: Transverse deflection $w(x)$ is taken as positive down, and in-plane deflection $u(x)$ is taken as positive outward from the center (defined as $x = 0$), as shown in Figure 1.

The membrane strains of the baseplate and patch, $e_i(x)$ and $e_{pi}(x)$, are defined at the centerlines of their respective components. The range of the index i is 1-3; each representing the domains previously defined in Section 2.1. Deformations shall be described by the geometrically nonlinear strain-displacement relations (for small strain and moderate rotation)

$$e_i = u_i' + \frac{1}{2} w_i'^2 \tag{10a}$$

$$e_{pi} = u_{pi}' + \frac{1}{2} w_{pi}'^2 \tag{10b}$$

where the primes indicate differentiation with respect to the x-coordinate.

Adhering to the Kirchhoff assumptions for each component, the respective strains at the reference surface are described by the following relations

$$\hat{e}_i = e_i + \frac{h}{2} \kappa_i \tag{11a}$$

$$\hat{e}_{pi} = e_{pi} - \frac{h_p}{2} \kappa_{pi} \tag{11b}$$

where the carats denote strain measured at the reference surface. The terms κ_i and κ_{pi} refer to the curvatures for each region (depending on the index value) in the baseplate and the patch, respectively. We use the approximation $\kappa = w''$.

As evident from Figure 3, the resultant axial force in Region 1 is

$$N_1^* = N_1 + N_{p1}. \tag{12}$$

Keeping in mind that there is also an applied temperature field, the axial forces for the baseplate and patch are given by

$$N_1 = C\left(e_1 - \alpha\Theta\right) \tag{13a}$$

$$N_{p1} = C_p\left(e_{p1} - \alpha_p\Theta\right) \tag{13b}$$

The force N_1* acts along the centroidal axis of the composite structure. Inserting Eqs. (11a)-(11b) for the membrane strain into Eq. (12) gives

$$N_1 * = C\left(\hat{e}_1 - \frac{h}{2}\kappa_1 - \alpha\Theta\right) + C_p\left(\hat{e}_{p1} + \frac{h_p}{2}\kappa_{p1} - \alpha_p\Theta\right) \tag{14}$$

If we assume perfect bonding, and since this relation corresponds to the bond zone, the following *matching conditions* hold at the reference surface

$$\hat{u}_1 = \hat{u}_{p1} = u_1* \tag{15a}$$

$$w_1 = w_{p1} = w_1* \tag{15b}$$

and hence

$$\hat{e}_1 = \hat{e}_{p_1} = e_1* \tag{15c}$$

$$\kappa_1 = \kappa_{p1} = \kappa_1* \tag{15d}$$

Incorporating these equalities into Eq. (14), permits N_1* to be rewritten in terms of deformations at the reference surface as follows

$$N_1 * = (C + C_p)e_1 * + \left(\frac{h_p}{2}C_p - \frac{h}{2}C\right)\kappa_1 * - \left(C\alpha + C_p\,\alpha_p\right)\Theta \tag{16}$$

We shall define the terms in parentheses as

$$C* = C + C_p \tag{17}$$

$$B* = \frac{h_p}{2}C_p - \frac{h}{2}C \tag{18}$$

$$n^* = C\alpha + C_p\alpha_p \tag{19}$$

This results in the constitutive relation for N_1^*

$$N_1^* = C^* e_1^* + B^* \kappa_1^* - n^* \Theta \tag{20}$$

We follow a similar procedure to obtain expressions for the bending moment in Region 1. The resultant bending moment of the composite structure M_1^* is

$$M_1^* = M_1 + M_{p1} \tag{21}$$

where M_1 and M_{p1} are the resultant bending moments of the baseplate and the patch, respectively. They are taken about an axis through the reference surface. Thus,

$$M_1 = D\kappa_1 - \frac{h}{2} N_1 \tag{22a}$$

$$M_{p1} = D_p \kappa_{p1} + \frac{h_p}{2} N_{p1} \tag{22b}$$

Utilizing the matching conditions (Eqs. (15a)-(15d)) and the expressions for N_1 and N_{p1} (Eqs. (13a)-(13b)) in Eq. (21) we obtain

$$M_1^* = \left[D + \left(\frac{h}{2}\right)^2 C + D_p + \left(\frac{h_p}{2}\right)^2 C_p \right] \kappa_1^* + \left(\frac{h_p}{2} C_p - \frac{h}{2} C \right) e_1^* - \left(\frac{h_p}{2} C_p \alpha_p - \frac{h}{2} C\alpha \right) \Theta \tag{23}$$

Once again, by renaming the quantities in parentheses, and noting that the second term in parentheses is B^* as defined in Eq. (18), we obtain the stiffness of the composite structure in terms of the material/structural/geometric parameters of its constituents. Hence,

$$A^* = \left[D + \left(\frac{h}{2}\right)^2 C + D_p + \left(\frac{h_p}{2}\right)^2 C_p \right] \tag{24}$$

$$\mu^* = \frac{h_p}{2} C_p \alpha_p - \frac{h}{2} C\alpha \tag{25}$$

It follows that

$$M_1^* = A^* \kappa_1^* + B^* e_1^* - \mu^* \Theta \tag{26}$$

Using Eq. (20), we can eliminate the strain from Eq. (26) to give

$$M_1^* = A^* \kappa_1^* + B^* \left[\frac{N^*}{C^*} - \frac{B^*}{C^*} \kappa_1^* + \frac{n^*}{C^*} \Theta \right] - \mu^* \Theta \tag{27}$$

Defining

$$\rho^* = \frac{B^*}{C^*} \tag{28}$$

$$D^* = A^* - \rho^* B^* \tag{29}$$

$$m^* = \mu^* - \rho^* n^* \tag{30}$$

$$\beta^* = \frac{m^*}{D^*} \tag{31}$$

where the quantity ρ^* may be seen to give the location of the centroidal axis with respect to the reference surface, we can now rewrite the resultant moment in the bonded region in the alternate form

$$M_1^* = D^* \left[\kappa_1^* - \beta^* \Theta \right] + \rho^* N_1^* \tag{32}$$

Now, for the *contact zone*, the resultant membrane force is given by

$$N_2^* = N_2 + N_{p2} \tag{33}$$

No further reduction is possible since the constituents slide freely with respect to each other in this domain. The expression for the resultant bending moment in this region is

$$M_2^* = M_2 + M_{p2} \tag{34}$$

where M_2 and M_{p2} represent the resultant bending moments of the baseplate and the patch, about an axis through the reference surface, such that

$$M_2 = D\kappa_2 - \frac{h}{2} N_2 \tag{35a}$$

$$M_{p2} = D_p \kappa_{p2} + \frac{h_p}{2} N_{p2} \tag{35b}$$

Hence,

$$M_2^* = D\kappa_2 + D_p \kappa_{p2} + \frac{h_p}{2} N_{p2} - \frac{h}{2} N_2 \tag{36}$$

In the contact zone, the plates move together transversely, and thus, along the interface, we have the following matching condition for curvature

$$\kappa_2 = \kappa_{p2} = \kappa_2^* \tag{37}$$

Substituting Eq. (37) into Eq. (36) gives the resultant moment in the contact zone as

$$M_2^* = D_c \kappa_2^* + \frac{h_p}{2} N_{p2} - \frac{h}{2} N_2 \tag{38}$$

where

$$D_c = D + D_p \tag{39}$$

The last domain of consideration is the *region of separation*. It is important to note that there are no composite terms in the following expressions since the structure behaves as two separate plates in this region. We have, for the resultant bending moment of the baseplate and the patch, about an axis through the reference surface

$$M_3 = D\kappa_3 - \frac{h}{2} N_3 \tag{40a}$$

$$M_{p3} = D_p \kappa_{p3} + \frac{h_p}{2} N_{p3} \tag{40b}$$

2.3.2 Equilibrium Equations

The governing equations may be derived by consideration of the free body diagram of an element in each domain. Figure 4 shows regions S_1 and S_2, while Figure 5

shows region S_3. Not pictured is the patch of region S_3, which is analogous to the baseplate without the applied transverse pressure.

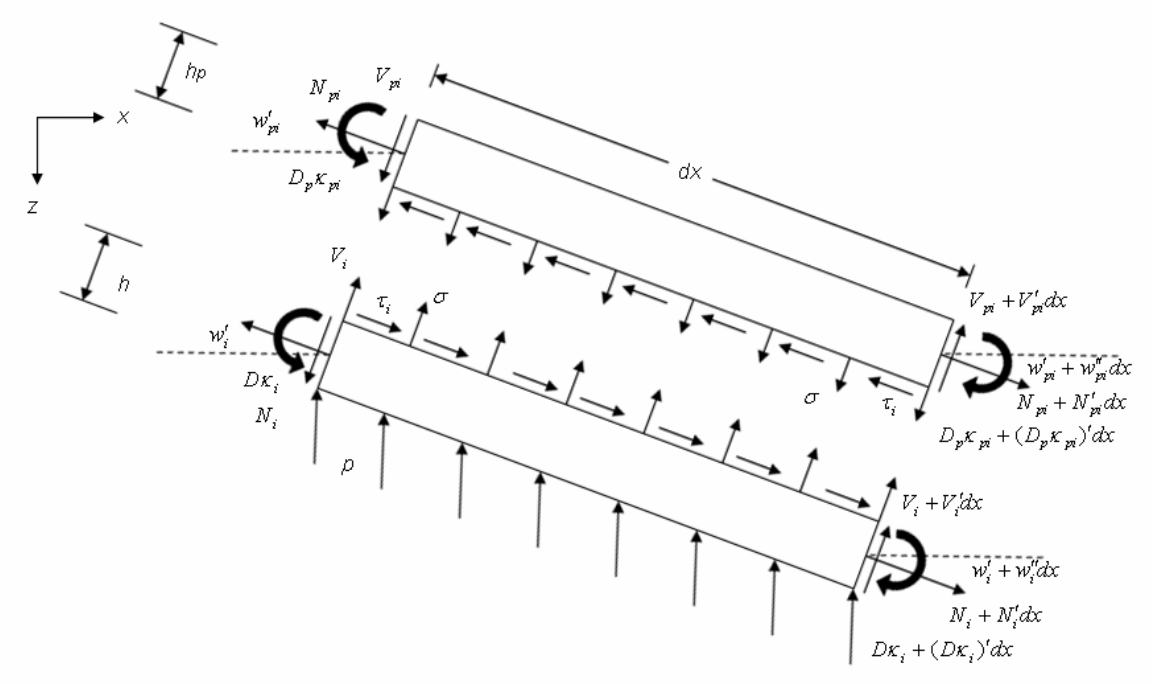


Figure 4. Free-body diagram of regions S_1 and S_2 (Index range *i*=1-2) Note that $\tau_2 = 0$

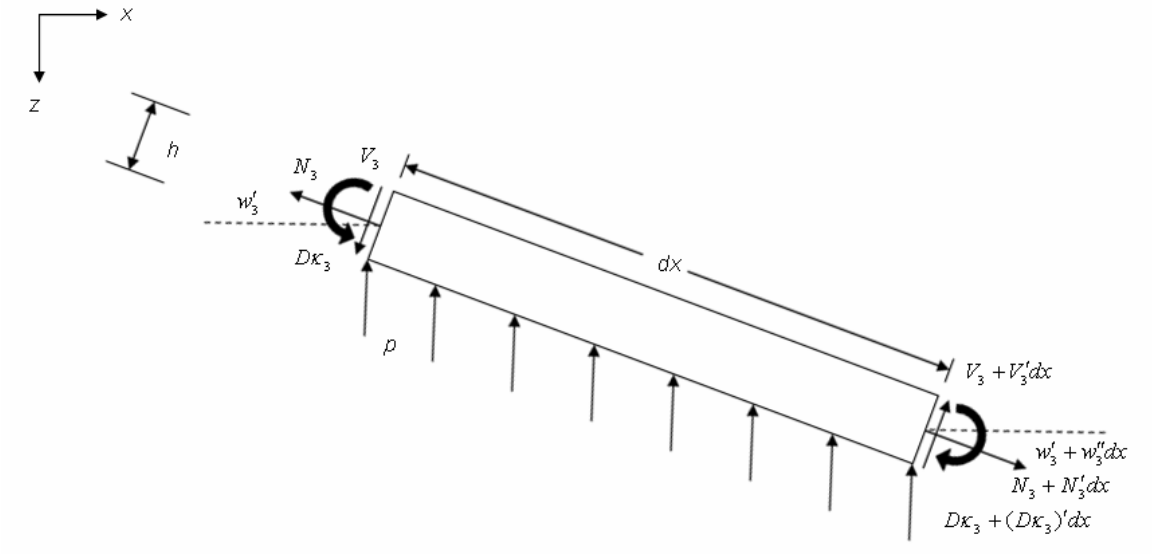


Figure 5. Free-body diagram of region S_3

Once again the primed variables refer to differentiation with respect to the *x*-coordinate. Based on the implicit assumptions made in thin structure theory, we will approximate the pressure as a distributed transverse force. Our strain-displacement relations, Eqs. (10a)-(10b), imply that the angles will be small. By applying the appropriate equilibrium equations, we arrive at the governing equations of the system. Summing forces in the *z*-direction (and combining equations for the patch and the baseplate) gives, for region S_1

$$V_1{}^{*\prime} = \left(N_1{}^* w_1{}^{*\prime}\right)' - p \tag{41}$$

where we recall that $N_1{}^* = N_1 + N_{p1}$, $w_1 = w_{p1} = w_1{}^*$, and also introduce the resultant transverse shear

$$V_1{}^* = V_1 + V_{p1} \tag{42}$$

Taking moments about the center point of the reference surface, and once again combining the equations gives for S_1

$$V_1{}^* = \left(D\kappa_1\right)' + \left(D_p\kappa_{p1}\right)' + N_{p1}{}' \frac{h_p}{2} - N_1{}' \frac{h}{2} \tag{43}$$

Recalling our definition of $M_1{}^*$ as defined in Eq. (21), we see that

$$V_1{}^* = M_1{}^{*\prime} \tag{44}$$

Substituting this equation into Eq. (41) gives the equation of equilibrium of the composite structure in region S_1 as

$$M_1{}^{*\prime\prime} - \left(N_1{}^* w_1{}^{*\prime}\right)' = -p \tag{45}$$

Summing forces and combining equations in the *x*-direction gives

$$\left(V_1{}^* w_1{}^{*\prime}\right)' + N_1{}^{*\prime} = 0 \tag{46}$$

Considering Eqs. (44) and (45), we are implicitly allowing $Nw' \sim O(V)$. It also follows from the assumption in plate theory, $\sigma_{xz} << \sigma_{zz}$, that $V<<N$. And, we have $w' << 1$ (small angle assumption). Thus, the term Vw' is so much less than N that we neglect the corresponding term in Eq. (46) to be consistent with the small strain/moderate rotation assumption. Eq. (46) thus reduces to the final form

$$N_1{}^{*\prime} = 0 \tag{47}$$

Following a similar procedure, we now derive the equations of motion for the remaining two regions. For region S_2 we have

$$M_2{}^{*\prime\prime} - \left(N_2 w_2{}^{*\prime}\right)' = -p \tag{48}$$

where M_2* is defined as in Eq. (38) and $w_2 = w_{p2} = w_2$*. Using the same argument as before, for the membrane force and transverse shear, we have the equation of motion in the plane of the plate as

$$N_2{}' = 0 \tag{49}$$

In region S_3 the bending moment is

$$M_3{}'' - \left(N_3 w_3'\right)' = -p \tag{50}$$

where M_3 is defined as in Eq. (40a). Again, for the membrane force

$$N_3{}' = 0 \tag{51}$$

Finally, the equation for the bending moment of the flap of the patch in region S_{p3}

$$M_{p3}{}'' = 0 \tag{52}$$

and in the plane of the plate

$$N_{p3}{}' = 0 \tag{53}$$

The membrane force in the patch outside of Region S_1 is taken to be zero, due to the fact that the patch is unloaded along the axial direction. Hence,

$$N_{p2} = N_{p3} = 0 \tag{54}$$

Integrating Eqs. (47), (49), and (51), and considering the matching conditions for the axial force on the domain boundaries, we have

$$N_1{}^* = N_2 = N_3 = N_0 \tag{55}$$

Thus the membrane force is uniform along the structure.

Using this result, and substituting in the appropriate expressions for the moments in terms of transverse deflections, we obtain a set of fourth-order differential equations describing the transverse motion of the structure. The governing equations are then

$$D^* w_1{}^{*iv} - N_0 w_1{}^{*\prime\prime} = -p \tag{56a}$$

$$D_c w_2{}^{*iv} - N_0 w_2{}^{*\prime\prime} = -p \tag{56b}$$

$$D w_3{}^{iv} - N_0 w_3{}^{*\prime\prime} = -p \tag{56c}$$

$$D_p w_{p3}{}^{iv} = 0 \tag{56d}$$

It is important to note that we are using a mixed formulation, solving for (w, N) pairs rather than (w, u) pairs to describe the system. This is done to render the governing equations to a form that is readily solved analytically.

Considering a symmetric deflection, the pertinent boundary and matching conditions for transverse motion take the form:

At $x = 0$

$$w_1{}^{*\prime}(0) = 0 \tag{57a}$$

$$w_1{}^{*\prime\prime\prime}(0) = 0 \tag{57b}$$

At $x = a$

$$w_1^*(a) = w_2^*(a) \tag{58a}$$

$$w_1^{*\prime}(a) = w_2^{*\prime}(a) \tag{58b}$$

$$M_1^*(a) = M_2^*(a) \tag{58c}$$

$$\left[M_1^{*\prime} - N_1^* w_1^{*\prime}\right]_{x=a} = \left[M_2^{*\prime} - N_2 w_2^{*\prime}\right]_{x=a} \tag{58d}$$

At $x = b$

$$w_2^*(b) = w_3(b) = w_{p3}(b) \tag{59a}$$

$$w_2^{*\prime}(b) = w_3^{\prime}(b) = w_{p3}^{\prime}(b) \tag{59b}$$

$$M_2^*(b) = M_3(b) + M_{p3}(b) \tag{59c}$$

$$\left[M_2^{*\prime} - N_2 w_2^{*\prime}\right]_{x=b} = \left[M_3^{\prime} - N_3 w_3^{\prime}\right]_{x=b} + \left[M_{p3}^{\prime} - N_{p3} w_{p3}^{\prime}\right]_{x=b} \tag{59d}$$

At $x = L_p$

$$w_{p3}^{\prime\prime}(L_p) = 0 \tag{60a}$$

$$\left[M_{p3}^{\prime} - N_{p3} w_{p3}^{\prime}\right]_{x=Lp} = 0 \tag{60b}$$

At $x = 1$

$$w_3(1) = 0 \tag{61a}$$

$$w_3^{\prime}(1) = 0 \text{ (clamped ends) or } w_3^{\prime\prime}(1) = 0 \text{ (hinged ends)} \tag{61b}$$

2.4 Delamination Criterion

From Bottega (1997), who approaches the problem as a moving boundary problem in the calculus of variations, we obtain the transversality condition. It represents

equilibrium configurations associated with the propagating bond zone boundary *a*. It should be noted that these equations are normalized in the same manner as our formulation. The condition for *x = a* implies our *delamination criterion*. There are two versions; one to be used when a contact zone is present, and the other for when a contact zone is not present. For a *contact zone*

$$G\{a\} = \frac{1}{2}\left[D_c \kappa^{*2}_2 - D^* \kappa^{*2}_1 + \frac{N_0^2}{C_e} + 2N_0(1-\alpha_1)\Theta + \frac{\eta\Theta^2}{2} \right]_{x=a} = 2\gamma\,,\ (b \geq a^+) \quad (62a)$$

In the *absence of a contact zone*

$$G\{a\} = \frac{1}{2}\left[D\kappa_3^2 - D^* \kappa^{*2}_1 + \frac{N_0^2}{C_e} + 2N_0(1-\alpha_1)\Theta + \frac{\eta\Theta^2}{2} \right]_{x=a} = 2\gamma\,,\ (b = a) \quad (62b)$$

where

$$C_e = \frac{C^*}{(C_p / C)} \quad (63)$$

$$\alpha_1 = \frac{n^*}{C^*} \quad (64)$$

and

$$\eta = C + \alpha_p^2 C_p - \alpha_1^2 C^* \quad (65)$$

The parameter γ appearing in Eqs. (62a)-(62b) is the bond strength for the patch-baseplate interface (a material property) while the expressions for *G* are the energy release rates. The latter are functions of the initial bond length a_0 and establish the following delamination growth criterion: *if $G\{a_0\} < 2\gamma$, where a_0 is the initial value of the bond zone boundary a, then delamination will not occur. The length of the bond zone will remain unchanged. If however, $G\{a_0\} \geq 2\gamma$, the delamination will propagate, and continue to do so until $G\{a\} = 2\gamma$.* The value of G differs, depending on if a contact zone

is present or not. The two cases are given by Eqs. (62a)-(62b). Our energy release rates are described in exact functional form, and hence apply to the entire class of problems considered.

2.5 Integrability Condition

Since we are using a mixed formulation, the governing equations describing the system do not give any information about the in-plane deflections. We must however, establish a relationship between the in-plane deflections in the respective regions and the membrane force N_0. This equation is analogous to the compatibility equation and is necessary whether the ends are free or fixed. If the ends are free and hence the membrane force is equal to zero, we have a relation for the in-plane edge deflection. If the ends are fixed and hence the edge deflection is zero, we have an equation for the membrane force corresponding to equilibrium positions of the evolving structure in terms of the loading parameters of the system. We obtain this *integrability condition* by integrating the strain-displacement relationships and imposing the appropriate boundary and matching conditions on the in-plane displacements along the beams. We begin with the expression for the membrane force in region S_1, Eq. (20). In this we substitute in the expression for geometrically nonlinear strain, Eq. (10), and also make use of the kinematic relation $\kappa = w''$. Doing this gives

$$N_1^* = C^*\left(u_1^{*\prime} + \frac{1}{2}w_1^{*\prime}\right) + B^* w_1^{*\prime\prime} - n^*\Theta \tag{66}$$

Dividing this expression by C^*, and then solving for $u_1^{*\prime}$ and integrating gives

$$u_1^*(a) - u_1^*(0) = \frac{N_1^*}{C^*}a - \rho^*\int_0^a w_1^{*\prime\prime}\,dx - \frac{1}{2}\int_0^a w_1^{*\prime 2}\,dx + \frac{n^*\Theta a}{C^*} \tag{67}$$

Similarly, in region S_2, we have

$$u_2(b) - \left[u_2{}^*(a) - \frac{h}{2} w_2{}'(a) \right] = \frac{N_2(b-a)}{C} - \frac{1}{2}\int_a^b w_2{}'^2 dx + \alpha\Theta(b-a) \qquad (68)$$

Likewise, for region S_3

$$u_3(L) - u_3(b) = \frac{N_3(L-b)}{C} - \frac{1}{2}\int_b^L w_3{}'^2 dx + \alpha\Theta(L-b) \qquad (69)$$

We now make use of the following conditions, starting with the matching conditions at the boundaries of the regions on the in-plane displacements

$$u_1{}^*(a) = u_2{}^*(a) \text{ and } u_2(b) = u_3(b) \qquad (70)$$

From Eq. (55), the membrane force is an unknown constant, N_0. Finally, since we are considering symmetric deformation, and hence the rotation and in-plane deflection of the center point of the structure are zero,

$$w_1{}^{*\prime}(0) = 0 \text{ and } u_1{}^*(0) = 0 \qquad (71)$$

Imposing these conditions and adding Eqs. (67), (68), and (69), results in the *integrability condition* given by

$$\Delta_L \equiv u_3(1) = \frac{N_0}{C_a} + (1 - a + a\alpha_1)\Theta - rw'(a) - \sum_{i=1}^{3}\int_{Si} \frac{1}{2} w_i'^2 dx \qquad (72)$$

where,

$$\frac{1}{C_a(a)} = \left(\frac{a^*}{C} + \frac{a}{C^*} \right) \qquad (73)$$

and

$$r = \frac{h}{2} + \rho^* \qquad (74)$$

Eqs. (56)-(62) and (72) comprise the complete set of equations for the class of problems considered, and are summarized below.

$$D^{*}\, {w_1}^{*iv} - N_0 {w_1}^{*\prime\prime} = -p \tag{56a}$$

$$D_c {w_2}^{*iv} - N_0 {w_2}^{*\prime\prime} = -p \tag{56b}$$

$$D {w_3}^{iv} - N_0 {w_3}^{*\prime\prime} = -p \tag{56c}$$

$$D_p {w_{p3}}^{iv} = 0 \tag{56d}$$

$${w_1}^{*\prime}(0) = 0 \tag{57a}$$

$${w_1}^{*\prime\prime\prime}(0) = 0 \tag{57b}$$

$${w_1}^{*}(a) = {w_2}^{*}(a) \tag{58a}$$

$${w_1}^{*\prime}(a) = {w_2}^{*\prime}(a) \tag{58b}$$

$${M_1}^{*}(a) = {M_2}^{*}(a) \tag{58c}$$

$$\left[{M_1}^{*\prime} - N_1 {}^{*} {w_1}^{*\prime}\right]_{x=a} = \left[{M_2}^{*\prime} - N_2 {w_2}^{*\prime}\right]_{x=a} \tag{58d}$$

$${w_2}^{*}(b) = w_3(b) = w_{p3}(b) \tag{59a}$$

$${w_2}^{*\prime}(b) = {w_3}^{\prime}(b) = {w_{p3}}^{\prime}(b) \tag{59b}$$

$${M_2}^{*}(b) = M_3(b) + M_{p3}(b) \tag{59c}$$

$$\left[{M_2}^{*\prime} - N_2 {w_2}^{*\prime}\right]_{x=b} = \left[{M_3}^{\prime} - N_3 {w_3}^{\prime}\right]_{x=b} + \left[{M_{p3}}^{\prime} - N_{p3} {w_{p3}}^{\prime}\right]_{x=b} \tag{59d}$$

$${w_{p3}}^{\prime\prime}(L_p) = 0 \tag{60a}$$

$$\left[{M_{p3}}^{\prime} - N_{p3} {w_{p3}}^{\prime}\right]_{x=Lp} = 0 \tag{60b}$$

$$w_3(1) = 0 \tag{61a}$$

$$w_3'(1) = 0 \text{ (clamped ends) or } w_3''(1) = 0 \text{ (hinged ends)} \qquad (61b)$$

$$G\{a\} = \frac{1}{2}\left[D_c \kappa_2^{*2} - D^* \kappa_1^{*2} + \frac{N_0^2}{C_e} + 2N_0(1-\alpha_1)\Theta + \frac{\eta\Theta^2}{2} \right]_{x=a} = 2\gamma \text{ , } (b \geq a^+) \quad (62a)$$

$$G\{a\} = \frac{1}{2}\left[D\kappa_3^2 - D^* \kappa_1^{*2} + \frac{N_0^2}{C_e} + 2N_0(1-\alpha_1)\Theta + \frac{\eta\Theta^2}{2} \right]_{x=a} = 2\gamma \text{ , } (b = a) \quad (62b)$$

$$\Delta_L \equiv u_3(1) = \frac{N_0}{C_a} + (1 - a + a\alpha_1)\Theta - rw'(a) - \sum_{i=1}^{3} \int_{Si} \frac{1}{2} w_i'^2 dx \qquad (72)$$

This concludes the derivation of the governing equations for this class of problems. Analytical solutions to these governing equations will be presented in Chapter 3, while the results of numerical simulations based on these equations is presented in Chapters 4 and 5.

Chapter 3

Analysis

3.1 Analytical Solutions of the Governing Equations

The first step in obtaining the response of the system is to solve the governing equations. We will perform a linear as well as a nonlinear analysis and compare the results in Chapter 4.

Since we are using the geometrically nonlinear strain-displacement relation, Eq. (10), and noting that the membrane force depends on that strain, we see that the $(Nw')'$ term is indeed non-linear. So the linear solution will neglect terms of that nature in Eqs. (56a)-(56d). It should be noted that when the ends of the beams are free to move in the *x*-direction, the membrane force vanishes, and hence the problem is linear for that case. It is when the ends are fixed with respect to in-plane translation that the nonlinear terms are pertinent.

The solutions to the linearized versions of Eqs. (56a)-(56d) are obtained by direct integration, giving

$$w_1^* = -\frac{px^4}{24D^*} + \frac{a_1x^3}{6} + \frac{a_2x^2}{2} + a_3x + a_4 \tag{75a}$$

$$w_2^* = -\frac{px^4}{24D_C} + \frac{b_1x^3}{6} + \frac{b_2x^2}{2} + b_3x + b_4 \tag{75b}$$

$$w_3 = -\frac{px^4}{24D} + \frac{c_1x^3}{6} + \frac{c_2x^2}{2} + c_3x + c_4 \tag{75c}$$

$$w_{p3} = \frac{d_1x^3}{6} + \frac{d_2x^2}{2} + d_3x + d_4 \tag{75d}$$

The constants of integration a_1-a_4, b_1-b_4, c_1-c_4, and d_1-d_4 are found by imposing the boundary and matching conditions for transverse motion given by Eqs. (57)-(61).

In contrast, the full geometrically nonlinear response is found by solving the ordinary differential equations, Eqs. (56a)-(56d), giving

$$w_1^* = a_1 \cosh(k_1 x) + a_2 \sinh(k_1 x) + a_3 x + a_4 + \frac{p x^2}{2N_0} \tag{76a}$$

$$w_2^* = b_1 \cosh(k_2 x) + b_2 \sinh(k_2 x) + b_3 x + b_4 + \frac{p x^2}{2N_0} \tag{76b}$$

$$w_3 = c_1 \cosh(k_3 x) + c_2 \sinh(k_3 x) + c_3 x + c_4 + \frac{p x^2}{2N_0} \tag{76c}$$

$$w_{p3} = \frac{d_1 x^3}{6} + \frac{d_2 x^2}{2} + d_3 x + d_4 \tag{76d}$$

where we define

$$k_1^2 = \frac{N_0}{D^*} \tag{77a}$$

$$k_2^2 = \frac{N_0}{D_c} \tag{77b}$$

and

$$k_3^2 = \frac{N_0}{D} \tag{77c}$$

Recall that the membrane force N_0 vanishes when the ends are free (and hence the problem is linear). However, for the case of fixed ends, N_0 is an unknown, and is found via the integrability condition.

3.2 The Contact Zone

The existence of a contact zone will influence the behavior of the evolving structure. In this region, the baseplate and the patch are in sliding contact, yet transverse displacements are still equal. To determine if a contact zone exists, Bottega (1995) solves for the contact stress in S_2

$$\sigma_2 = -\left(\frac{D_p}{D_c}\right)p \tag{78}$$

This expression is equal to a negative constant. Clearly, when $p = 0$, $\sigma_2 = 0$, however, the class of problem considered does indeed include an applied pressure. When this pressure term is present, a contact zone exists if the curvature in region S_2 is negative. Figure 6 shows such a deflection of the structure.

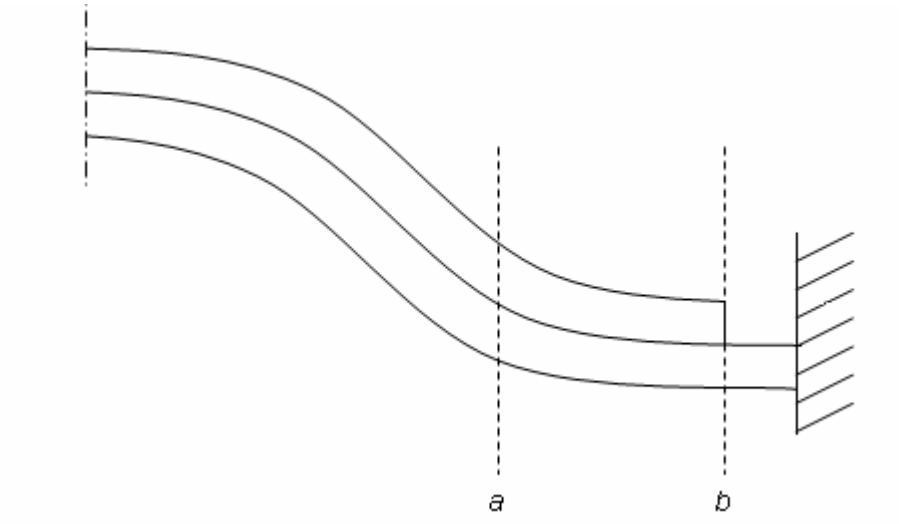


Figure 6. Half-span deflection of structure with a contact zone

It is proven in Bottega (1995) that for structures of this class, if a contact zone exists, it is a full contact zone, or at most edge-point contact. That is, independent propagation of the contact zone is not an issue in this study. It is also proved for such structures, that a

contact zone only exists for the case of clamped-fixed supports. Due to the effects of a temperature difference, however, we may expect a contact zone to be present for the case of free ends, depending on the sense of the deflection. The existence of this contact zone affects the predicted behavior of the delaminating structure.

3.3 Delamination Growth Paths

The goal of this analysis is to assess the delamination propagation for the structure under various loading scenarios. This will be done by examination of the *delamination growth paths* for the system of plates. These "paths" are curves that correspond to values of the bond zone boundary associated with equilibrium configurations of the evolving structure. From these one can predict the onset, extent and stability of delamination of the two plates. We examine the delamination growth paths with respect to the applied load, displacement, as well as the stiffness of the structure. The curves are such that we can determine, depending on the initial size of the flaw, a_0*, when debonding will begin based on the applied pressure and temperature field, and the centerspan deflection. We can also determine, based on the shape of the curves, if the ensuing delamination will be stable, unstable, or catastrophic. These results have profound implications on the design and effectiveness of such patched structures.

3.4 Linear Model

We begin the linear analysis by imposing the linear versions (i.e. those excluding the Nw' terms) of the boundary and matching conditions for the transverse deflection (as described at the end of Section 2.3) on the governing equations given by Eqs. (75a)-

(75d). This will give a system of 16x16 equations, with 16 unknowns. From here, a numerical analysis is performed using MATLAB (see Appendix). The program requires that the material parameters (E_0, α_p, and γ) be prescribed at the outset, and then cycles through all values of the bond length a within a certain step size on [0, 0.9]. In addition, to explore the effect of a temperature change on the pressure loaded plates, Θ will be prescribed. The free end programs use the transversality equation to solve for p as a function of a (and hence a^*), and the integrability condition to solve for the edge-point in-plane deflection. Should the centerspan deflection be negative, we run a "contact zone" version of the same program. The fixed-end programs use the integrability condition to express N in terms of p, and then the appropriate transversality equation (Eq. (62a) for a contact zone, Eq. (62b) for no contact zone) to solve for p. (It should be noted that the nonlinear integral term in the integrability condition (Eq. (72)) is neglected for the linear model.) For the problem of clamped-fixed ends, the program tests to see if a contact zone solution is valid for the given set of constraints, based on the arguments given in Section 3.2. From these programs we can import the output data into Microsoft Excel to obtain the desired plots of λ vs. a^*, Δ vs. a^*, and K vs. a^*. These plots show when we expect delamination growth to occur for a given loading or displacement of the structure, based on the initial size of the flaw. They also show the extent of growth, and whether the growth is stable, unstable, or catastrophic. In addition, they will show the effect of varying the prescribed parameters, as well as the influence of a temperature change on the delamination behavior of the system.

3.5 Geometrically Nonlinear Model

Since introducing nonlinearities greatly increases the complexity of the system, the approach must be modified here. First, the full nonlinear boundary and matching conditions (developed in Section 2.3), as well as the integrability condition (developed in Section 2.5), are applied to the full nonlinear solutions (Eqs. (76a)-(76d)) of the governing differential equations. This produces a system of 16x16 equations, which is solved analytically, with the aid of Maple (see Appendix). From this solution we obtain the constants of integration as functions of the given parameters as well as the pressure and temperature field. These expressions are then substituted into the appropriate transversality equation and the full nonlinear integrability condition. These two equations are combined and give a function of both N_0 and p. The computations are then performed, using MATLAB. The MATLAB program for this case is similar to its linear counterpart in that all the prescribed parameters are to be entered at the beginning. Once all the basic parameters are found, the next step is to cycle through values of the bond length a. In this loop, values of N_0 are cycled and the built-in root-solving function "fzero" is used on the function of p and N_0 to find solutions for p. Once a solution is found, it is saved along with the corresponding values of a and N_0, as well as the centerspan deflection $w_1(0)$. For the case of a contact zone, the validity of the solution is also tested again based on the arguments presented in Section 3.2. Once the data is collected, the same procedure is performed as was done for the linear case – to obtain the delamination paths as functions of the flaw size a^*. It is from this point that we can make comparisons between the results corresponding to the linear and the nonlinear solutions. Figure 7 shows a general schematic of the process for both linear and nonlinear models.

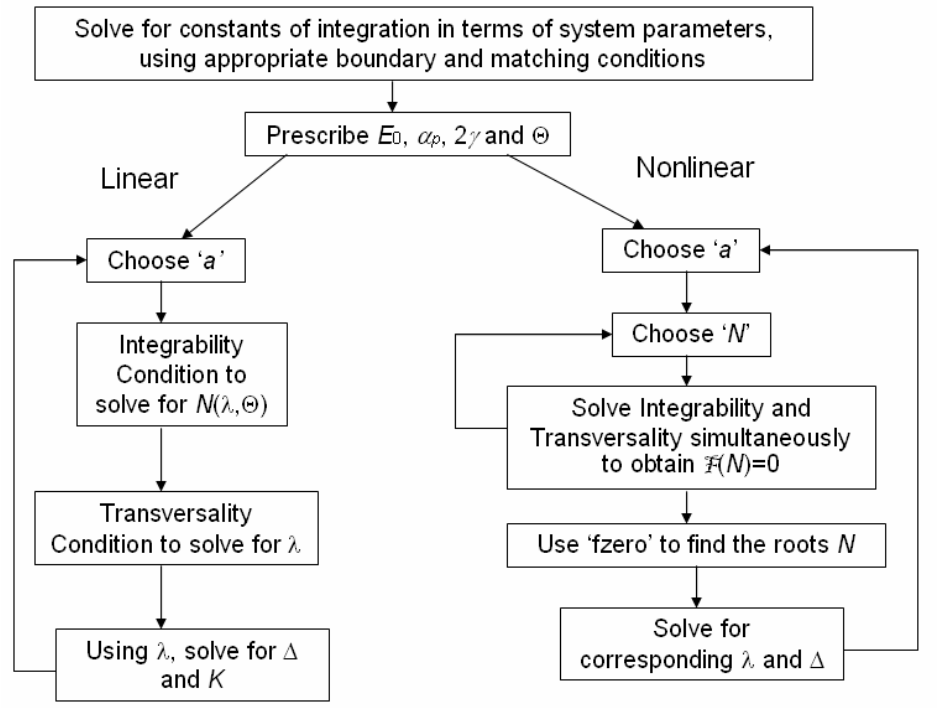


Figure 7. General schematic depicting the algorithms for solving the linear and nonlinear models

Chapters 4 and 5 present results in the form of load-deflection curves and the delamination growth paths obtained from the preceding analysis. The growth paths will be shown in terms of applied pressure and center-span deflection. The structural degradation is also characterized (for the linear model) by the corresponding curves of the effective stiffness of the evolving structure. Chapter 4 details the results obtained from the linear model, while Chapter 5 contains results from the nonlinear model.

Chapter 4

Results Based on the Linear Model

4.1 Preliminary

The purpose of this investigation is to model the delamination of patched structures under thermo-mechanical loads and assess characteristic behavior. This is facilitated by the production and interpretation of the corresponding delamination "growth paths." These paths are plotted for various configurations of the patched plate system. In this chapter we will examine the results of the linear analysis.

4.2 Interpreting the Delamination Growth Paths

Figure 8 depicts a generic delamination growth path. We can conclude a great deal about the behavior of the structure by examination of such a path.

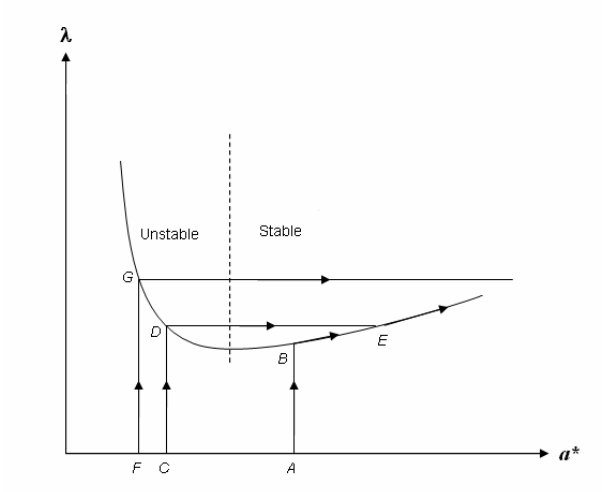


Figure 8. A generic delamination growth path

This path is plotted for the loading parameter λ (see Section 4.3) as a function of the conjugate bond zone size, a^*. Increasing conjugate bond size represents increasing

delamination size. The curve represents a threshold: the point at which the loaded patched plate assembly begins to delaminate. Each point on the path corresponds to an equilibrium configuration associated with the critical energy release rate, $G = 2\gamma$. The shape of the curve also gives information as to the stability and extent of delamination growth.

The *stable region* (as shown in the figure) corresponds to the section of the path with a positive slope. The structure begins to delaminate when the loading curve intersects the delamination growth path. Consider the loading path for a structure beginning with an initial flaw a_0* denoted by point *A* on Figure 8. It is loaded until reaching the delamination growth path at point *B.* The structure will not delaminate any further at point *B,* due to the increasing slope in the growth path. The load must be increased in order for the damage to grow. An increment in load produces a likewise increment in delamination, and hence, growth is stable.

In the *unstable region* (negative slope), no additional loading is required for damage to propagate. The onset of unstable delamination growth begins upon reaching a certain critical load, and continues at the same load level, until a stable configuration is reached. Now consider the loading path beginning at the initial flaw designated by point *C* on Figure 8. It is loaded until reaching the delamination growth path at point *D.* For that specific load designated by point *D* on the curve, the structure will exhibit unstable debonding along the path *DE*. It is at point *E* where delamination growth becomes stable, and additional loading is necessary for delamination growth to propagate any further. *Catastrophic* delamination growth occurs when there is no stable configuration possible at the critical load level, and the delamination propagates the full extent of the patch. For

example, consider loading a structure that has an initial flaw denoted by point *F*. The structure remains intact, until the loading path intersects the delamination path at point *G*, upon which delamination begins. This is catastrophic delamination growth, as the loading path cannot intersect a stable equilibrium position on the growth path again.

Delamination growth paths can also be expressed in terms of a characteristic deflection. Growth can then be predicted using similar arguments to those above, for displacement controlled loading. It is via these interpretations that we may predict the onset, extent, and stability of delamination growth in such a structure, given its initial flaw size. In addition, the deterioration of the structure can be quantified by monitoring the corresponding change in global stiffness at the critical level as the conjugate bond size increases.

4.3 System Parameters

To illustrate characteristic behavior, we will consider representative structures for which the transverse thicknesses of the baseplate and patch will be taken as $h = h_p = 0.05$. (All of the parameters are normalized as shown in Section 2.1). For the thermal parameters, we consider representative thermal expansion coefficient mismatches, such that $\alpha_p = 0.5$ or 2. The latter represents the case when the baseplate has twice the thermal expansion coefficient of the patch, and vice versa for the former. (This is to observe the effect of thermally induced stresses of the patch and baseplate in addition to the mechanical stresses due to the applied transverse pressure.) For the present analysis, the *loading parameter* λ will be the applied pressure, acting on the lower surface of the baseplate. Hence,

$$\lambda = p \tag{79}$$

Since we formulated our problem using a "positive down" sign convention, the *characteristic deflection* Δ is defined as the negative of the transverse center-span deflection, to be consistent with the direction of the applied mechanical load. That is,

$$\Delta = -w_1(0) \tag{80}$$

Using this definition of characteristic deflection, the delamination growth paths for displacement may be interpreted as such: positive values indicate deflection up, while negative values indicate deflection down. We can also define a *global stiffness* as the ratio of the loading parameter to the characteristic deflection

$$K = \frac{\lambda}{\Delta} = \frac{p}{-w_1(0)} \tag{81}$$

Should the structure experience a change in the sign of the deflection, it will be manifested as an asymptote in the stiffness plot. This will occur at the critical value of the initial flaw size such that the deflection crosses zero in changing to the opposite sense. The stiffness plot only applies to the linear model.

The ratio of elastic moduli E_0 and energy release rate 2γ will be studied for differing orders of magnitude. The bond strength γ can be interpreted as the energy required to produce one unit area of delamination. The total bond strength (or "toughness") is 2γ since there are two surfaces being created as delamination progresses. The energy release rate at the threshold of delamination propagation is equal to the total bond strength (see Section 2.4). The applied uniform temperature field Θ will be varied since we wish to explore the effect of temperature change on delamination behavior due to applied transverse pressure. Results corresponding to the linear model will be

presented in this chapter, followed by those associated with the geometrically nonlinear model in the following chapter.

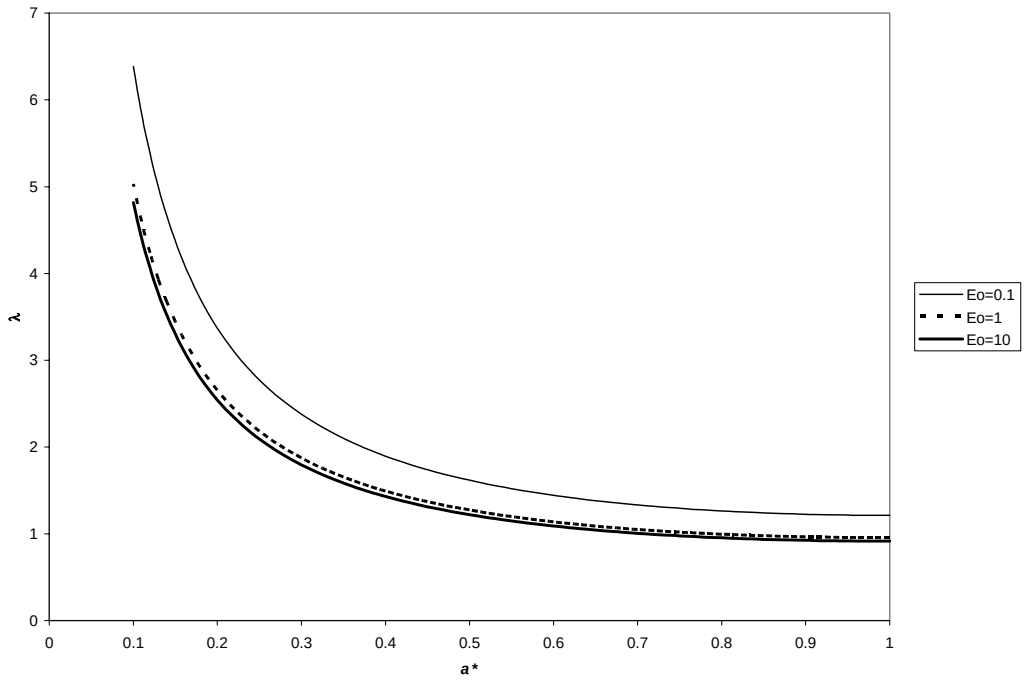
4.4 Results

4.4.1 Hinged-Free End Supports

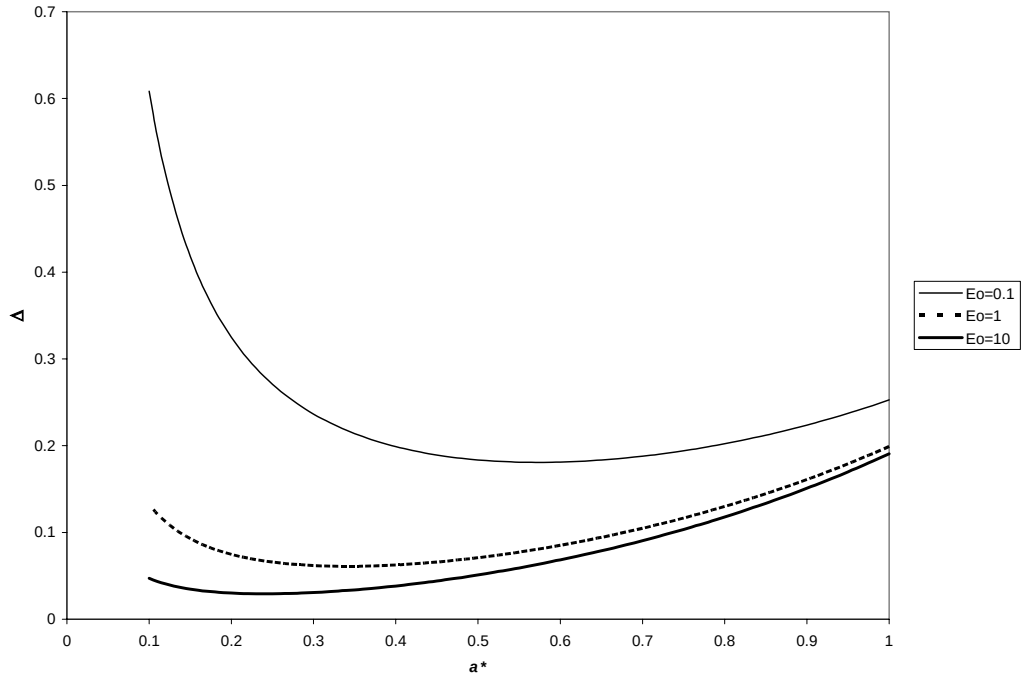
We begin by examining the simplest case: results based on the linearized equilibrium equations for the patched plate with hinged-free ends.

Pure Pressure Loading, Θ = 0

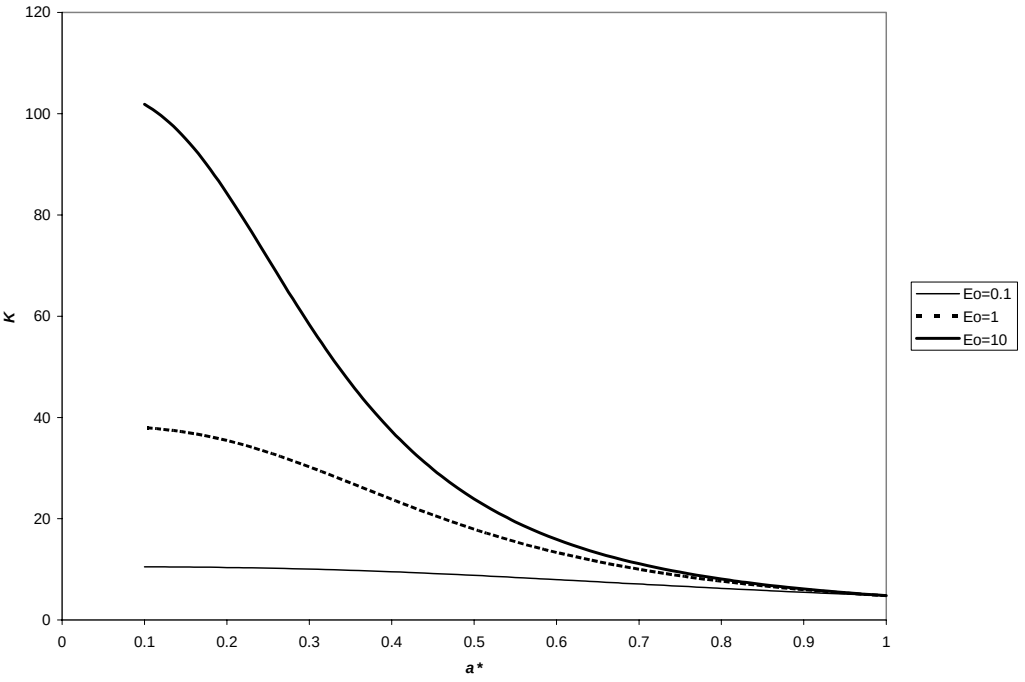
Figure 9 shows the delamination growth paths comparing the effect of different orders of magnitude on E_0, when there is no thermal load present, and $2\gamma = 0.1$.



(a)



(b)



(c)

Figure 9. Delamination growth paths and stiffness degradation for hinged-free ends without applied thermal load, 2γ = 0.1 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

We observe the effect of increasing the modulus ratio for a structure under these conditions from Figure 9. It is seen that increasing E_0 lowers the threshold for

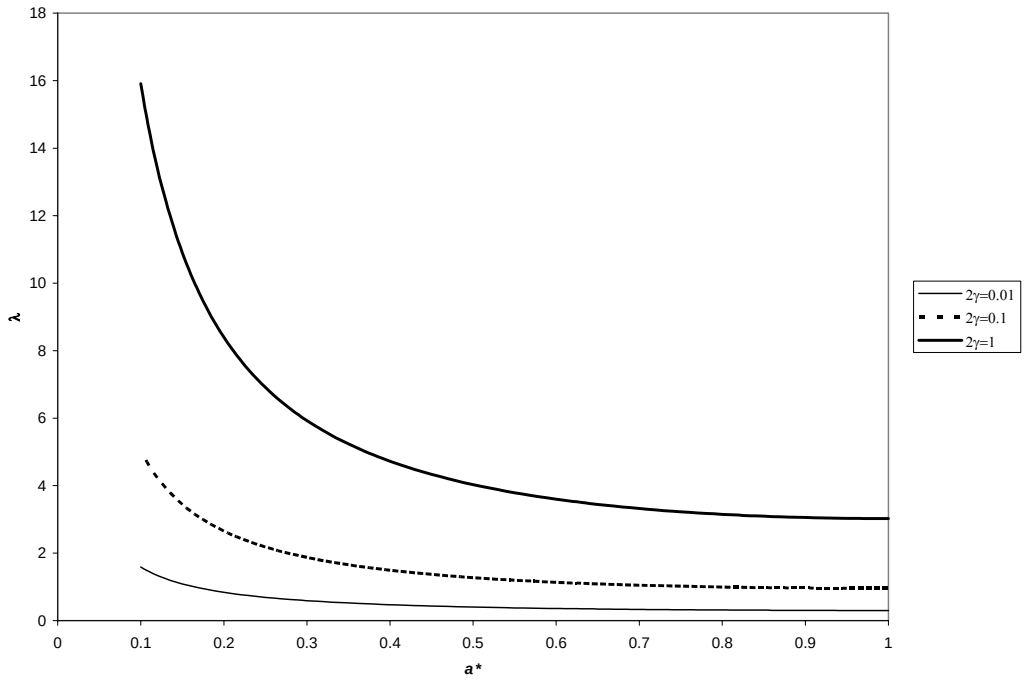
delamination growth for a load or displacement controlled test (Figures 9a and 9b). Since the modulus ratio is defined as the elastic modulus of the patch to that of the baseplate, lesser values of E_0 denote a patch that is more compliant (or flexible). Thus, the patched plate assembly can more easily bend as a unit into the deformed shape. For a stiff patch, it is more likely that it will delaminate than bend along with the baseplate. A stiffer structure has more "pent up" strain energy. Since we are examining cases where the energy release rates are equal, the stiffer structure will reach the critical value at lower load levels than the more compliant structure.

Debonding is catastrophic for the load controlled test, whereas it is unstable (for large bond zones) followed by stable (for small bond zones) for a displacement controlled test. In Figure 9b, as the modulus ratio increases, the onset of stable delamination growth shifts to the left, representing a smaller initial flaw size for which growth will still be stable. The critical initial flaw sizes (the points at which the growth paths go from unstable to stable) are a_0* = 0.57, 0.34, and 0.24 for E_0 = 0.1, 1, and 10, respectively.

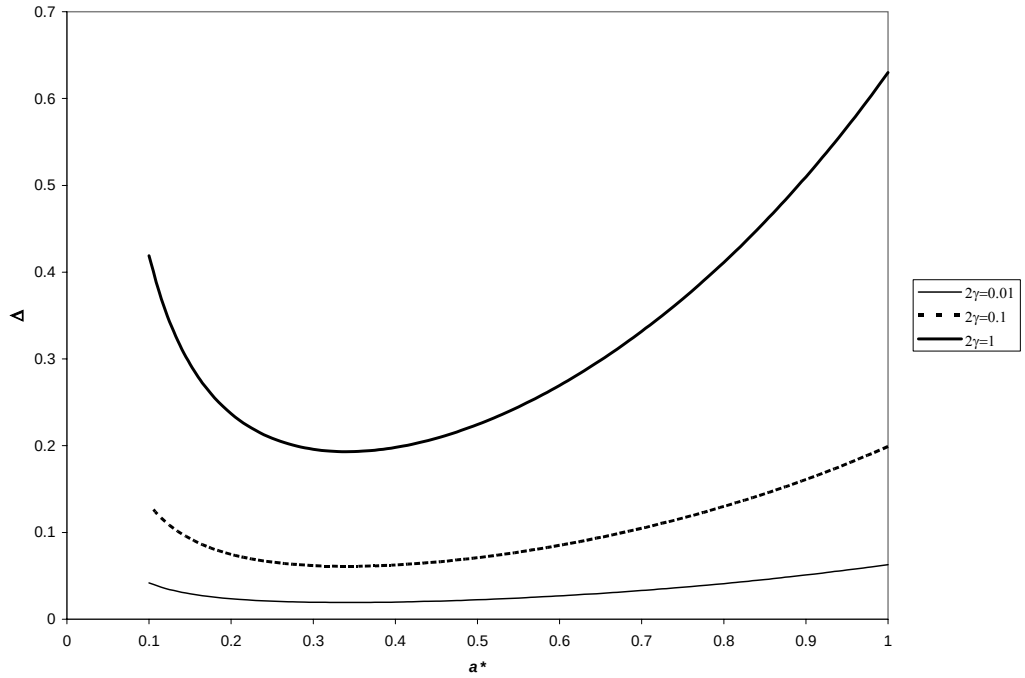
Upon consideration of Figure 9c it is seen that the largest order of magnitude on the modulus ratio (E_0 = 10) shows the most extreme deterioration in global stiffness as the patch (or bond zone) size decreases, or delamination progresses. However, when E_0 = 0.1, we see that the deterioration in stiffness is minimal along progressing delamination. For lesser values of the flaw size a*, the modulus ratio has a significant effect on the global stiffness. However, as the flaw size increases, the effect of the modulus ratio on stiffness is reduced significantly. This is because a small flaw size corresponds to a large bond size for the patch. Thus, when the patch is large (or mostly bonded), its elastic modulus becomes very significant in determining the overall stiffness of the structure.

When the patch is very small (or has almost completely debonded), its modulus is less significant. The structure is mostly just the baseplate, with only a small portion affected by whatever the modulus is of the patch.

Figure 10 shows the delamination growth paths for a structure such that $E_0 = 1$, and compares the effect of different orders of magnitude on the bond strength 2γ, in the absence of a thermal load.



(a)



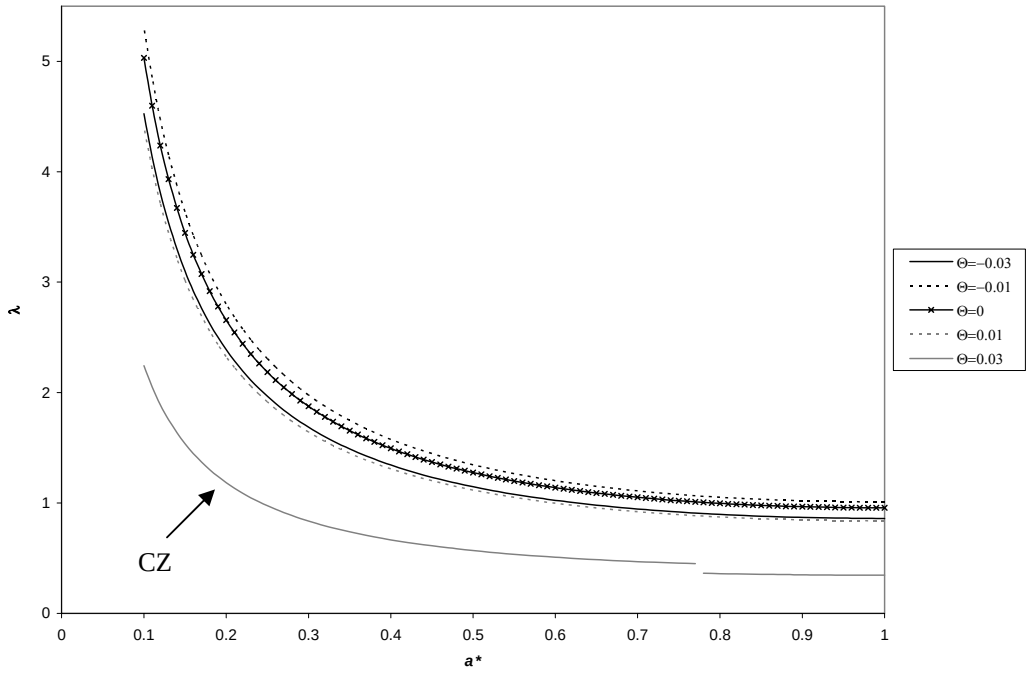
(b)

Figure 10. Delamination growth paths for hinged-free ends without applied thermal load, E_0 = 1 a) λ vs. a*, b) Δ vs. a*

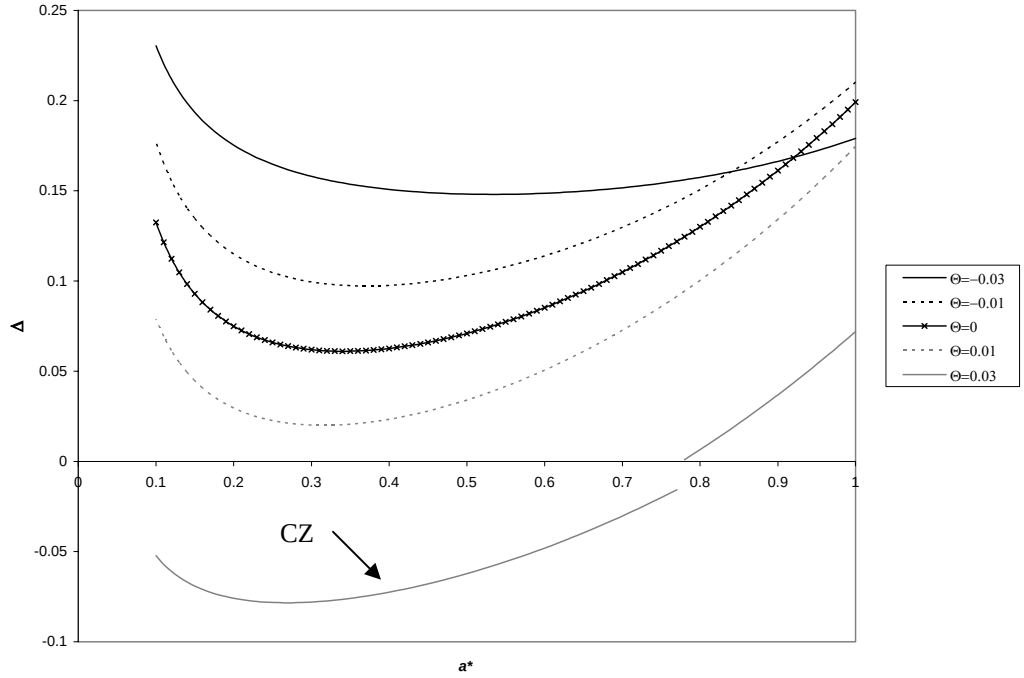
As one might anticipate, increasing the bond strength greatly increases the threshold for delamination growth for a load or displacement controlled test. A greater strength of bond would obviously require a greater load or displacement to separate the layers. We observe unstable debonding for all cases of load controlled specimens. Once again for displacement controlled specimens, we see unstable followed by purely stable delamination growth, centered around a_0* = 0.34. The onset of stable growth occurs at the same point for all three curves, since only the order of magnitude of each parameter is affected by the change in bond strength. The bond strength has no effect on the global stiffness of the structure.

Effect of Uniform Temperature Field

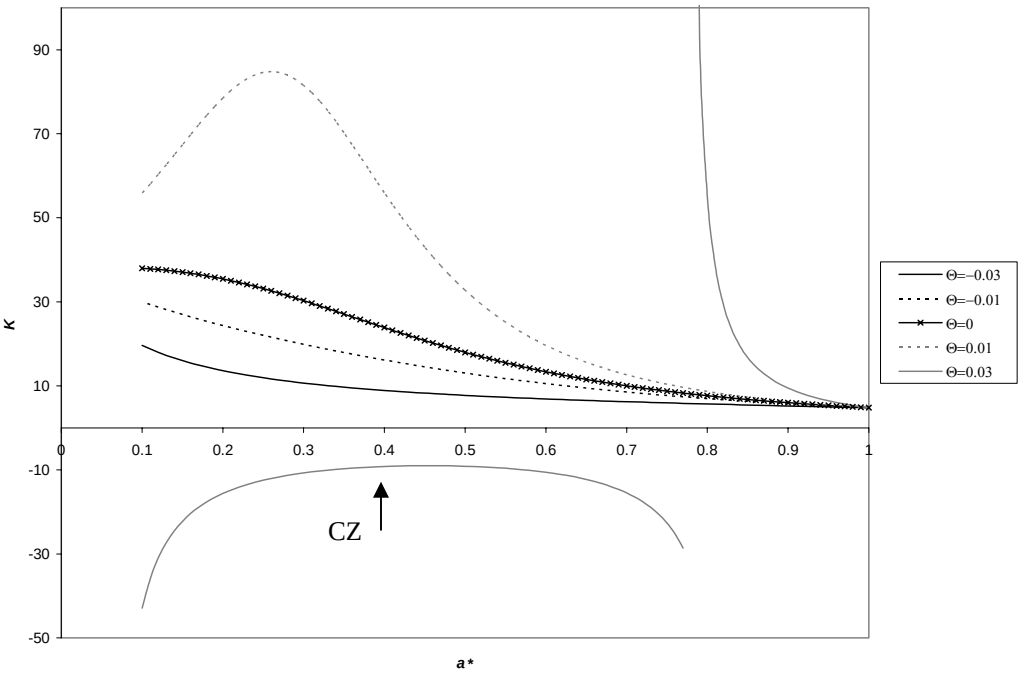
The next phase is to study the effect of a uniform temperature field applied in addition to the already present transverse pressure. Recall the temperature scale is that used by Rutgerson and Bottega (2002) and is detailed in Section 2.1. Figure 11 compares delamination growth paths for various values of the applied temperature field, both positive and negative. In this figure, α_p = 0.5; also E_0 = 1 (to isolate the temperature effects) and 2γ = 0.1.



(a)



(b)



(c)

Figure 11. Delamination growth paths and stiffness degradation for hinged-free ends with applied thermal load, E_0 = 1, 2γ = 0.1, α_p = 0.5 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

When α_p = 0.5, this corresponds to the baseplate having twice the coefficient of thermal expansion than that of the patch. Since we are considering a hinged-free specimen, there

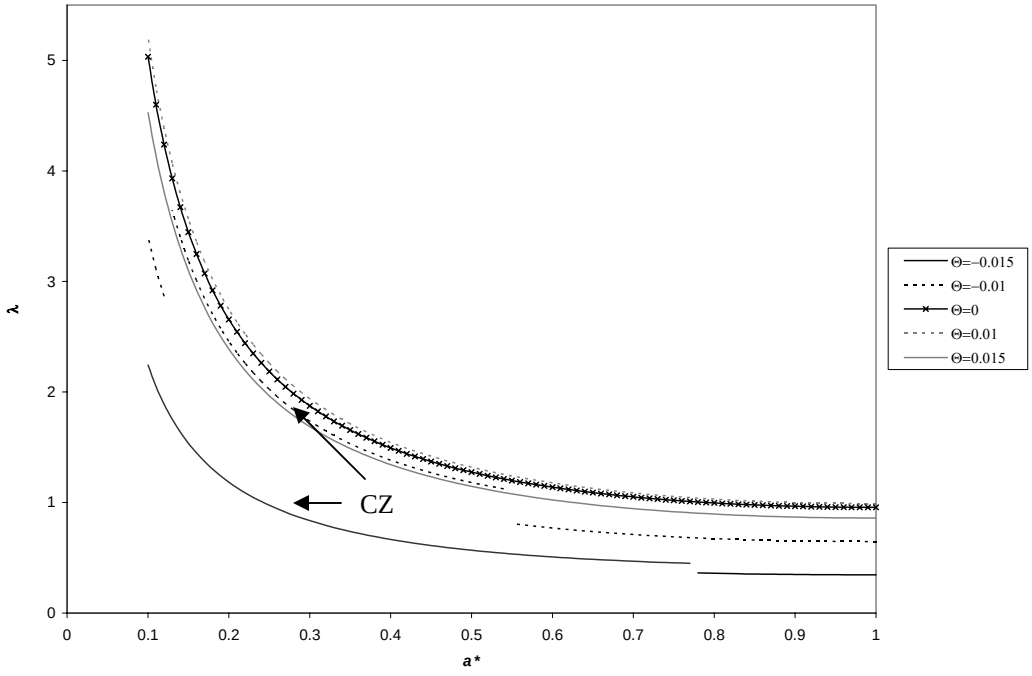
is a possibility of a contact zone. We observe in Figure 11b that for greater values of a positive temperature field, the deflection occurs in the opposite sense, down. Deflection in this sense suggests the presence of a contact zone, where the patch and baseplate remain in sliding contact. If the structure deflects up, we would expect no contact zone, since there will be zero curvature outside of the bond zone, due to the hinged ends. In this case, a hinged-free specimen will deflect down depending on the magnitude and sign of the temperature field. When α_p = 0.5, a positive temperature difference causes the baseplate to expand more than the patch, and since the ends are free to translate in the in-plane direction, the structure will deflect down. This opposes the direction of the applied transverse pressure. For this particular case, we see this behavior for Θ = 0.03 and the portion on the curve where a contact zone is present is denoted by "CZ" in the figure. The gap in the path indicates where the contact zone is no longer present.

At first glance it appears that heating the structure lowers the thresholds for delamination, while cooling achieves the opposite effect. However, this trend is not entirely observed in Figure 11a, where the curve representing Θ = -0.03 is actually at lower values than the case of no thermal load. The effect of a temperature difference is to induce thermal stress, which results in bending-stretching coupling in the bond zone. The structure bends more readily due to the thermal moment than the mechanical moment. This can have a profound impact on the delamination behavior, as indicated by the plots. The aforementioned "trend" is also observed in Figure 11b, until approximately a^* = 0.78.

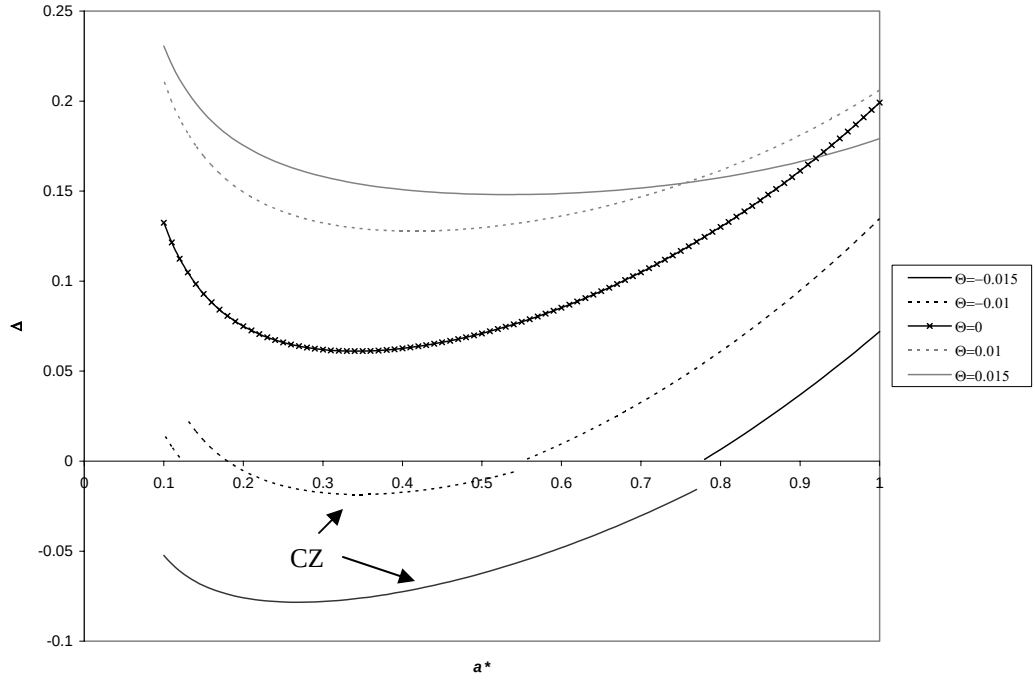
The opposite deflection is evident in the stiffness plot, Figure 11c, as the curve for Θ = 0.03 shows an asymptote where the sign of the deflection changes. Beyond this

asymptote, we see that structure is stiffer with a more positive temperature field, since the deflections are positive.

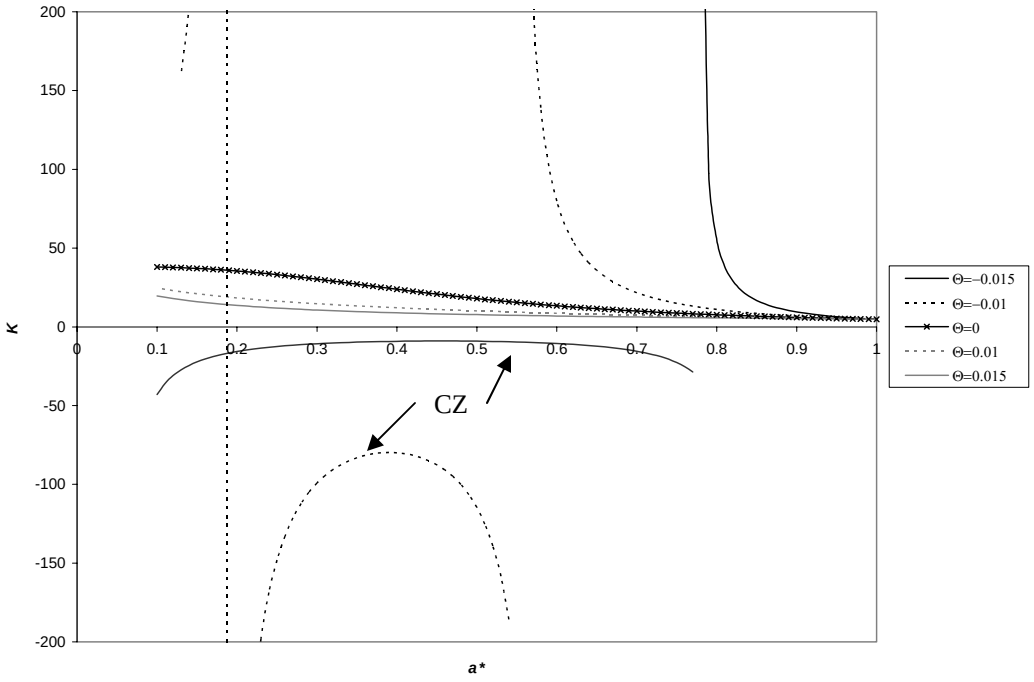
Figure 12 shows the delamination growth paths for a system under the same conditions as Figure 11, however the ratio of coefficients of thermal expansion is taken to be $\alpha_p = 2$. This means that the patch has a thermal expansion coefficient twice that of the baseplate. Thus, the patch is more responsive to the presence of the temperature difference. The growth paths are shown for varying thermal loads.



(a)



(b)



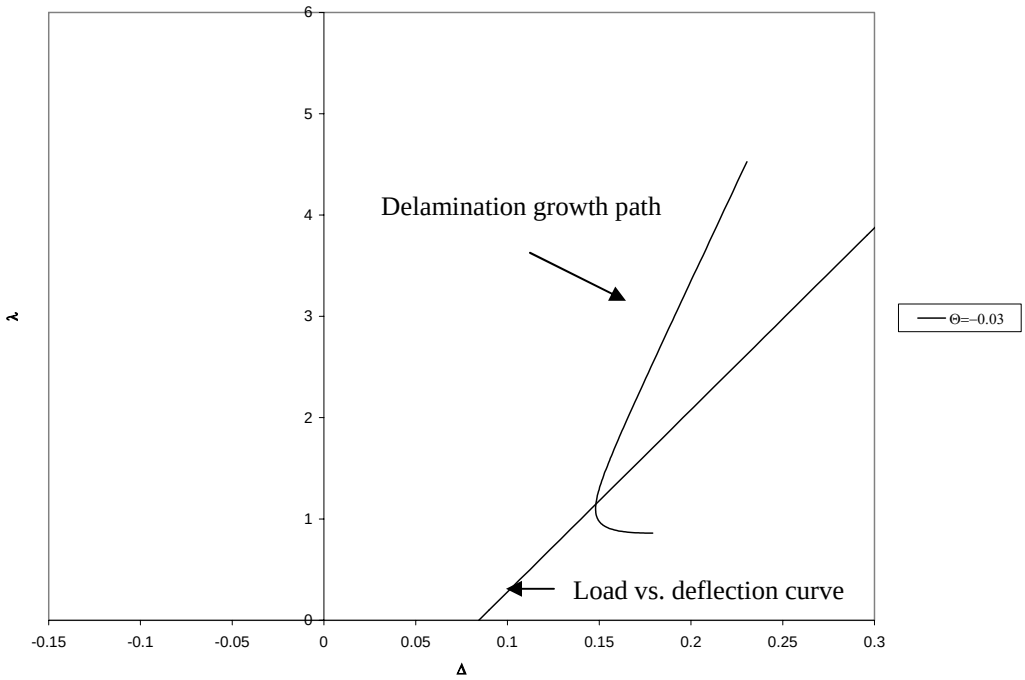
(c)

Figure 12. Delamination growth paths and stiffness degradation for hinged-free ends with applied thermal load, $E_0 = 1$, $2\gamma = 0.1$, $\alpha_p = 2$ a) λ vs. a*, b) Δ vs. a*, c) K vs. a*

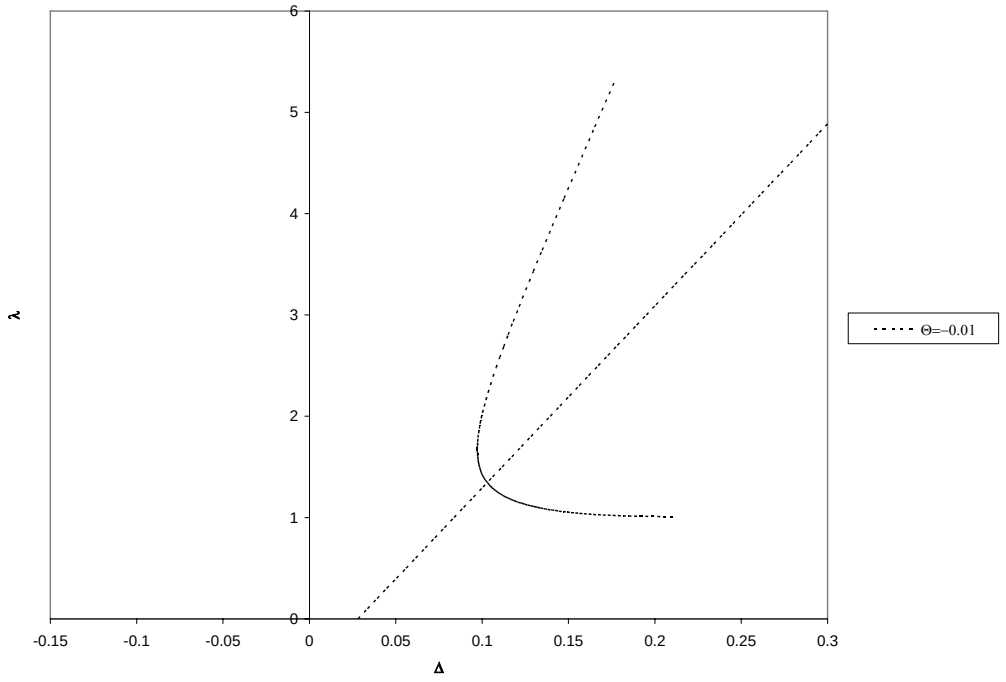
For the most part, we observe the opposite trends to those established for Figure 11a in Figure 12a. Figure 12b shows both delamination paths for a negative temperature field

exhibit deflection in the opposite sense, again suggesting that there is a contact zone present. Negative temperature fields may cause the structure to deflect downward; the reverse effect of when the baseplate has the greater coefficient of thermal expansion. The existence of a contact zone is once again denoted by the curves being labeled with a "CZ". Unlabeled curves indicate that there is no contact zone present. The portions of the growth paths for which contact occurs are separated from the other case by the gaps in the plots. Figure 12c shows how the stiffness changes due to these thermal effects, where the asymptotes are once again indicative of a change in the sense of deflection of the structure.

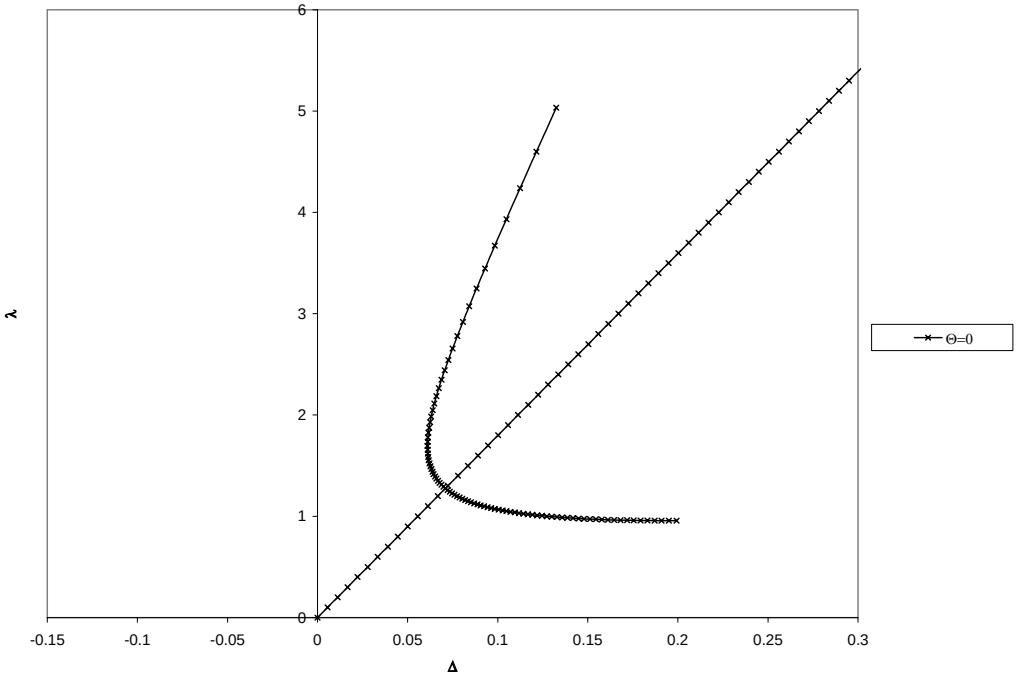
To establish a sense of the delamination growth path as it pertains to the regular quasi-static loading curve of such a structure, we superimpose the former onto the latter. Figure 13 depicts the load-deflection curve for the patched plate structure when $E_0 = 1$, $2\gamma = 0.1$, $\alpha_p = 0.5$.



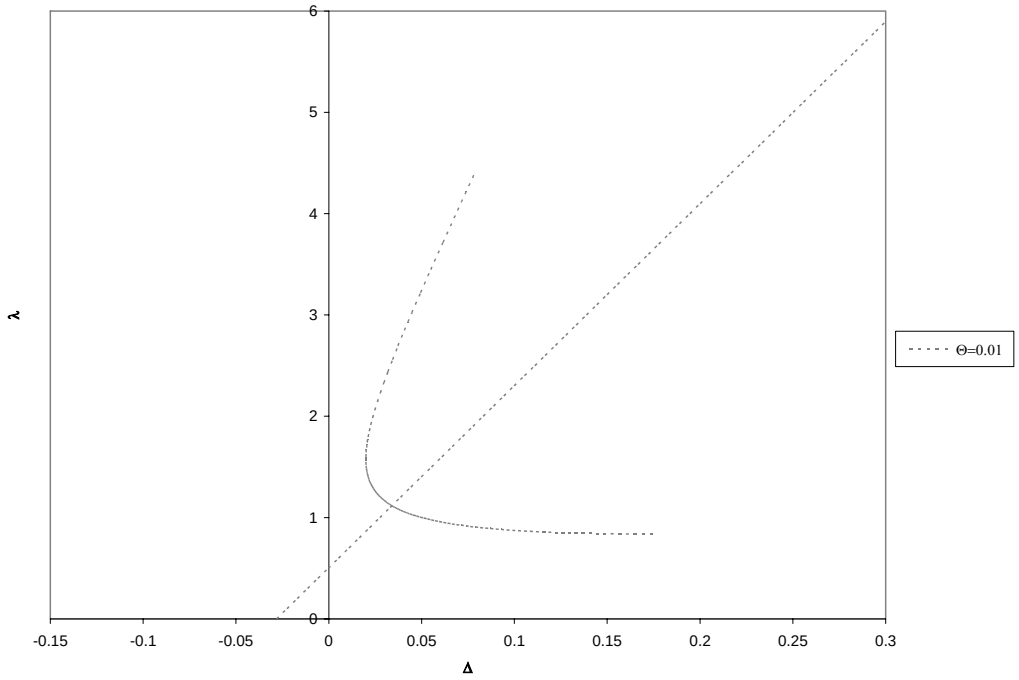
(a)


λ
Δ
Θ=–0.01
0
1
2
3
4
5
6
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3

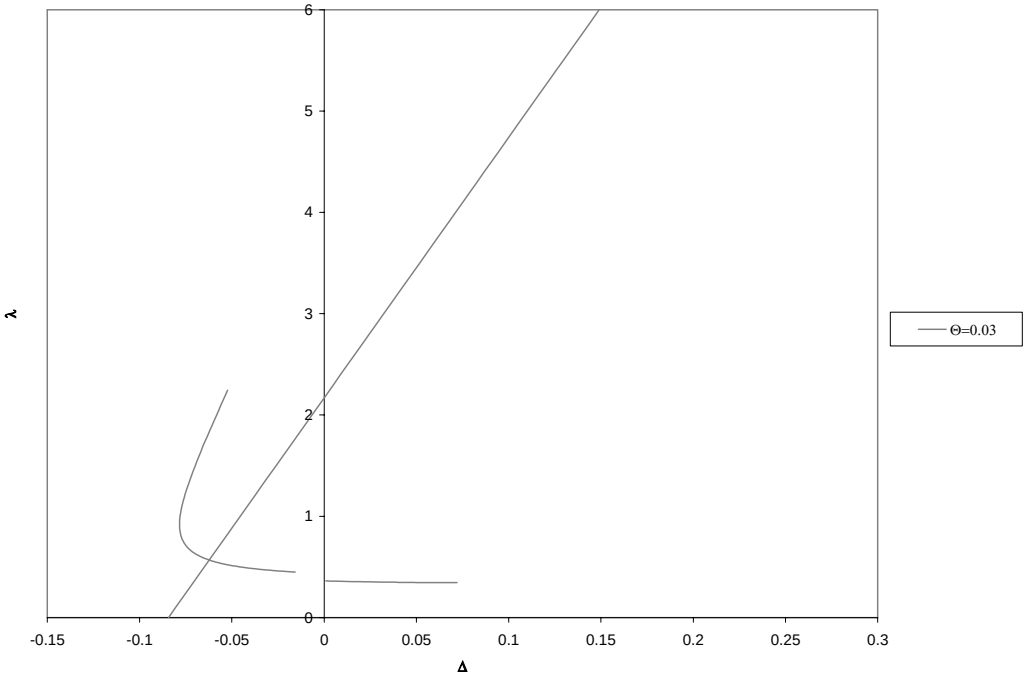
(b)


λ
Δ
Θ=0
0
1
2
3
4
5
6
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3

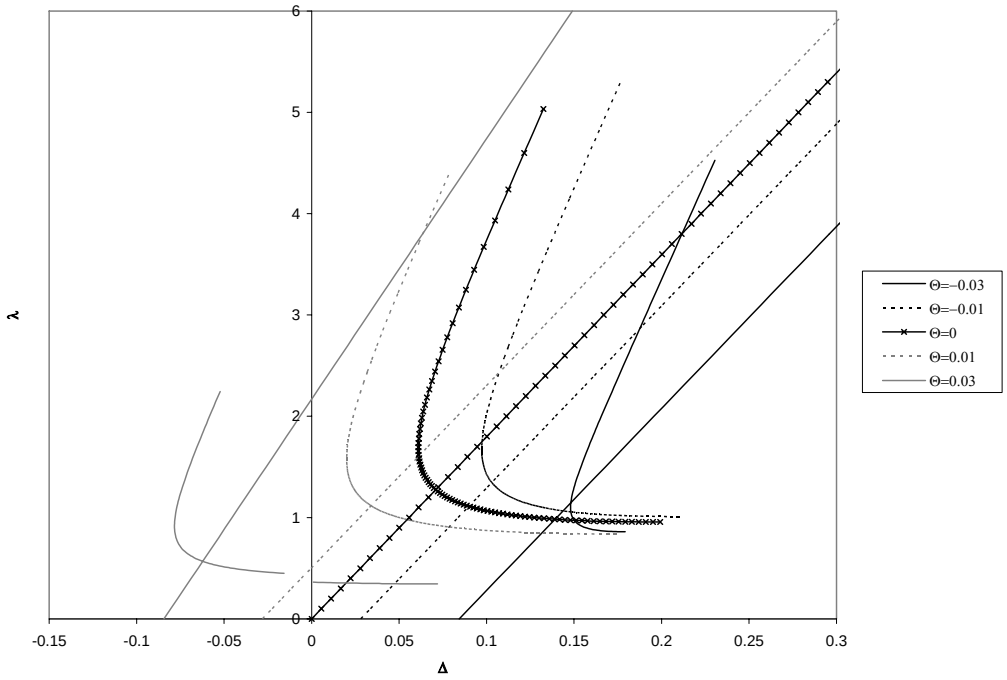
(c)


λ
Δ
Θ=0.01
6
5
4
3
2
1
0
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3

(d)


λ
Δ
Θ=0.03
6
5
4
3
2
1
0
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3

(e)



(f)

Figure 13. Load vs. deflection curves and corresponding delamination growth paths for hinged-free ends, E_0 = 1, 2γ = 0.1, α_p = 0.5, a) Θ =-0.03, b) Θ =-0.01, c) Θ = 0, d) Θ =0.01, e) Θ =0.03, f) summary

The load vs. deflection line is plotted for an initial conjugate bond zone (flaw) size of a_0* = 0.5. The loading path intersects the delamination growth path at the corresponding value of the loading parameter λ previously established for that initial value a_0* = 0.5. The point where the two paths intersect represents the onset of delamination for the structure. The value of the flaw size decreases along the delamination path as it is traversed in a counterclockwise fashion, from left to right. The offsets from zero deflection of the loading paths observed at λ = 0 are due to the presence of the temperature fields. As shown in Figure 13f, in the absence of pressure, the negative thermal load causes the hinged-free structure to deflect up, and vice versa.

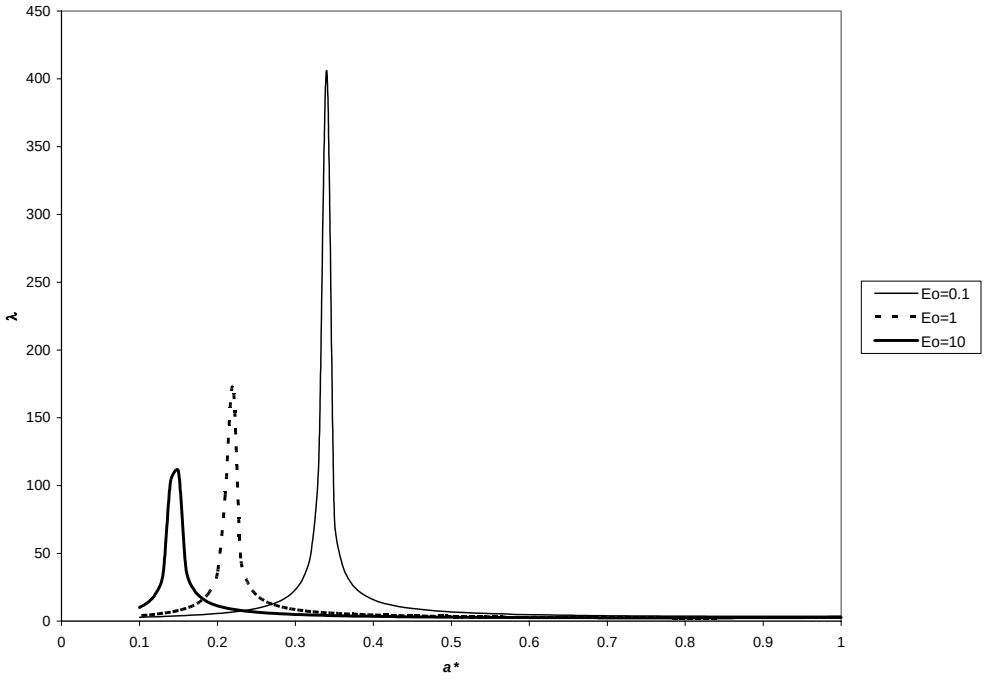
In the following section we will explore delamination behavior for the case of *clamped-free* end supports.

4.4.2 Clamped-Free End Supports

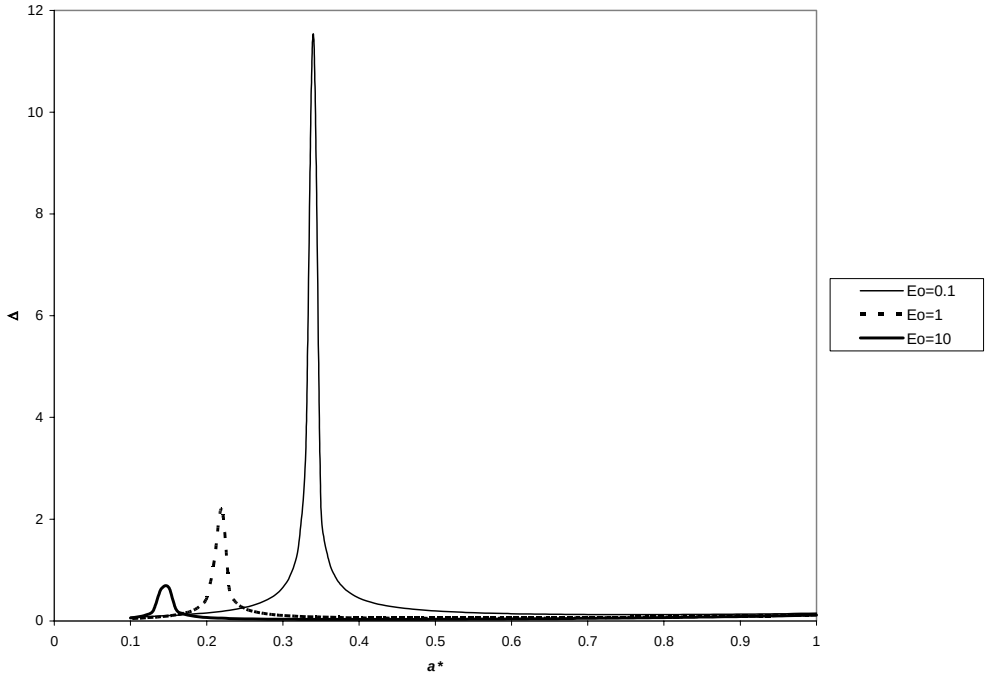
We now add the restriction that the structure may not rotate about its end supports; however, the ends may still translate freely in the in-plane direction. Since the ends are free, we expect no contact zone, unless the deflection should occur in the negative direction.

Pure Pressure Loading, Θ = 0

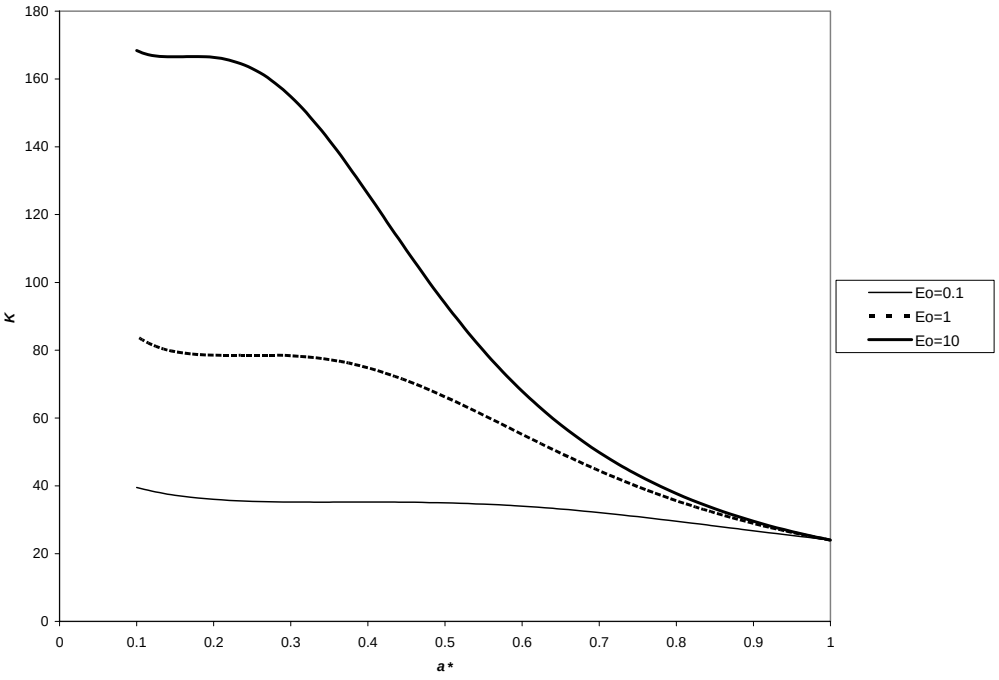
Figure 14 shows the delamination paths for this case comparing the effect of different orders of magnitude of E_0, when there is no thermal load present, and $2\gamma = 0.1$.



(a)



(b)



(c)

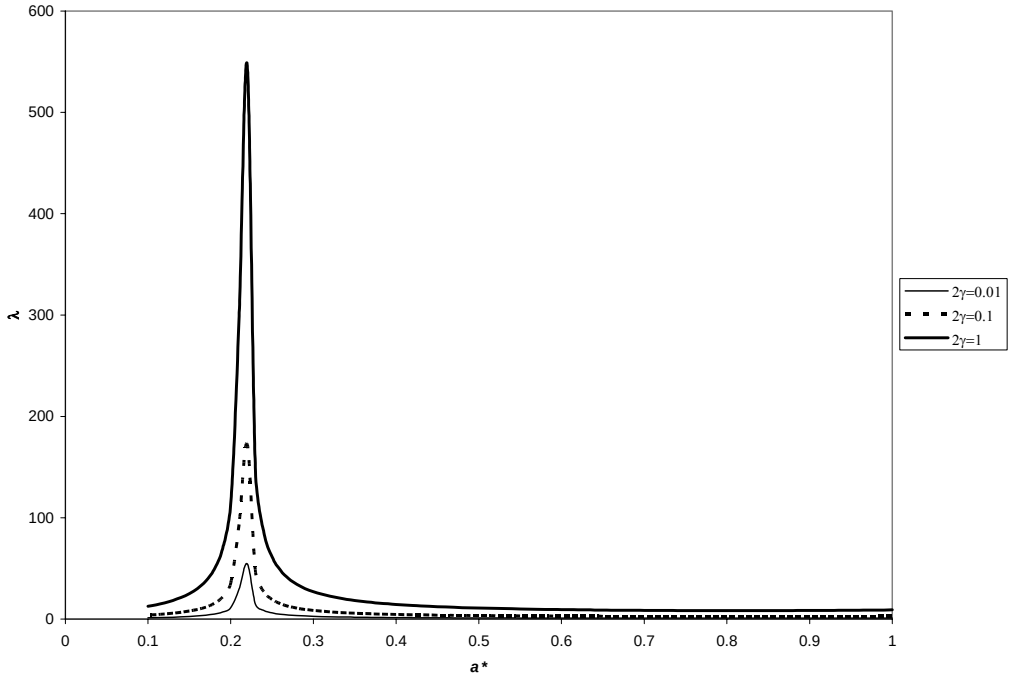
Figure 14. Delamination growth paths and stiffness degradation for clamped-free ends without applied thermal load, 2γ = 0.1 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

Figure 14 suggests that, like the hinged-free case, increasing the modulus ratio decreases the threshold for the onset of delamination. However, in this case we observe peaks

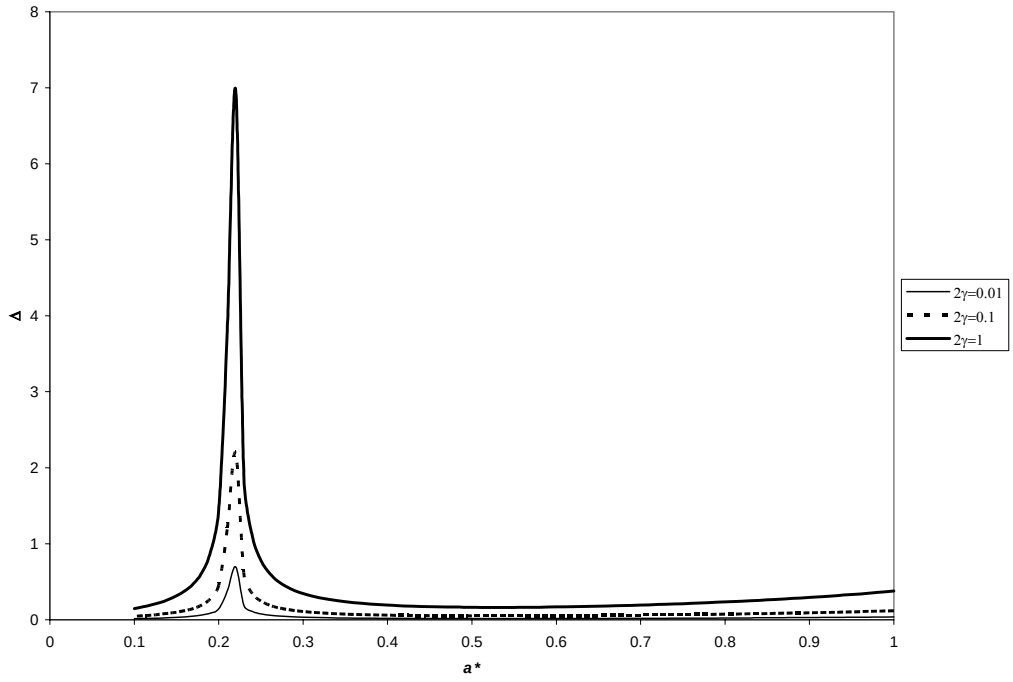
separating regions of stable and unstable delamination growth on Figures 14a and 14b. What is interesting regarding the peaks is that we may now observe the phenomena of stable debonding followed by catastrophic debonding. Take for example, a clamped-free patched plate with an initial flaw size located somewhere to the left of the peak. Upon loading up to the delamination growth path, we will first achieve stable delamination growth. As the loading increases and the structure evolves, we will eventually reach the apex of the peak, upon which the structure suddenly delaminates catastrophically.

Upon loading up to the delamination curve from a position after the peak, the structure will delaminate catastrophically, unless it intersects the growth path again at a greater value of a^*. If that occurs, then stable debonding resumes (this occurs in Figure 14b). These are the scenarios depicted by Figure 8 of Section 4.2. The effect of order of magnitude on the modulus ratio is to shift the peaks to the left and lower the amplitude as the modulus ratio increases. Thus, depending on the initial flaw size, catastrophic delamination growth begins much sooner for a very stiff structure. This reflects that delamination growth onset occurs at lower values of the loading parameter. This is because of the increased strain energy stored in the stiffer structure. The global stiffness degradation curve (Figure 14c) is similar to its hinged-free counterpart, Figure 9c.

Figure 15 shows the effect of the order of magnitude of the bond strength, 2γ, on the delamination behavior of the structure, when E_0=1.



(a)



(b)

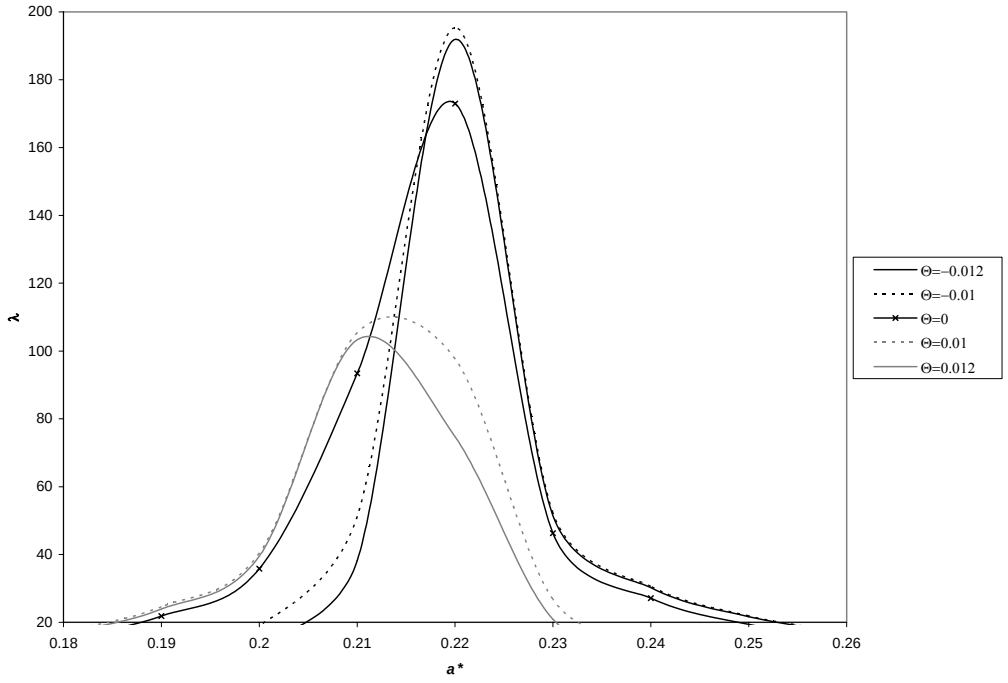
Figure 15. Delamination growth paths for clamped-free ends without applied thermal load, E_0 = 1 a) λ vs. a*, b) Δ vs. a*

The bond strength does not alter the position of the peaks on Figures 15a and 15b; rather, it alters the magnitude. The threshold of delamination increases as the bond strength

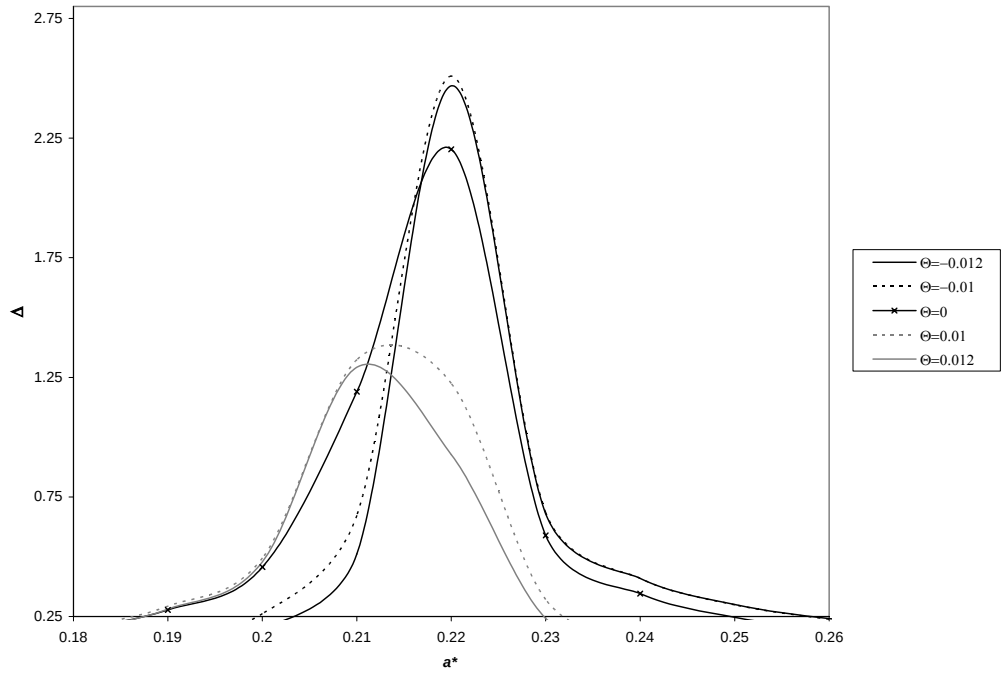
increases for a load or displacement controlled specimen. Again, we observe peaks in the growth paths, which can be interpreted as in Figure 14. Like the hinged-free specimen, the order of magnitude of the bond strength has no effect on the global stiffness of the system.

Effect of Uniform Temperature Field

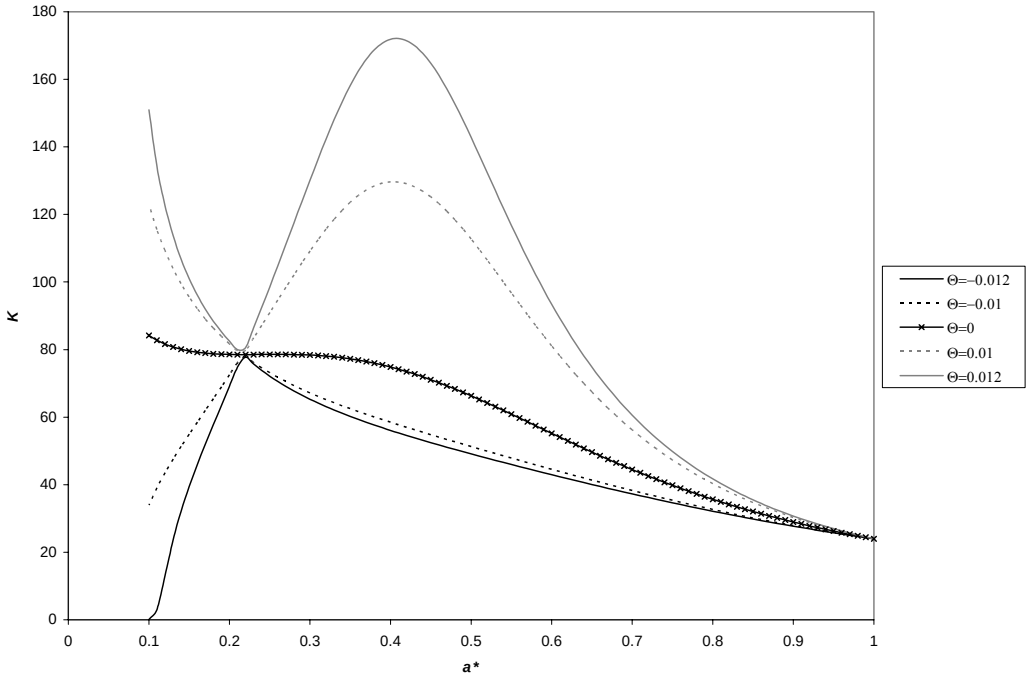
Following here we now examine the delamination behavior of the clamped-free structure exposed to applied transverse pressure as well as a uniform temperature field. Figure 16 compares delamination growth paths for positive and negative temperature fields. For this comparison, $\alpha_p = 0.5$; also $E_0 = 1$ and $2\gamma = 0.1$



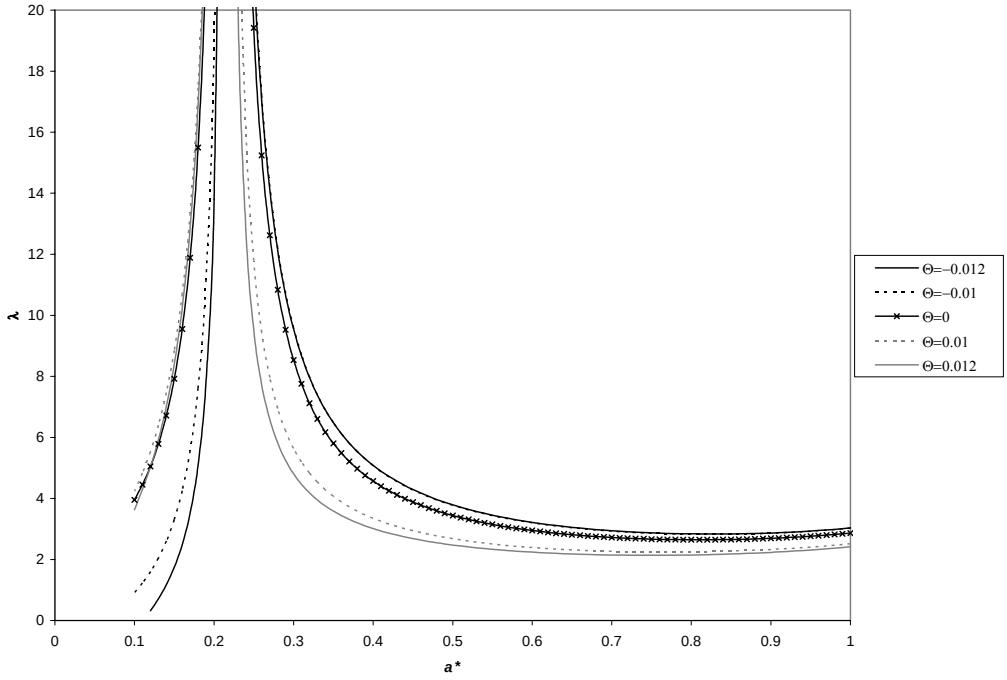
(a)


2.75
2.25
1.75
1.25
0.75
0.25
Δ
0.18
0.19
0.2
0.21
0.22
0.23
0.24
0.25
0.26
a*
Θ=–0.012
Θ=–0.01
Θ=0
Θ=0.01
Θ=0.012

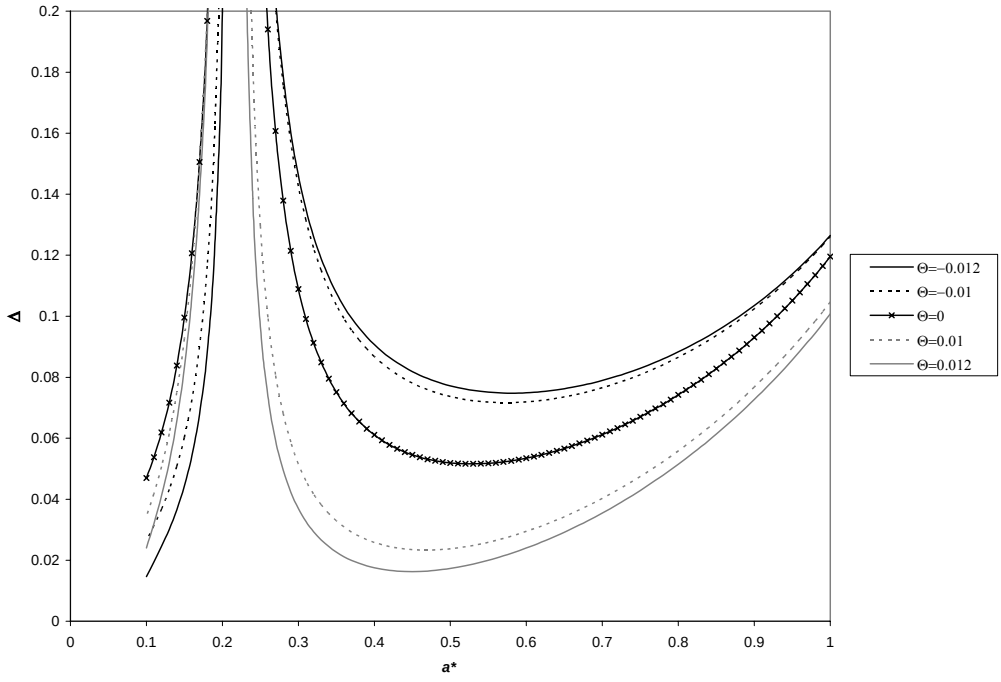
(b)


180
160
140
120
100
80
60
40
20
0
K
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a*
Θ=–0.012
Θ=–0.01
Θ=0
Θ=0.01
Θ=0.012

(c)



(d)



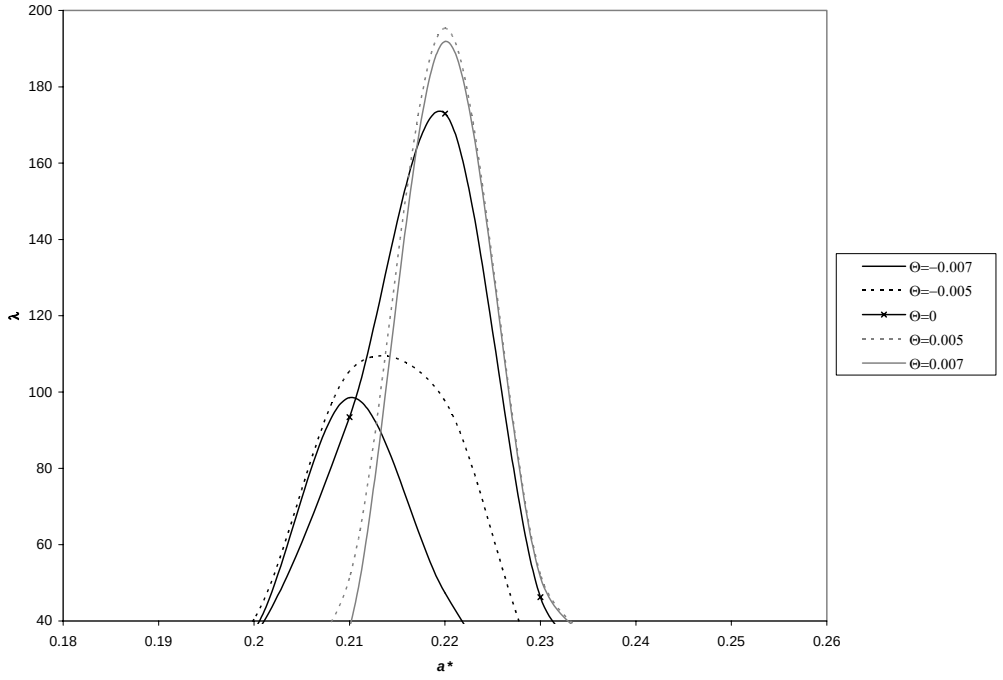
(e)

Figure 16. Delamination growth paths and stiffness degradation for clamped-free ends with applied thermal load, E_0 = 1, 2γ = 0.1, α_p = 0.5 a) close-up of the peaks of λ vs. a*, b) close-up of the peaks of Δ vs. a*, c) *K* vs. a*, d) general λ vs. a*, e) general Δ vs. a*

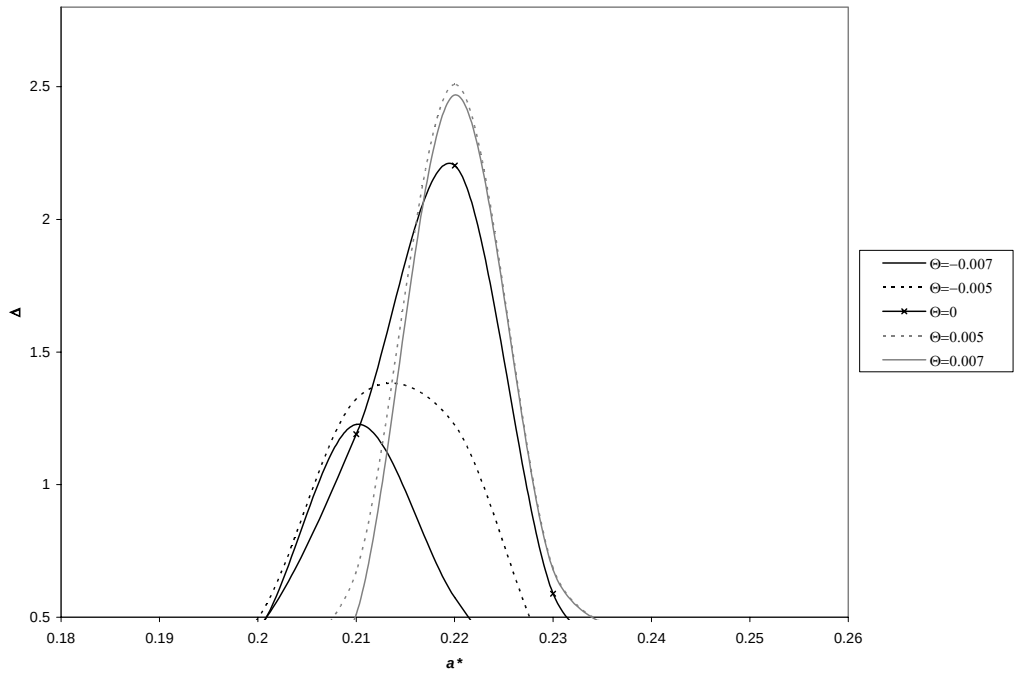
In contrast to the hinged-free case, we once again observe peaks in the delamination growth paths for a clamped-free specimen. Negative temperature fields shift the peaks up and to the right, but again, nonmonotonically. So, initial flaw size may be larger before catastrophic delamination begins, in the presence of such a field. Positive temperature fields shift the peaks down and to the left. The onset of catastrophic delamination will occur sooner for the structure subject to a positive temperature field. All peaks occur in the close vicinity of a critical value $a^* = 0.2$. As evidenced by the plots depicted in Figures 16d and 16e, the behavior differs before and after that critical value of the flaw size is passed. It also appears that stable delamination growth is recovered for much smaller initial flaw size values for the displacement controlled specimen, in comparison to the load controlled. The temperature field also plays a role in the recovery of stable delamination. For instance, if a structure with $a_0^* = 0.5$ is displaced to the point of delamination growth, the stability depends on the temperature change. If $\Theta = 0.12$, the delamination growth is already stable. If, however, $\Theta = -0.12$, the growth is unstable, and will not become stable again until the flaw has progressed to about $a^* = 0.7$. This is a significant difference, and a very important consideration for designing with such a structure.

The stiffness degradation curve (Figure 16c) also shows the area around $a^* = 0.2$ to be critical. In the vicinity of this point, the stiffnesses for varying temperature differences seem to converge. Away from this point, the positive temperature fields create a stiffer structure, and vice versa. All plots approach the same limit as delamination propagates.

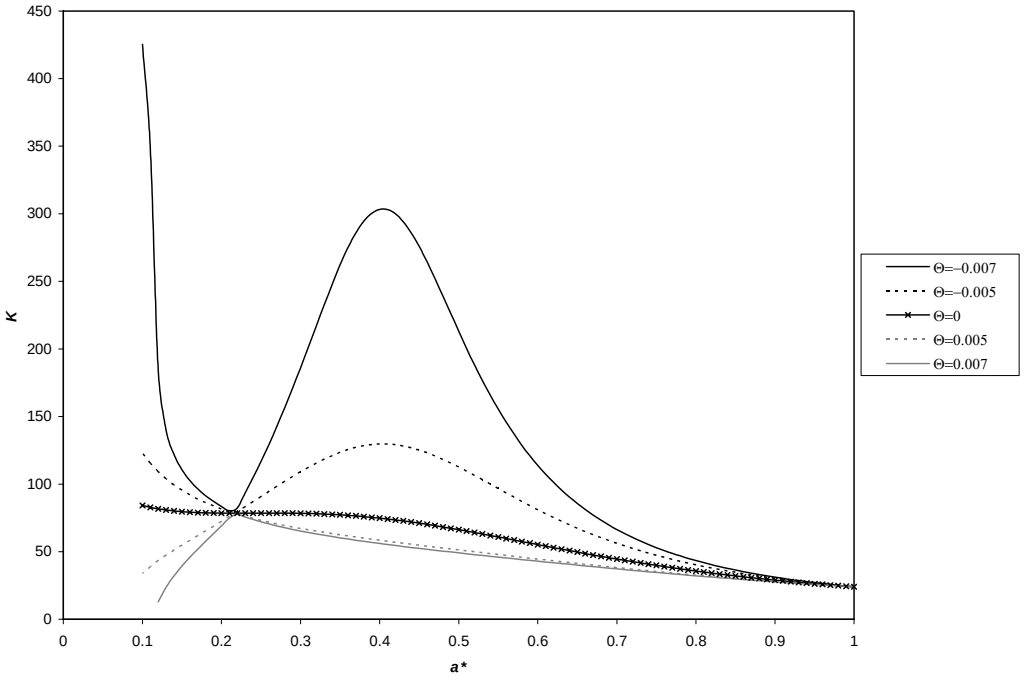
Figure 17 explores the effect of changing the thermal expansion coefficient ratio to $\alpha_p = 2$. (Also for the parameters $E_0 = 1$ and $2\gamma = 0.1$)



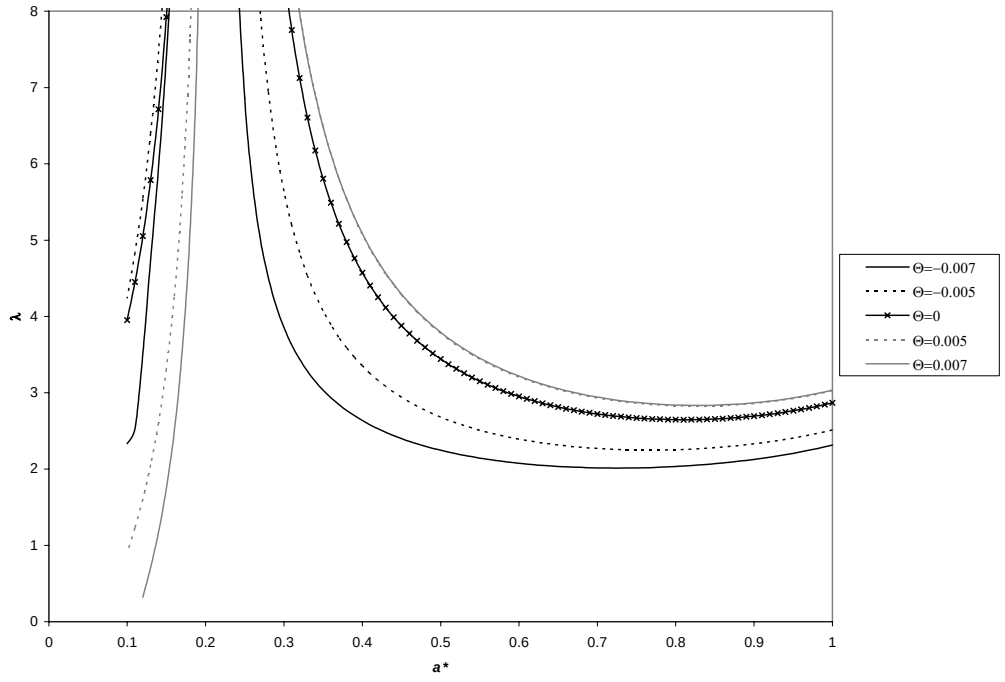
(a)



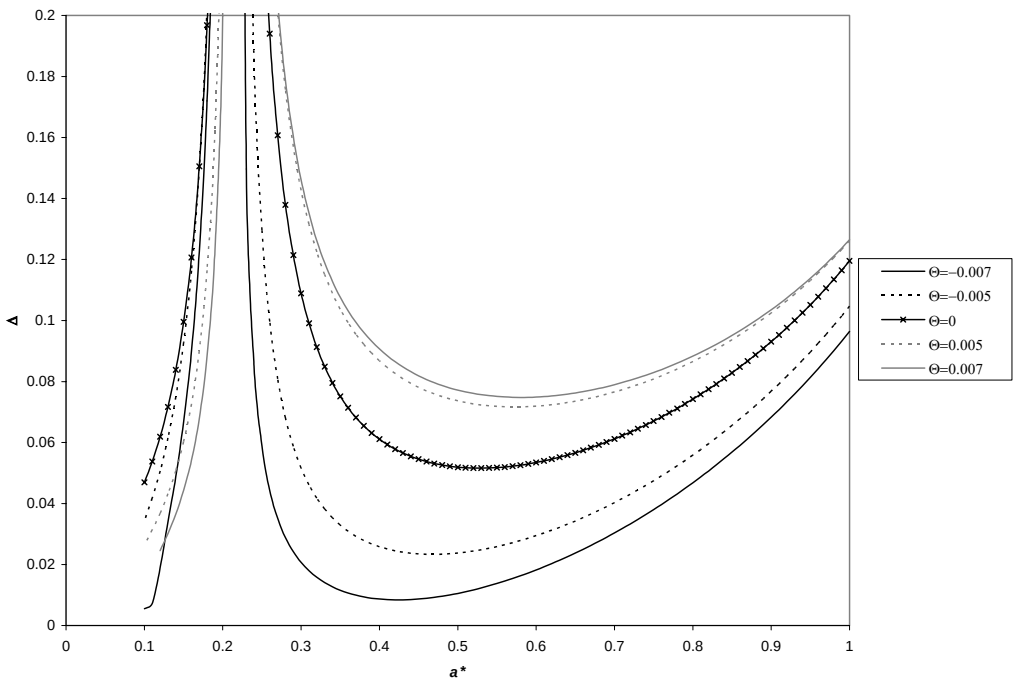
(b)


450
400
350
300
250
200
150
100
50
0
K
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a*
Θ=−0.007
Θ=−0.005
Θ=0
Θ=0.005
Θ=0.007

(c)


8
7
6
5
4
3
2
1
0
λ
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a*
Θ=−0.007
Θ=−0.005
Θ=0
Θ=0.005
Θ=0.007

(d)

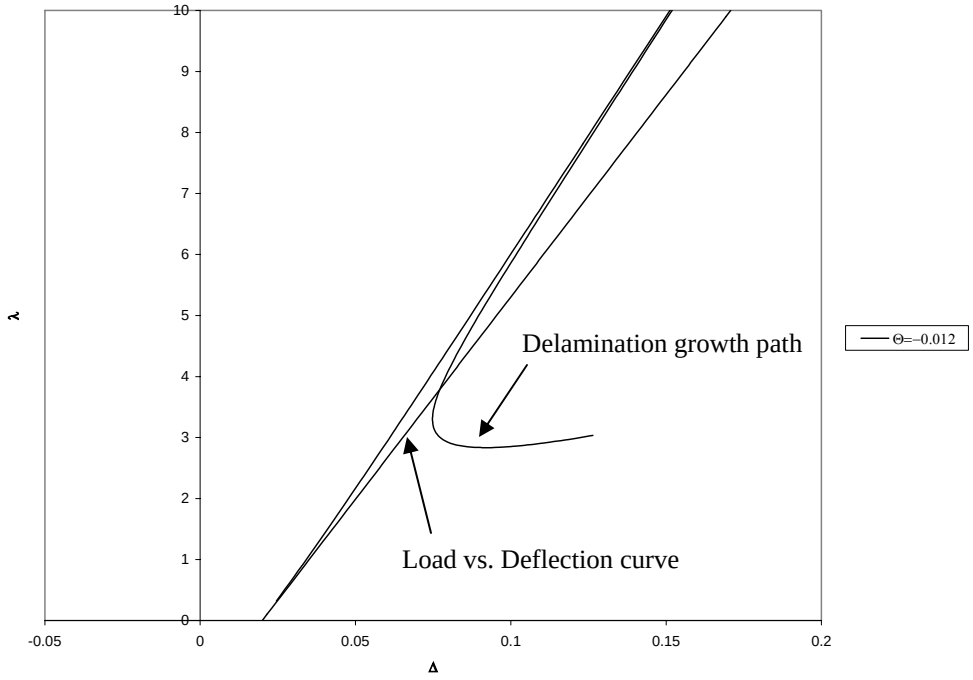


(e)

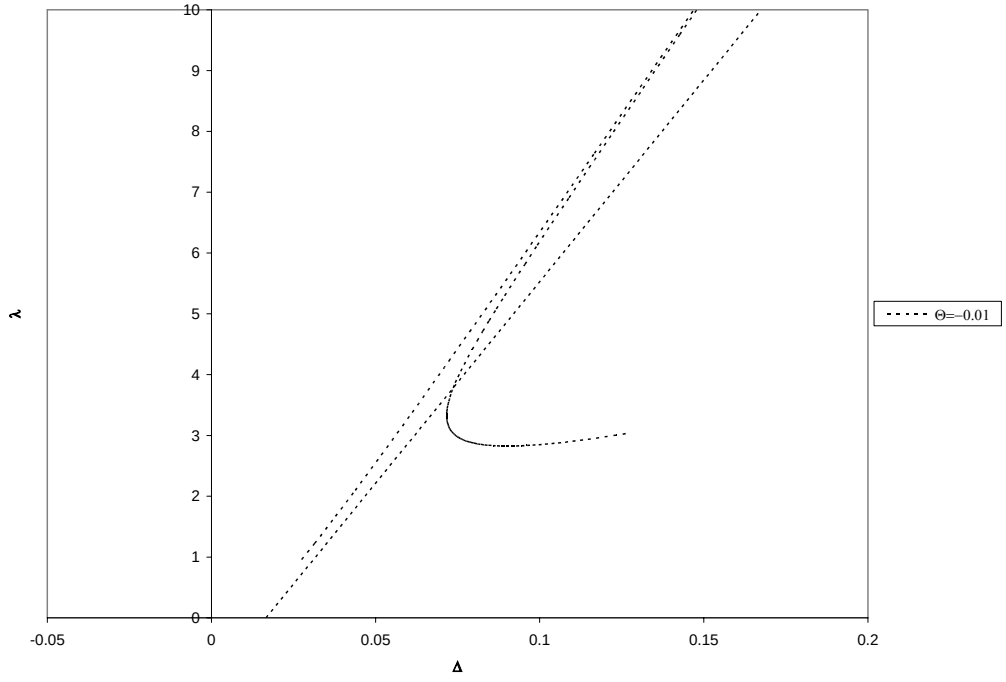
Figure 12. Delamination growth paths and stiffness degradation for clamped-free ends with applied thermal load, E_0 = 1, 2γ = 0.1, α_p = 2 a) close-up of the peaks of λ vs. a*, b) close-up of the peaks of Δ vs. a*, c) *K* vs. a*, d) general λ vs. a*, e) general Δ vs. a*

The change in the ratio of coefficients of thermal expansion reverses the trends observed in Figure 16. Positive temperature fields shift the peaks slightly up and to the right, whereas positive temperature fields shift the peaks down and to the left. All peaks again occur near $a^* = 0.2$. In Figures 17d and 17e, positive temperature fields are seen to lower the threshold for delamination before the peaks, and raise it after the peaks. Similar trends in stability are observed for Figure 17 in comparison to Figure 16, the difference being the reverse effect of a temperature change.

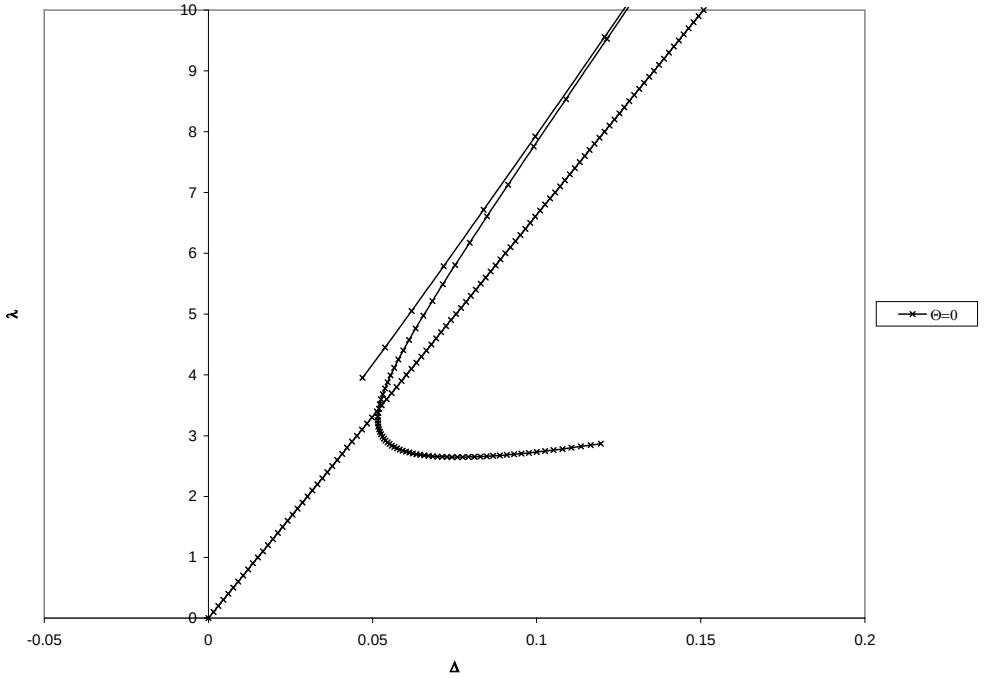
Figure 18 shows the load-deflection curve superimposed onto the delamination growth path in λ-Δ space for each temperature field presented from Figure 16. The parameters are taken as $E_0 = 1$, $2\gamma = 0.1$, and $\alpha_p = 0.5$. They are also plotted with an initial flaw size of $a_0^* = 0.5$.


λ
Δ
Delamination growth path
Load vs. Deflection curve
Θ=−0.012

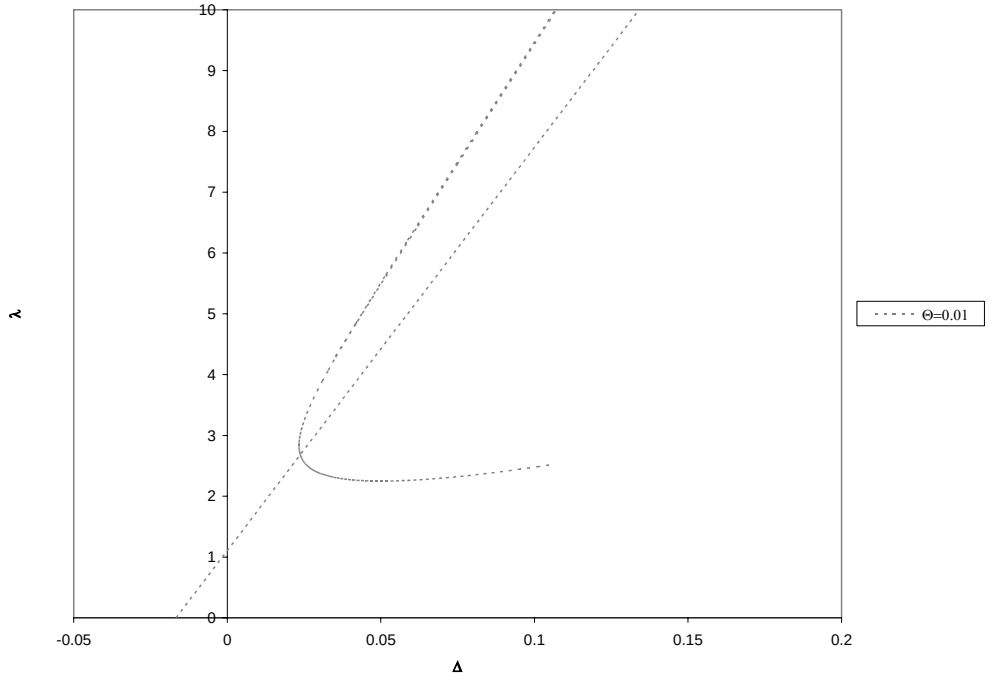
(a)


λ
Δ
Θ=−0.01

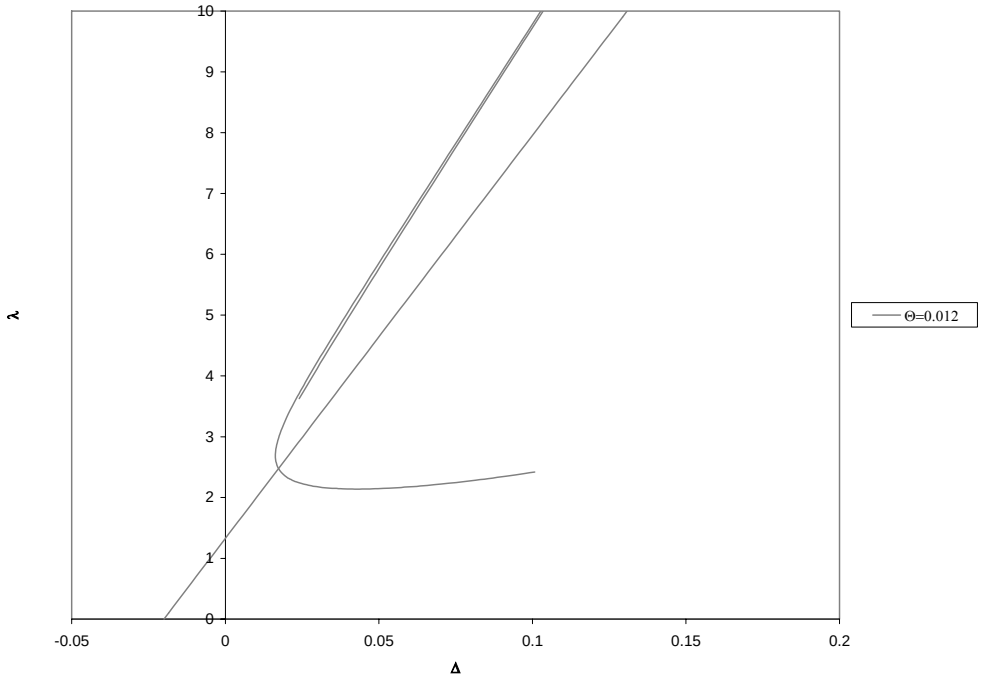
(b)


10
9
8
7
6
5
4
3
2
1
0
λ
-0.05
0
0.05
0.1
0.15
0.2
Δ
Θ=0

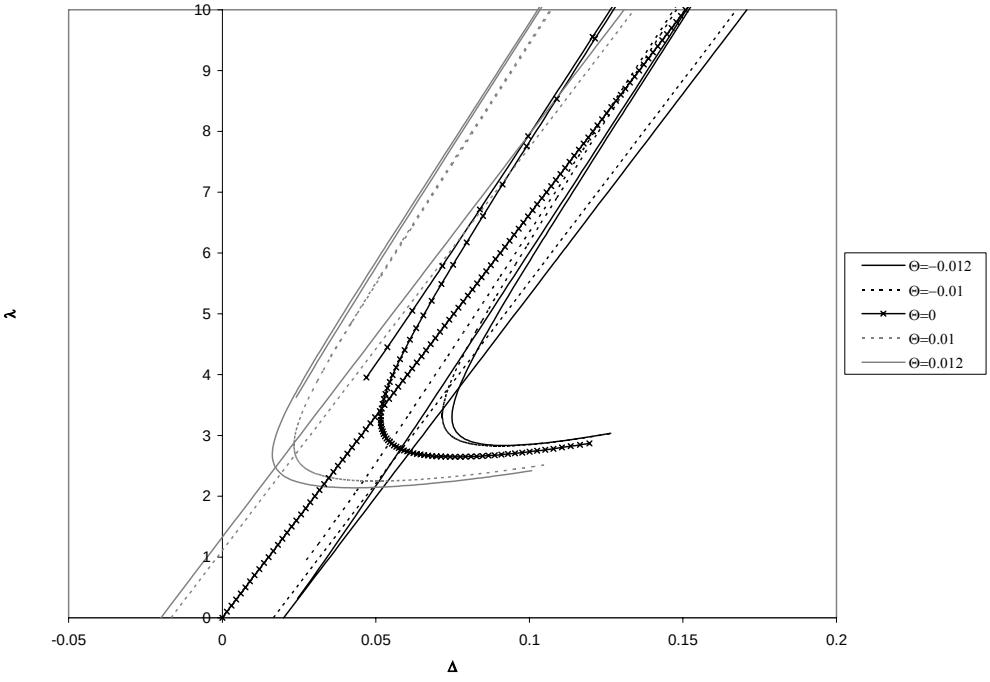
(c)


10
9
8
7
6
5
4
3
2
1
0
λ
-0.05
0
0.05
0.1
0.15
0.2
Δ
Θ=0.01

(d)



(e)



(f)

Figure 18. Load vs. deflection curves and corresponding delamination growth paths for clamped-free ends, E_0 = 1, 2γ = 0.1, α_p = 0.5, a) Θ=-0.012, b) Θ =-0.01, c) Θ = 0, d) Θ =0.01, e) Θ =0.012, f) summary

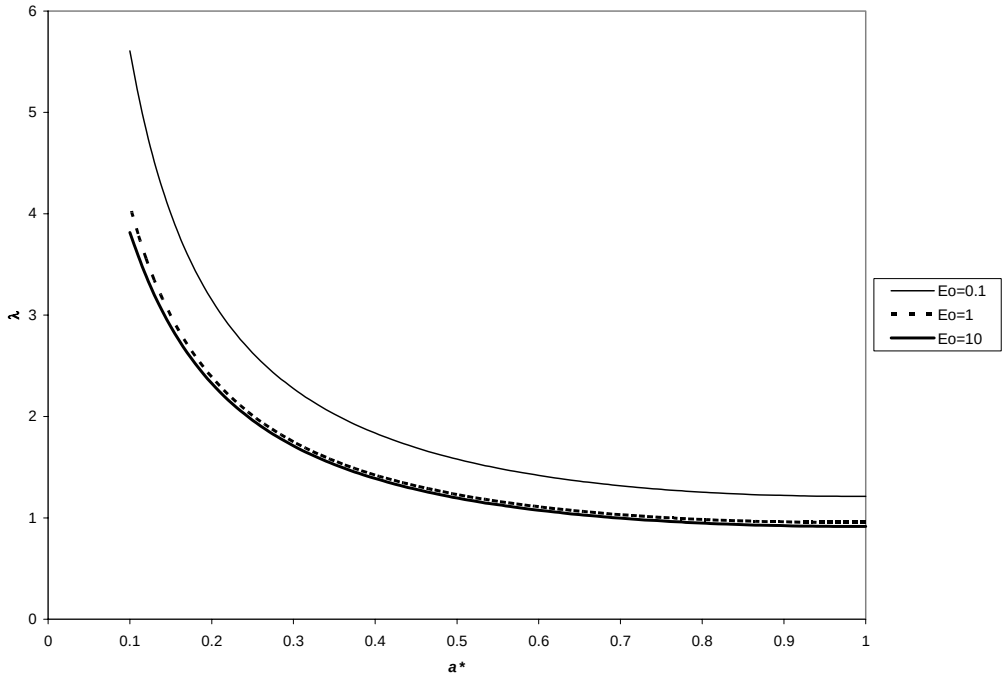
The interpretation of Figure 18 follows from that of Figure 13. In Figure 18f a similar offshoot from zero deflection is observed that is the effect of the thermal load, even in the absence of pressure. (Note: the load vs. deflection curve and the delamination growth path do *not* intersect at all in the vicinity of $\Delta = 0.02$ in Figure 18a)

4.4.3 Hinged-Fixed End Supports

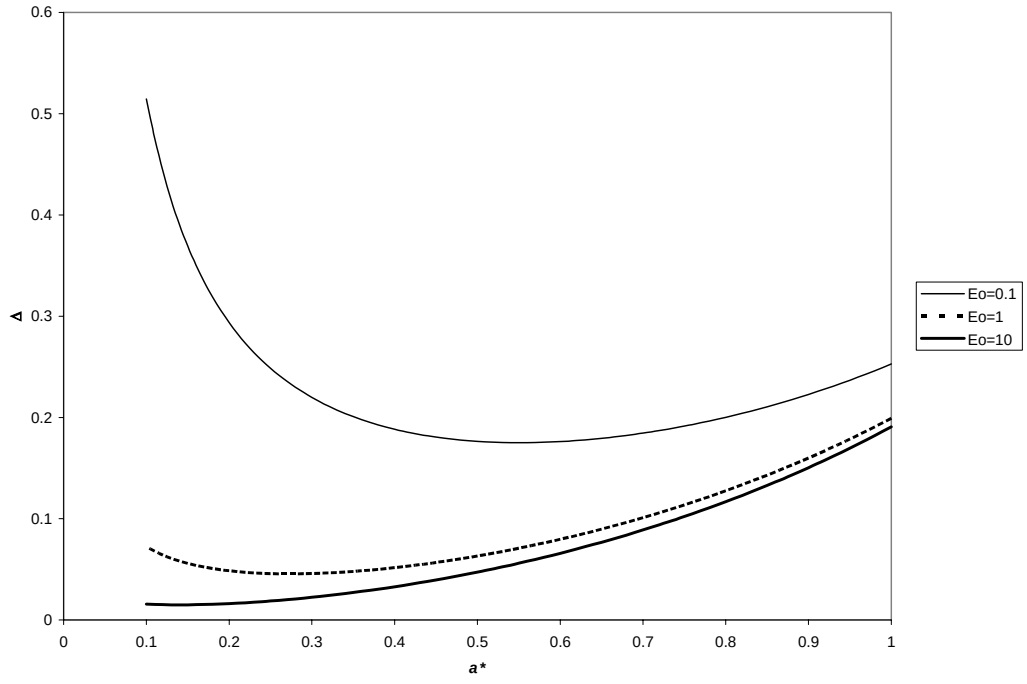
The following section studies the structure with hinged supports, however, this time with the added restriction that the ends of the structure are not free to move in the in-plane direction.

Pure Pressure Loading, $\Theta = 0$

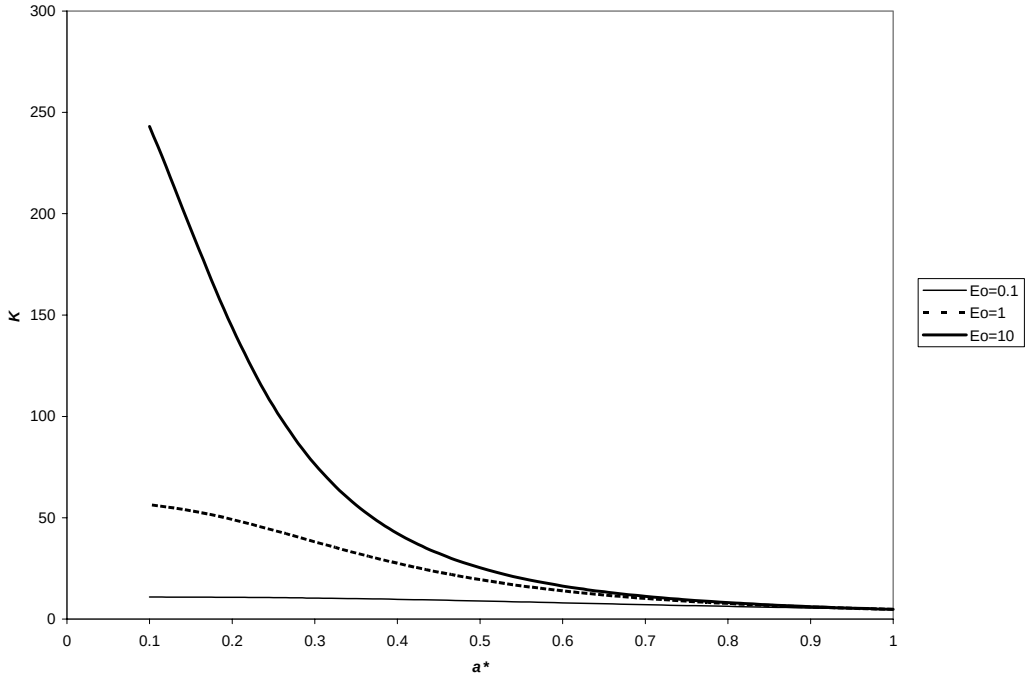
Figure 19 shows a comparison of the delamination growth paths subject to different values of the elastic modulus ratio, when there is no temperature field present and $2\gamma = 0.1$.



(a)



(b)

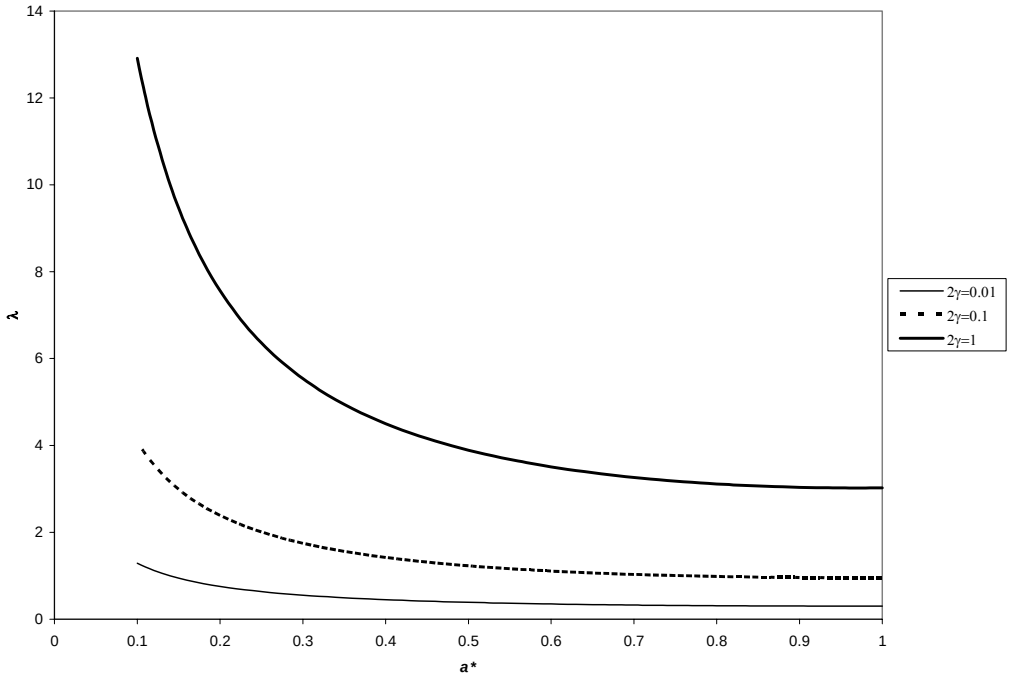


(c)

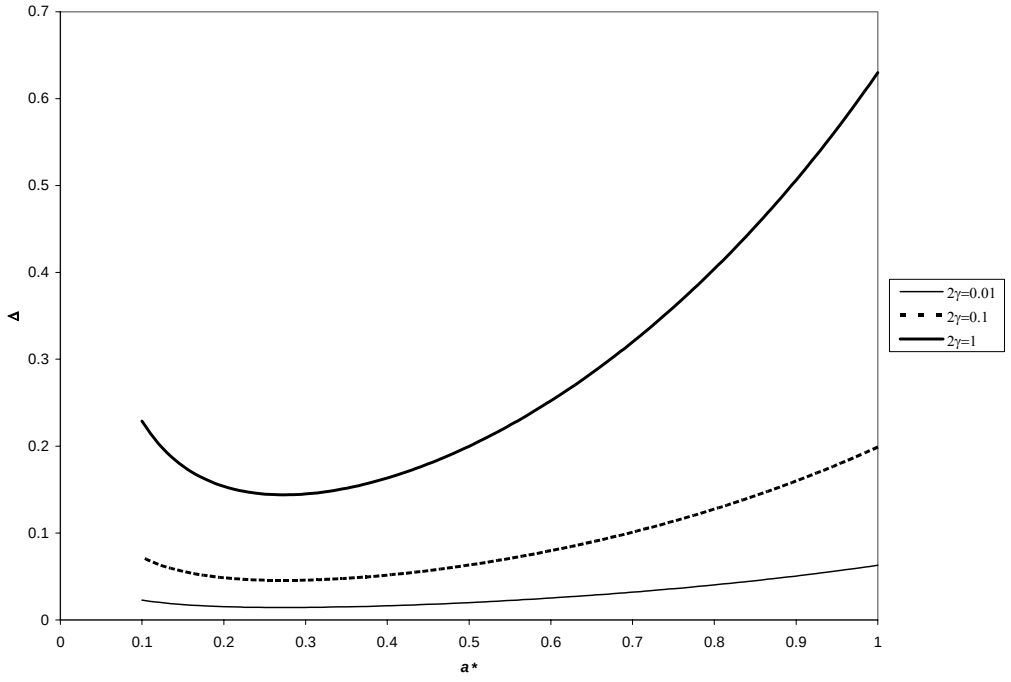
Figure 19. Delamination growth paths and stiffness degradation for hinged-fixed ends without applied thermal load, 2γ = 0.1 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

The trends observed in Figure 19 for hinged-fixed ends are similar to those observed in Figure 9 for hinged-free ends. Smaller values of the modulus ratio are seen to raise the threshold of delamination in Figures 19a and 19b. The load controlled test shows unstable delamination growth for all initial flaw sizes, while a displacement controlled test shows either unstable followed by stable growth, or purely stable growth, depending on the initial flaw size. It is also seen from Figure 19b that if the modulus ratio is high enough, the delamination will be stable for almost all initial flaw sizes. Figure 19c may be interpreted the same as Figure 9c.

Figure 20 shows the effect of the bond strength on the delamination behavior of the hinged-fixed specimen in the absence of a temperature field and when E_0=1. Trends are similar to those seen for the hinged-free case in Figure 10.



(a)



(b)

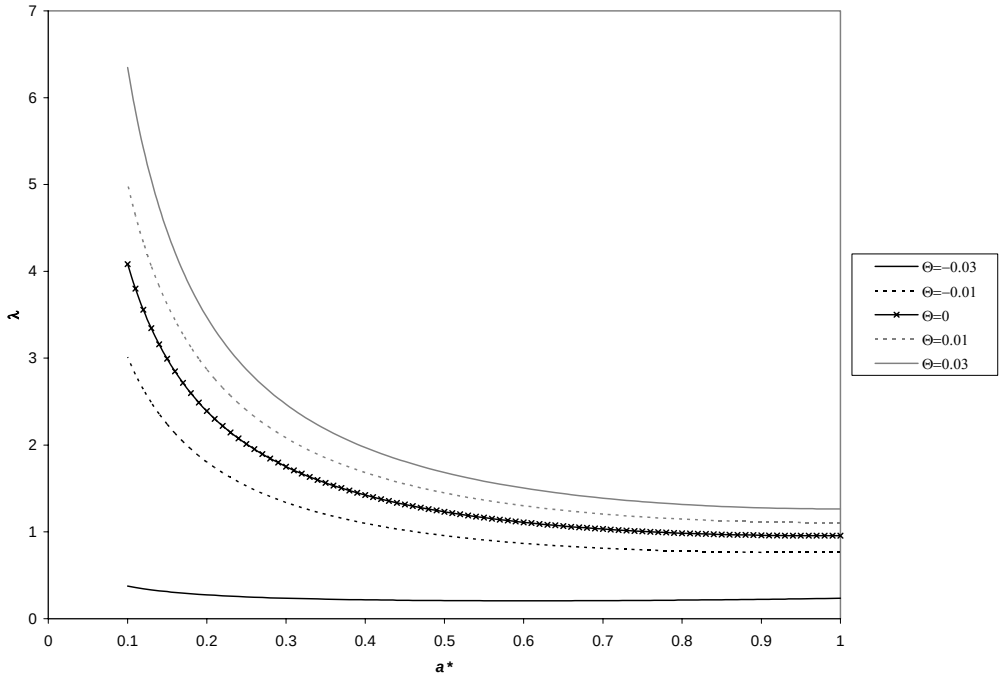
Figure 20. Delamination growth paths for hinged-fixed ends without applied thermal load, E_0 = 1 a) λ vs. a*, b) Δ vs. a*

Greater values of the bond strength expectedly increase the thresholds for delamination. A load controlled specimen delaminates in an unstable manner, regardless of the initial

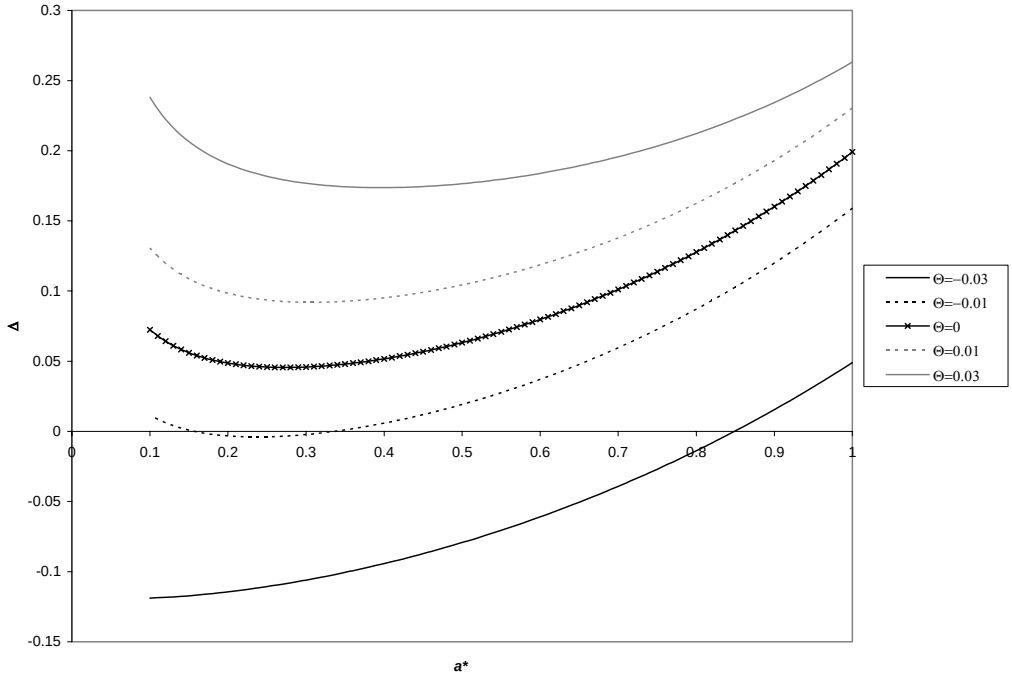
flaw size, whereas the displacement controlled specimen will experience unstable followed by stable delamination depending on the size of that flaw. The magnitude of the bond strength serves only to alter the amplitude of the delamination path. The stiffness is unaffected by the bond strength.

Effect of Uniform Temperature Field

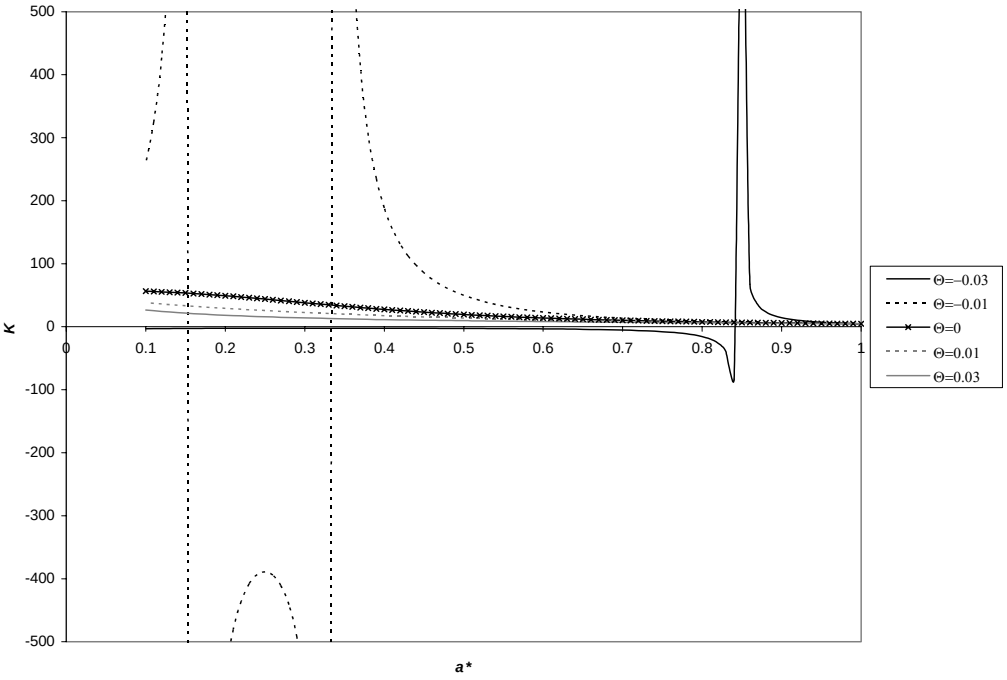
Figure 21 is a plot of the delamination growth paths for hinged-fixed ends under a uniform temperature field as well as transverse pressure. These paths are plotted for parameters $E_0 = 1$, $2\gamma = 0.1$, and $\alpha_p = 0.5$. For hinged ends, we do not expect a contact zone unless the deflection is downward.



(a)



(b)



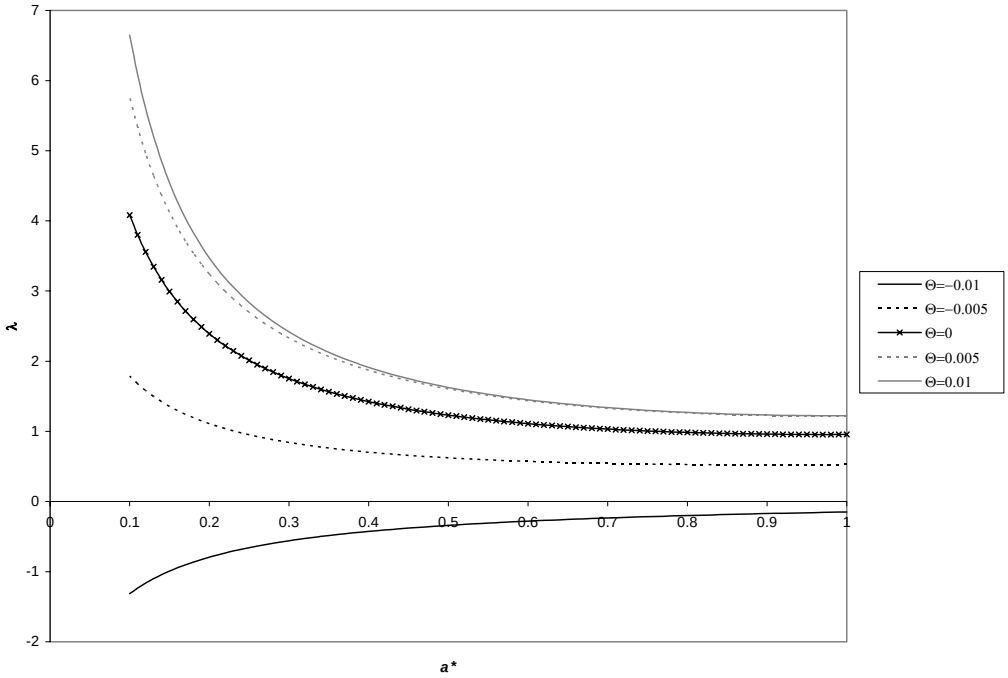
(c)

Figure 21. Delamination growth paths and stiffness degradation for hinged-fixed ends with applied thermal load, $E_0 = 1$, $2\gamma = 0.1$, $\alpha_p = 0.5$ a) λ vs. a*, b) Δ vs. a*, c) K vs. a*

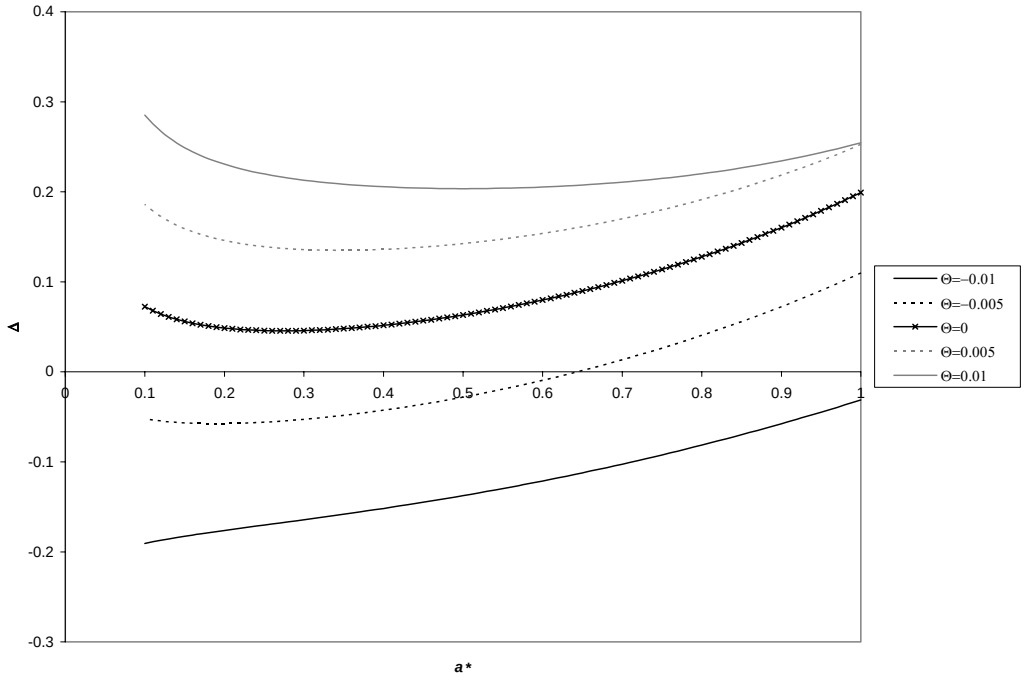
Unlike the case of hinged-free end supports, in Figures 21a and 21b we observe that positive temperature fields actually increase the delamination threshold, and negative

fields decrease it. Debonding is almost entirely unstable for a load controlled specimen, save for the case of Θ = -0.03. The delamination growth for this case is unstable followed by stable. The structure is much more restricted now, since it can not move in the in-plane direction. So, in this case, when a positive temperature difference is applied, the baseplate will expand. However, since the ends are fixed, this generates a compressive membrane force in the baseplate. This causes bending-stretching coupling at the reference surface in the bonded region, and hence, upward deflection. When the temperature difference is negative however, the baseplate will contract. The fixed ends will then create a tensile membrane force, which ultimately results in the reverse deflection. This is reflected in Figure 21b for the characteristic deflection. This also suggests the presence of a contact zone. Our research on this particular case indicates that the behavior is more complex however; as it seems to imply that the structure may exhibit a propagating contact zone boundary. This requires additional future study to further confirm. The opposite deflections are also noted in the stiffness plot, Figure 21c. The asymptotes occur due to the change in sign of deflection.

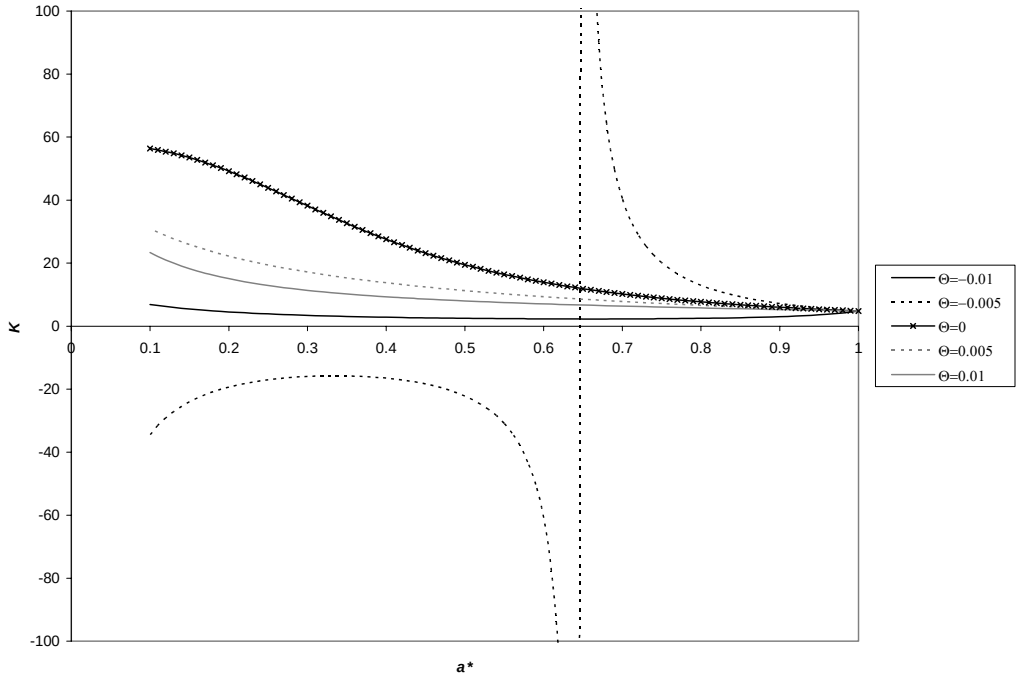
Figure 22 presents the reverse scenario from Figure 21 for the ratio of thermal expansion coefficient (i.e. $\alpha_p = 2$).



(a)



(b)



(b)

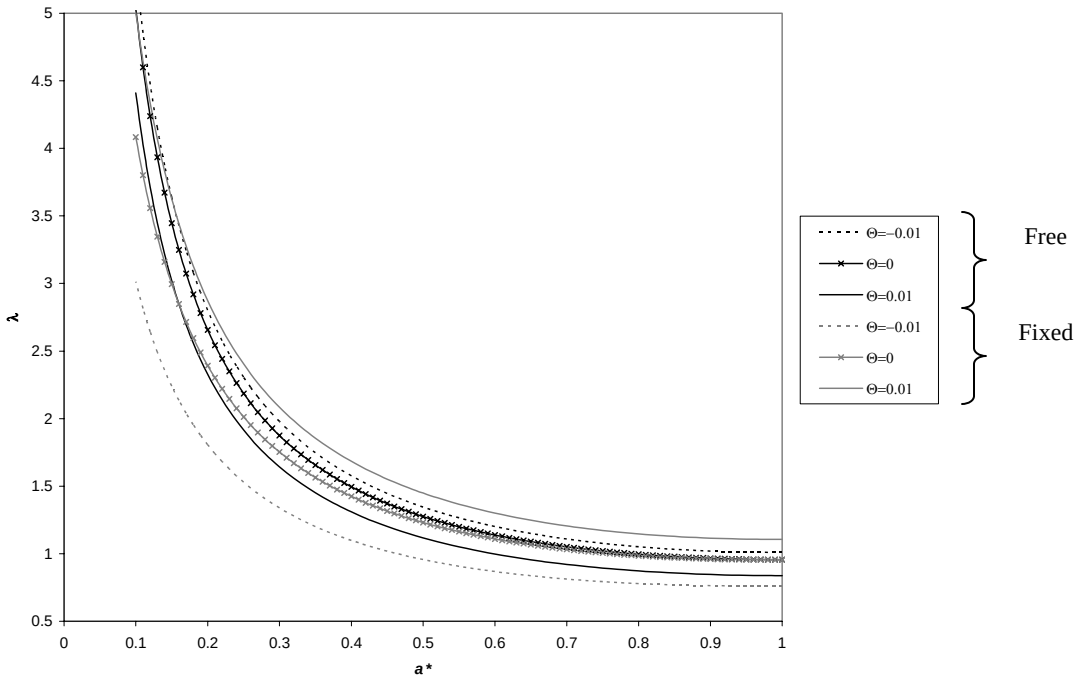
Figure 22. Delamination growth paths and stiffness degradation for hinged-fixed ends with applied thermal load, $E_0 = 1$, $2\gamma = 0.1$, $\alpha_p = 2$ a) λ vs. a*, b) Δ vs. a*, c) K vs. a*

The trends observed actually follow the trends observed in Figure 21. That is, it appears that the thermal expansion coefficient ratio does not alter the trend of the more negative thermal loads greatly decreasing the thresholds of delamination growth. It is interesting to note that the λ delamination growth path for $\Theta = -0.01$ is entirely negative. So the structure will delaminate before the transverse pressure can even be applied. The temperature field is strong enough by itself to cause delamination propagation.

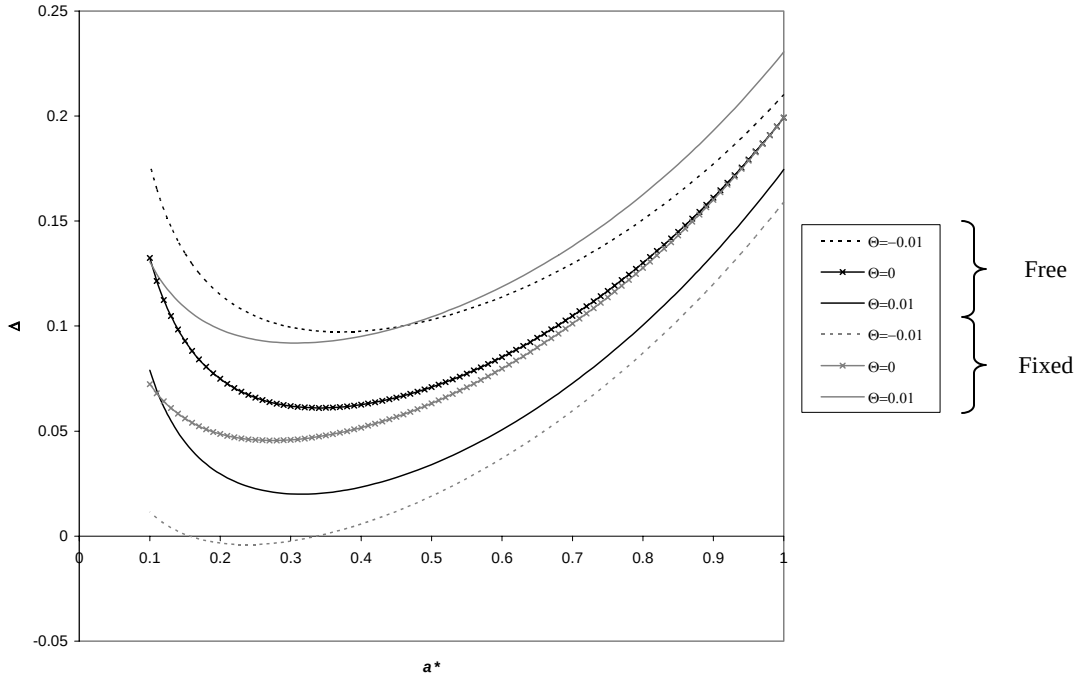
We once again observe that the structure deflects downward for certain values of the initial flaw size. This also suggests that a contact zone is present. Again, our tests show us that the behavior is rather complex, and must be further analyzed before it can be truly characterized. This is unlike the hinged-free case, where the reciprocal values of the thermal expansion coefficient produced reverse trends. Again, this is due to the fixed-

ends, which cause a membrane force to be generated when the structure expands or contacts. This membrane force affects the sense of how the structure can deflect.

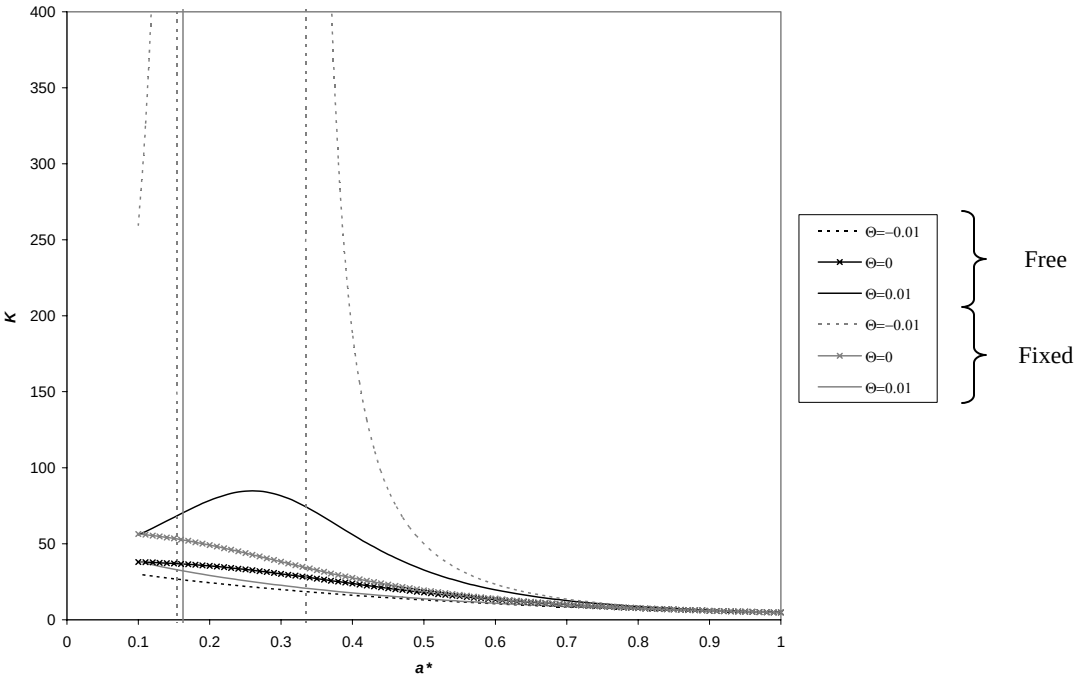
Figure 23 presents a comparison of delamination growth paths for hinged-free ends and hinged-fixed ends. The parameters are taken as $E_0 = 1$, $2\gamma = 0.1$, and $\alpha_p = 0.5$.



(a)



(b)



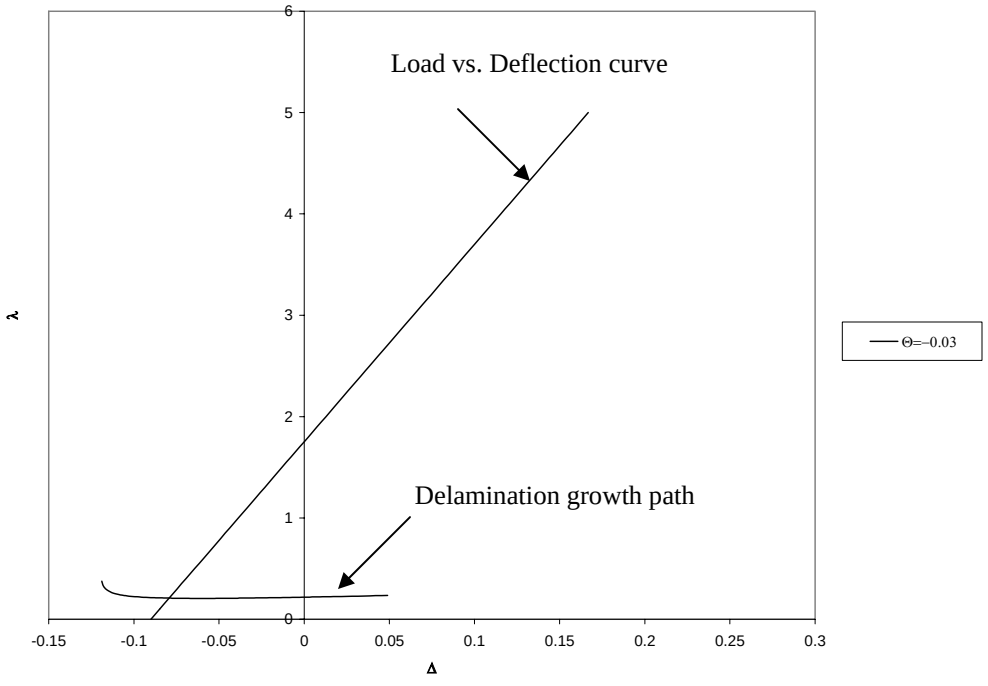
(c)

Figure 23. Comparison of delamination growth paths and stiffness degradation for hinged-free and hinged fixed ends, where E_0=1, 2γ=0.1, and α_p=0.5 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

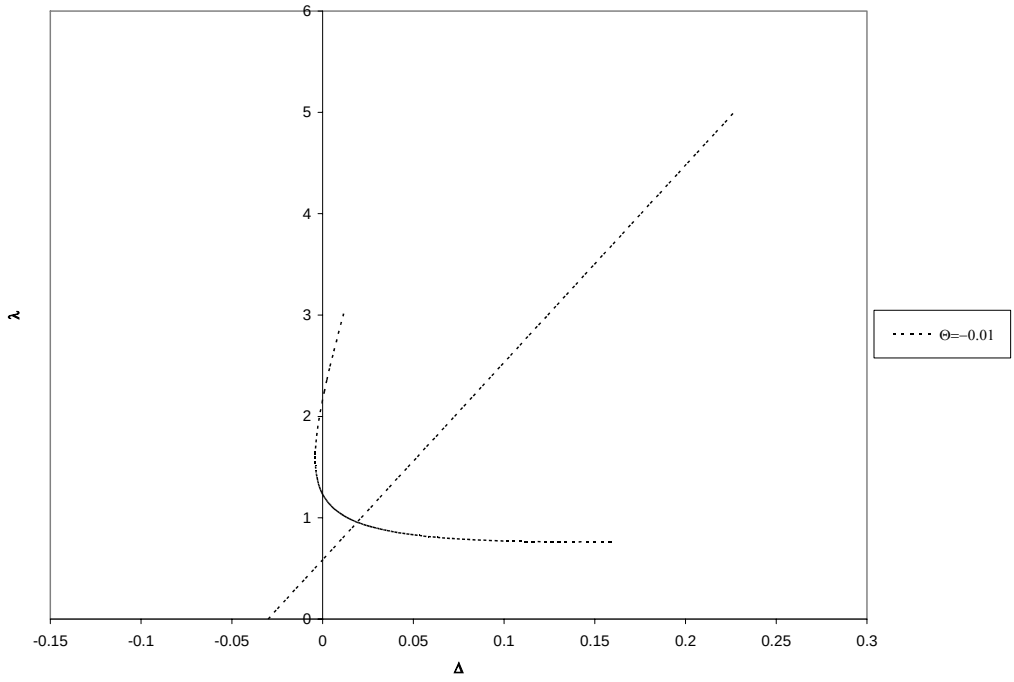
The plots show that the effect of temperature on the structure is dependent on the type of support conditions for these parameters. For free ends, the more positive temperature

field lowered the delamination threshold, while the more negative temperature field raised it. The trend is reversed for the fixed ends. The positive temperature field causes the hinged-free structure to deflect down, whereas it causes the hinged-fixed structure to deflect up. This is because the thermally induced compressive membrane force creates a mechanical moment in the bond zone, which is further amplified by the thermal moment in that same zone. This seems to render the hinged-fixed structure more compliant, while it renders the hinged-free structure less compliant, thus affecting the thresholds for delamination growth.

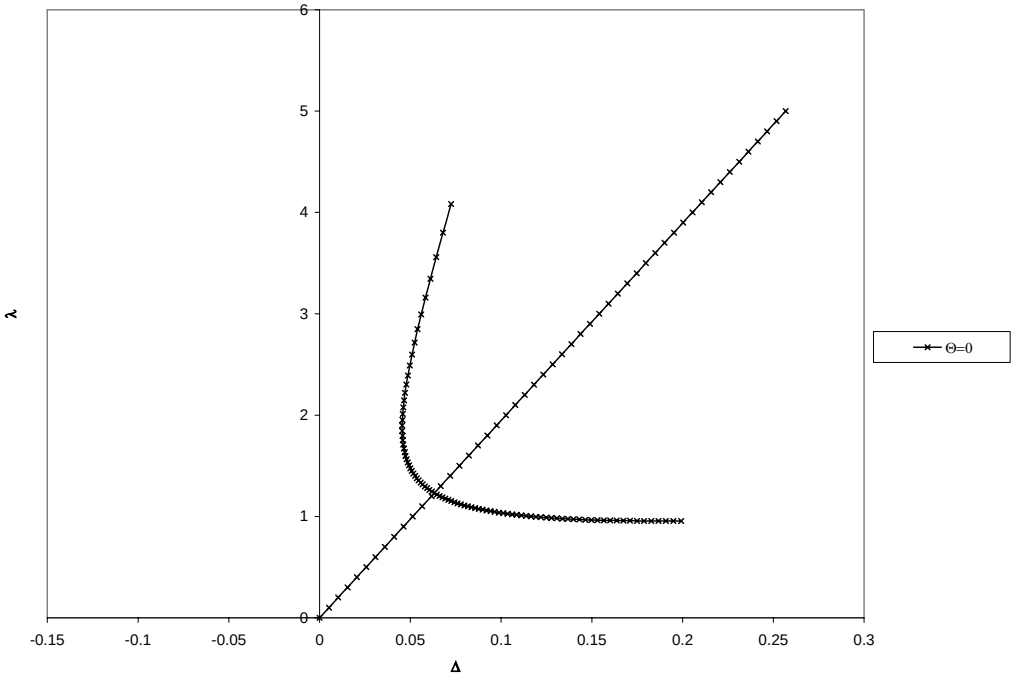
Figure 24 shows the load vs. deflection curve and delamination growth paths for all temperature fields presented in Figure 21. The initial flaw size is taken to be $a_0^* = 0.5$. All parameters are the same as in Figure 21. Again, the interpretation follows from Figure 13.



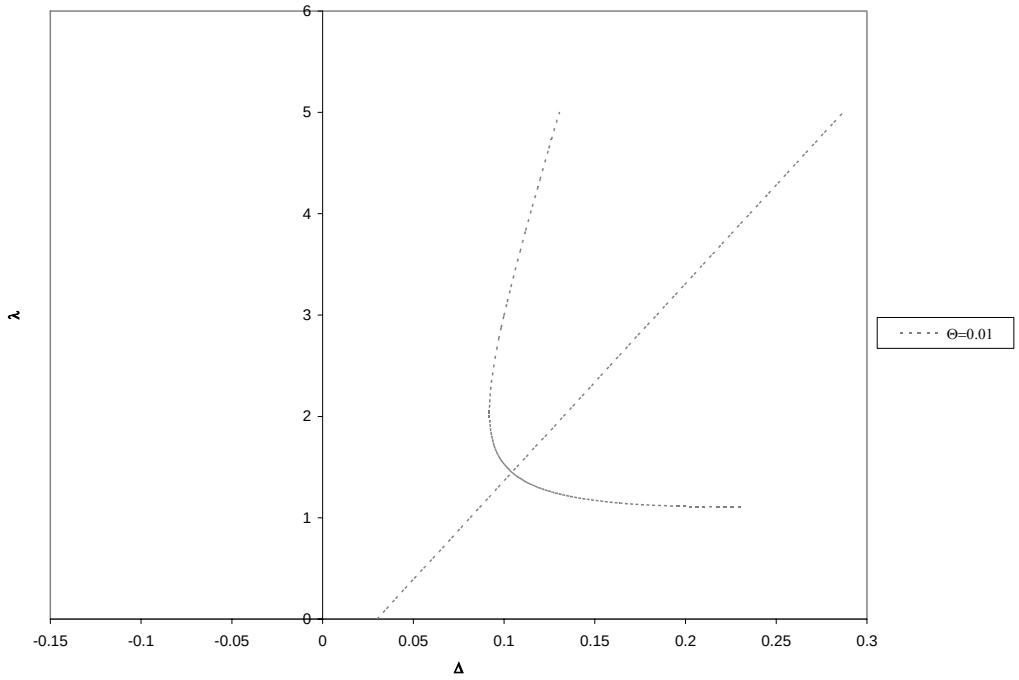
(a)



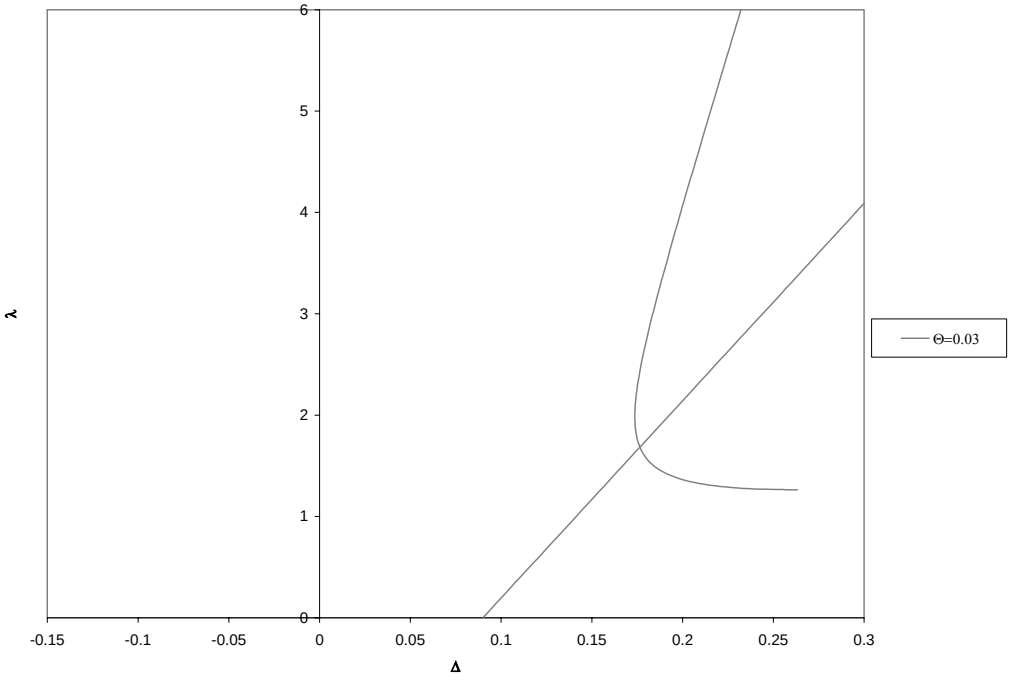
(b)



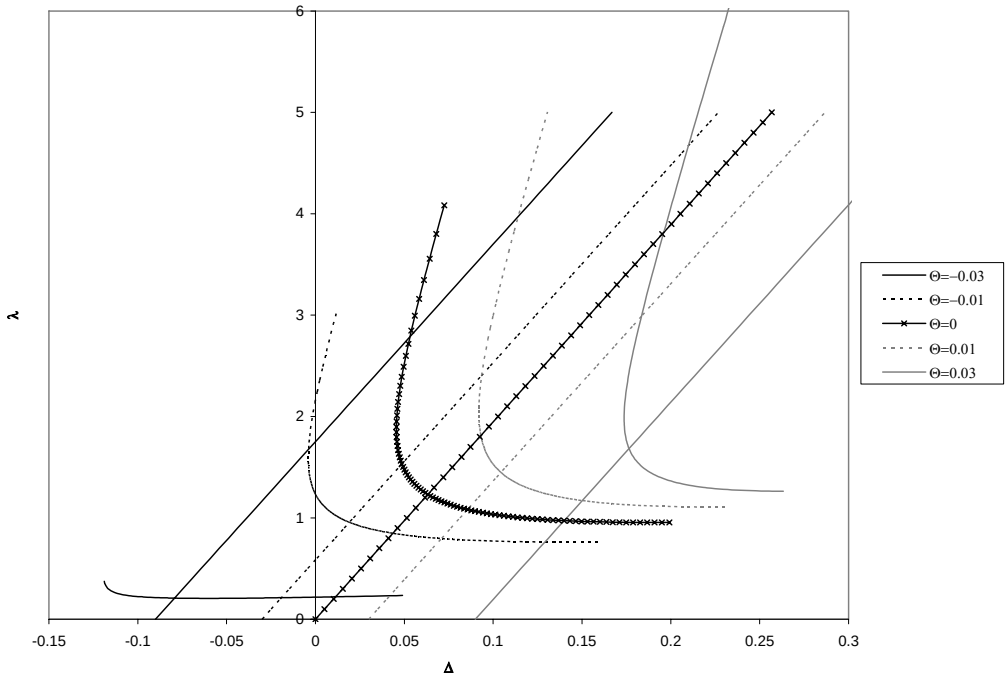
(c)


λ
Δ
Θ=0.01
0
1
2
3
4
5
6
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3

(d)


λ
Δ
Θ=0.03
0
1
2
3
4
5
6
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2
0.25
0.3

(e)



(f)

Figure 24. Load vs. deflection curves and corresponding delamination growth paths for hinged-fixed ends, E_0 = 1, 2γ = 0.1, α_p = 0.5, a) Θ=-0.012, b) Θ =-0.01, c) Θ = 0, d) Θ =0.01, e) Θ =0.012, f) summary

From Figure 24f it can be seen how the positive temperature field greatly increases the threshold for delamination growth. We can also see the effect of temperature in the absence of pressure; recognized as the offset of the load-deflection curve at the zero point of pressure. The offsets are reversed with respect to what was observed in Figure 13. Since α_p = 0.5, the baseplate has twice the thermal expansion coefficient of the patch, and hence expands under a positive temperature field more than the patch. Since the ends are fixed, the baseplate cannot expand, which results in a compressive membrane force. In the bond zone, there is a bending-stretching coupling effect at the reference surface. This is depicted in Figure 25. The effect is to deflect upward. Under a negative field, there is a resultant tensile force, and the deflection is downward.

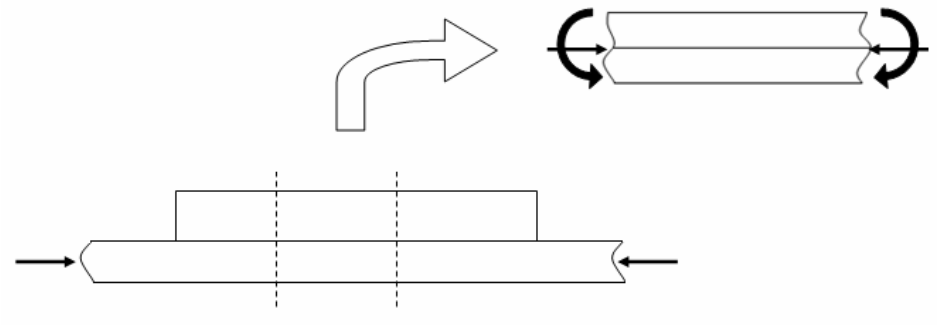


Figure 25. Bending-stretching coupling in the bond zone due to thermal expansion with fixed ends

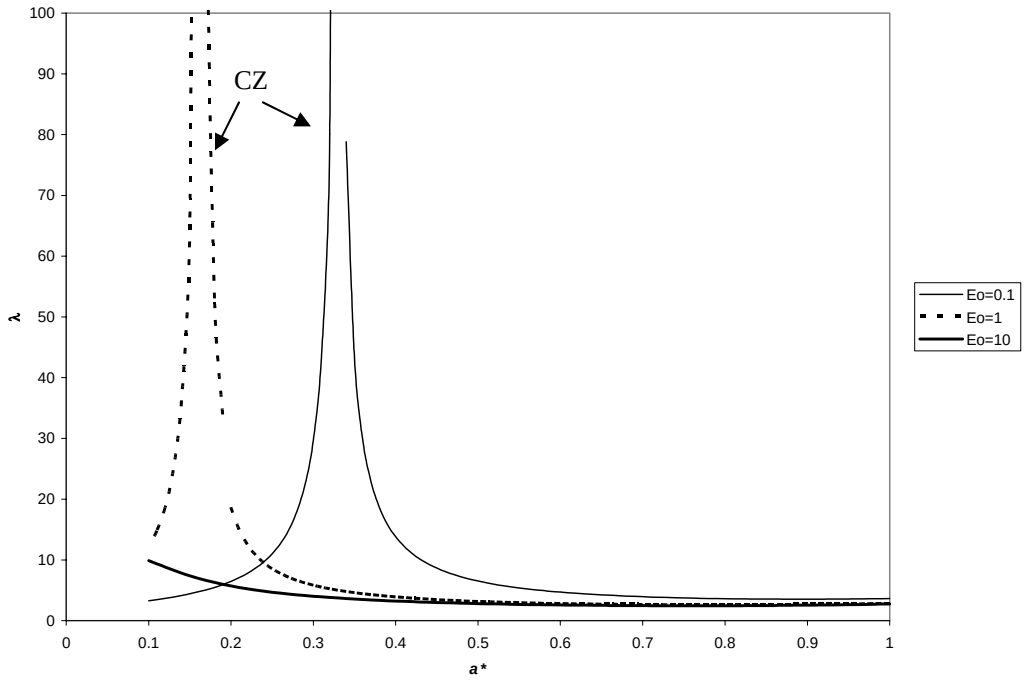
4.4.4 Clamped-Fixed End Supports

In the following section we study the structure with clamped supports, however, this time with the added restriction that the ends of the structure are not free to move in the in-plane direction. It is in this case that we examine the effect of the presence of a contact zone (See Section 3.2). For this case we take the patch size to be $L_p = 0.9$. This parallels the structure studied by Bottega (1995), who proves that for a patched plate under mechanical loading, the contact zone is full ($b = L_p$), or nonexistent ($b = a$). The validity of the contact zone solution must be verified, and this is done by imposing the condition that the curvature in Region S_2 is negative, which is consistent with our sign convention.

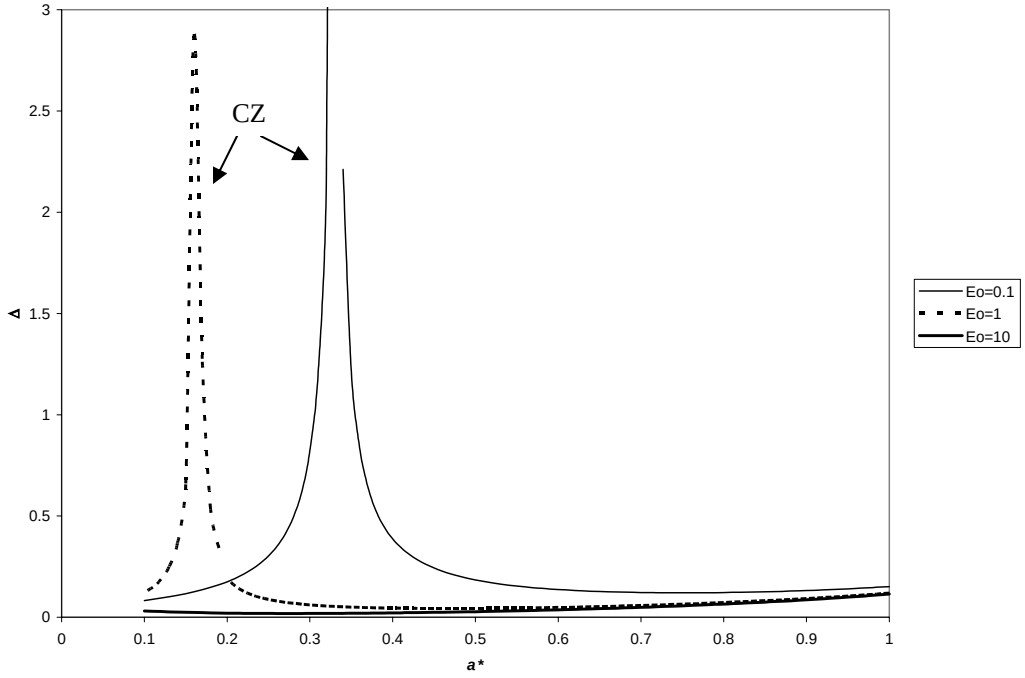
Pure Pressure Loading, $\Theta = 0$

Figure 26 shows a comparison of the delamination growth paths subject to different values of the elastic modulus ratio, when there is no temperature field present and $2\gamma=0.1$. For this case of zero temperature, the equilibrium configurations with a full contact zone were found for values of a^* on [0.1, 0.33] for $E_0 = 0.1$, and on [0.1, 0.19] for $E_0 = 1$. No equilibrium configurations with a contact zone were found for $E_0 = 10$. This

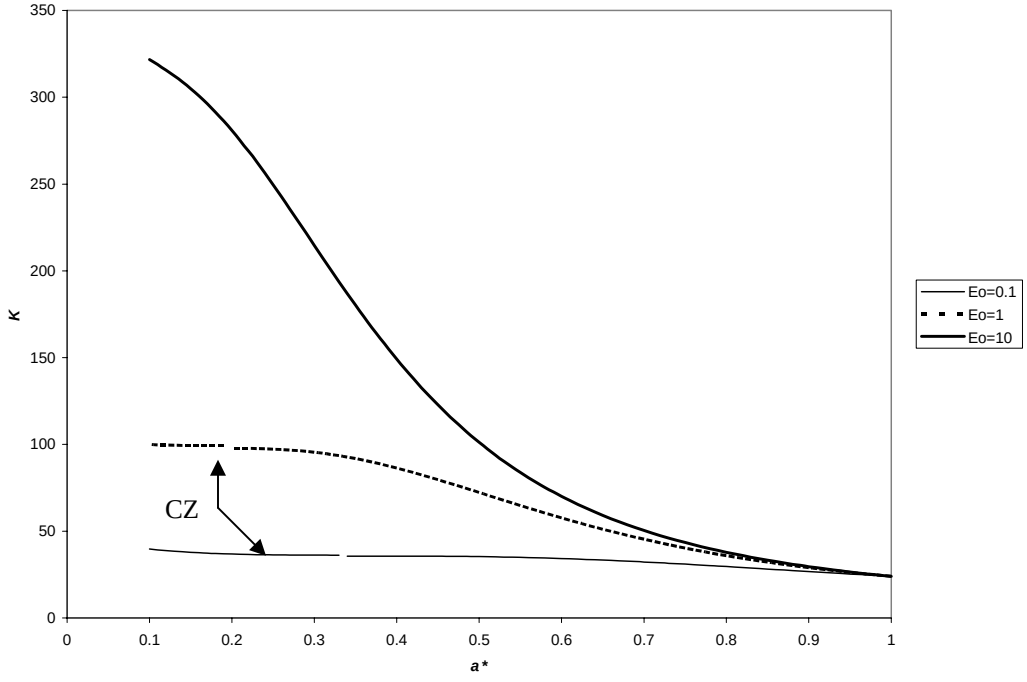
suggests that as the structure becomes more compliant, contact is more likely to occur. A more flexible structure is likely to be able to bend in a way that promotes the contact zone, in other words, in a way such that the curvature "opens up" in region S_2, as shown in Figure 25.



(a)



(b)



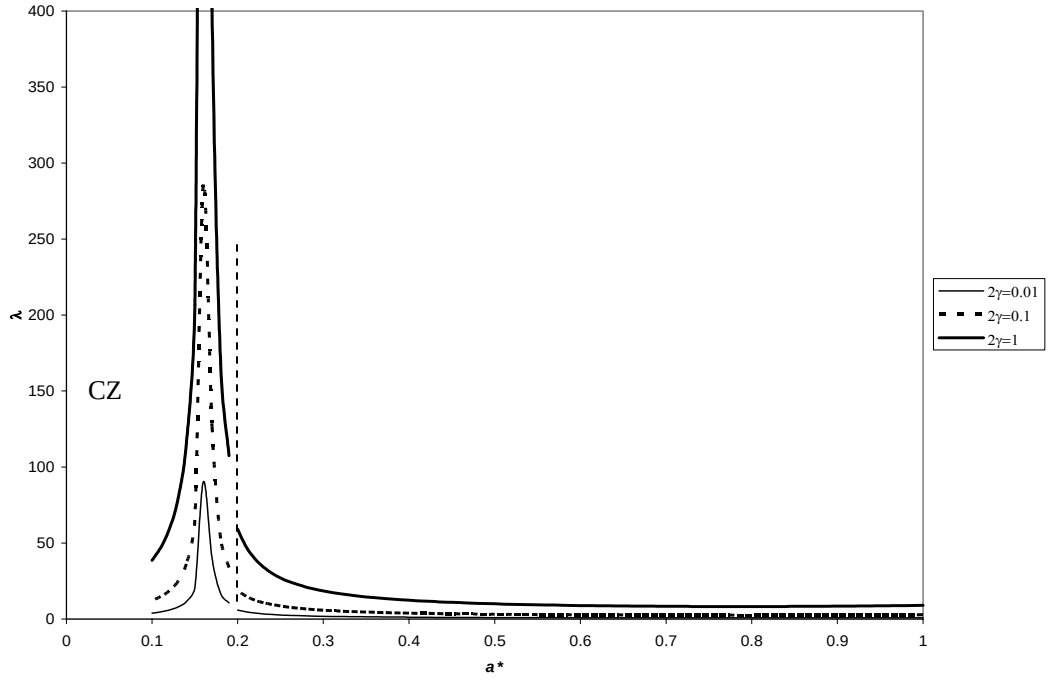
(c)

Figure 26. Delamination growth paths and stiffness degradation for clamped-fixed ends without applied thermal load, 2γ = 0.1 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

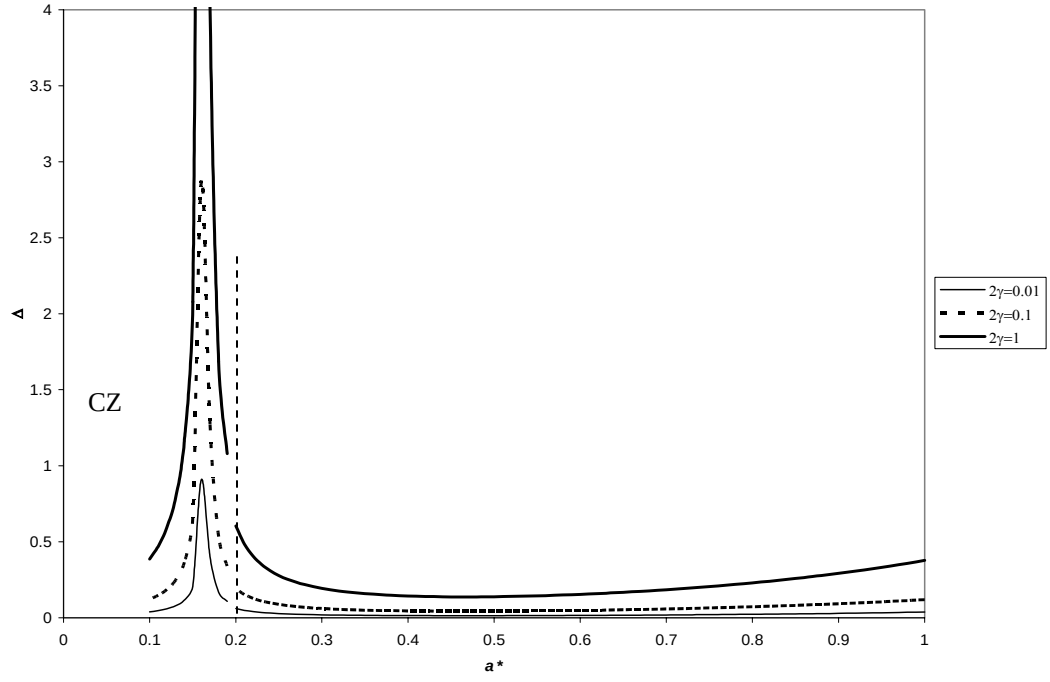
It is to be assumed that no contact zone is present on the curves unless indicated by a "CZ." The paths are separated by gaps to represent at which point along the evolving

structure the portion of the patch in contact with the baseplate lifts off. Once again these paths are plotted for the case when $2\gamma = 0.1$. We observe a shift to the left in the peaks of the delamination growth paths, to the point that there is no peak in the domain for $E_0 = 10$. Growth is stable until reaching the peak, after which it will be unstable. Stable growth is seen to be recovered for a range of initial flaw sizes after the peak in the displacement controlled test. It is seen that, like the other cases considered to this point, the more compliant structure has a lower threshold for delamination growth based on the loading parameter and characteristic deflection. Once again, the global stiffness degradation can be interpreted in an identical fashion to Figure 9c.

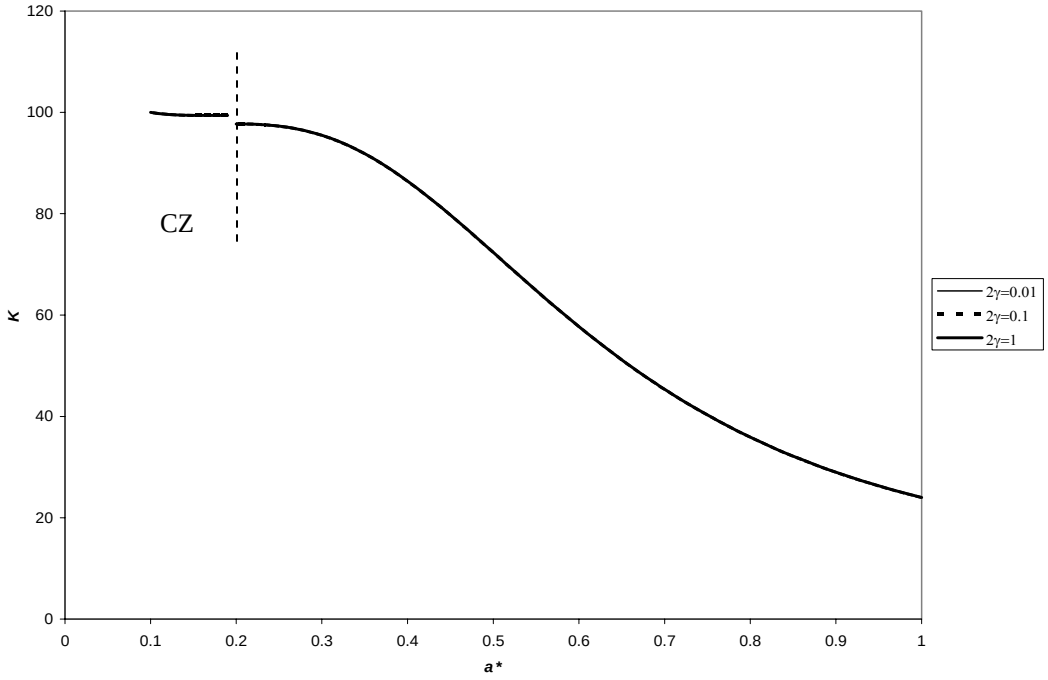
Figure 27 shows the comparison of delamination growth paths for different bond strengths, again in the absence of a temperature field. The contact zone is present for [0.1, 0.19] for this case. All points on the plots to the left of the dotted line indicate where the contact zone is present. The bond strength does not affect the existence of the contact zone, for this linear solution.


400
350
300
250
200
150
100
50
0
λ
CZ
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a*
2γ=0.01
2γ=0.1
2γ=1

(a)


4
3.5
3
2.5
2
1.5
1
0.5
0
Δ
CZ
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a*
2γ=0.01
2γ=0.1
2γ=1

(b)



(c)

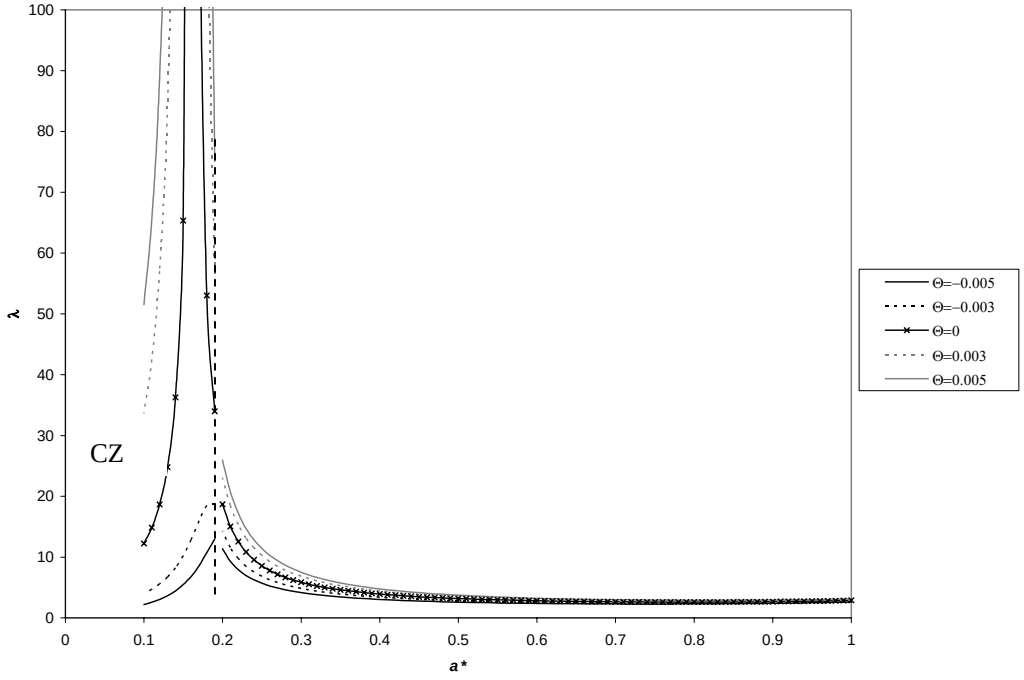
Figure 27. Delamination growth paths and stiffness degradation for clamped-fixed ends without applied thermal load, E_0 = 1 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

From Figures 27a and 27b, the effect of increasing the bond strength is to increase the threshold of delamination growth. This is expected, as is the path in Figure 27c. Although the bond strength does not affect the global stiffness, it has been included in Figure 27 to highlight the effect of the contact zone. The contact zone is evident by the jump in the stiffness plot at $a^* = 0.19$ in Figure 27c. It makes the structure stiffer in that region. This is because the plates are touching for a greater range, which clearly would be stiffer than the baseplate alone.

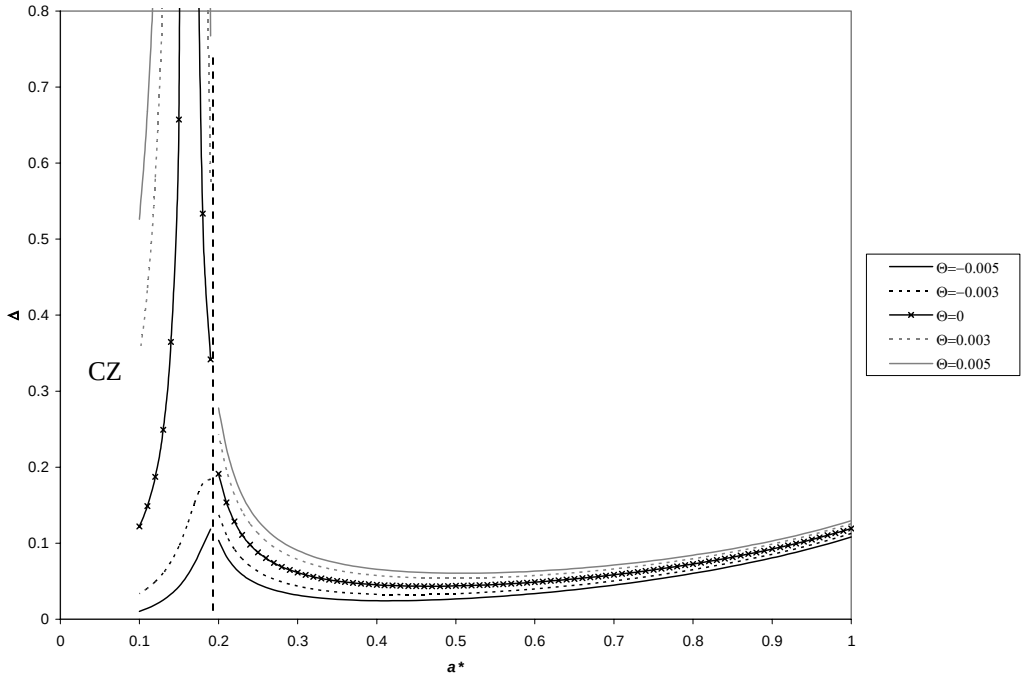
Effect of Uniform Temperature Field

The following series of plots explores the effect of a uniform temperature field in addition to the transverse pressure. When $\alpha_p = 0.5$, a contact zone is present on the domain [0, 0.19] on a^*, for all values of Θ. This seemingly suggests that the existence

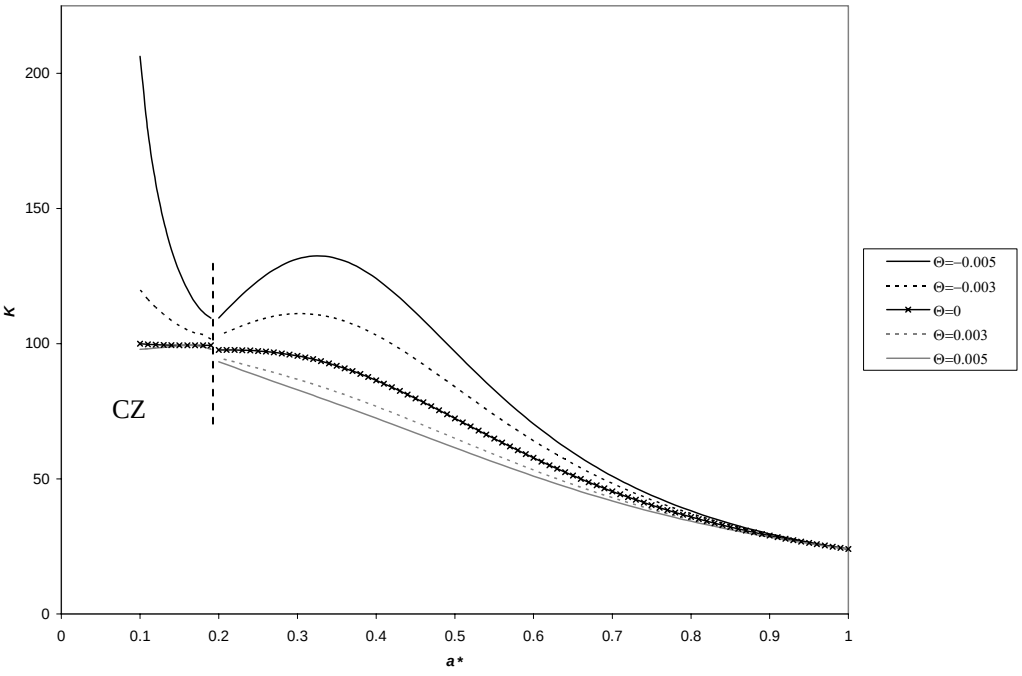
of the contact zone is independent of the temperature; however, in the following section we will see that this is not the case. As usual, for Figure 28, we consider structures for which $E_0 = 1$ and $2\gamma = 0.1$. Again, the contact zone is present on the left of the dotted line. As delamination grows, the patch lifts off the baseplate, to the point where it no longer touches at all.



(a)



(b)



(c)

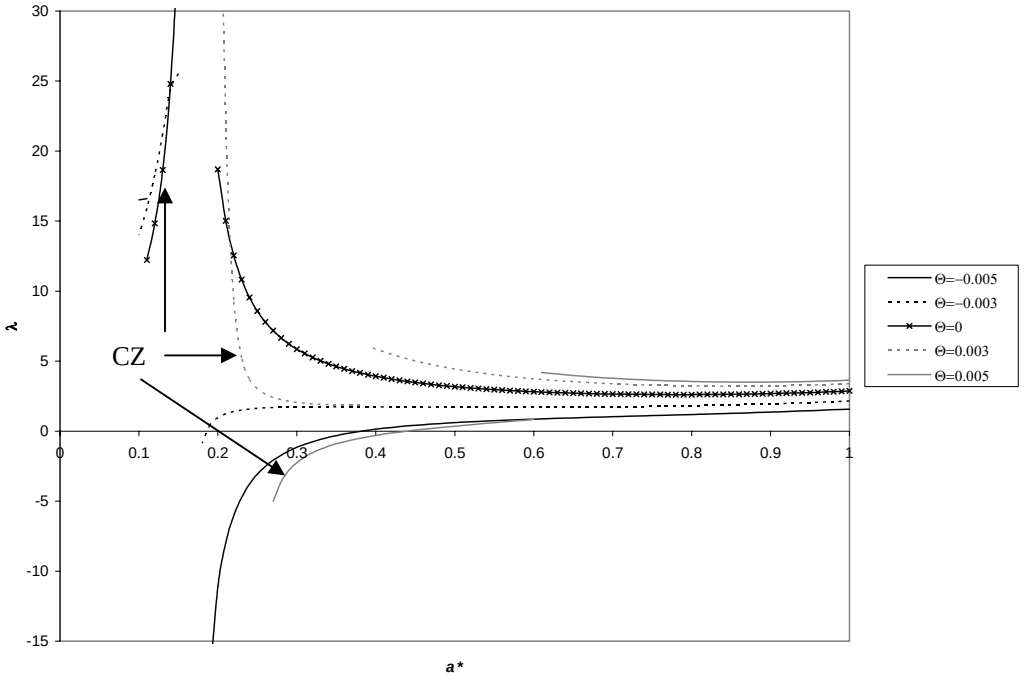
Figure 28. Delamination growth paths and stiffness degradation for clamped-fixed ends with applied thermal load, $E_0 = 1$, $2\gamma = 0.1$, $\alpha_p = 0.5$ a) λ vs. a*, b) Δ vs. a*, c) K vs. a*

There are peaks on the delamination growth paths in Figures 28a and 28b that increase dramatically with increasingly positive temperature fields. For the loading controlled

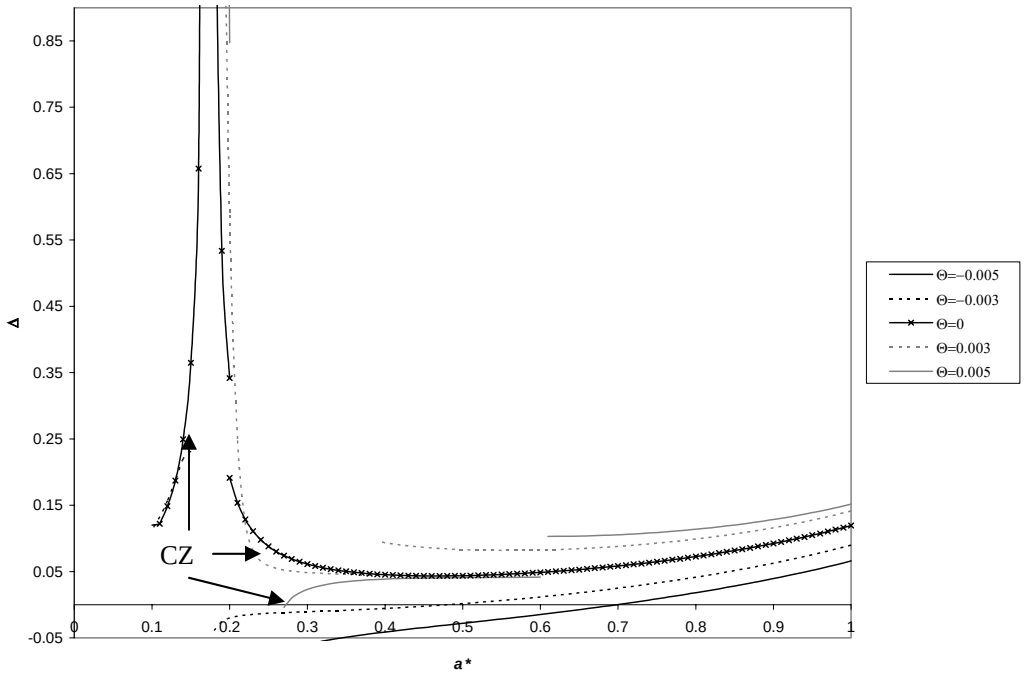
test, it is seen that for initial flaw sizes greater than about 0.3, all temperatures increase the threshold of delamination growth. For the displacement controlled test, the more negative temperature fields decrease the thresholds, whereas the positive increase them. The specimens experiencing a temperature field all approach the same limit for stiffness as the bond zone gets smaller.

Figure 29 presents the opposite case of Figure 28, when $\alpha_p = 2$. The presence of a contact zone differs vastly from that of the previous case. No contact zone exists when the temperature fields are negative, and the domain increases with increasing positive temperature fields. Recall that the contact zone is present for the domain [0, 0.19] on a^* in the pure pressure case. When $\Theta = 0.003$, the domain of contact is [0, 0.38], and when $\Theta = 0.005$, the domain of contact is [0, 0.6]. The positive temperature values cause the structure to deflect upward while rendering it more compliant, thus producing a curvature in region S_2 that is conducive to a contact zone.

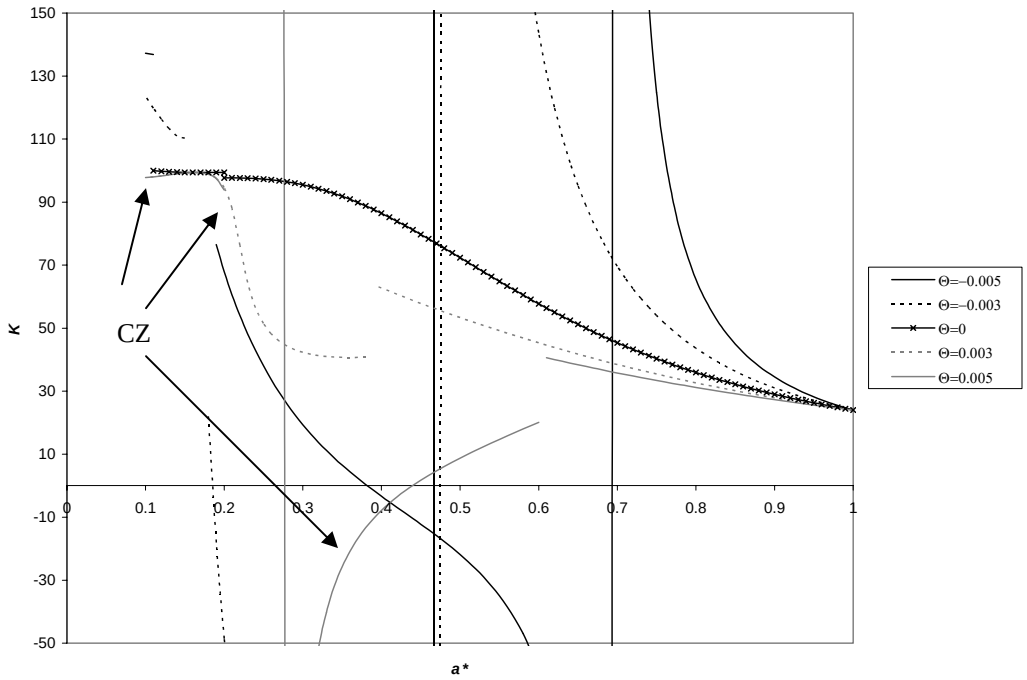
The effects of the contact zone are apparent on the delamination growth paths. There are large discontinuities, as the presence of the contact zone lowers the delamination growth thresholds significantly. The portions of the plot marked “CZ” indicate a contact zone occurs there. We see for the case of $\alpha_p = 2$, the linear model appears to break down for portions of the delamination growth paths. This could possibly mean arrest of all growth in those regions.


30
25
20
15
10
5
0
-5
-10
-15
λ
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a*
CZ
Θ=−0.005
Θ=−0.003
Θ=0
Θ=0.003
Θ=0.005

(a)


0.85
0.75
0.65
0.55
0.45
0.35
0.25
0.15
0.05
-0.05
Δ
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a*
CZ
Θ=−0.005
Θ=−0.003
Θ=0
Θ=0.003
Θ=0.005

(b)



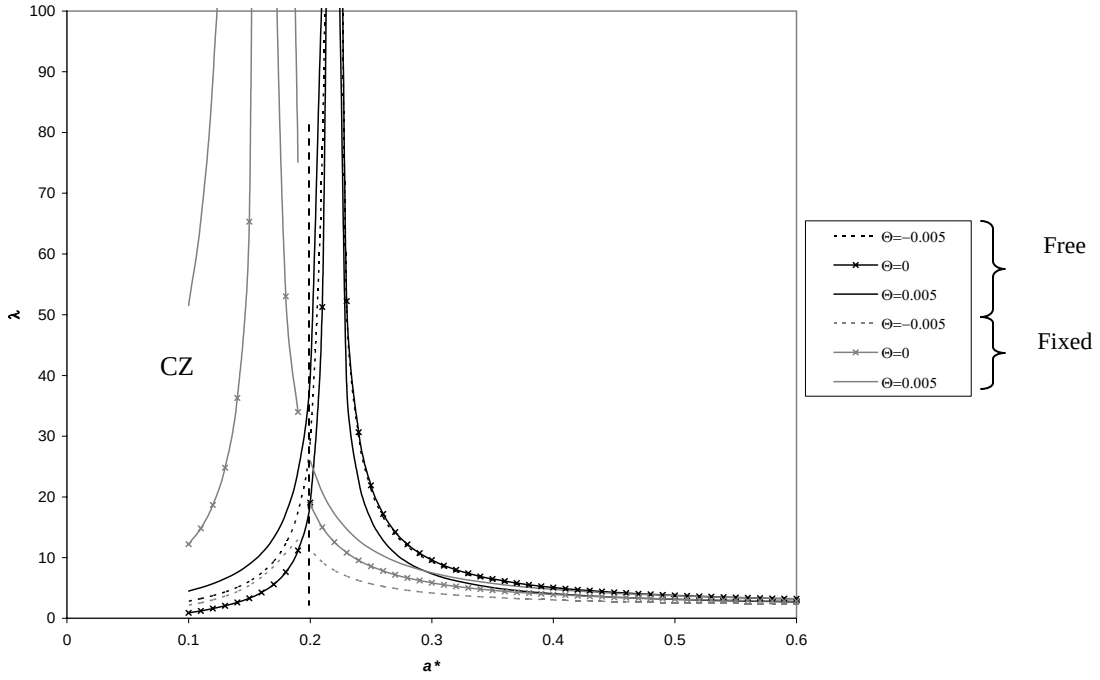
(c)

Figure 29. Delamination growth paths and stiffness degradation for clamped-fixed ends with applied thermal load, $E_0 = 1$, $2\gamma = 0.1$, $\alpha_p = 2$ a) λ vs. a*, b) Δ vs. a*, c) K vs. a*

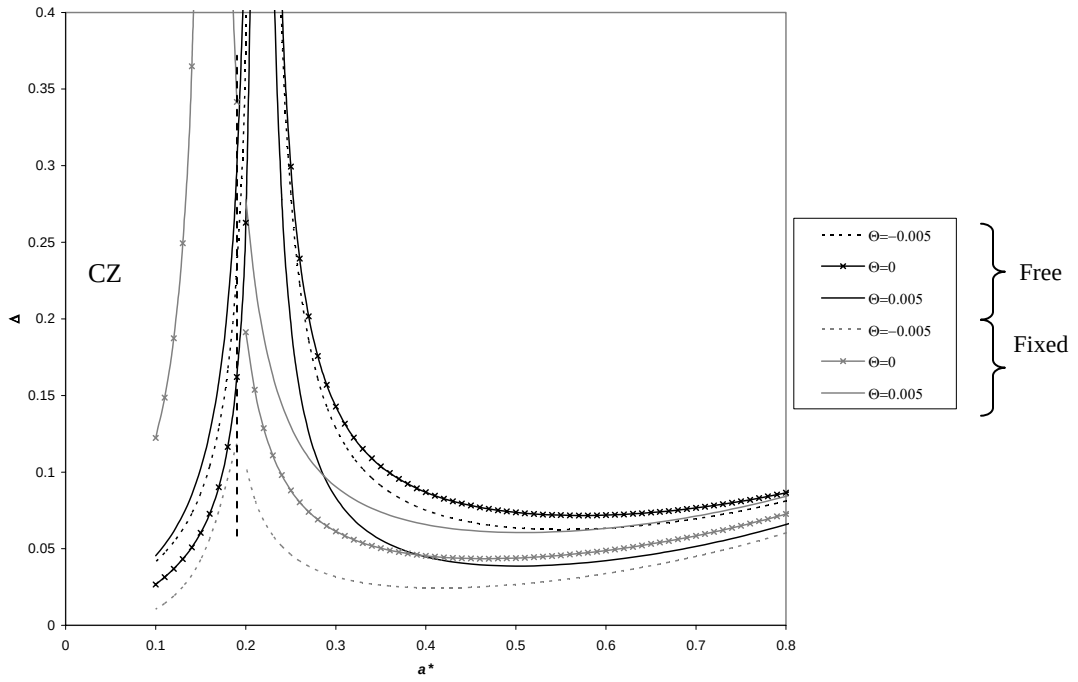
In Figure 29a, flaw sizes greater than 0.6, the more positive thermal loads decrease the threshold of delamination growth. This trend is halted by the presence of the contact zone for the positive values of Θ. We also see more negative values of the loading parameter on the delamination path. This again suggests that delamination growth occurs solely due to the thermal load, in the absence of pressure. In Figure 29b, the displacement controlled test, the opposite effect is observed post $a^* = 0.6$. The asymptotes in the stiffness plot are once again due to the change of sign in the deflection plots. We also observe that the delamination growth paths approach the same stiffness limit as the patch size decreases.

Figure 30 shows the effect that the end supports have on the system. It is a side-by-side look at the delamination paths of clamped-free ends and those of clamped-fixed

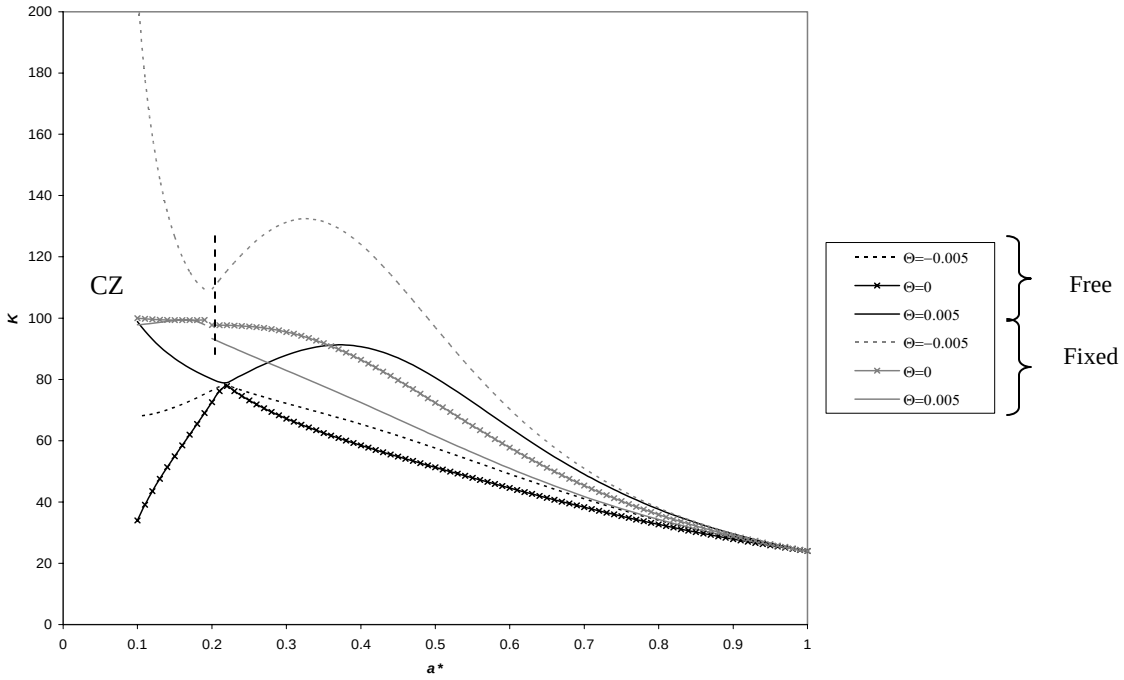
ends. These are plotted for $\alpha_p = 0.5$, $E_0 = 1$, and $2\gamma = 0.1$. A contact zone is present for the fixed end case to the left of the dotted line.



(a)



(b)

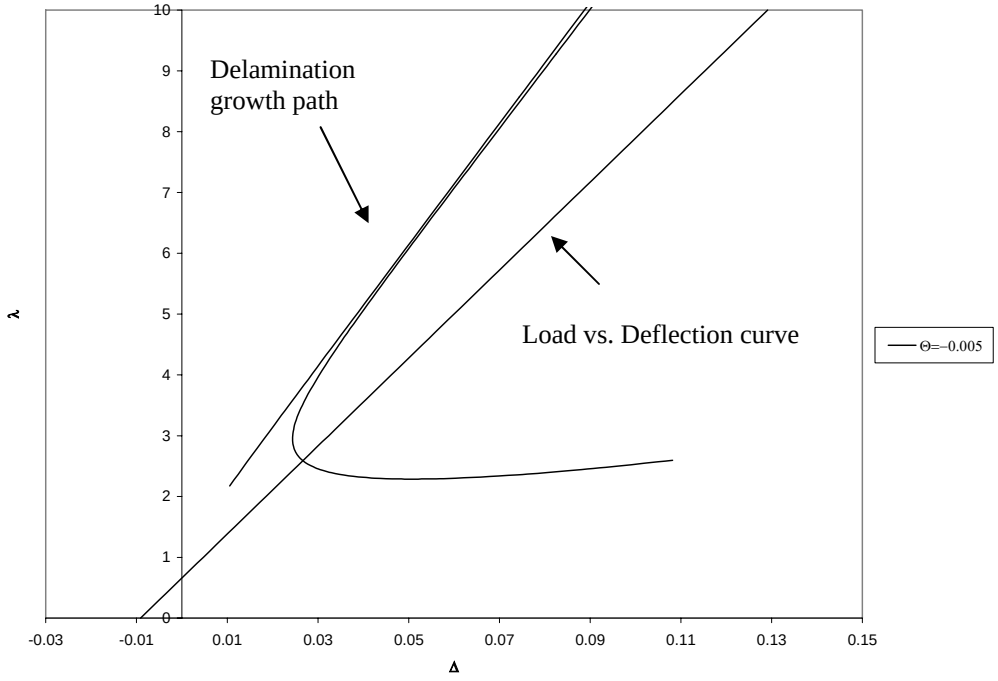


(c)

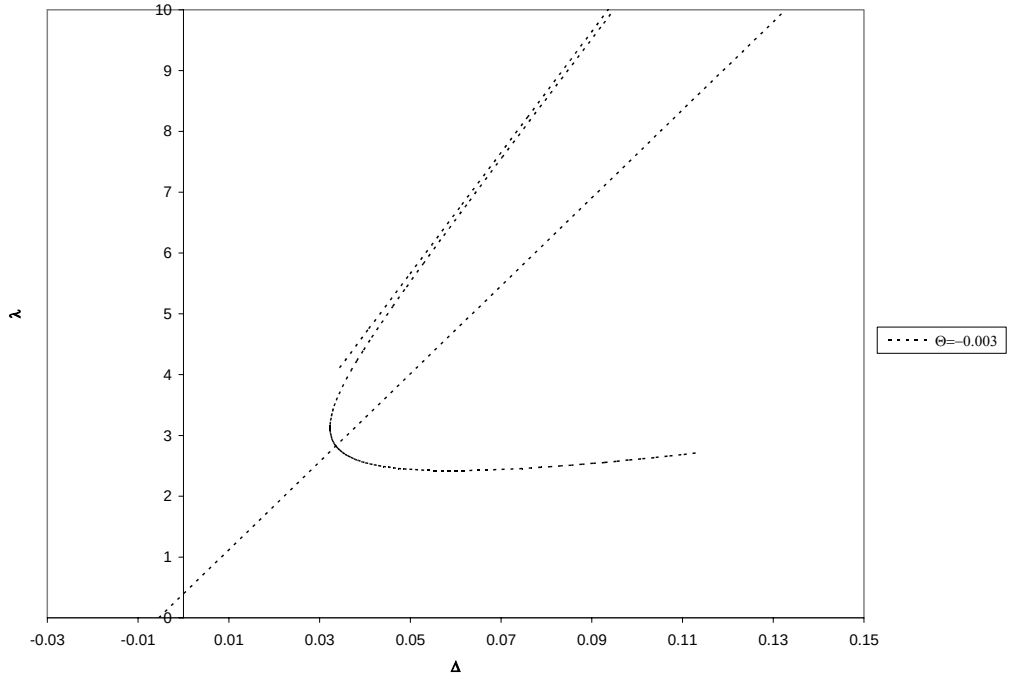
Figure 30. Comparison of delamination growth paths and stiffness degradation for clamped-free and clamped-fixed ends, where E_0=1, 2γ=0.1, and α_p=0.5 a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

The effect of the fixed ends with a contact zone is to shift the peaks in the path to the left. This means that catastrophic delamination growth occurs for larger bonded portions of the patch. We also see that fixing the ends effectively lowers the thresholds for delamination growth beyond approximately a_0*= 0.2, with the exception of the more positive thermal load. Temperature differences have the reverse effect on the structures, like the hinged case. This is because temperature differences induce opposite deflections for free and fixed ends. The stiffness is increased by fixing the ends; however, the thermal effect for the positive temperature field actually causes the fixed end specimen to be less stiff than the free end specimen.

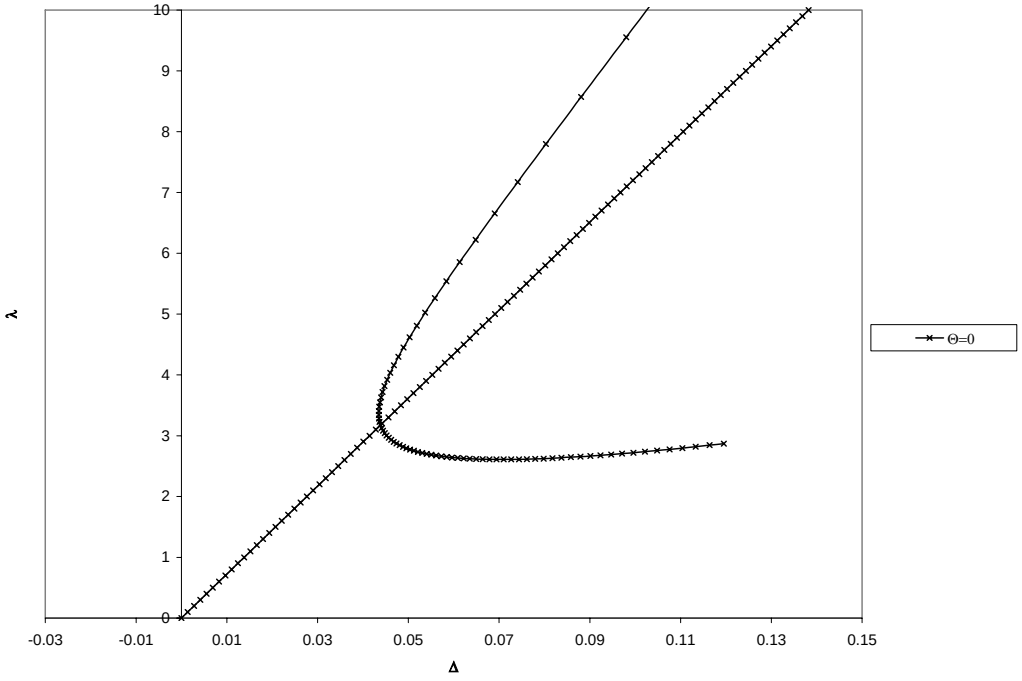
Figure 31, shows the load vs. deflection curve and delamination growth paths for the temperature fields used in Figure 28. The initial flaw size is taken to be a_0*=0.5. All parameters are the same as in Figure 28.


Delamination
growth path
Load vs. Deflection curve
λ
Δ
Θ=-0.005

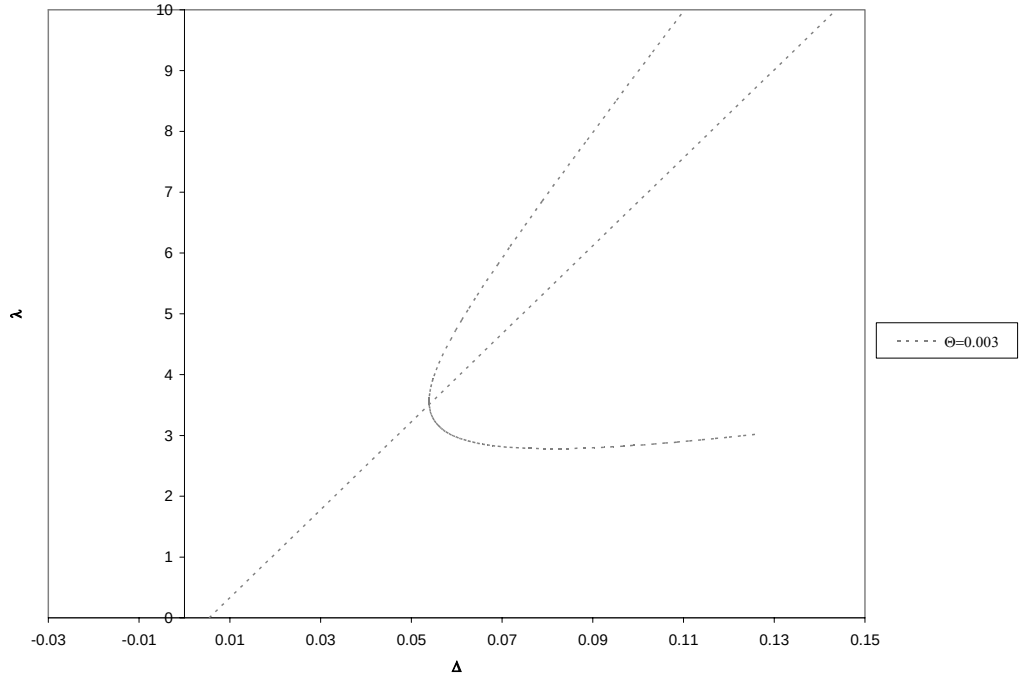
(a)


λ
Δ
Θ=-0.003

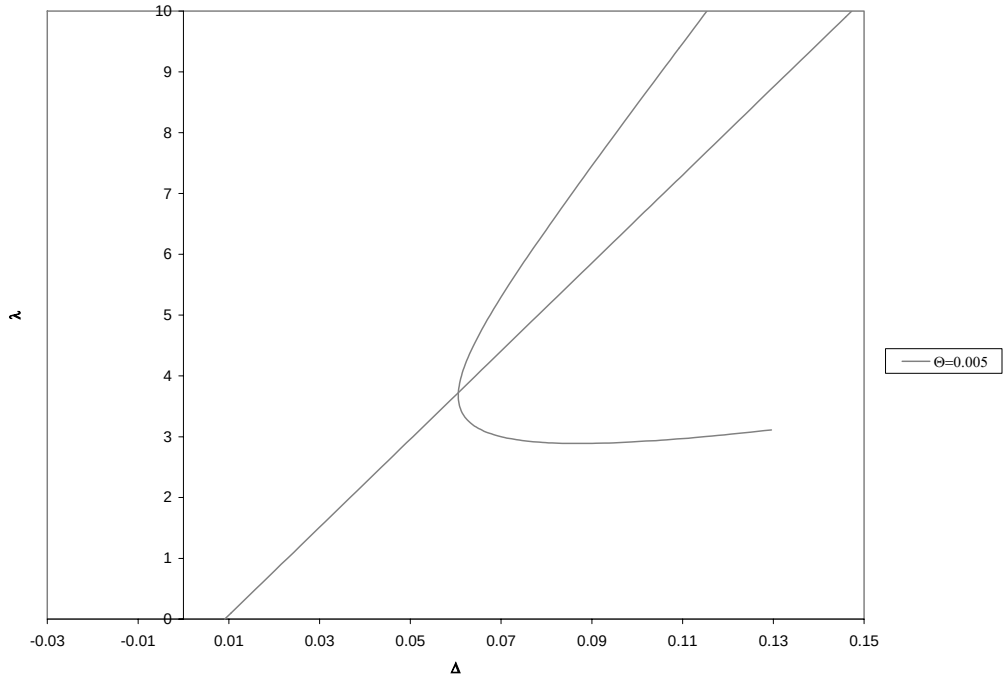
(b)


λ
Δ
Θ=0
10
9
8
7
6
5
4
3
2
1
0
-0.03
-0.01
0.01
0.03
0.05
0.07
0.09
0.11
0.13
0.15

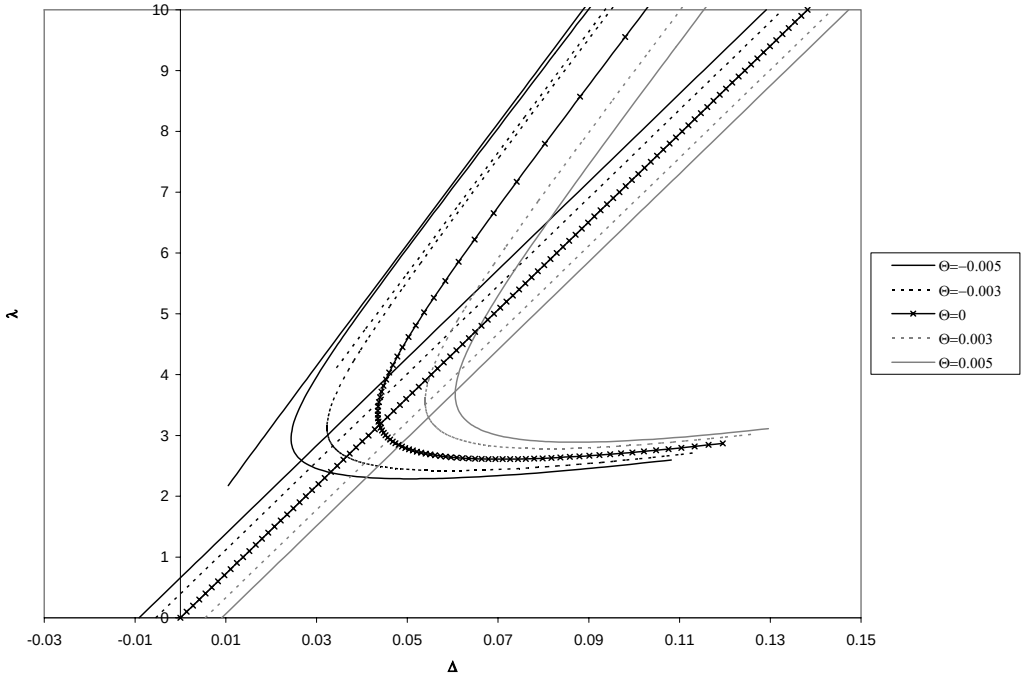
(c)


λ
Δ
Θ=0.003
10
9
8
7
6
5
4
3
2
1
0
-0.03
-0.01
0.01
0.03
0.05
0.07
0.09
0.11
0.13
0.15

(d)



(e)



(f)

Figure 31. Load vs. deflection curves and corresponding delamination growth paths for clamped fixed ends, E_0 = 1, 2γ = 0.1, α_p = 0.5, a) Θ=-0.005, b) Θ =-0.003, c) Θ = 0, d) Θ =0.003, e) Θ =0.005, f) summary

The plots contained in Figure 31 show the typical loading curves as they increase, and ultimately, intersect the delamination growth paths. The interpretation follows that of Figure 13. The directions in which the structure begins to deflect before the transverse pressure is applied, is the same as in the hinged-fixed case, which is the reverse of either free end case.

4.5 Concluding Remarks

We see that various parameters can affect the delamination behavior of the patched plate structure. In the absence of a temperature field, structures that are stiffer (i.e. have a high modulus ratio) experience delamination propagation much more easily than more compliant structures. Bond strength of course affects delamination growth as well, in a manner that could be expected. A temperature difference has a profound impact on the delamination behavior, which also depends on the support conditions. We also see that including the possibility of a contact zone can alter behavior. A contact zone tends to make a structure stiffer, which will affect the thresholds for delamination growth. A contact zone may be present for the hinged-fixed case, when the deflection is opposite to the sense of the applied pressure. However, our preliminary examination of this case is inconclusive, and requires further investigation. Current results assume the contact zone is either full, or nonexistent. Other configurations may be possible in the presence of a temperature field.

In the following chapter we present the results obtained from the nonlinear analysis. These will then be compared to the linear results.

Chapter 5

Results Based on the Geometrically Nonlinear Model

5.1 Preliminary

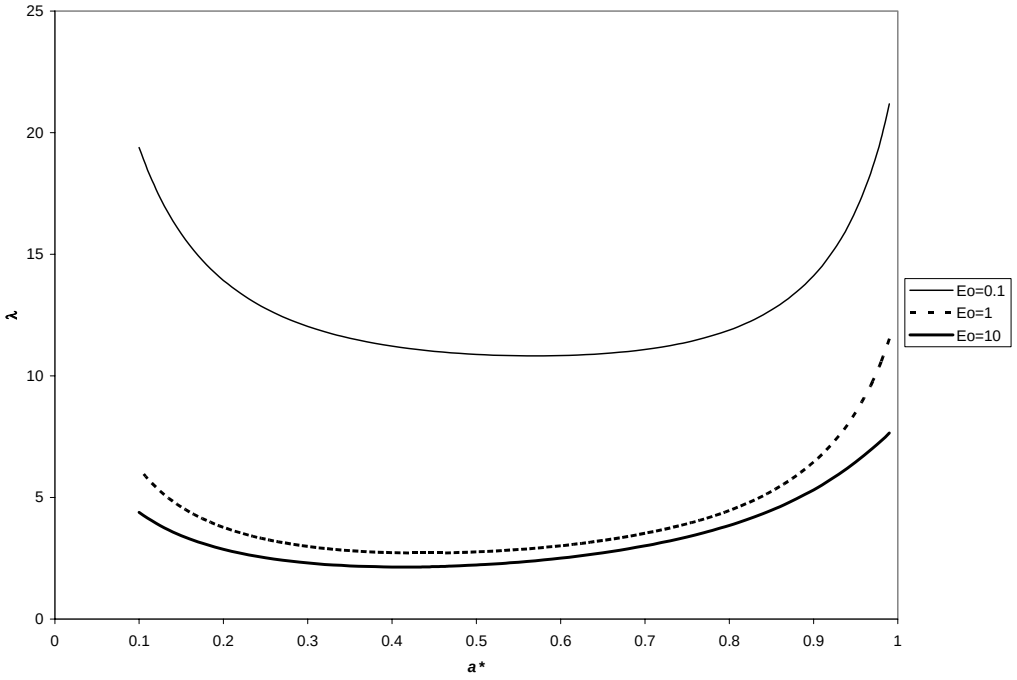
Having established the linear response of the system, we now turn out attention to the nonlinear response. The nonlinear response should provide a more complete picture of the delamination behavior of the patched plates. We may then compare and make inferences on the results based on the linear and geometrically nonlinear models. Recall that the nonlinear response is obtained by way of retaining the $(Nw')'$ terms in Eqs. (56a)-(56d). This leads to the solution given by Eqs. (76a)-(76d). Since the nonlinear effect enters the formulation via the membrane force term in the governing equations, and since $N_0 = 0$ when the ends are free, there is no effect of nonlinearities on the response. In this section we will detail the cases for hinged-fixed and clamped-fixed ends.

5.2 Results

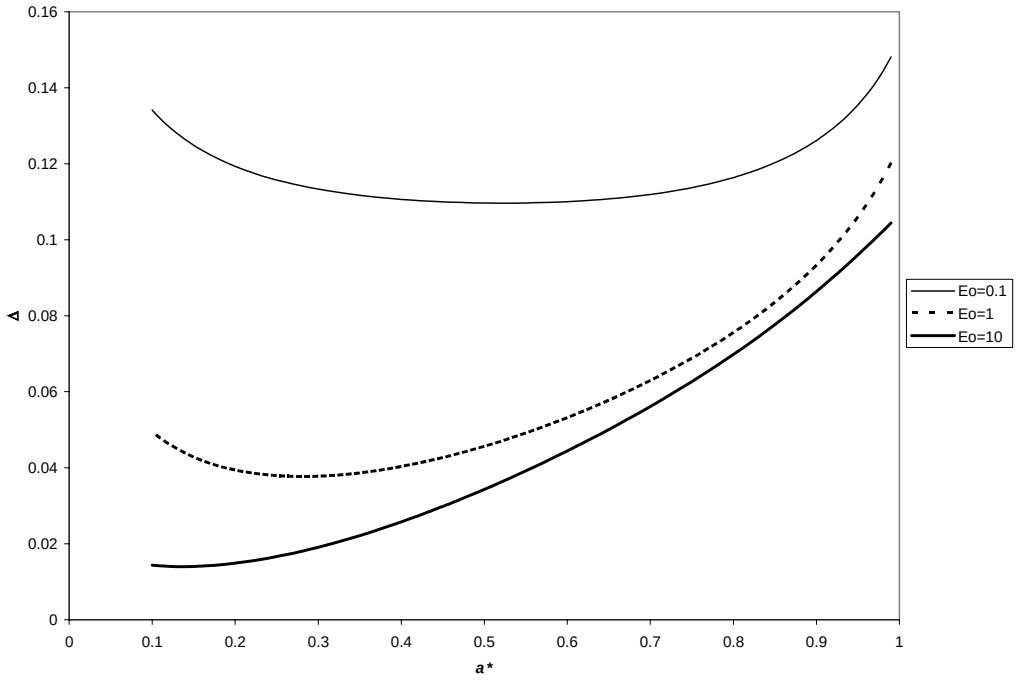
5.2.1 Hinged-Fixed End Supports

Pure Pressure Loading, $\Theta = 0$

We begin by examining the case of hinged-fixed ends. As mentioned in Chapter 4, we expect no contact zone, unless the deflection should be down. Figure 32 shows the comparison of the delamination growth paths plotted for various values of the elastic modulus ratio, E_0. There is no uniform thermal load present, and $2\gamma = 0.1$.



(a)



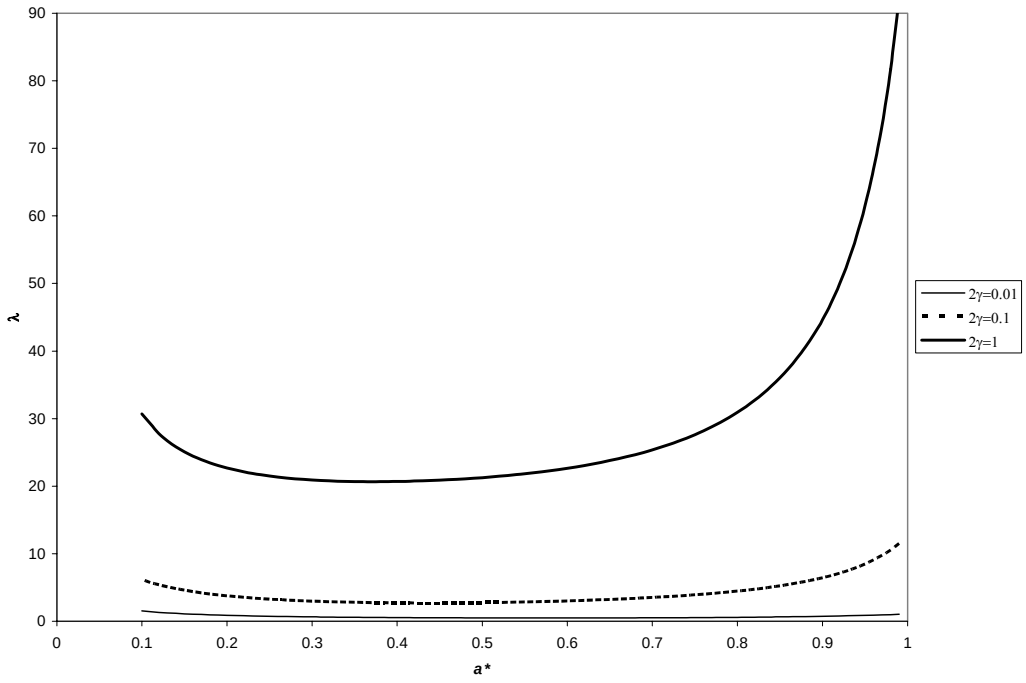
(b)

Figure 32. Nonlinear delamination growth paths for hinged-fixed ends without applied thermal load, $2\gamma = 0.1$ a) λ vs. a*, b) Δ vs. a*

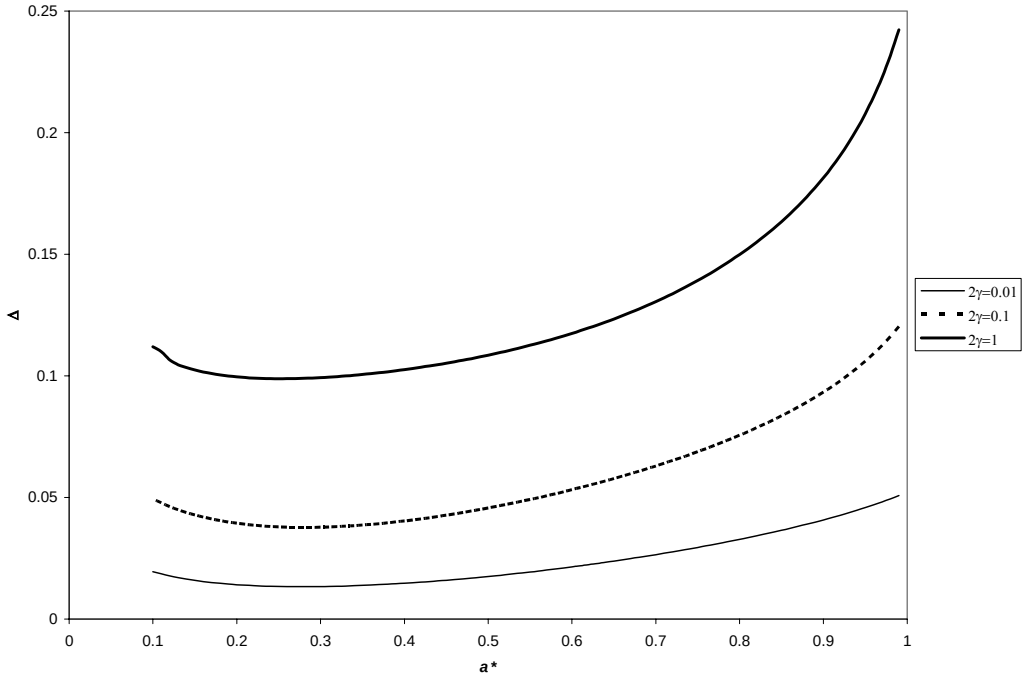
Like the previously covered linear cases, we see that in Figures 32a and 32b, the effect of decreasing the order of magnitude of the modulus ratio is to raise the threshold for

delamination growth. For the load controlled specimen, we see that the nonlinear model reveals stable debonding as the flaw size increases. The nonlinear model introduces bending-stretching coupling under compressive loading, which tends to have a stabilizing effect. This effect was not observed with the linear model. Stable debonding is also observed for the displacement controlled test, Figure 32b. This effect was predicted by the linear model. It begins at smaller flaw sizes than the for the load controlled test.

Figure 33 is a plot of the delamination growth paths for varying orders of magnitude on the bond strength, 2γ. For this figure, $E_0 = 1$, and there is no temperature field.



(a)



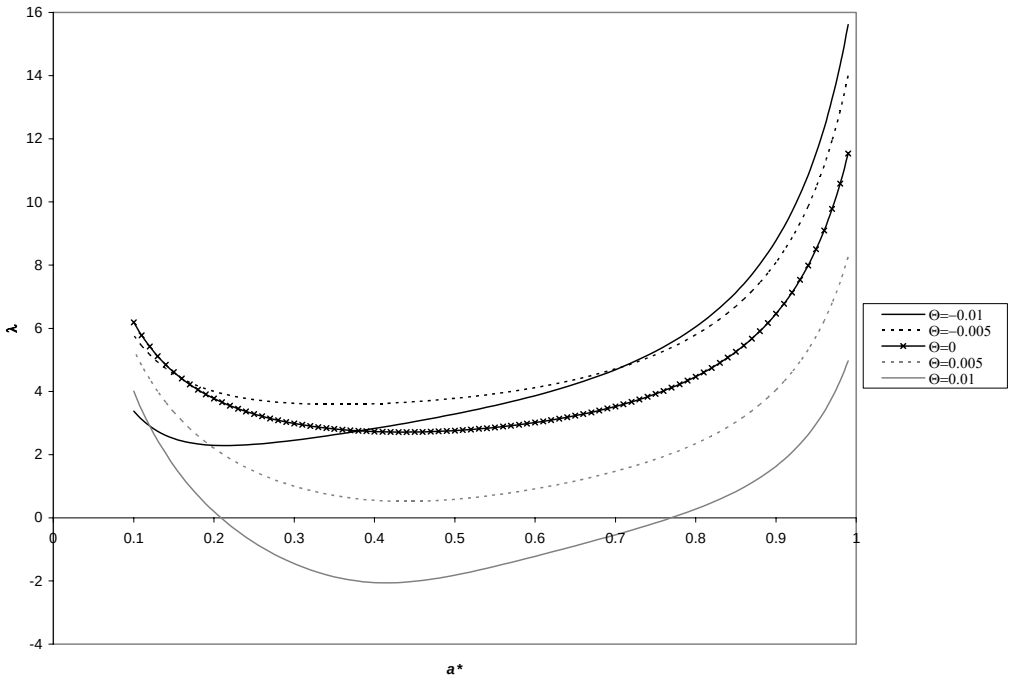
(b)

Figure 33. Nonlinear delamination growth paths for hinged-fixed ends without applied thermal load, E_0 = 1 a) λ vs. a*, b) Δ vs. a*

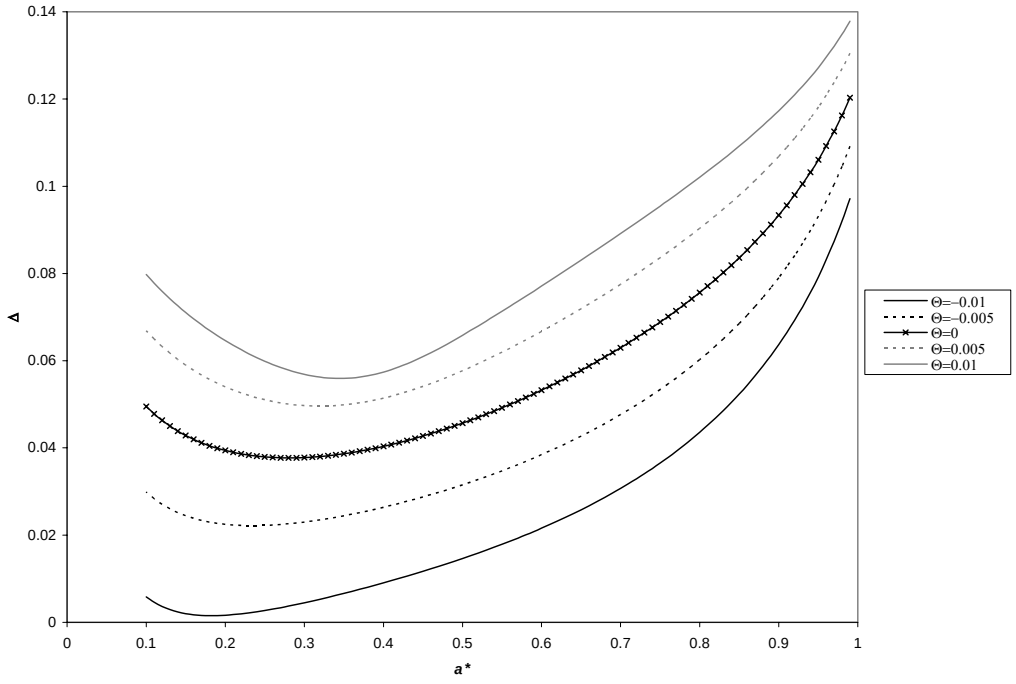
The trend observed is the expected; increasing the bond strength increases the delamination thresholds. The nonlinear model once again reveals a stability for the load controlled test (Figure 33a) as in the previous plot. The delamination growth becomes stable as the flaw size is increased for all plots.

Effect of Uniform Temperature Field

We turn our attention now to the effect of a temperature change on delamination growth due to transverse pressure. Figure 34 shows the delamination growth paths for the parameters $E_0 = 1$, $2\gamma = 0.1$, and $\alpha_p = 0.5$. This represents the case when the baseplate has twice the coefficient of thermal expansion to that of the patch.



(a)



(b)

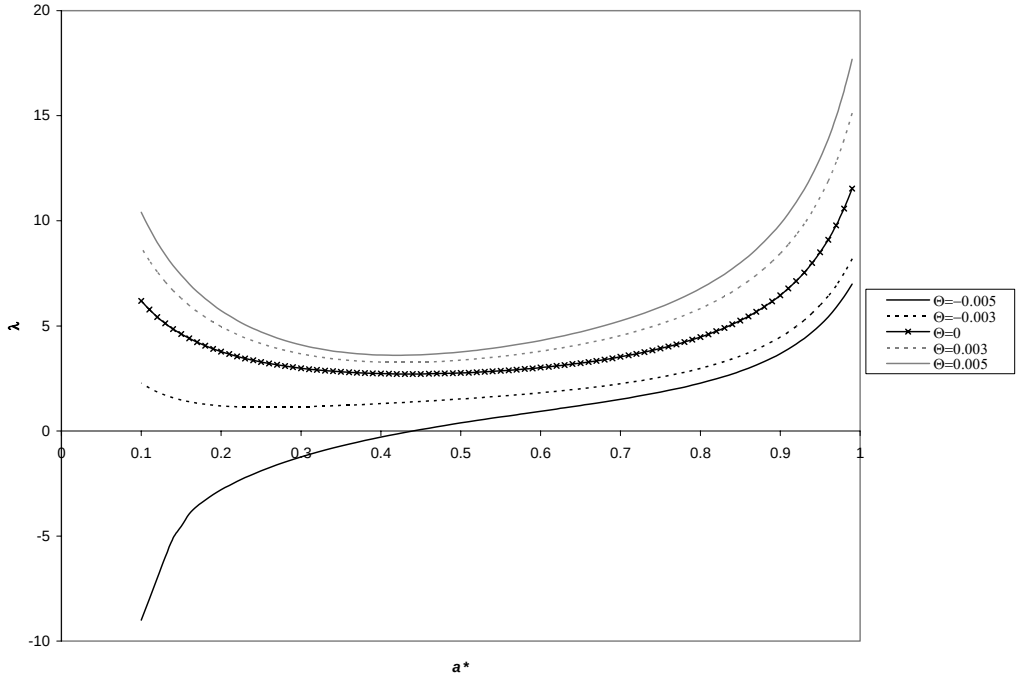
Figure 34. Nonlinear delamination growth paths for hinged-fixed ends with applied thermal load, E_0 = 1, 2γ = 0.1, α_p = 0.5 a) λ vs. a*, b) Δ vs. a*

Figure 34a shows that heating the structure lowers the delamination threshold, for initial flaw sizes greater than approximately 0.7. For $a^* < 0.7$, we see that the delamination

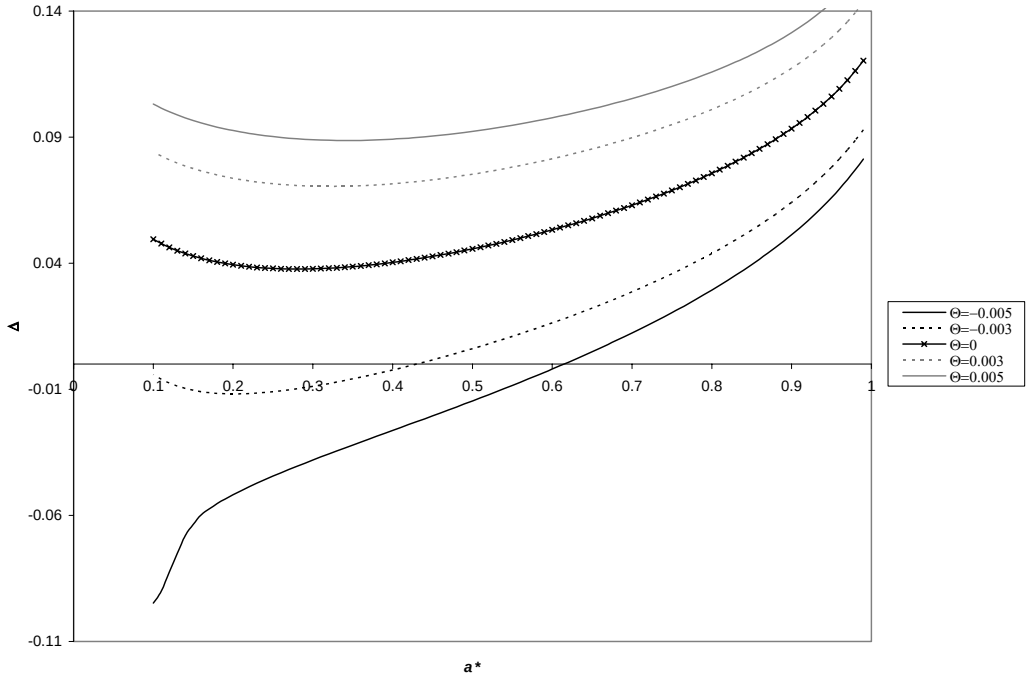
path for $\Theta = 0.01$ actually begins to fall below the other paths. In this case, positive temperatures cause the baseplate to expand, which results in a compressive axial membrane force throughout the structure. In the bond zone, this membrane force is coupled with bending at the reference surface. We also in observe in Figure 34a the phenomenon of the delamination growth path for the force controlled loading falling below zero. This case corresponds to a situation where the thermal load is so extreme that the structure will delaminate due to the temperature difference alone, in the complete absence of pressure.

Figure 34b shows that cooling the structure lowers the delamination threshold, and vice versa. A negative temperature difference makes the baseplate contract, which results in a tensile force and opposite sense of deflection. According to our results however, the downward deflection induced by thermal stress is not enough to overcome the upward deflection induced by pressure. The structure is seen to deflect up for all scenarios in Figure 34b. The deflection of the structure due to negative temperature fields competes with the effect of pressure, and hence delamination propagation occurs at a much lower threshold. The structure is essentially less compliant. The positive temperature fields cause a deflection which enhances the pressure effect, rendering the structure more compliant, and thus requiring higher loads for propagation. The thermal stresses induced by the increasing (positive or negative) temperature fields have a profound impact on the delamination paths, and hence the behavior of the structure. This suggests that a separate study solely on the effects due to temperature is warranted for this type of problem.

Following the plots of Figure 34, we examine the reverse case ($\alpha_p = 2$) in Figure 35. This corresponds to the patch having twice the thermal expansion coefficient to that of the baseplate. The other parameters are taken as in Figure 34.



(a)

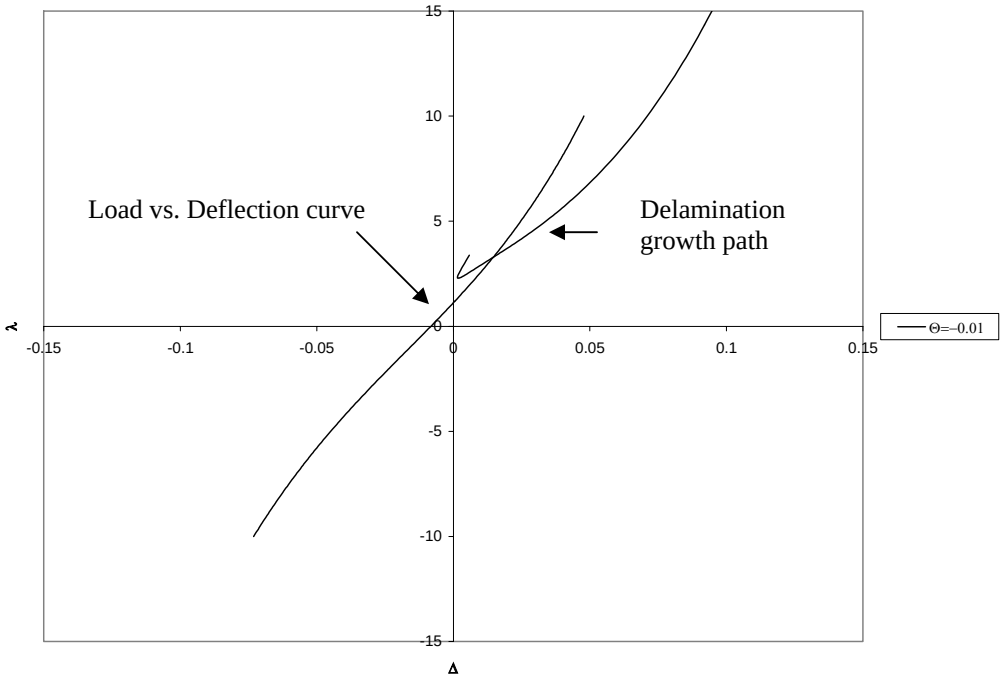


(b)

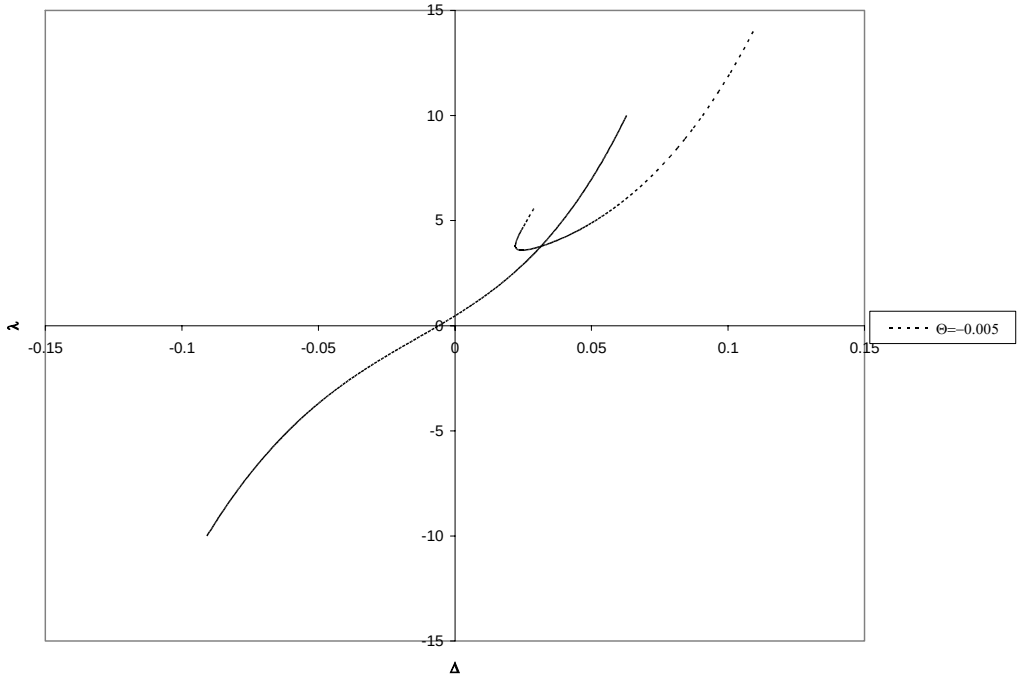
Figure 35. Nonlinear delamination growth paths for hinged-fixed ends with applied thermal load, E_0 = 1, 2γ = 0.1, α_p = 2 a) λ vs. a*, b) Δ vs. a*

In Figure 35a, we see the reverse trend from Figure 34a. Cooling the structure tends to lower the delamination thresholds, and vice versa. The most negative value of the thermal load once again, follows a path not like the others, suggesting that this is the effect of the thermal stress induced by a cooling temperature. The path extends below the zero axis, again showing that a temperature difference alone is enough to induce delamination propagation. The other paths show unstable debonding followed by stable, again a deviation from the linear model for the load controlled test. The displacement controlled test (Figure 35b) shows the same trends as established in Figure 34b, however, it also suggests that a contact zone may be present when the deflection is negative. Like the linear case presented in Chapter 4, this phenomena requires additional research.

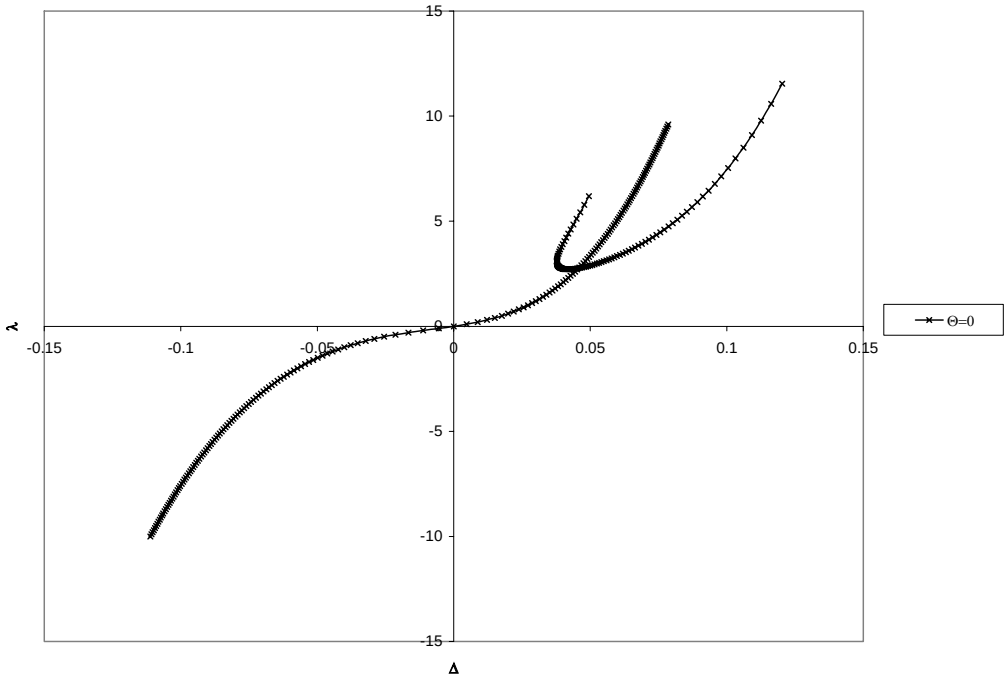
Figure 36 shows the regular nonlinear quasi-static loading curves of such a structure superimposed onto the delamination growth paths for the system.


Load vs. Deflection curve
Delamination growth path
15
10
5
0
-5
-10
-15
-0.15
-0.1
-0.05
0.05
0.1
0.15
λ
Δ
Θ=–0.01

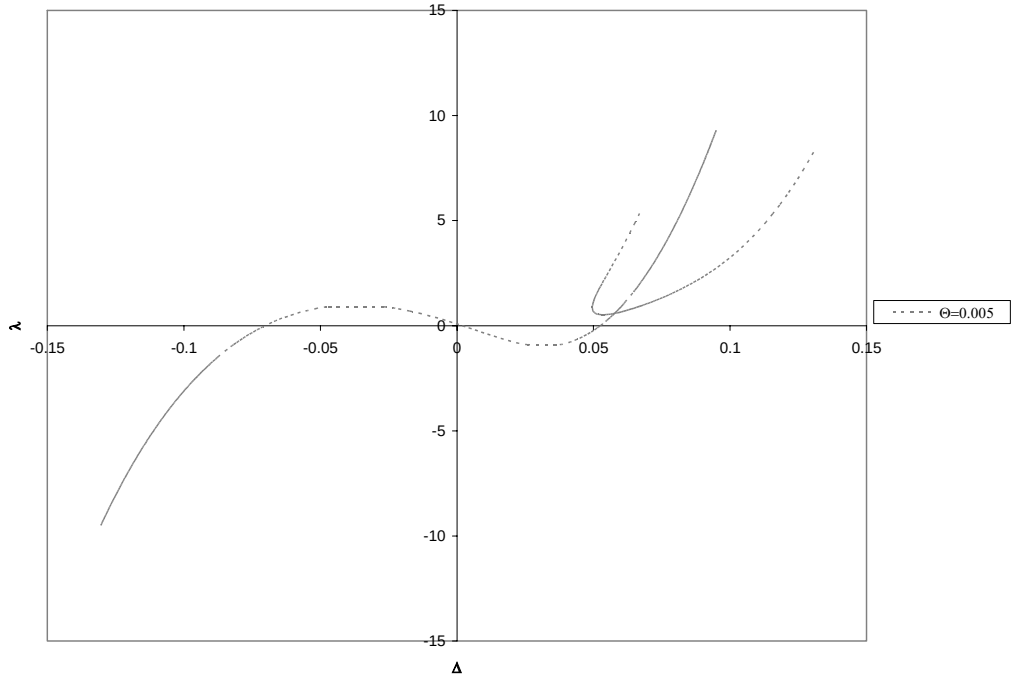
(a)


15
10
5
0
-5
-10
-15
-0.15
-0.1
-0.05
0.05
0.1
0.15
λ
Δ
Θ=–0.005

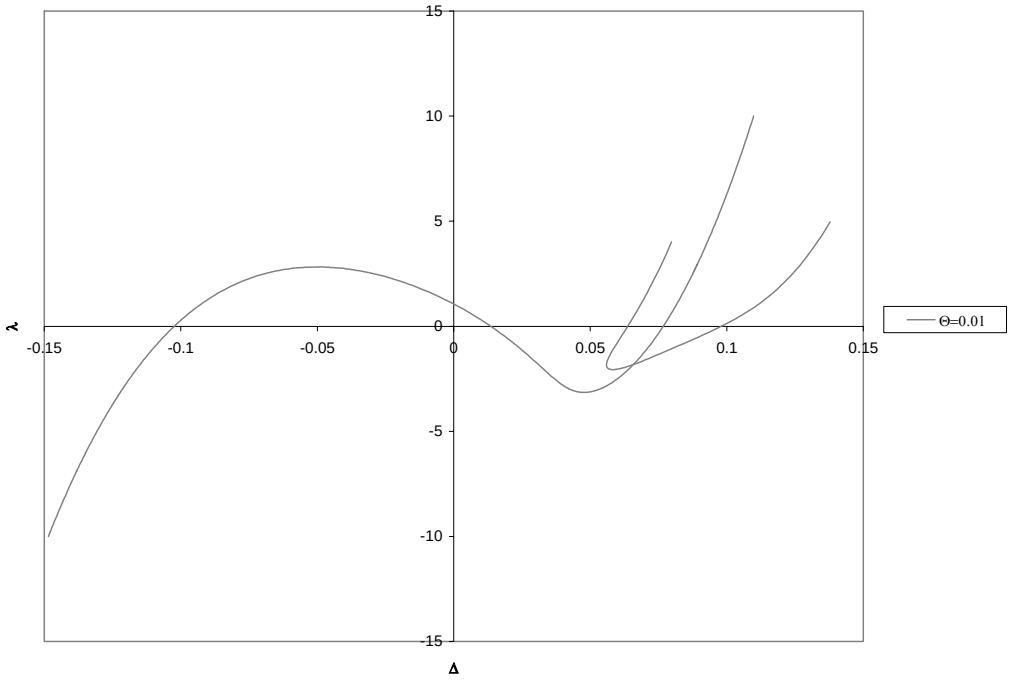
(b)


15
10
5
0
-5
-10
-15
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
λ
Δ
Θ=0

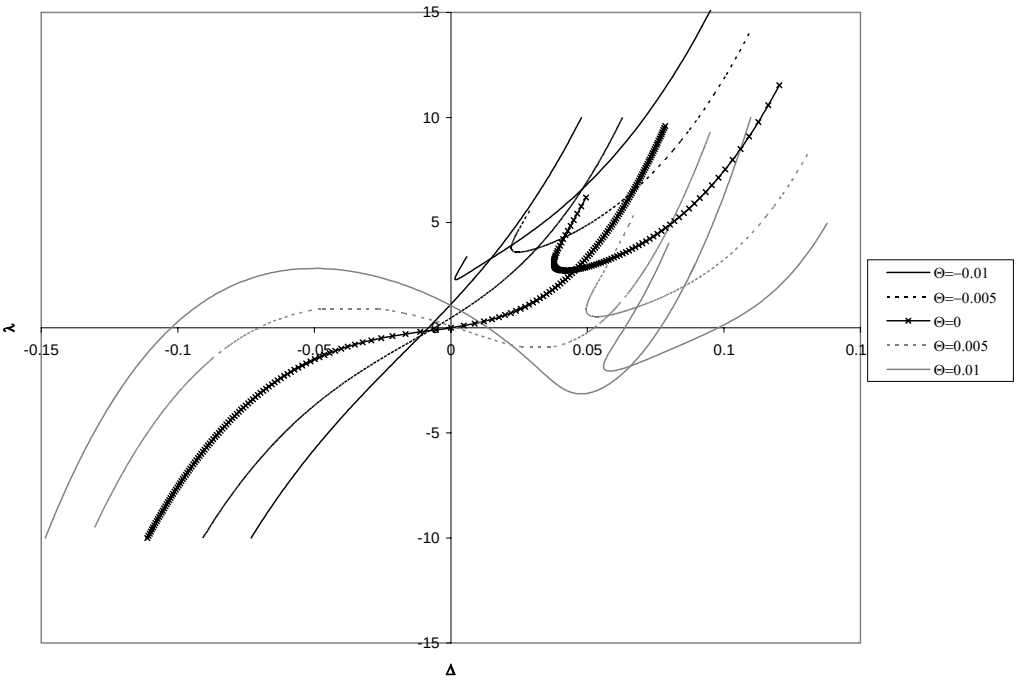
(c)


15
10
5
0
-5
-10
-15
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
λ
Δ
Θ=0.005

(d)



(e)



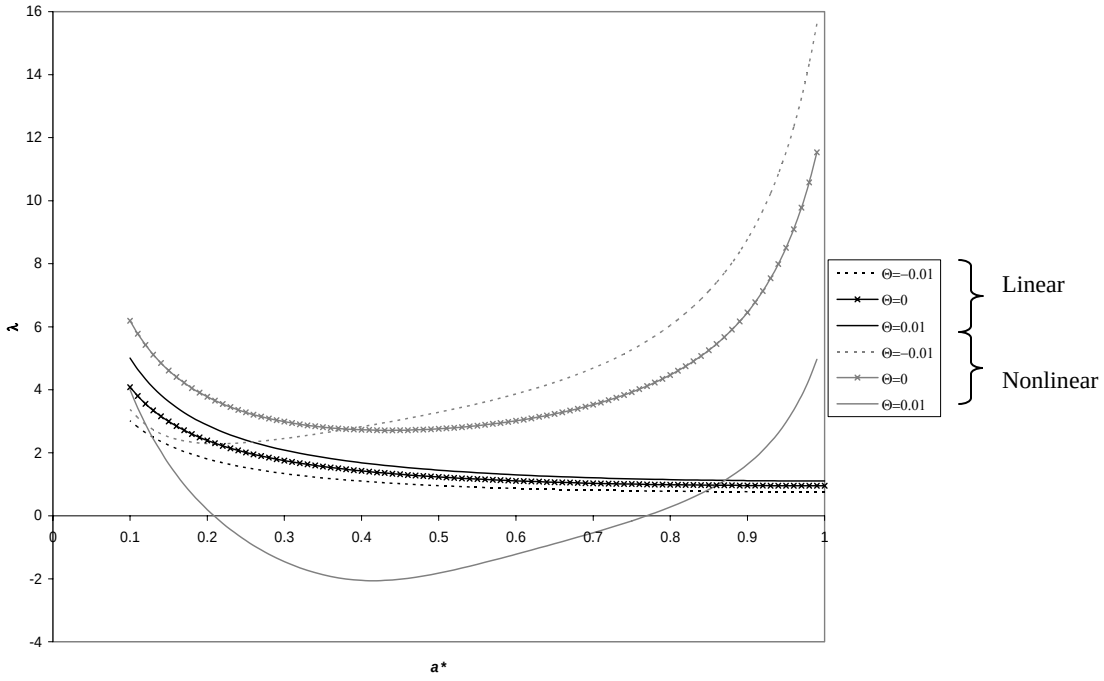
(f)

Figure 36. Nonlinear load vs. deflection curves and corresponding delamination growth paths for hinged-fixed ends, E_0 = 1, 2γ = 0.1, α_p = 0.5, a) Θ=-0.01, b) Θ =-0.005, c) Θ = 0, d) Θ =0.005, e) Θ =0.01, f) summary

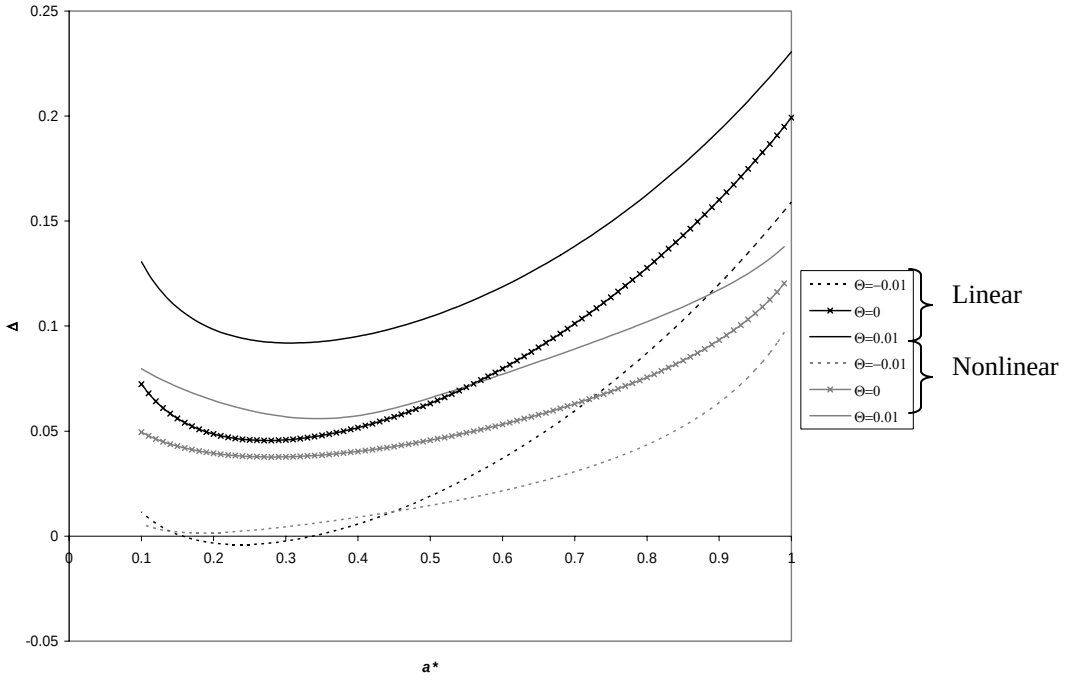
Since we are now considering a nonlinear model, the quasi-static loading curves are no longer straight lines. There may be multiple equilibrium positions associated with a single load value. This is evident in Figures 36d and 36e, or the plots for the positive temperature fields. The results based on the linear model suggest that the loading curve actually begins at the positive root of Δ when $\lambda = 0$. This is further examined in the following section.

Comparison of Linear and Nonlinear Models

We now draw a comparison between the results based on the linear simulation and the results based on the nonlinear simulation. The difference in the governing equations for each case is detailed in Chapter 3. The comparison is shown for $\alpha_p = 0.5$, $E_0 = 1$, and $2\gamma = 0.1$.



(a)



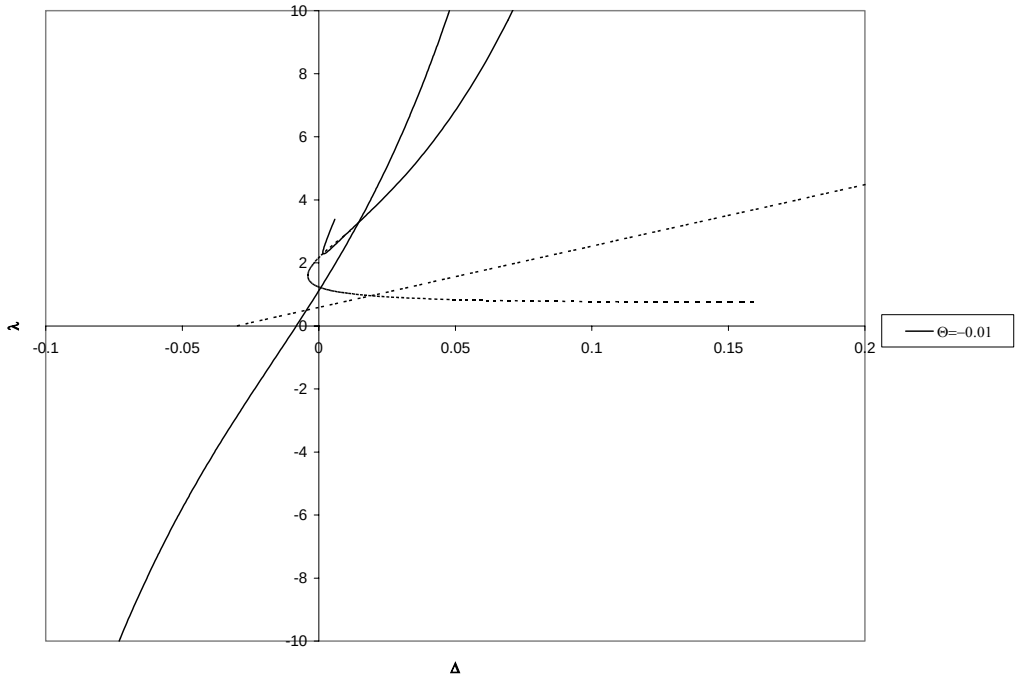
(b)

Figure 37. Comparison of hinged-fixed delamination growth paths for linear and nonlinear simulations, where E_0=1, 2γ=0.1, and α_p=0.5, a) λ vs. a*, b) Δ vs. a*

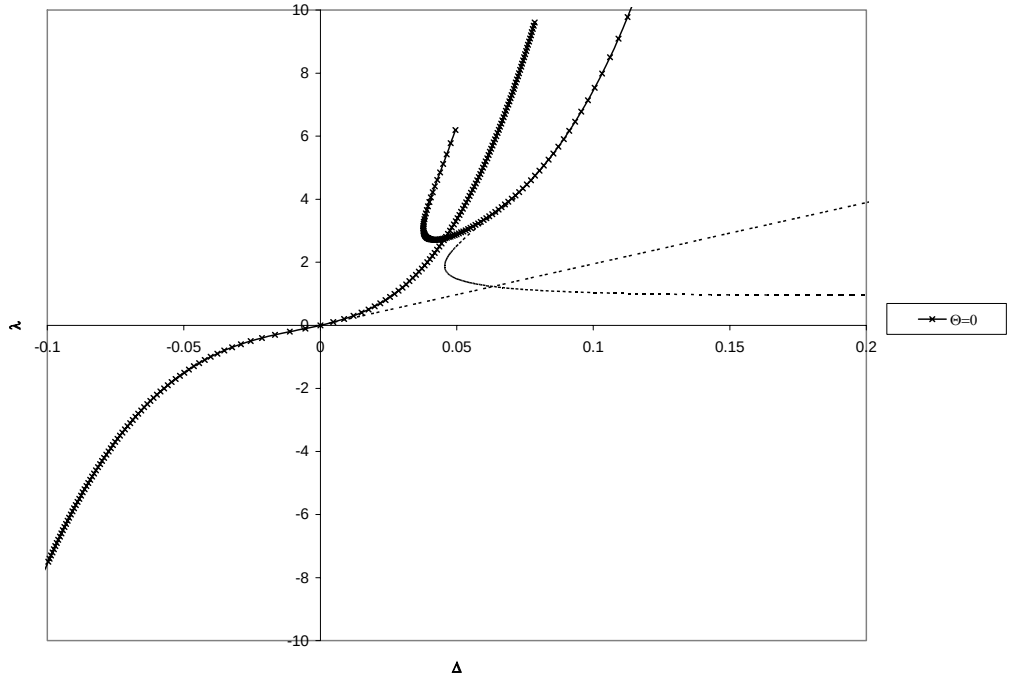
With the exception of the positive temperature change, Figure 37a suggests that the linear model generally offers a more conservative view of the problem from a pressure-loading standpoint. The linear model predicts lower, but unstable delamination growth thresholds. The nonlinear model stabilizes the behavior due to enhanced bending-stretching coupling. However, when Θ = 0.01, the nonlinear model reveals that this is a case where the effect of thermal stress becomes too powerful, and the structure begins to delaminate before it can be loaded with transverse pressure. Figure 37b, the displacement controlled test, suggests that the linear model generally offers a *less* conservative prediction of delamination growth, except for a region of a^* between 0.15 and 0.45 when Θ = -0.01.

When a thermal load is present, the linear model seems to be even less accurate. This is illustrated in our next plot, Figure 38. This figure shows a comparison of the

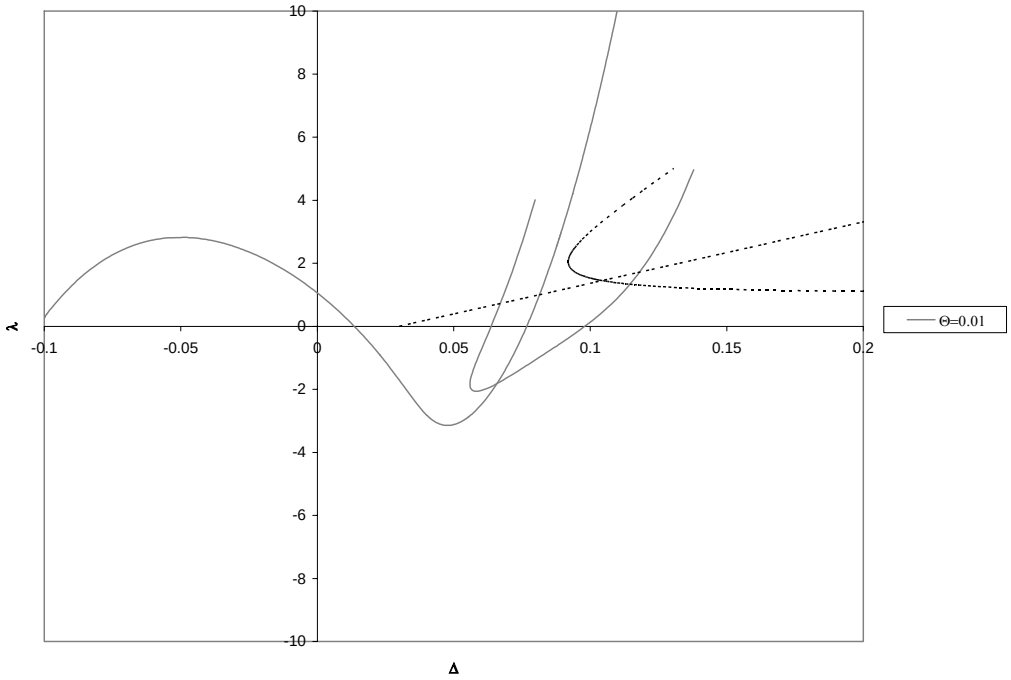
linear and nonlinear loading curves, with the corresponding delamination growth paths superimposed. The parameters are α_p =0.5, E_0 = 1, and 2γ = 0.1. The dotted black lines correspond to the response based on the linear model.



(a)



(b)



(c)

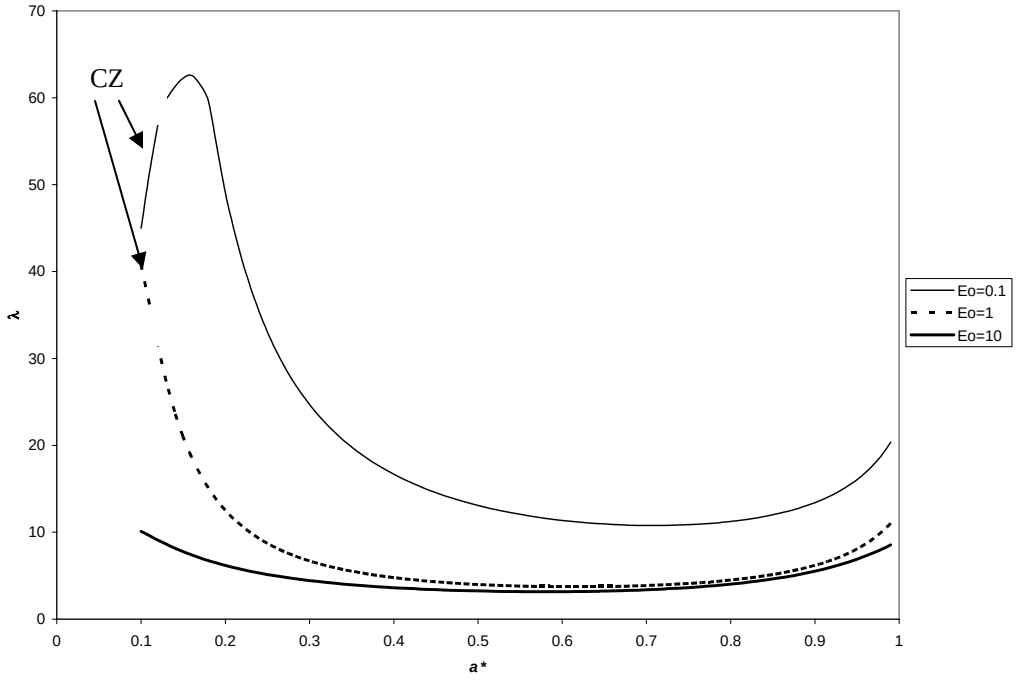
Figure 38. Comparison of linear and nonlinear load-deflection curves (with corresponding delamination growth paths) for hinged-fixed ends a) Θ=-0.01, b) Θ=0, c) Θ=0.01

The plots suggest that the linear model does not accurately capture the essence of the uniform temperature field. While the linear loading path is an accurate simplification for a small range when $\Theta = 0$, the corresponding delamination growth paths are still far apart. In general, the linear model may not be a good predictor of the full delamination evolution. However, as mentioned in the previous section, these results indicate that the nonlinear loading curve begins at the most positive root for the deflection in the absence of pressure. This is further expounded in Bottega (2006) and Karlsson and Bottega (2000). Therefore, snap-through buckling is only of concern if we are applying "negative" pressure (i.e. on the top surface of the patch).

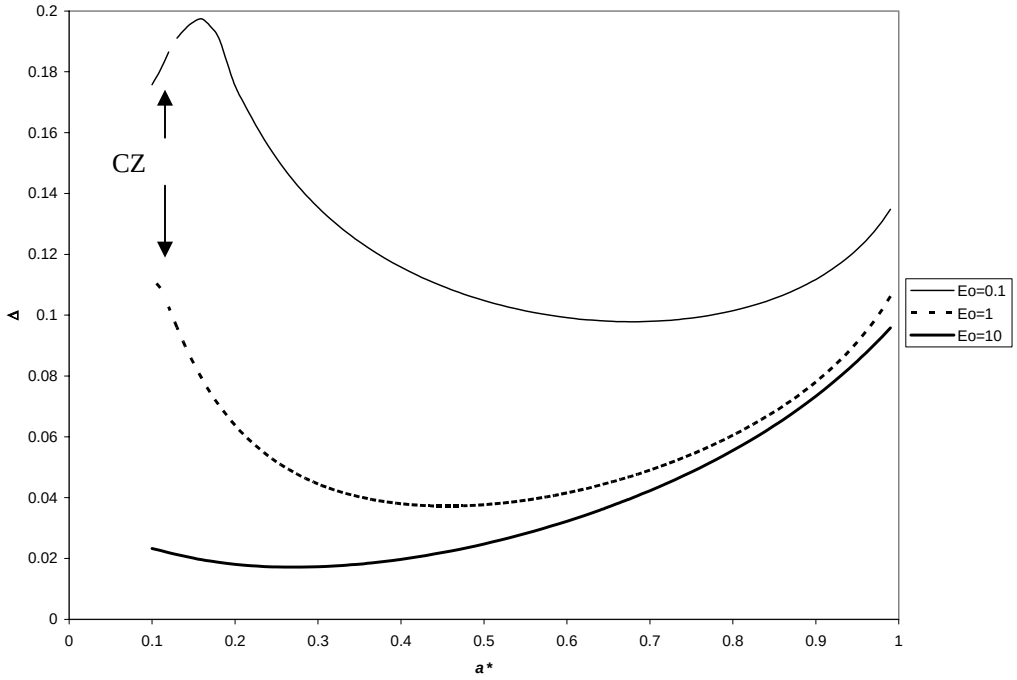
5.2.2 Clamped-Fixed End Supports

Pure Pressure Loading, Θ = 0

Finally, we explore the delamination behavior of structures with clamped-fixed ends. Figure 39 gives the comparison on the effect of various values of the elastic modulus ratio, E_0. As before, there is no uniform thermal load present, and $2\gamma = 0.1$. In the clamped-fixed case, we have taken the possibility of a contact zone into consideration. The pattern in the presence of the contact zone for the varying modulus ratios is similar to that established for the linear case. We see that the domain has shrunk, with the contact zone being present on [0, 0.12] for a^* when $E_0 = 0.1$, and [0, 0.11] when $E_0 = 1$. For the most stiff structure, $E_0 = 10$, no contact zone is present over the entire domain of initial conjugate bond sizes. Again, it should be assumed that there is no contact zone present unless otherwise marked with a "CZ" on each plot.



(a)

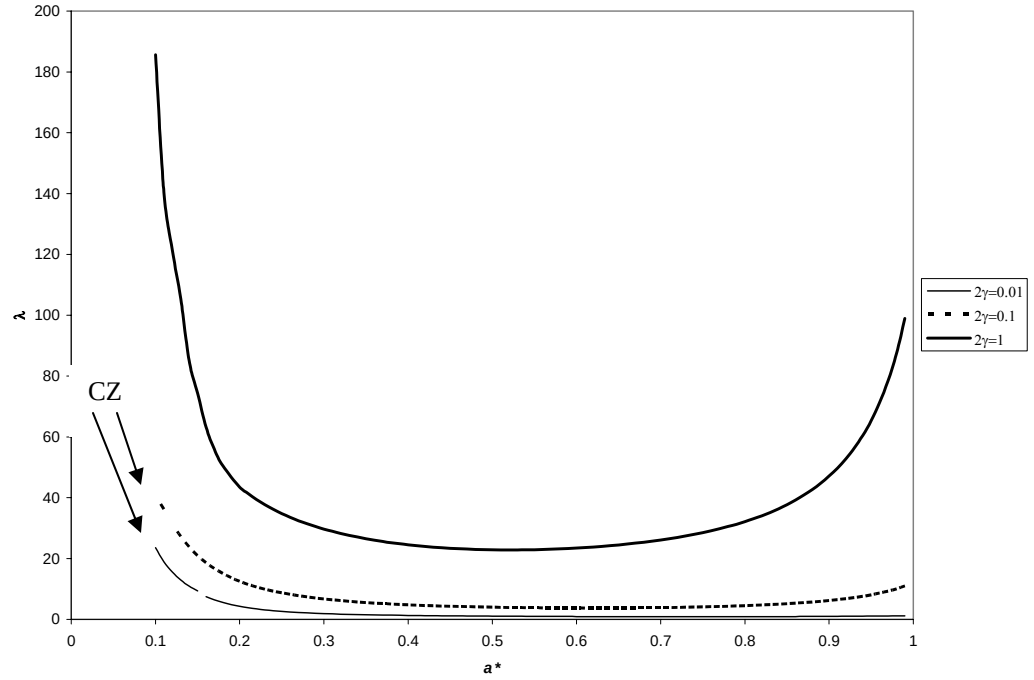


(b)

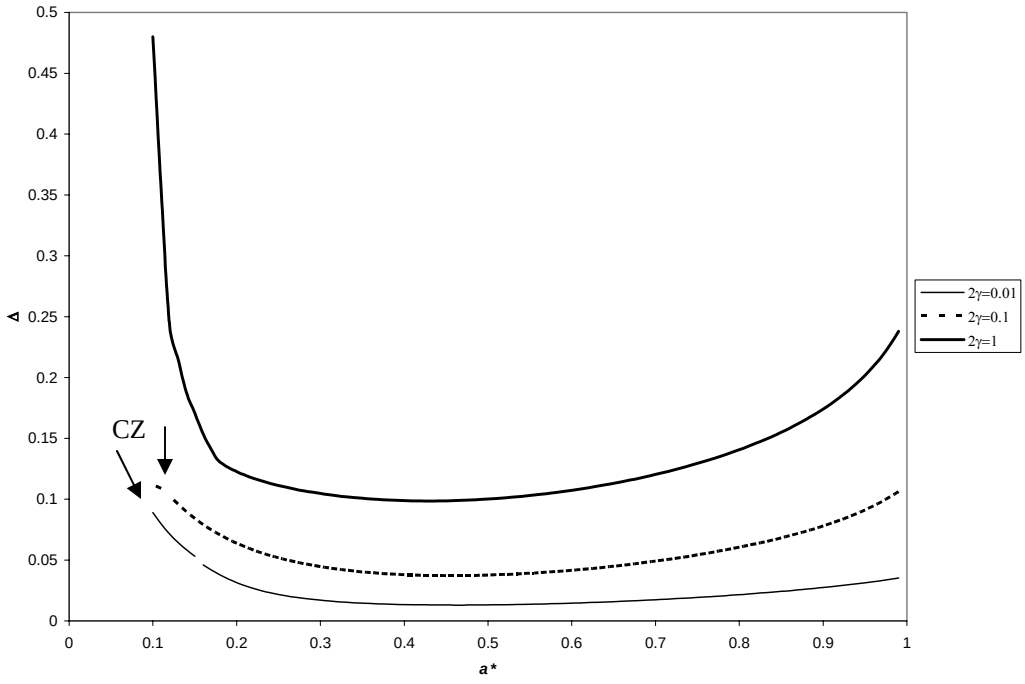
Figure 39. Nonlinear delamination growth paths for clamped-fixed ends without applied thermal load, 2γ = 0.1 a) λ vs. a*, b) Δ vs. a*

The peaks that were present in the linear solution are no longer there, except for the case of E_0 = 0.1, which appears to have a lower peak, of considerably less magnitude. For Figures 39a and 39b, the same trends as seen throughout are visible.

Figure 40 shows the effect that the bond strength, 2γ, has on the structure with clamped-fixed end supports. For the linear model, the domain of the contact zone was unaltered by the bond strength. That is not the case here, however. We observed that for weaker bond strength, the contact zone was more likely to be present. The domain of the contact zone on a^* is [0, 0.15] when 2γ = 0.01, and [0, 0.12] when 2γ = 0.1. No contact zone is present for the case of 2γ = 1.



(a)



(b)

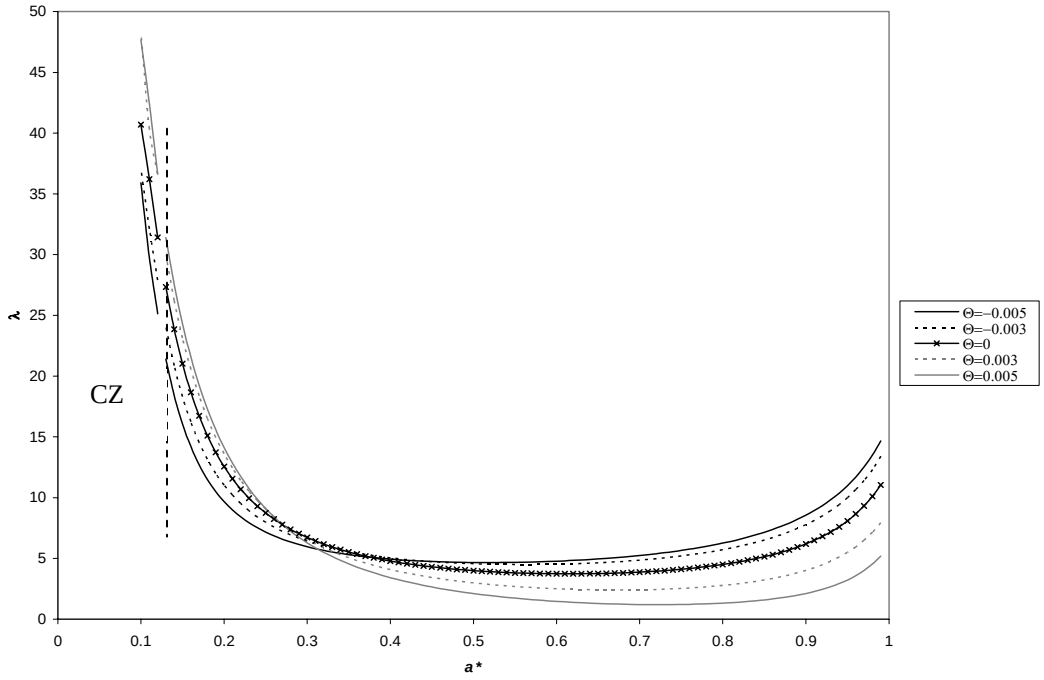
Figure 40. Nonlinear delamination growth paths for clamped-fixed ends without applied thermal load, $E_0 = 1$ a) λ vs. a*, b) Δ vs. a*

There are no peaks, as there were for the linear clamped-fixed specimen. It appears as though the peak has been shifted to the left, to the point where it is no longer on the

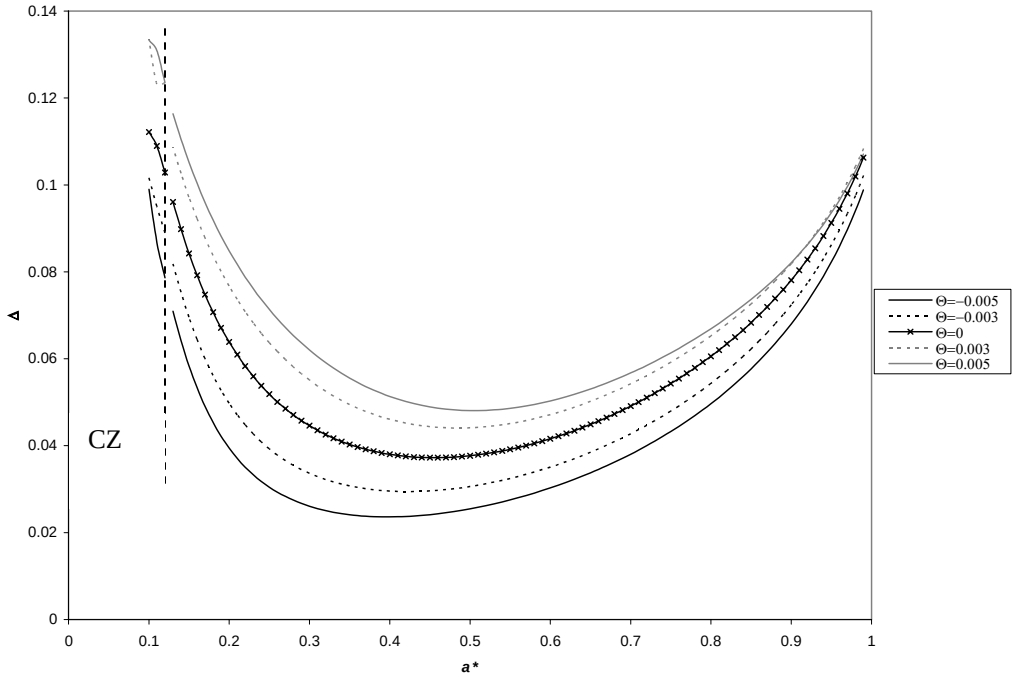
domain of the plates. The trends otherwise follows that of the linear model. The nonlinear model however predicts that stable delamination growth is recovered in the load controlled test, just like the hinged-fixed case.

Effect of Uniform Temperature Field

We now turn our attention to the effect that a uniform temperature field has on the delamination behavior of the patched plate system subject to transverse pressure. A contact zone was found to exist on [0, 0.12] for a^*, when $\alpha_p = 0.5$, for all values of Θ. This is similar to the linear case, in that the domain is the same for which a contact zone is present. The nonlinear model, however, suggests that the region of initial flaw sizes for which there may be a contact zone, is smaller. This could be due to the peaks having been shifted off the domain of the plates. A contact zone is present for any structure with a flaw size located to the left of the dotted line.



(a)



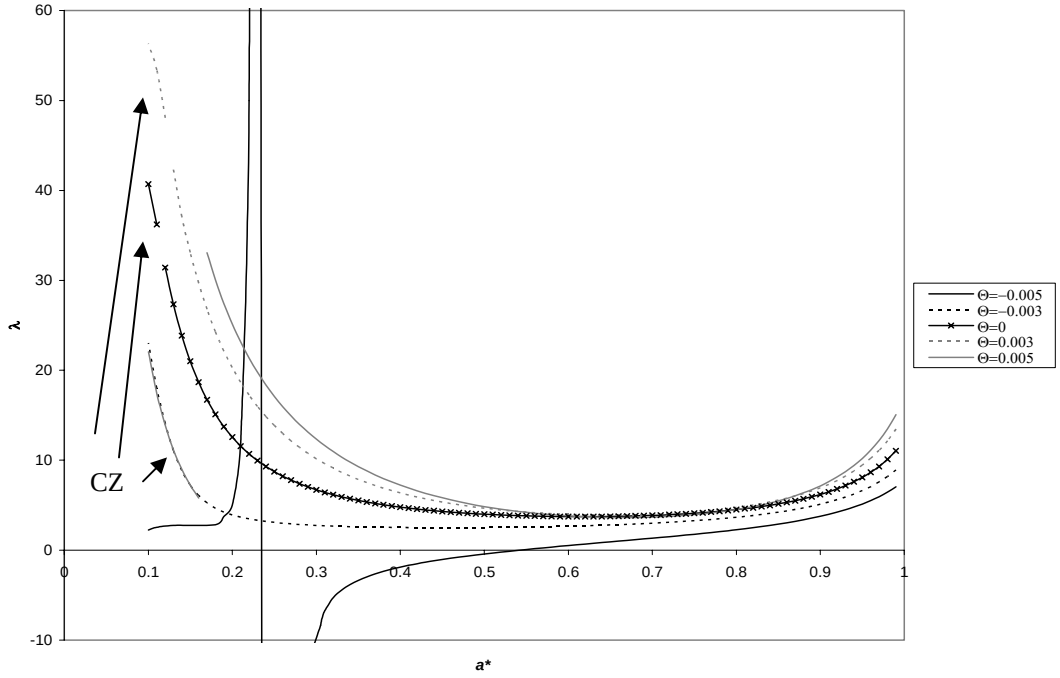
(b)

Figure 41. Nonlinear delamination growth paths for clamped-fixed ends with applied thermal load, $E_0 = 1$, $2\gamma = 0.1$, $\alpha_p = 0.5$ a) λ vs. a*, b) Δ vs. a*

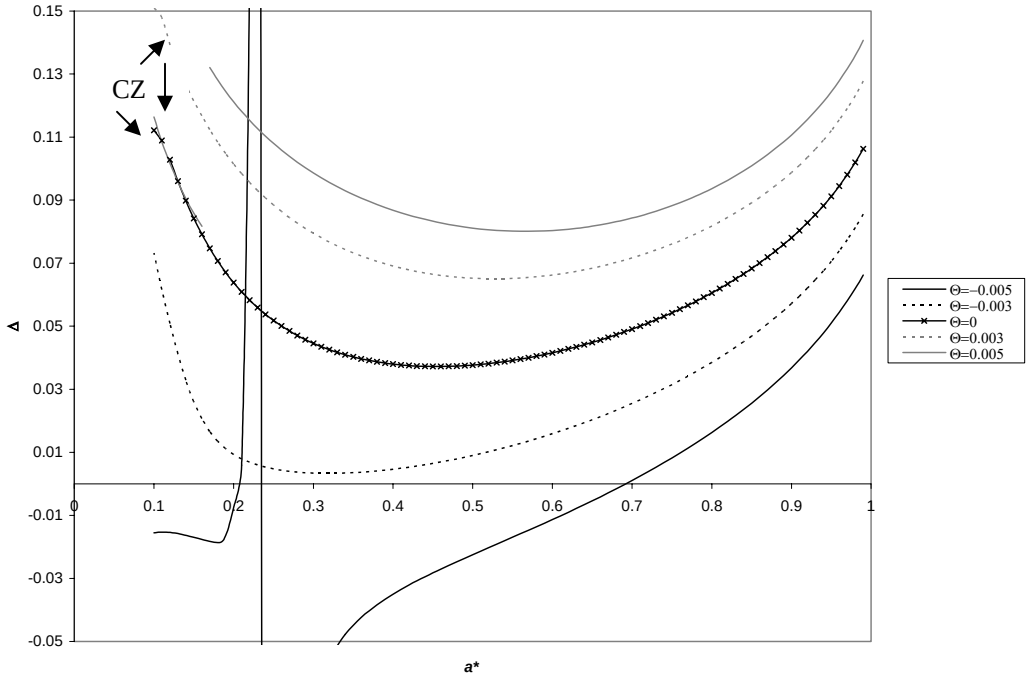
It once again looks as though there are no peaks on the domain of a^* for the nonlinear model. Like the previous case, they seem to have been shifted left, which is off the physical domain of the plates. In Figure 41a, we see that positive temperature differences tend to raise the delamination growth threshold, while negative temperature differences tend to lower it. This trend continues until about $a^* = 0.3$, where it then reverses. The displacement controlled test shows more positive temperatures raise the delamination growth threshold, and vice versa.

In Figure 42, we reverse the coefficient of thermal expansion ratio, such that $\alpha_p = 2$. Like the linear case before, the presence of the contact zone solution differs vastly, based on the uniform temperature field. Also like the linear case, the more positive temperature differences are more likely to induce a contact zone. A contact zone exists on a* for [0, 0.16] when $\Theta = 0.005$; [0, 0.12] when $\Theta = 0.003$ and $\Theta = 0$; and there is no

contact zone when Θ = -0.003 and Θ = -0.005. The effect of the contact zone is most drastic for the most positive temperature field, as it causes a sudden drop in delamination growth threshold.



(a)

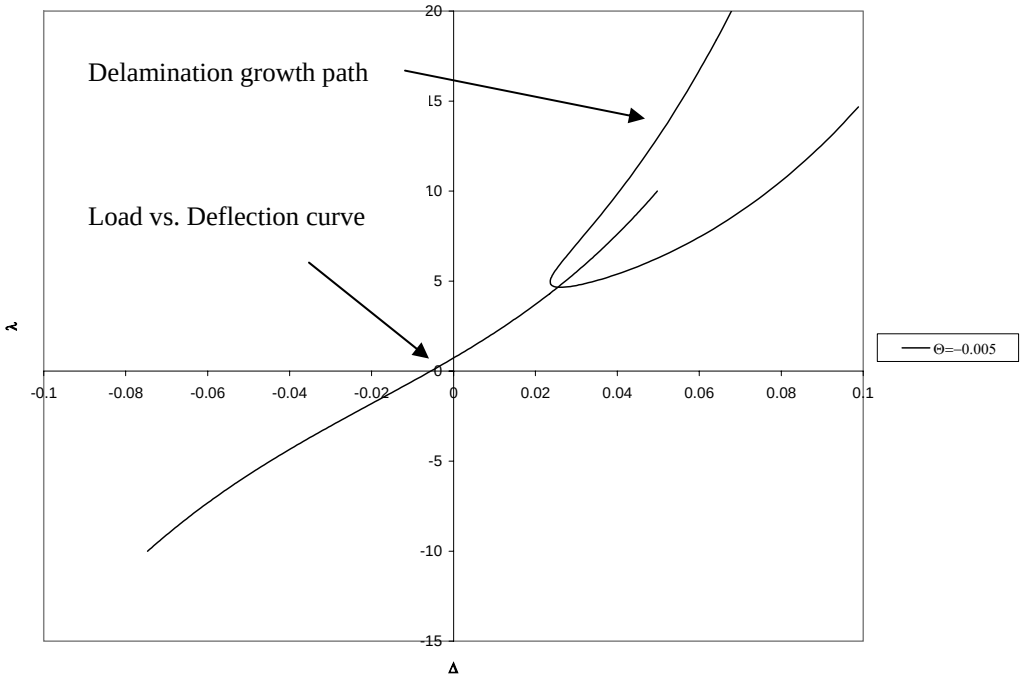


(b)

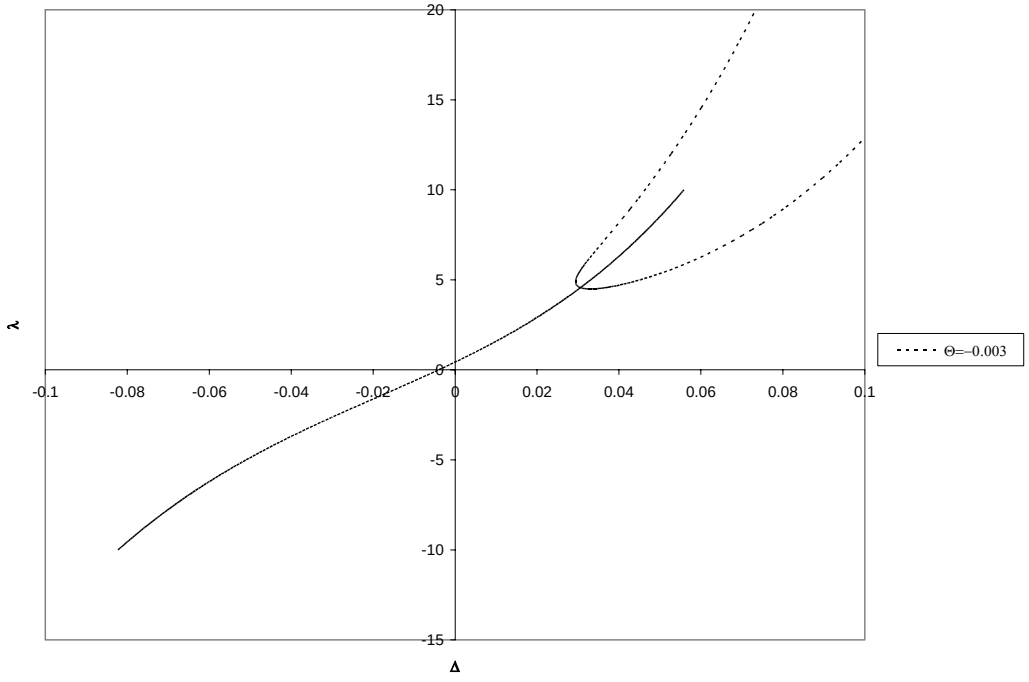
Figure 42. Nonlinear delamination growth paths for clamped-fixed ends with applied thermal load, E_0 = 1, 2γ = 0.1, α_p = 2 a) λ vs. a*, b) Δ vs. a*

If the initial flaw size is past a^* = 0.24, then the trends of Figures 42a and 42b suggest that more positive temperatures will increase the delamination thresholds. For the most negative value of the uniform temperature field, there is a range of values on a^* where delamination growth will propagate based on the thermal load alone, without the aid of the applied pressure.

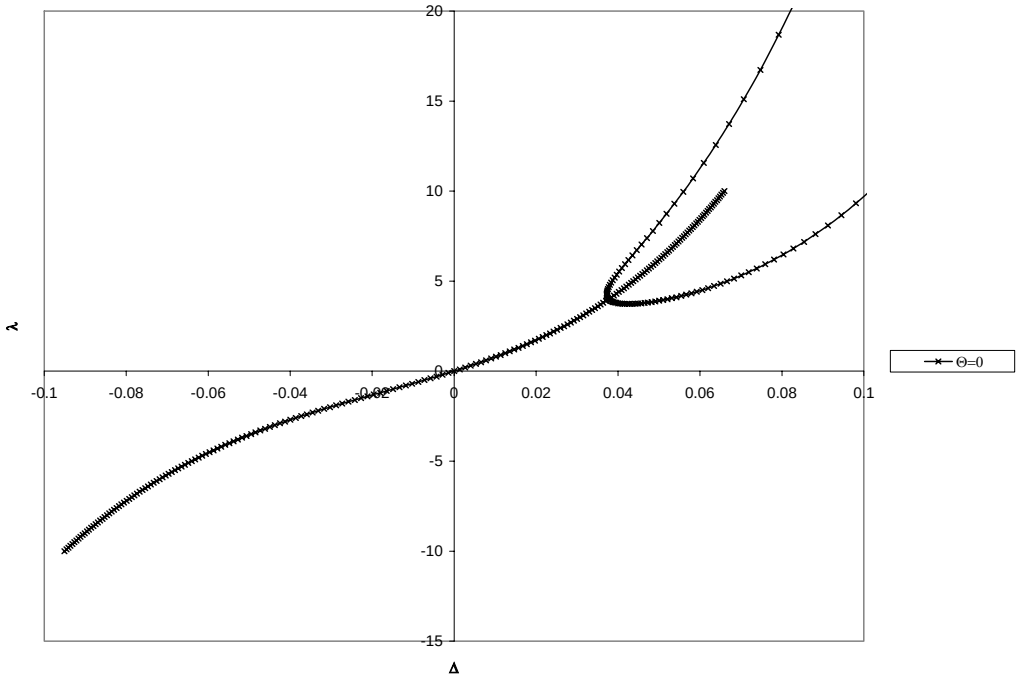
Figure 43 shows the regular nonlinear quasi-static load-deflection curves superimposed onto the delamination growth paths for the case of α_p = 0.5. The loading curves are plotted for the value of a^* = 0.5, and hence intersect the delamination path at that specific value. These are plotted for parameters for which there is no contact zone.


Delamination growth path
Load vs. Deflection curve
λ
Δ
Θ=–0.005
20
15
10
5
0
-5
-10
-15
-0.1
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1

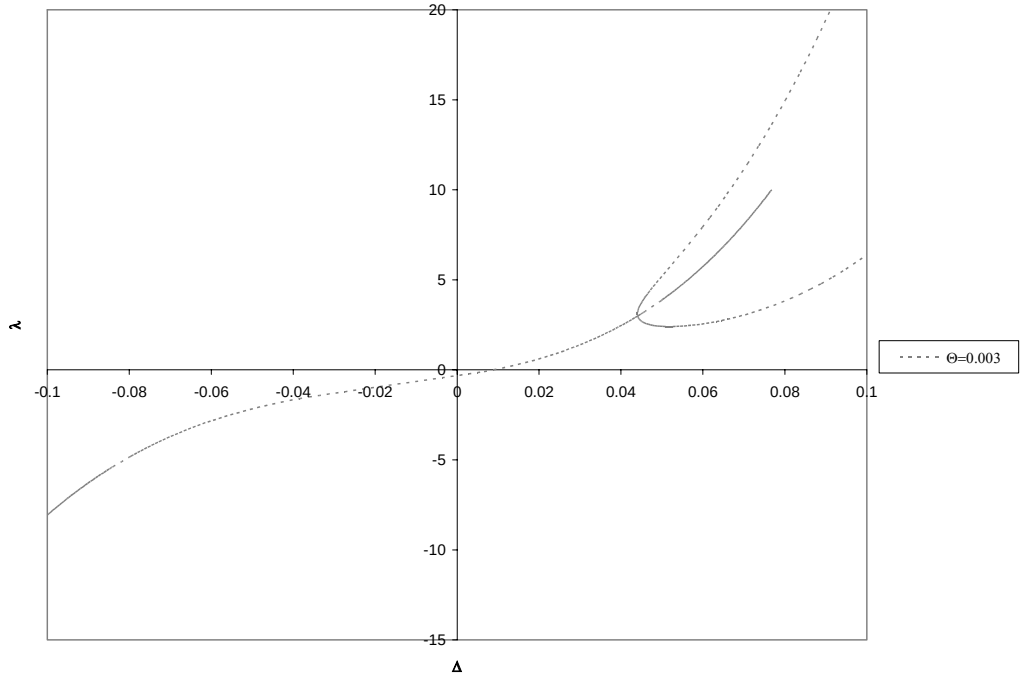
(a)


λ
Δ
Θ=–0.003
20
15
10
5
0
-5
-10
-15
-0.1
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1

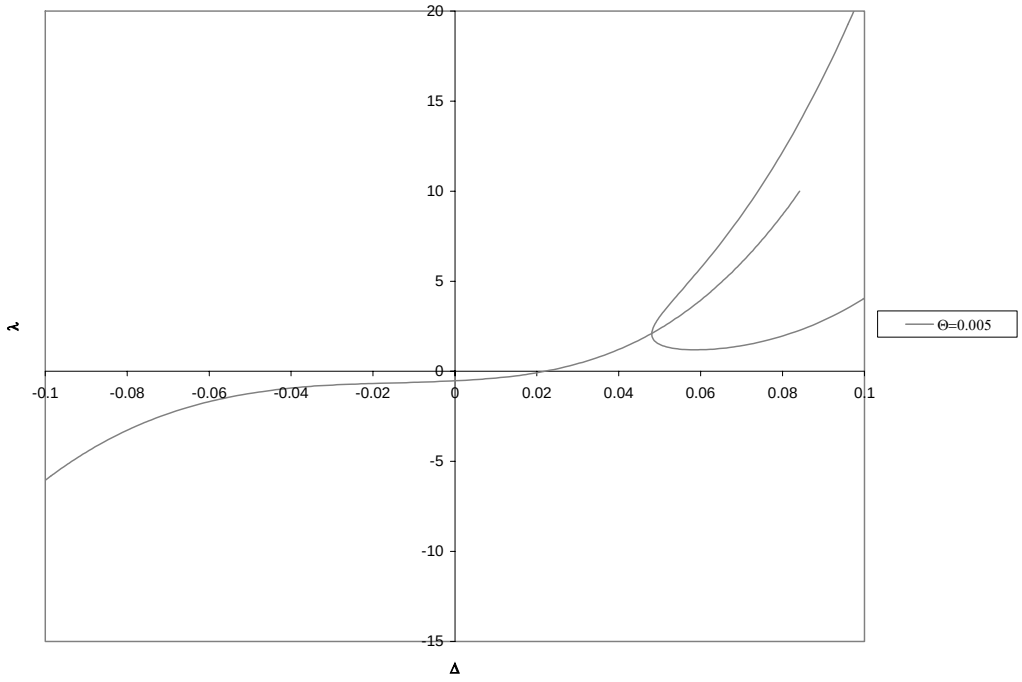
(b)


20
15
10
5
0
-5
-10
-15
-0.1
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1
λ
Δ
Θ=0

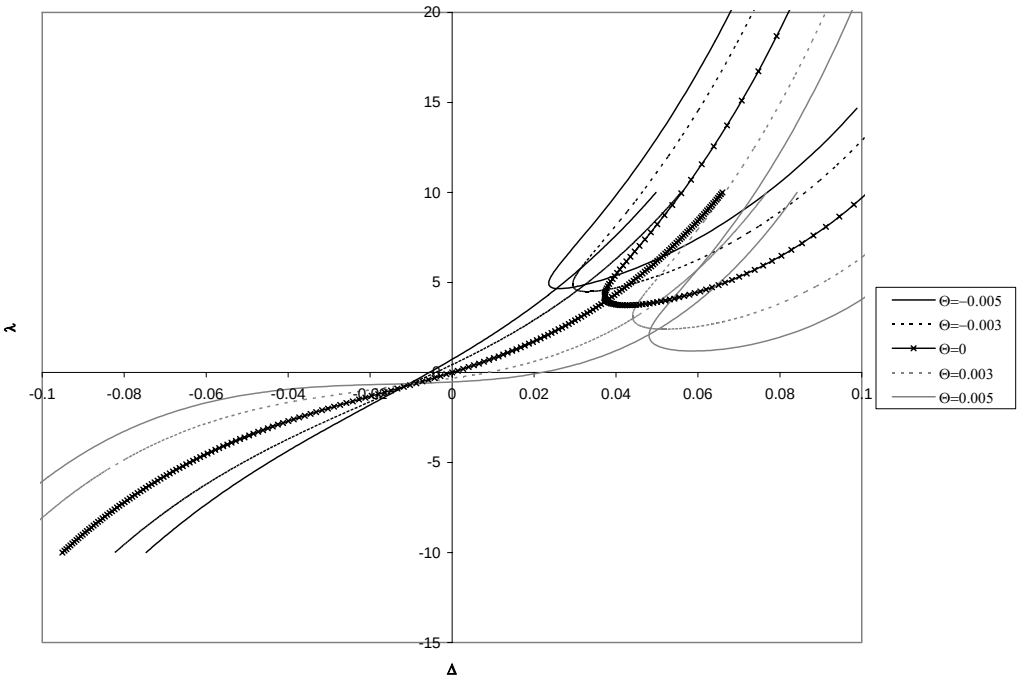
(c)


20
15
10
5
0
-5
-10
-15
-0.1
-0.08
-0.06
-0.04
-0.02
0
0.02
0.04
0.06
0.08
0.1
λ
Δ
Θ=0.003

(d)



(e)



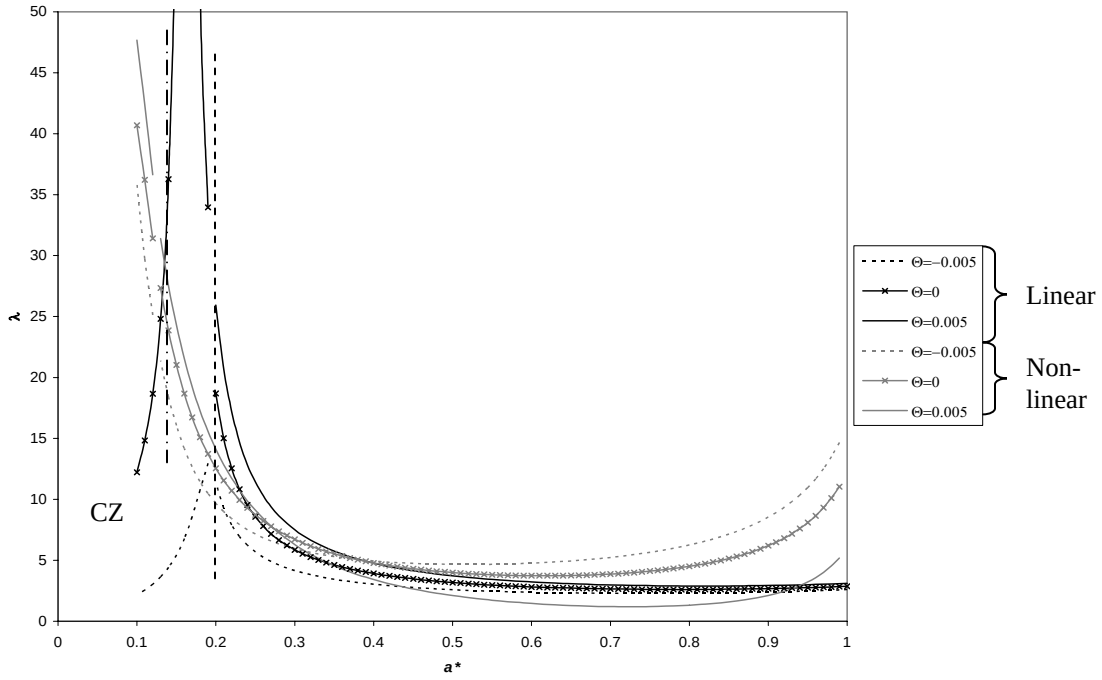
(f)

Figure 43. Load vs. deflection curves and corresponding delamination growth paths for clamped-fixed ends, E_0 = 1, 2γ = 0.1, α_p = 0.5, a) Θ=-0.01, b) Θ =-0.005, c) Θ = 0, d) Θ =0.005, e) Θ =0.01, f) summary

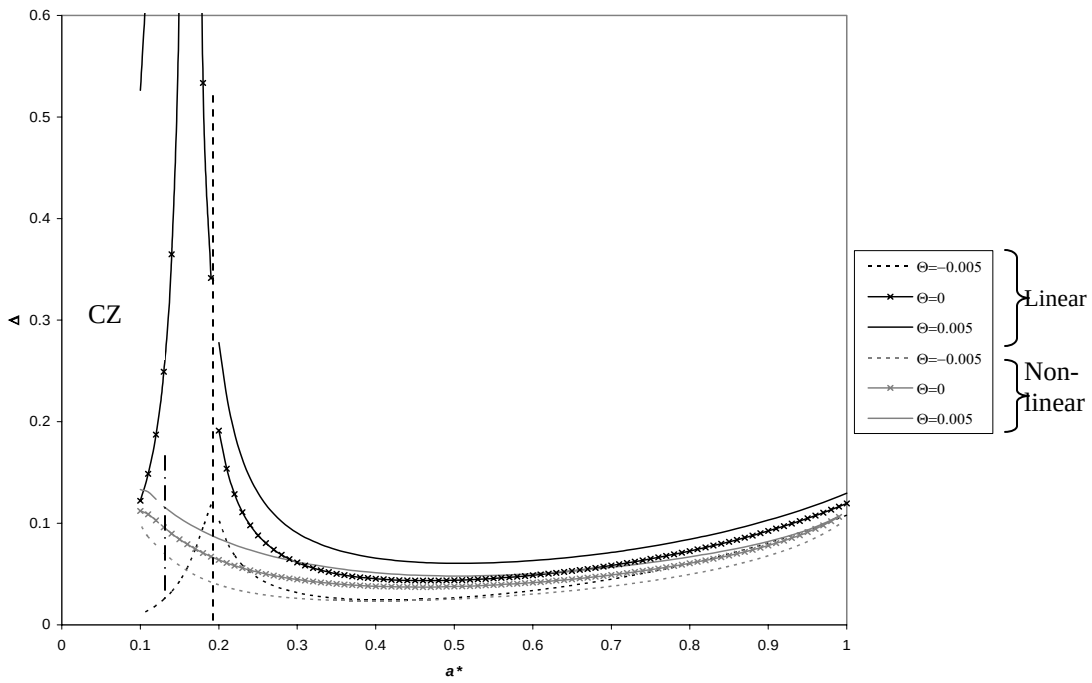
The interpretation follows that of the previously established cases. Unlike the nonlinear hinged-fixed case, we do not see the multiple equilibrium positions that are commonly associated with buckled configurations.

Comparison of Linear and Nonlinear Models

Figure 44 shows the comparison of the delamination growth paths for the linear and geometrically nonlinear solutions. These are plotted for the same parameters as Figure 34. The first dotted line on the left indicates the domain for a contact zone for the nonlinear model. The second line indicates the domain for the linear model.



(a)

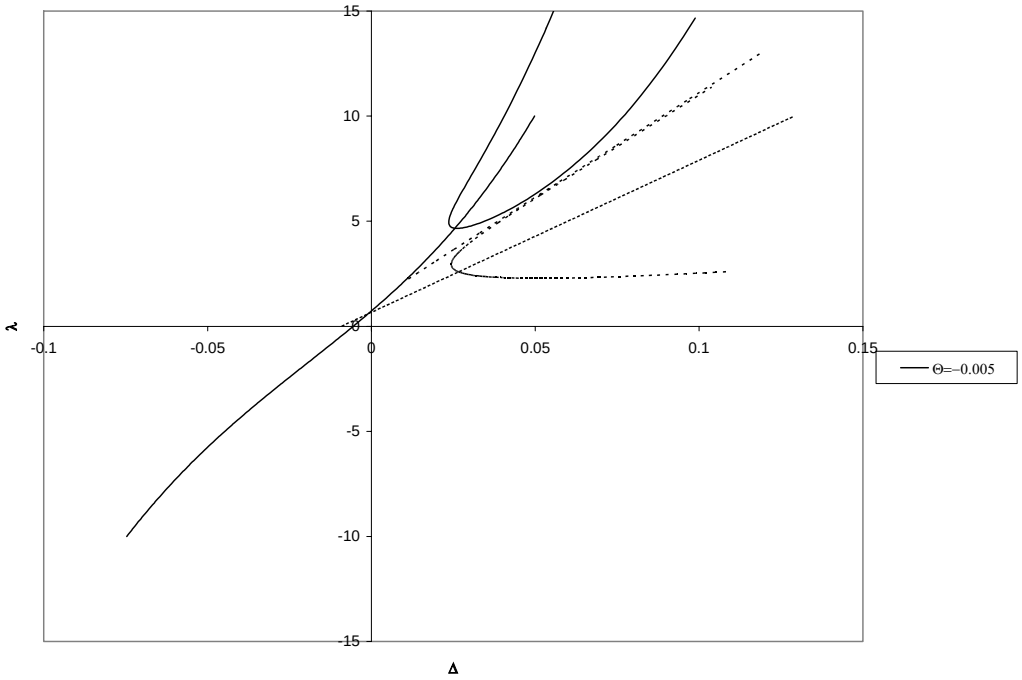


(b)

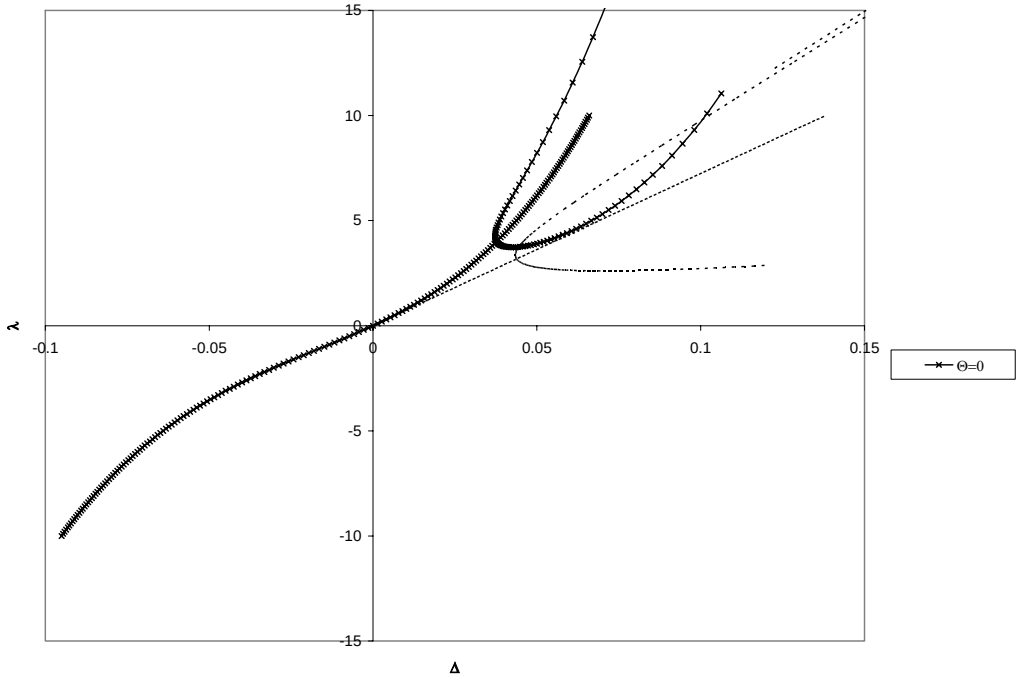
Figure 44. Comparison of clamped-fixed delamination paths for linear and nonlinear simulations, where E_0=1, 2γ=0.1, and α_p=0.5, a) λ vs. a*, b) Δ vs. a*, c) *K* vs. a*

Like the hinged-fixed case before, the linear model provides a more conservative view in Figure 44a, however, this only applies when $a^* \geq 0.35$. The linear model predicts a larger range of conjugate bond sizes for which a contact zone is present than the nonlinear model, and this plays a part in delamination growth. Since the nonlinear model seems to shift the peaks out of the domain, it suddenly provides the more conservative view in that area. Like Figure 37b, Figure 44b shows that the delamination thresholds are lowered by the introduction of the nonlinear terms.

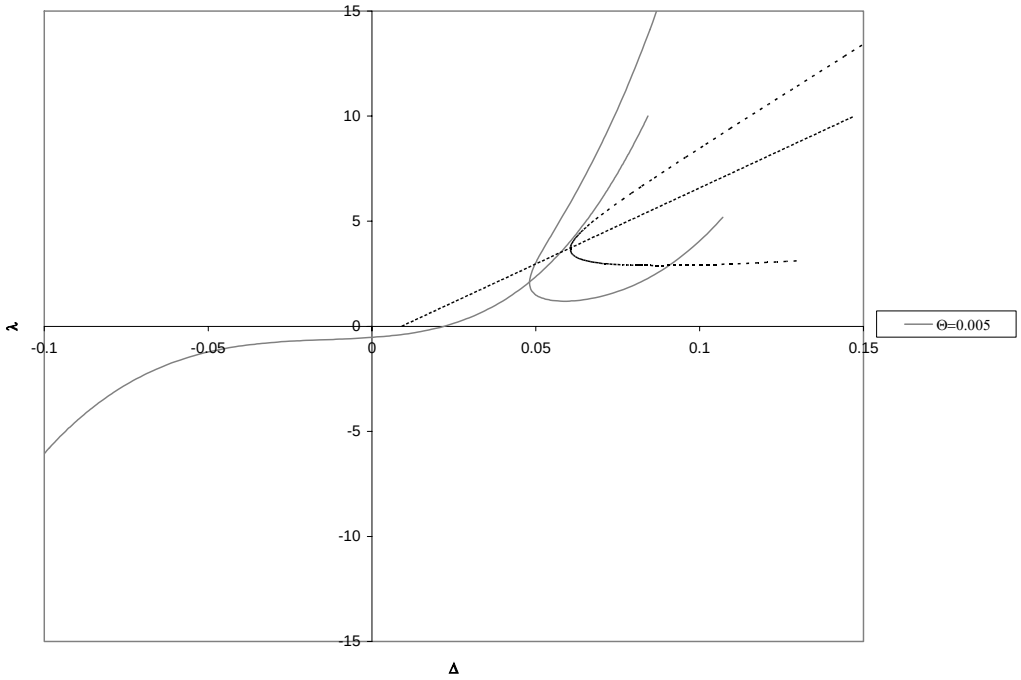
Figure 45 shows the comparison of the linear and nonlinear quasi-static load-displacement paths, with the superimposed delamination growth paths. The parameters are the same as presented for Figure 38. Once again, the black dotted lines correspond to the linear model. A contact zone is not present in this case.


15
10
5
0
-5
-10
-15
-0.1
-0.05
0
0.05
0.1
0.15
λ
Δ
Θ=−0.005

(a)


15
10
5
0
-5
-10
-15
-0.1
-0.05
0
0.05
0.1
0.15
λ
Δ
Θ=0

(b)



(c)

Figure 45. Comparison of linear and nonlinear load vs. deflection curves (with corresponding delamination growth paths) for clamped-fixed ends a) Θ=-0.01, b) Θ=0, c) Θ=0.01

The interpretation of Figure 45 follows directly from that of Figure 38. The zero case shows a good correlation between the linear and nonlinear loading curves, however, they quickly diverge in the region where delamination growth begins. The linear model is therefore generally a poor simplification when a temperature difference is introduced to the problem.

5.3 Concluding Remarks

In general, the nonlinear model reveals behavior that is not predicted by the linear model, such as the recovery of stable growth for the load controlled tests. We see that in the absence of temperature fields, parameters such as modulus ratio and stiffness have the same effect on delamination behavior. Like the linear model, we see behavior of the hinged-fixed specimen that requires additional consideration before it may be fully

characterized. The linear model tends to be more conservative with respect to loading, however it seems less conservative with respect to displacement. The comparison of linear and nonlinear results shows that the linear model may not be appropriate to describe the effect of a temperature difference. In addition, the effect of the temperature difference is so great, that it may be prudent to examine alone, in the absence of transverse pressure.

A discussion of the results obtained is presented in Chapter 6.

Chapter 6

Discussion

6.1 The Goal

The focus of our research has been to asses the effects of geometric nonlinearities and uniform temperature fields on the debonding of pressure-loaded patched plates. The structures of interest are loaded with transverse pressure applied to the lower surface of the baseplate, along with an applied uniform temperature field, and can be subject to various boundary conditions.

To accomplish this, we begin by establishing a mathematical model to describe the structure, which is viewed as an assemblage of two plates, bonded at a common interface. By considering structures under a general state of plane stress (or plane strain), we may perform a one-dimensional theoretical analysis. Using a local force balance, we can derive the governing equations for the system. Recasting the problem into a mixed formulation in terms of the transverse displacement and membrane force allows for an exact analytical solution to the geometrically nonlinear problem. By integrating the strain-displacement relationships along the domain of the beam, we establish the integrability condition, which is a necessary component and essentially a compatibility equation. A transversality condition is taken from Bottega (1997) and serves as our delamination growth condition. This equation arises from a moving boundary problem in the calculus of variations, and represents equilibrium positions of the structure for each value of the moving boundary, which in our case is the bond zone boundary. Along with

appropriate boundary and matching conditions this set of equations constitute a complete set to describe the problem of interest.

Using exact analytical solutions, we perform numerical simulations, using MATLAB. Simulations are performed comparing linear and nonlinear versions of the governing equations. We develop the linear model by neglecting the term involving the nonlinear strain-displacement relation in the governing equations. The results of these simulations give us the delamination growth paths, which can be interpreted to describe the physical delamination process behavior of the evolving composite plates.

6.2 Implications

In Chapter 4 we examine four different cases, each using the linear model as a basis. We begin by looking at the patched plate structure with hinged-free end supports, and then detail cases for end supports that are clamped-free, hinged-fixed, and clamped-fixed. It should be noted that the solution to the linear problem is the only solution when the ends are free to translate in the in-plane direction; the non-linearity arising from the membrane force is not present, since the membrane force is identically zero. The possibility of a contact zone is explored for certain cases. Recall that the contact zone exists when the plates have debonded, yet remain in sliding contact along the domain of the patch. Based on intuition and current literature (Bottega, 1995), for free ends we only expect a contact zone if the structure deflects downward. When the ends are hinged, it is expected that the curvature will not be negative, and hence contact will only occur if the deflection is down. This does actually occur, and it's proven in Bottega (1995) that only a full contact zone is present, if at all, for pressure loading alone. However, our

preliminary results suggest that the behavior may be more complex than a simple full contact zone in that region. Further research is necessary to explore this avenue. Finally, a full contact zone is considered for clamped-free ends.

Both hinged-free and clamped-free ends show similar trends with respect to how parameters such as the elastic modulus ratio and bond strength affect the delamination growth paths. It is found that, for pure pressure loading, the effect of increasing the elastic modulus ratio is to make the structure stiffer, and hence lower the thresholds for onset of delamination growth. Since the modulus ratio is defined as the ratio of elastic modulus of the patch to that of the baseplate, higher values of the modulus ratio suggest that the patch will not readily bend along with the baseplate, and hence it will begin to debond from the baseplate. As for the bond strength, increasing it expectedly increases those thresholds. A structure with greater bond strength would obviously require more load to cause it to debond.

It is seen for hinged-free ends that some values of the uniform temperature field, Θ, lend to the structure deflecting downward, opposing the presence of the transverse pressure. A contact zone solution has thus been included. The clamped-free case shows no evidence of the existence of a contact zone. The growth paths for the clamped structure exhibit peaks, which imply interesting behavior. Such peaks allow for the case of stable debonding followed immediately by catastrophic debonding upon suddenly reaching the peak position. We observe for free ends with large bond zones that the effect of a temperature difference on the response is not monotonic.

For the patched plate system with free ends (and when the coefficient of thermal expansion of the baseplate is twice that of the patch) it is seen that, in the absence of

transverse pressure, a positive temperature field causes the structure to deflect down, and vice versa. This is because the ends are free in the lateral direction, so no axial membrane force is generated, therefore, when the baseplate expands under a positive thermal load, the structure deflects down. We later witness the reverse phenomena for fixed ends.

The fixed end solutions for the linear model for both hinged and clamped cases show the same trends as their free end counterparts, with respect to the modulus ratio and the bond strength. Again, the hinged case displays behavior that suggests the occurrence of a contact zone adjacent to the bond zone. We also observe the phenomena of the loading parameter falling below the zero axis which implies that, under certain conditions, the structure will delaminate due to the temperature difference alone, before any pressure is applied. Such a finding has great importance to the practical use of these structures.

For clamped-fixed end supports, peaks in the threshold curves are observed. A contact zone is found to occur over a greater range of initial flaw size values as the modulus ratio decreased. This is because a more compliant structure can more readily achieve the curvature necessary to promote contact. The contact zone presence is unaffected by the bond strength. Its presence is also unaffected by the temperature difference when $\alpha_p = 0.5$, for the range of temperature fields considered. This is not the case when $\alpha_p = 2$. The latter value describes the case when the patch has twice the greater coefficient of thermal expansion than the baseplate. Contact zones are found to be more prevalent for increasingly positive thermal loads. This represents the situation

when the patch expands more than the baseplate due to the applied temperature difference.

The load-deflection paths (with corresponding delamination growth paths) for hinged-fixed and clamped-fixed ends, are also plotted for situations when the coefficient of thermal expansion of the baseplate is twice that of the patch, when no contact zone is present. When the pressure is zero, the reverse trend is seen on the sense of the deflection, in comparison to the free end case. Since the ends are fixed, the baseplate cannot expand under the temperature difference. This causes a resultant compressive axial force on the structure. In the composite bond zone, this axial force is coupled with a bending moment generated at the reference surface. This causes a deflection up. When the baseplate contracts, a resultant tensile membrane force is generated, producing the opposite effect. This is in addition to the thermal moment generated in the bond zone. The mechanical and thermal moments are competing effects, as shown in the relationship [Eq. (32)] for the total moment in region S_1.

The nonlinear model provides a more robust solution than the linear model. In general, we observe mostly similar relative trends for the effect of such parameters as the modulus ratio and the bond strength on the delamination growth behavior. When the ends are hinged-fixed, we once again observe that the effect of temperature is not necessarily monotonic. The effect of the bending-stretching coupling due to the temperature difference becomes overpowering, depending on the value of Θ. We also observed that the loading parameter is sometimes negative, suggesting that delamination growth begins solely due to the presence of the uniform temperature field.

Upon examination of the quasi-static loading paths (with superimposed delamination paths) we see interesting behavior for positive temperature fields when α_p = 0.5. It appears as though delamination growth may occur post-buckling. However, comparison of this load-deflection curve with that predicted by the linear model suggests that is not the case. It is also discussed in Bottega (2006) and Karlsson and Bottega (2000). It is possible that if the structure were loaded in negative pressure (i.e. applied on the top) that there could be buckling behavior.

Results for the nonlinear clamped-fixed model also show the effect of temperature to be nonmonotonic. The deflections at zero pressure are in the same sense as the nonlinear hinged-fixed solution. As for the existence of the contact zone, parallels may be drawn between linear and nonlinear. In general the domains where contact zones are present are much smaller for the nonlinear model. We see with the modulus ratio, that a contact zone is present over a much wider range as the structure becomes more compliant. Unlike for the linear model however, the results corresponding to the full nonlinear model suggest that the bond strength does have an effect on the presence of a contact zone. We observe that contact zones exist for a greater domain when the bond strength is weaker. Like the linear case, the contact zone is seen to occur for the same range of values of the conjugate bond zone regardless of temperature when the baseplate has twice the coefficient of thermal expansion of the patch. When the patch has twice the thermal expansion coefficient of the baseplate, the results also correlate. For more positive temperature differences, contact zones are observed to exist over a greater domain. The patch is more responsive to the application of a uniform temperature field

when $\alpha_p = 2$. The existence of the contact zone depends largely on the deformed shape of the patch; hence this configuration more greatly affects it.

The comparison of the results for linear and nonlinear models shows that the linear model predicts a much earlier onset of delamination in the load controlled test. There is an exception of the case of the more positive temperature change. The nonlinear model suggests that at this value the effect of the thermal stress is to reduce the delamination growth threshold significantly, which is an important consideration for design. Also revealed by the delamination growth paths for the nonlinear model is that stable debonding is recovered for force-controlled loading. Thus, additional load would be required to separate the plates, while the linear model predicts mostly catastrophic delamination growth. The displacement controlled test suggests the opposite; that the linear model is less conservative.

In general, we see that the nonlinear model predicts a stiffer structure, with some areas of exception, of course. This is because of increased bending-stretching coupling for the nonlinear model. It explains why the contact zones are smaller for the nonlinear model. Stiffness has an impact on the possibility of a contact zone, so it makes sense that the model predicting a stiffer structure would have smaller domains of contact. Most importantly, it is seen that contact zone does have a dramatic effect on the delamination behavior of the system. Furthermore, it is present for more cases than initially anticipated.

When the quasi-static loading paths for the linear and nonlinear models are compared, we see that there is good correspondence between the cases in the absence of temperature for low values of the normalized pressure. The curves, however, diverge

before any significant delamination behavior takes place. This striking result suggests that the linear model is not a good approximation around the critical values on the loading curve. We also see that when a temperature field is present, the linear model is not a good approximation at any point on the loading curves. The effect of temperature cannot be properly identified by linear means. This is because temperature enters the problem via an axial force, which is directly dependent on the strain. Since nonlinear strain-displacement relations are used in our formulation, the nonlinear model more accurately captures the salient behavior.

In the pantheon of delamination research, it is always important to contribute a simulation of delamination behavior based entirely on exact mathematical results. The computational aspect only serves as a tool in this work, rather than an approximation/numerical solution. Our research offers many detailed and subtle nuances about patched plates that have been detected by use of full nonlinear analytical solutions. Debonding of patched plates subject to pressure loading [for example, Bottega (1995)], or thermal *buckling* of patched plates have been studied before [for example, Karlsson and Bottega (2000)], in a very similar fashion. But our purpose was to explore the combination of effects that these two types of realistic loading have on the *delamination growth* of a patched structure. This exact problem has not been studied in any great capacity, using exact analytical methods.

6.3 Future Considerations

The problem of delamination propagation is of critical importance to design, utilization, and survival of patched plate structures. This is a step taken in an effort to understand such behavior of these important structures.

It is shown that the linear model is a rather crude approximation of the true delamination behavior of the system. It was observed that the linear model does not accurately predict the effects of an applied uniform temperature field, in addition to the transverse pressure. To get a clearer view of the behavior, a more mathematically robust model is necessary.

The problem does not end here – there are other considerations of importance. From our analysis, we found that the effect of a temperature difference on delamination growth is so potent, that it warrants its own study to be more fully characterized. We also may wish to more extensively explore the stability of the equilibrium positions found, by consideration of the second variation of an appropriate potential energy functional. This also lends itself to consideration of thermal buckling, and its interaction with delamination propagation.

It is of interest to observe a wider range of values on the parameter α_p. By looking at a full spectrum, we could further characterize the effects of thermal expansion coefficient mismatches.

We also uncovered behavior of the hinged-fixed specimen that appears to more complex than simply having a full contact zone. This requires additional consideration as well. It is possible that the contact zone could be propagating, rather than stationary.

In addition, we have not explored the effect of the mode mix on delamination growth. This generally affects our delamination criterion, since we have not decomposed our critical energy release rate into Mode I and Mode II. This may affect the thresholds, and will provide a more accurate physical interpretation of the debonding phenomena.

References

Altus, E., Ishai, O. (1992) Delamination Buckling Criterion for Composite Laminates: A Macro Approach, *Engineering Fracture Mechanics* **41** (5), 737–751.

Blanco, N., Gamstedt, E.K., Asp, L.E., Costa, J. (2004) Mixed-Mode Delamination Growth in Carbon-Fibre Composite Laminates Under Cyclic Loading, *Int. J. Solids Structures* **41**, 4219–4235.

Bois, C., Herzog, P., Hochard, C. (2007) Monitoring a Delamination in a Laminated Composite Beam Using In-Situ Measurements and Parametric Identification, *Journal of Sound and Vibration* **299**, 786–805.

Borg, R., Nilsson, L., Simonsson, K. (2004) Simulating DCB, ENF, and MMB Experiments Using Shell Elements and a Cohesive Zone Model, *Composites Science and Technology* **64**, 269–278.

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. (1997) On the Detachment of Patched Panels Under Themo-Mechanical Loading, unpublished notes.

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., Karlsson, A.M. (1999) On the Detachment of Step-Tapered Doublers: Part 1 – Foundations, *Int. J. Solids and Structures* **36**, 1597–1623.

Carpinteri, A., Spagnoli, A., Vantadori, S. (2004) A Fracture Mechanics Model for a Composite Beam with Multiple Reinforcements Under Cyclic Loading, *Int. J. Solids Structures* **41**, 5499–5515.

Duong, C.N., Yu, J. (2002) An Analytical Estimate of Thermal Effects in a Composite Bonded Repair: Plane Stress Analysis, *Int. J. Solids Structures* **39**, 1003–1014.

Giannakopoulos, A.E. (1994) Energy Release Rate of Dynamic Delamination, *Engineering Fracture Mechanics* **47** (4), 465–471.

Gregory J.R., Spearing S.M. (2006) Modeling Inelastic Matrix Crack Tip Deformation in a Double Cantilever Beam Specimen, *Journal of Composite Materials* **40** (2), 142–156.

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.

Hwu, C., Hu, J.S. (1992) Stress Intensity Factors and Energy Release Rates of Delaminations in Composite Laminates, *Engineering Fracture Mechanics* **42** (6), 977–988.

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. (1999a) The Presence of Edge Contact and Its Influence on the Debonding of Patched Panels, *International Journal of Fracture* **96**, 381–404.

Karlsson, A.M., Bottega, W.J. (1999b) On the Detachment of Step-Tapered Doublers: Part 2 – Evolution of the Pressure Loaded Structures, *Int. J. Solids and Structures* **36**, 1625–1651.

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

Karlsson, A.M., Bottega, W.J. (2000) Thermo-Mechanical Response of Patched Plates, *AIAA Journal* **38** (6), 1055–1062.

Kim, H. (2006) Closed Form Solution for Strain Energy Release Rate Distribution in Debonded One-Edge Free Postbuckled Composite Flanged Joints, *Composites Science and Technology* **66**, 2456–2464.

Klingbeil, N.W., Bontha, S. (2003) A Maximum Allowable Flaw Size for Debond-Resistant Bimaterial Layers, *Engineering Fracture Mechanics* **70**, 2103–2114.

La Saponara, V., Muliana, H., Haj-Ali, R., Kardomateas, G.A. (2002) Experimental and Numerical Analysis of Delamination Growth in Double Cantilever Laminated Beams, *Engineering Fracture Mechanics* **69**, 687–699.

Liu, M.L., Yu, J. (2003) Finite Element Modeling of Delamination by Layerwise Shell Element Allowing for Interlaminar Displacements, *Composites Science and Technology* **63**, 517–529.

Moore, T.D. (2005) Thermomechanical Peeling in Multilayer Beams and Plates – a Solution from First Principles, *Int. J. Solids Structures* **42**, 271–285.

Müller, I. (2007) Clapping in Delaminated Sandwich-Beams Due to Forced Oscillations, *Comput. Mech.* **39**, 113–126.

Panigrahi, S.K., Pradhan, B. (2007) Three Dimensional Failure Analysis and Damage Propagation Behavior of Adhesively Bonded Single Lap Joints in Laminated FRP Composites, *J. Reinforced Plastics and Composites* **26** (2), 183–200.

Qingchun, M., Xing, Z. (1993) Analytical-Generalized Variational Method of Solution for Delamination of Laminates, *Engineering Fracture Mechanics* **46** (5), 797–805.

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, S.J., Waas, A.M. (1994) Mode I Failure of Laminated Polymeric Composites, *Engineering Fracture Mechanics* **49** (1), 17–27.

Storåkers, B., Larsson, P.L., Rohart, C. (2004) On Delamination Growth in Shallow Shells, *J. Applied Mech.* **71**, 247–254.

Tafreshi, A. (2006) Delamination Buckling and Postbuckling in Composite Cylindrical Shells Under Combined Axial Compression and External Pressure, *Composite Structures* **72**, 401–418.

Tamuzs, V., Tarasovs, S. Vilks, U. (2001) Progressive Delamination and Fiber Bridging Modeling in Double Cantilever Beam Composite Specimens, *Engineering Fracture Mechanics* **68**, 513–525.

Tarn, J., Shek, K. (1991) Analysis of Cracked Plates With a Bonded Patch, *Engineering Fracture Mechanics* **40** (6), 1055–1065.

Tsamasphyros, G.J., Kanderakis, G.N., Marioli-Riga, Z.P. (2003) Thermal Analysis by Numerical Methods of Debonding Effects Near the Crack Tip Under Composite Repairs, *Applied Composite Materials* **10**, 149–158.

Wang, J., Qiao, P. (2004) Interface Crack Between Two Shear Deformable Elastic Layers, *Journal of the Mechanics and Physics of Solids* **52**, 891–905.

Appendix

The appendix presents a listing of all code created to generate the results for each scenario. It includes all the relevant MATLAB and Maple code.

A.1 MATLAB Code for Free Ends

```
% Separation of Patched Structure
% Patched Plate, transverse pressure/thermal load, hinged-free ends

% This program uses the functions:  HFREE_system1, HFREE_system2,  Pstar_HFREE
clear

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

% 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;
rhostar = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Dstar = Astar-rhostar*Bstar;
Ds = Dstar;
Dc = D+Dp; % bending stiffness of debonded segment
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);

%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 2; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rhostar*nstar;
alpha_1 = nstar/Cstar;
etaTilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
ms = mstar;

% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure
```

```
a = [0:0.01:0.9]; % values of a (bond length)
b=0.9;
resultp = zeros(16,length(a)); % allocate memory for the answers
resultT = zeros(16,length(a)); % Change to 16 for CZ
global theta gamma
theta = -0.01;
gamma = 0.1/2;

for count = 1:length(a);
   %No CZ
  %resultp(:,count) = HFREE_system1(a(count),Dstar,D);
  %resultT(:,count) = HFREE_system2(a(count),Dstar,D,mstar); %zero when theta=0
   %CZ
  resultp(:,count) = HFREE_CZsys1(a(count),b,Dstar,Dc,D);
  resultT(:,count) = HFREE_CZsys2(a(count),b,Dstar,Dc,D,mstar); %zero when theta = 0

  %P_star(count) =
Pstar_HFREE(a(count),squeeze(resultp(2,count)),squeeze(resultT(2,count)),squeeze(resultp(6,count)),sque
eze(resultT(6,count)),Dstar,D,etaTilda);

P(count)=P_HFREE(a(count),squeeze(resultp(2,count)),squeeze(resultT(2,count)),squeeze(resultp(6,count)
),squeeze(resultT(6,count)),Dstar,D,Dc,etaTilda);
  %P(count)=P_star(count)*sqrt(2*gamma);
  Delta0(count) = -1*HFREE_DELTA(squeeze(resultp(4,count)),squeeze(resultT(4,count)),P(count));
  K(count) = P(count)/Delta0(count);
end

Results(:,1)=P';
Results(:,2)=Delta0';
Results(:,3)=K';

astar = ones(1,length(a)) - a;

figure
plot(astar,P)
title('Loading Parameter')
ylabel('P')
xlabel('Conjugate Bond Length')

%Choose value of a, I choose a=0.5
q=51;  %when a=0.5
Pc = [0:.1:10];
for ic = 1:length(Pc);
   def(ic)=-1*(squeeze(resultp(4,q))*Pc(ic)+squeeze(resultT(4,q))*theta);
end


% Separation of Patched Structure
% Patched Plate, transverse pressure/thermal load, clamped-free ends

% This program uses the functions:  CFREE_system1, CFREE_system2,  Pstar_HFREE
clear
```

```
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

% 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;
rhostar = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Dstar = Astar-rhostar*Bstar;

Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);

%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-rhostar*nstar;
alpha_1 = nstar/Cstar;
etaTilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;

% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure

a = [0:0.01:0.9]; % values of a (bond length)
resultp = zeros(12,length(a)); % allocate memory for the answers
resultT = zeros(12,length(a));

global gamma theta
gamma = 0.1/2;
theta = 0.005;

for count = 1:length(a);
  resultp(:,count) = CFREE_system1(a(count),Dstar,D);
  resultT(:,count) = CFREE_system2(a(count),Dstar,D,mstar);
  %P_star(count) =
Pstar_HFREE(a(count),squeeze(resultp(2,count)),squeeze(resultT(2,count)),squeeze(resultp(6,count)),sque
eze(resultT(6,count)),Dstar,D,etaTilda);

P(count)=P_HFREE(a(count),squeeze(resultp(2,count)),squeeze(resultT(2,count)),squeeze(resultp(6,count)
),squeeze(resultT(6,count)),Dstar,D,etaTilda);
  Delta0(count) = -HFREE_DELTA(squeeze(resultp(4,count)),squeeze(resultT(4,count)),P(count));
  K(count) = P(count)/Delta0(count);
```

```
end

Pc=[0:0.1:10];
ind=51; %corresponds to a=0.5
for jj=1:length(Pc);
   def(jj)=squeeze(resultp(4,ind))*Pc(jj)+squeeze(resultT(4,ind))*theta;
   deflection(jj) = -def(jj);
end

astar = ones(1,length(a)) - a;

figure
plot(astar,P)
title('Loading Parameter')
ylabel('Pstar')
xlabel('Conjugate Bond Length')

function[CONST] = HFREE_system1(a,Dstar,D)
% System for P

Fp1 = [0,0,1,0,0,0,0,0,0,0,0,0];
Fp2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Fp3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Fp4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Fp5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Fp6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Fp7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Fp8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Fp9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Fp10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Fp11 = [0,0,0,0,0,.5,1,1,0,0,0,0];
Fp12 = [0,0,0,0,0,1,0,0,0,0,0,0];

Fptotal = [Fp1;Fp2;Fp3;Fp4;Fp5;Fp6;Fp7;Fp8;Fp9;Fp10;Fp11;Fp12];
Rp = [0;0;(a^4)*(1/Dstar-1/D)/24;(a^4)/(24*Dstar);(a^3)*(1/Dstar-
1/D)/6;(a^3)/(6*Dstar);0;0;0;0;1/(24*D);1/(2*D)];%times p
CONST = inv(Fptotal)*Rp; % Matrix composed of integration constants, p factored out

function[CON] = HFREE_system2(a,Dstar,D,mstar)
% System for Thetatilda

F1 = [0,0,1,0,0,0,0,0,0,0,0,0];
F2 = [1,0,0,0,0,0,0,0,0,0,0,0];
F3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
F4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
F5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
F6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
F7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
F8 = [0,0,0,0,1,0,0,0,0,0,0,0];
F9 = [0,0,0,0,0,0,0,0,0,1,0,0];
F10 = [0,0,0,0,0,0,0,0,1,0,0,0];
F11 = [0,0,0,0,0,.5,1,1,0,0,0,0];
F12 = [0,0,0,0,0,1,0,0,0,0,0,0];
```

```
Ftotal = [F1;F2;F3;F4;F5;F6;F7;F8;F9;F10;F11;F12];
R = [0;0;0;0;0;0;mstar;0;0;0;0;0];%times thetatilda
CON = inv(Ftotal)*R; % Matrix composed of integration constants, thetatilda factored out

function[CONSTp] = HFREE_CZsys1(a,b,Dstar,Dc,D)
% This function creates the system of 16 equations and solves it for the
% integration constants.  PRESSURE

Fp1 = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0];
Fp2 = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
Fp3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0,0,0,0,0];
Fp4 = [0,a,0,0,0,-a,-1,0,0,0,0,0,0,0,0,0];
Fp5 = [0,Dstar,0,0,0,-Dc,0,0,0,0,0,0,0,0,0,0];
Fp6 = [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0];
Fp7 = [0,0,0,0,0,(b^2)/2,b,1,0,-(b^2)/2,-b,-1,0,0,0,0];
Fp8 = [0,0,0,0,0,(b^2)/2,b,1,0,0,0,0,0,0,-b,-1];
Fp9 = [0,0,0,0,0,b,1,0,0,-b,-1,0,0,0,0,0];
Fp10 = [0,0,0,0,0,b,1,0,0,0,0,0,0,0,-1,0];
Fp11 = [0,0,0,0,0,Dc,0,0,0,-D,0,0,0,0,0,0];
Fp12 = [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0];
Fp13 = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0];
Fp14 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0];
Fp15 = [0,0,0,0,0,0,0,0,0,0.5,1,1,0,0,0,0];
%Fp16 = [0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0]; %Clamped
Fp16 = [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0];  % Hinged

Fptotal = [Fp1;Fp2;Fp3;Fp4;Fp5;Fp6;Fp7;Fp8;Fp9;Fp10;Fp11;Fp12;Fp13;Fp14;Fp15;Fp16];
%Rp = [0;0;((a^4)/24)*(1/Dstar-1/Dc);((a^3)/6)*(1/Dstar-1/Dc);0;0;((b^4)/24)*(1/Dc-
1/D);(b^4)/(24*Dc);((b^3)/6)*(1/Dc-1/D);(b^3)/(6*Dc);0;0;0;0;1/(24*D);1/(6*D)]; %clamped
Rp = [0;0;((a^4)/24)*(1/Dstar-1/Dc);((a^3)/6)*(1/Dstar-1/Dc);0;0;((b^4)/24)*(1/Dc-
1/D);(b^4)/(24*Dc);((b^3)/6)*(1/Dc-1/D);(b^3)/(6*Dc);0;0;0;0;1/(24*D);1/(2*D)]; %hinged
CONSTp = inv(Fptotal)*Rp; % Matrix composed of integration constants, P factored out

function[CONSTth] = HFREE_CZsys2(a,b,Dstar,Dc,D,mstar)
% This function creates the system of 16 equations and solves it for the
% integration constants.  THERMAL

Ft1 = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0];
Ft2 = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
Ft3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0,0,0,0,0];
Ft4 = [0,a,0,0,0,-a,-1,0,0,0,0,0,0,0,0,0];
Ft5 = [0,Dstar,0,0,0,-Dc,0,0,0,0,0,0,0,0,0,0];
Ft6 = [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0];
Ft7 = [0,0,0,0,0,(b^2)/2,b,1,0,-(b^2)/2,-b,-1,0,0,0,0];
Ft8 = [0,0,0,0,0,(b^2)/2,b,1,0,0,0,0,0,0,-b,-1];
Ft9 = [0,0,0,0,0,b,1,0,0,-b,-1,0,0,0,0,0];
Ft10 = [0,0,0,0,0,b,1,0,0,0,0,0,0,0,-1,0];
Ft11 = [0,0,0,0,0,Dc,0,0,0,-D,0,0,0,0,0,0];
Ft12 = [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0];
Ft13 = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0];
Ft14 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0];
Ft15 = [0,0,0,0,0,0,0,0,0,0.5,1,1,0,0,0,0];
%Ft16 = [0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0]; %Clamped
Ft16 = [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0];  % Hinged
```

```
Fttotal = [Ft1;Ft2;Ft3;Ft4;Ft5;Ft6;Ft7;Ft8;Ft9;Ft10;Ft11;Ft12;Ft13;Ft14;Ft15;Ft16];
Rt = [0;0;0;0;mstar;0;0;0;0;0;0;0;0;0;0;0];
CONSTth = inv(Fttotal)*Rt; % Matrix composed of integration constants, thetatilda factored out

function[CONST] = CFREE_system1(a,Dstar,D)
% System for P

Fp1 = [0,0,1,0,0,0,0,0,0,0,0,0];
Fp2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Fp3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Fp4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Fp5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Fp6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Fp7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Fp8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Fp9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Fp10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Fp11 = [0,0,0,0,0,.5,1,1,0,0,0,0];
Fp12 = [0,0,0,0,0,1,1,0,0,0,0,0];

Fptotal = [Fp1;Fp2;Fp3;Fp4;Fp5;Fp6;Fp7;Fp8;Fp9;Fp10;Fp11;Fp12];
Rp = [0;0;(a^4)*(1/Dstar-1/D)/24;(a^4)/(24*Dstar);(a^3)*(1/Dstar-
1/D)/6;(a^3)/(6*Dstar);0;0;0;0;1/(24*D);1/(6*D)];%times p
CONST = inv(Fptotal)*Rp; % Matrix composed of integration constants, p factored out

function[CON] = CFREE_system2(a,Dstar,D,mstar)
% System for Thetatilda

F1 = [0,0,1,0,0,0,0,0,0,0,0,0];
F2 = [1,0,0,0,0,0,0,0,0,0,0,0];
F3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
F4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
F5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
F6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
F7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
F8 = [0,0,0,0,1,0,0,0,0,0,0,0];
F9 = [0,0,0,0,0,0,0,0,0,1,0,0];
F10 = [0,0,0,0,0,0,0,0,1,0,0,0];
F11 = [0,0,0,0,0,.5,1,1,0,0,0,0];
F12 = [0,0,0,0,0,1,1,0,0,0,0,0];

Ftotal = [F1;F2;F3;F4;F5;F6;F7;F8;F9;F10;F11;F12];
R = [0;0;0;0;0;0;mstar;0;0;0;0;0];%times thetatilda
CON = inv(Ftotal)*R; % Matrix composed of integration constants, thetatilda factored out

function[Pugh] = P_HFREE(a,A2P,A2T,B2P,B2T,Dstar,D,Dc,etatilda)
% This function calculates P, not Pstar....dont forget to change C and B

global gamma theta

%ugh3 = C2P-((a^2)/(2*D));
ugh2 = B2P-((a^2)/(2*Dc));
ugh1 = A2P-((a^2)/(2*Dstar));
```

```
%Z1 = (1/2)*(D*(ugh3^2)-Dstar*(ugh1^2));
%Z2 = (1/2)*(D*ugh3*C2T*theta-Dstar*ugh1*A2T*theta);
%Z3 = (1/2)*(D*C2T^2*theta^2-Dstar*A2T^2*theta^2+etatilda*theta^2)-2*gamma;

Z1 = (1/2)*(Dc*(ugh2^2)-Dstar*(ugh1^2));
Z2 = (1/2)*(Dc*ugh2*B2T*theta-Dstar*ugh1*A2T*theta);
Z3 = (1/2)*(Dc*B2T^2*theta^2-Dstar*A2T^2*theta^2+etatilda*theta^2)-2*gamma;

Z4 = sqrt(Z2^2-4*Z1*Z3);

Pugh = (-Z2+Z4)/(2*Z1);
%Pugh = (-Z2-Z4)/(2*Z1);

function[Delta] = HFREE_DELTA(CONST4p,CONST4T,Pin)
%This function calculates the characteristic deflection, delta, NOT
%DELTASTAR
global theta
w1star_0p = CONST4p;
w1star_0T = CONST4T;
Delta = w1star_0p*Pin+w1star_0T*theta; % characteristic deflection (in this case, at x=0)
```

A.2 MATLAB Code for Fixed Ends – Linear Model

```
% Separation of Patched Structure
% Patched Plate, transverse pressure/thermal load, hinged-fixed ends
% NO CONTACT ZONE!!!!

% This program uses the functions:
clear

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

% 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;
rhostar = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Dstar = Astar-rhostar*Bstar;

Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
```

```
%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-rhostar*nstar;
alpha_1 = nstar/Cstar;
etaTilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;

% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure

a = [0:0.01:0.90]; % values of a (bond length)
resultTH = zeros(12,length(a)); % allocate memory for the answers
resultN = zeros(12,length(a));
resultP = zeros(12,length(a));
global TTstar theta gamma
gamma = 0.1/2;
theta = -0.01;
TTstar = theta/sqrt(2*gamma); %Dont forget to change Pstar_HFIX function!
for count = 1:length(a);
  resultTH(:,count) = HFIXsys_TH(a(count),Dstar,D,mstar); %block tt=0
  resultN(:,count) = HFIXsys_N(a(count),Dstar,D,rhostar,h);
  resultP(:,count) = HFIXsys_P(a(count),Dstar,D);
  N_zero(count) =
N_ZEROHFIX(a(count),squeeze(resultTH(2,count)),squeeze(resultN(2,count)),squeeze(resultP(2,count)),
Dstar,alpha_1,h,rhostar,Cstar,C);
  %N_zero(count) = 0;  %When tt=0
  N_zero2(count) =
N_ZEROHFIX2(a(count),squeeze(resultTH(2,count)),squeeze(resultN(2,count)),squeeze(resultP(2,count)),
Dstar,alpha_1,h,rhostar,Cstar,C);
  %pstar(count) =
Pstar_HFIX(a(count),squeeze(resultP(2,count)),squeeze(resultN(2,count)),squeeze(resultTH(2,count)),sque
eze(resultP(6,count)),squeeze(resultN(6,count)),squeeze(resultTH(6,count)),N_zero(count),N_zero2(count)
,Dstar,D,Ce,alpha_1,etaTilda);
  P(count) =
P_HorC_FIX(a(count),squeeze(resultP(2,count)),squeeze(resultN(2,count)),squeeze(resultTH(2,count)),squ
eeze(resultP(6,count)),squeeze(resultN(6,count)),squeeze(resultTH(6,count)),N_zero(count),N_zero2(count
),Dstar,D,Ce,alpha_1,etaTilda);
  %P(count) = pstar(count)*sqrt(2*gamma);
  delta(count) = -
1*delta_HFIX(squeeze(resultN(4,count)),squeeze(resultTH(4,count)),squeeze(resultP(4,count)),N_zero(co
unt),N_zero2(count),P(count));
  Ncheck(count) = N_HFIX(N_zero(count),N_zero2(count),P(count));
  K(count)=P(count)/delta(count);
end
Results(:,1)=P';
Results(:,2)=delta';
Results(:,3)=K';

astar = ones(1,length(a)) - a;
q = 51; %when a=0.5
```

```
figure
plot(astar,P)
title('Loading Parameter')
xlabel('astar')
ylabel('P')

Pc=[0:0.1:10];

for jj=1:length(Pc);
   deflection(jj) = -
1*delta_HFIX(squeeze(resultN(4,q)),squeeze(resultTH(4,q)),squeeze(resultP(4,q)),N_zero(q),N_zero2(q),P
c(jj));
end
Pcfinal=Pc';
deflectionfinal=deflection';



% Separation of Patched Structure
% Patched Plate, transverse pressure/thermal load, CLAMPED-fixed ends
% NO CONTACT ZONE!!!!

% This program uses the functions:
clear

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

% 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;
rhostar = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Dstar = Astar-rhostar*Bstar;

Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);

%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 2; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rhostar*nstar;
alpha_1 = nstar/Cstar;
etaTilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;
```

```
% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure

a = [0:0.01:0.9]; % values of a (bond length)
resultTH = zeros(12,length(a)); % allocate memory for the answers
resultN = zeros(12,length(a));
resultP = zeros(12,length(a));

global TTstar theta gamma
gamma = 0.1/2;
theta = -0.003;
TTstar = theta/sqrt(2*gamma); %Dont forget to change Pstar_HFIX function!
for count = 1:length(a);
  resultTH(:,count) = CFIXsys_THnocz(a(count),Dstar,D,mstar); %block tt=0
  resultN(:,count) = CFIXsys_Nnocz(a(count),Dstar,D,rhostar,h);
  resultP(:,count) = CFIXsys_Pnocz(a(count),Dstar,D);
  N_zero(count) =
N_ZEROHFIX(a(count),squeeze(resultTH(2,count)),squeeze(resultN(2,count)),squeeze(resultP(2,count)),
Dstar,alpha_1,h,rhostar,Cstar,C);
  %N_zero(count) = 0;  %When Theta=0
  N_zero2(count) =
N_ZEROHFIX2(a(count),squeeze(resultTH(2,count)),squeeze(resultN(2,count)),squeeze(resultP(2,count)),
Dstar,alpha_1,h,rhostar,Cstar,C);
      %pstar(count) =
Pstar_HFIX(a(count),squeeze(resultP(2,count)),squeeze(resultN(2,count)),squeeze(resultTH(2,count)),sque
eze(resultP(6,count)),squeeze(resultN(6,count)),squeeze(resultTH(6,count)),N_zero(count),N_zero2(count)
,Dstar,D,Ce,alpha_1,etaTilda);
      %P(count) = pstar(count)*sqrt(2*gamma);
  P(count) =
P_HorC_FIX(a(count),squeeze(resultP(2,count)),squeeze(resultN(2,count)),squeeze(resultTH(2,count)),squ
eeze(resultP(6,count)),squeeze(resultN(6,count)),squeeze(resultTH(6,count)),N_zero(count),N_zero2(count
),Dstar,D,Ce,alpha_1,etaTilda);
  delta(count) = -
1*delta_HFIX(squeeze(resultN(4,count)),squeeze(resultTH(4,count)),squeeze(resultP(4,count)),N_zero(co
unt),N_zero2(count),P(count));
  K(count) = P(count)/delta(count);
  %Ncheck(count) = N_HFIX(N_zero(count),N_zero2(count),pstarcheck(count));
end
Results(:,1)=P';
Results(:,2)=delta';
Results(:,3)=K';

astar = ones(1,length(a)) - a;
q = 75; %when a=0.5

%figure
%plot(astar,pstar)
%title('Loading Parameter')
%xlabel('astar')
%ylabel('Pstar')
```

```
Pc=[0:0.1:10];

for jj=1:length(Pc);
   deflection(jj) = -
1*delta_HFIX(squeeze(resultN(4,q)),squeeze(resultTH(4,q)),squeeze(resultP(4,q)),N_zero(q),N_zero2(q),P
c(jj));
end
Pcfinal=Pc';
deflectionfinal=deflection';



% Separation of Patched Structure
% Patched Plate, thermal load, CLAMPED-FIXED ends, this program includes a
% CONTACT ZONE!!!

% This program uses the functions:
clear

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

% 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;
rhostar = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Dstar = Astar-rhostar*Bstar;

Dc = D+Dp; % bending stiffness of debonded segment
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);

%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-rhostar*nstar;
alpha_1 = nstar/Cstar;
etaTilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;

% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure

b = 0.9;
```

```
a = [0:0.01:b]; % values of a (bond length)
resultTH = zeros(16,length(a)); % allocate memory for the answers
resultN = zeros(16,length(a));
resultP = zeros(16,length(a));
global TTstar gamma theta
gamma = 0.1/2;
theta = -0.03;
TTstar = theta/sqrt(2*gamma); %Dont forget to change Pstar_CFIX function!
for count = 1:length(a);
  resultTH(:,count) = CFIXsys_TH(a(count),b,Dstar,Dc,D,mstar); %block tt=0
  resultN(:,count) = CFIXsys_N(a(count),b,Dstar,Dc,D,rhostar,h);
  resultP(:,count) = CFIXsys_P(a(count),b,Dstar,Dc,D);
  N_zero(count) =
N_ZEROHFIX(a(count),squeeze(resultTH(2,count)),squeeze(resultN(2,count)),squeeze(resultP(2,count)),
Dstar,alpha_1,h,rhostar,Cstar,C);
  N_zero2(count) =
N_ZEROHFIX2(a(count),squeeze(resultTH(2,count)),squeeze(resultN(2,count)),squeeze(resultP(2,count)),
Dstar,alpha_1,h,rhostar,Cstar,C);
  pstar(count) =
Pstar_CFIX(a(count),squeeze(resultP(2,count)),squeeze(resultN(2,count)),squeeze(resultTH(2,count)),sque
eze(resultP(6,count)),squeeze(resultN(6,count)),squeeze(resultTH(6,count)),N_zero(count),N_zero2(count)
,Dstar,Dc,Ce,alpha_1,etaTilda);
  P(count) = pstar(count)*sqrt(2*gamma);
  Ncheck(count) = N_HFIX(N_zero(count),N_zero2(count),P(count));
  delta(count) = -
1*delta_HFIX(squeeze(resultN(4,count)),squeeze(resultTH(4,count)),squeeze(resultP(4,count)),N_zero(co
unt),N_zero2(count),P(count));
  K(count)=P(count)/delta(count);
end
Results(:,1)=P';
Results(:,2)=delta';
Results(:,3)=K';

astar = ones(1,length(a)) - a;
for q = 1:length(a);
   a2 =
(squeeze(resultTH(2,q))+squeeze(resultN(2,q))*N_zero(q))*theta+(squeeze(resultN(2,q))*N_zero2(q)+squ
eeze(resultP(2,q)))*P(q);
   a4 =
(squeeze(resultTH(4,q))+squeeze(resultN(4,q))*N_zero(q))*theta+(squeeze(resultN(4,q))*N_zero2(q)+squ
eeze(resultP(4,q)))*P(q);
   b2 =
(squeeze(resultTH(6,q))+squeeze(resultN(6,q))*N_zero(q))*theta+(squeeze(resultN(6,q))*N_zero2(q)+squ
eeze(resultP(6,q)))*P(q);
   b3 =
(squeeze(resultTH(7,q))+squeeze(resultN(7,q))*N_zero(q))*theta+(squeeze(resultN(7,q))*N_zero2(q)+squ
eeze(resultP(7,q)))*P(q);
   b4 =
(squeeze(resultTH(8,q))+squeeze(resultN(8,q))*N_zero(q))*theta+(squeeze(resultN(8,q))*N_zero2(q)+squ
eeze(resultP(8,q)))*P(q);
   c2 =
(squeeze(resultTH(10,q))+squeeze(resultN(10,q))*N_zero(q))*theta+(squeeze(resultN(10,q))*N_zero2(q)+
squeeze(resultP(10,q)))*P(q);
   c3 =
(squeeze(resultTH(11,q))+squeeze(resultN(11,q))*N_zero(q))*theta+(squeeze(resultN(11,q))*N_zero2(q)+
squeeze(resultP(11,q)))*P(q);
```

```
   c4 =
(squeeze(resultTH(12,q))+squeeze(resultN(12,q))*N_zero(q))*theta+(squeeze(resultN(12,q))*N_zero2(q)+
squeeze(resultP(12,q)))*P(q);
   d3 =
(squeeze(resultTH(15,q))+squeeze(resultN(15,q))*N_zero(q))*theta+(squeeze(resultN(15,q))*N_zero2(q)+
squeeze(resultP(15,q)))*P(q);
   d4 =
(squeeze(resultTH(16,q))+squeeze(resultN(16,q))*N_zero(q))*theta+(squeeze(resultN(16,q))*N_zero2(q)+
squeeze(resultP(16,q)))*P(q);

   x1 = [0:0.01:a(q)];
   for j1 = 1:length(x1);
      w1s(q,j1) = -(P(q)*(x1(j1)^4))/(24*Dstar)+a2*(x1(j1)^2)/2+a4;
      kappa1(q,j1) =  -(P(q)*(x1(j1)^2))/(2*Dstar)+a2;
   end
   x2 = [a(q):0.01:b];
   for j2 = 1:length(x2);
      w2s(q,j2) = -(P(q)*(x2(j2)^4))/(24*Dc)+b2*(x2(j2)^2)/2+b3*x2(j2)+b4;
      kappa2(q,j2) = -(P(q)*(x2(j2)^2))/(2*Dc)+b2;
   end
   x3 = [b:0.01:1];
   for j3 = 1:length(x3);
      w3(q,j3) = -(P(q)*(x3(j3)^4))/(24*D)+c2*(x3(j3)^2)/2+c3*x3(j3)+c4;
      kappa3(q,j3) = -(P(q)*(x3(j3)^2))/(2*D)+c2;
   end

   for ja = 1:length(x2);
      if kappa2(q,ja)<=0;
         CONBOND(q)=astar(q);
      else
         CONBOND(q)=0;
        break;
      end
   end

end
ii=51;
x1plot = [0:0.01:a(ii)];
x2plot = [a(ii):0.01:b];
x3plot = [b:0.01:1];
figure
plot(x1plot,-w1s(ii,1:length(x1plot)),'b');
hold on
plot(x2plot,-w2s(ii,1:length(x2plot)),'g');
hold on
plot(x3plot,-w3(ii,1:length(x3plot)),'r');

qq=51;
Pc=[0:0.1:25];
for jj=1:length(Pc);
   deflection(jj) = -
1*delta_HFIX(squeeze(resultN(4,qq)),squeeze(resultTH(4,qq)),squeeze(resultP(4,qq)),N_zero(qq),N_zero2
(qq),Pc(jj));
end
Pcfinal=Pc';
```

```
deflectionfinal=deflection';
ValidCZ=CONBOND';


function[CONSTn] = HFIXsys_N(a,Dstar,D,rhon,hn)
% This function creates the system of 12 equations and solves it for the
% integration constants.

Fn1 = [0,0,1,0,0,0,0,0,0,0,0,0];
Fn2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Fn3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Fn4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Fn5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Fn6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Fn7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Fn8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Fn9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Fn10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Fn11 = [0,0,0,0,0,0.5,1,1,0,0,0,0];
Fn12 = [0,0,0,0,0,1,0,0,0,0,0,0];

Fntotal = [Fn1;Fn2;Fn3;Fn4;Fn5;Fn6;Fn7;Fn8;Fn9;Fn10;Fn11;Fn12];
Rn = [0;0;0;0;0;0;-(rhon+hn/2);0;0;0;0;0];
CONSTn = inv(Fntotal)*Rn; % Matrix composed of integration constants, N0 factored out


function[CONSTp] = HFIXsys_P(a,Dstar,D)
% This function creates the system of 12 equations and solves it for the
% integration constants.

Fp1 = [0,0,1,0,0,0,0,0,0,0,0,0];
Fp2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Fp3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Fp4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Fp5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Fp6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Fp7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Fp8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Fp9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Fp10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Fp11 = [0,0,0,0,0,0.5,1,1,0,0,0,0];
Fp12 = [0,0,0,0,0,1,0,0,0,0,0,0];

Fptotal = [Fp1;Fp2;Fp3;Fp4;Fp5;Fp6;Fp7;Fp8;Fp9;Fp10;Fp11;Fp12];
Rp = [0;0;((a^4)/24)*(1/Dstar-1/D);(a^4)/(24*Dstar);((a^3)/6)*(1/Dstar-
1/D);(a^3)/(6*Dstar);0;0;0;0;1/(24*D);1/(2*D)]; %times P
CONSTp = inv(Fptotal)*Rp; % Matrix composed of integration constants, P factored out


function[CONSTth] = HFIXsys_TH(a,Dstar,D,mstar)
% This function creates the system of 12 equations and solves it for the
% integration constants.

Ft1 = [0,0,1,0,0,0,0,0,0,0,0,0];
```

```
Ft2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Ft3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Ft4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Ft5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Ft6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Ft7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Ft8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Ft9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Ft10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Ft11 = [0,0,0,0,0,0.5,1,1,0,0,0,0];
Ft12 = [0,0,0,0,0,1,0,0,0,0,0,0];

Fttotal = [Ft1;Ft2;Ft3;Ft4;Ft5;Ft6;Ft7;Ft8;Ft9;Ft10;Ft11;Ft12];
Rt = [0;0;0;0;0;0;mstar;0;0;0;0;0];
CONSTth = inv(Fttotal)*Rt; % Matrix composed of integration constants, thetatilda factored out


function[CONSTn] = CFIXsys_Nnocz(a,Dstar,D,rhon,hn)
% This function creates the system of 12 equations and solves it for the
% integration constants.

Fn1 = [0,0,1,0,0,0,0,0,0,0,0,0];
Fn2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Fn3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Fn4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Fn5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Fn6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Fn7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Fn8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Fn9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Fn10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Fn11 = [0,0,0,0,0,0.5,1,1,0,0,0,0];
Fn12 = [0,0,0,0,0,1,1,0,0,0,0,0];

Fntotal = [Fn1;Fn2;Fn3;Fn4;Fn5;Fn6;Fn7;Fn8;Fn9;Fn10;Fn11;Fn12];
Rn = [0;0;0;0;0;0;-(rhon+hn/2);0;0;0;0;0];
CONSTn = inv(Fntotal)*Rn; % Matrix composed of integration constants, N0 factored out


function[CONSTp] = CFIXsys_Pnocz(a,Dstar,D)
% This function creates the system of 12 equations and solves it for the
% integration constants.

Fp1 = [0,0,1,0,0,0,0,0,0,0,0,0];
Fp2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Fp3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Fp4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Fp5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Fp6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Fp7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Fp8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Fp9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Fp10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Fp11 = [0,0,0,0,0,0.5,1,1,0,0,0,0];
```

```
Fp12 = [0,0,0,0,0,1,1,0,0,0,0,0];

Fptotal = [Fp1;Fp2;Fp3;Fp4;Fp5;Fp6;Fp7;Fp8;Fp9;Fp10;Fp11;Fp12];
Rp = [0;0;((a^4)/24)*(1/Dstar-1/D);(a^4)/(24*Dstar);((a^3)/6)*(1/Dstar-
1/D);(a^3)/(6*Dstar);0;0;0;0;1/(24*D);1/(6*D)]; %times P
CONSTp = inv(Fptotal)*Rp; % Matrix composed of integration constants, P factored out


function[CONSTth] = CFIXsys_THnocz(a,Dstar,D,mstar)
% This function creates the system of 12 equations and solves it for the
% integration constants.

Ft1 = [0,0,1,0,0,0,0,0,0,0,0,0];
Ft2 = [1,0,0,0,0,0,0,0,0,0,0,0];
Ft3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0];
Ft4 = [0,(a^2)/2,0,1,0,0,0,0,0,0,-a,-1];
Ft5 = [0,a,0,0,0,-a,-1,0,0,0,0,0];
Ft6 = [0,a,0,0,0,0,0,0,0,0,-1,0];
Ft7 = [0,Dstar,0,0,0,-D,0,0,0,0,0,0];
Ft8 = [0,0,0,0,1,0,0,0,0,0,0,0];
Ft9 = [0,0,0,0,0,0,0,0,0,1,0,0];
Ft10 = [0,0,0,0,0,0,0,0,1,0,0,0];
Ft11 = [0,0,0,0,0,0.5,1,1,0,0,0,0];
Ft12 = [0,0,0,0,0,1,1,0,0,0,0,0];

Fttotal = [Ft1;Ft2;Ft3;Ft4;Ft5;Ft6;Ft7;Ft8;Ft9;Ft10;Ft11;Ft12];
Rt = [0;0;0;0;0;0;mstar;0;0;0;0;0];
CONSTth = inv(Fttotal)*Rt; % Matrix composed of integration constants, thetatilda factored out


function[CONSTn] = CFIXsys_N(a,b,Dstar,Dc,D,rhon,hn)
% This function creates the system of 16 equations and solves it for the
% integration constants.

Fn1 = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0];
Fn2 = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
Fn3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0,0,0,0,0];
Fn4 = [0,a,0,0,0,-a,-1,0,0,0,0,0,0,0,0,0];
Fn5 = [0,Dstar,0,0,0,-Dc,0,0,0,0,0,0,0,0,0,0];
Fn6 = [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0];
Fn7 = [0,0,0,0,0,(b^2)/2,b,1,0,-(b^2)/2,-b,-1,0,0,0,0];
Fn8 = [0,0,0,0,0,(b^2)/2,b,1,0,0,0,0,0,0,-b,-1];
Fn9 = [0,0,0,0,0,b,1,0,0,-b,-1,0,0,0,0,0];
Fn10 = [0,0,0,0,0,b,1,0,0,0,0,0,0,0,-1,0];
Fn11 = [0,0,0,0,0,Dc,0,0,0,-D,0,0,0,0,0,0];
Fn12 = [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0];
Fn13 = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0];
Fn14 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0];
Fn15 = [0,0,0,0,0,0,0,0,0,0.5,1,1,0,0,0,0];
%Fn16 = [0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0]; %Clamped
Fn16 = [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0];  % Hinged

Fntotal = [Fn1;Fn2;Fn3;Fn4;Fn5;Fn6;Fn7;Fn8;Fn9;Fn10;Fn11;Fn12;Fn13;Fn14;Fn15;Fn16];
Rn = [0;0;0;0;-(rhon+hn/2);0;0;0;0;0;0;0;0;0;0;0];
```

```
CONSTn = inv(Fntotal)*Rn; % Matrix composed of integration constants, N0 factored out


function[CONSTp] = CFIXsys_P(a,b,Dstar,Dc,D)
% This function creates the system of 16 equations and solves it for the
% integration constants.

Fp1 = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0];
Fp2 = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
Fp3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0,0,0,0,0];
Fp4 = [0,a,0,0,0,-a,-1,0,0,0,0,0,0,0,0,0];
Fp5 = [0,Dstar,0,0,0,-Dc,0,0,0,0,0,0,0,0,0,0];
Fp6 = [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0];
Fp7 = [0,0,0,0,0,(b^2)/2,b,1,0,-(b^2)/2,-b,-1,0,0,0,0];
Fp8 = [0,0,0,0,0,(b^2)/2,b,1,0,0,0,0,0,0,-b,-1];
Fp9 = [0,0,0,0,0,b,1,0,0,-b,-1,0,0,0,0,0];
Fp10 = [0,0,0,0,0,b,1,0,0,0,0,0,0,0,-1,0];
Fp11 = [0,0,0,0,0,Dc,0,0,0,-D,0,0,0,0,0,0];
Fp12 = [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0];
Fp13 = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0];
Fp14 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0];
Fp15 = [0,0,0,0,0,0,0,0,0,0.5,1,1,0,0,0,0];
%Fp16 = [0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0]; %Clamped
Fp16 = [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0];  % Hinged

Fptotal = [Fp1;Fp2;Fp3;Fp4;Fp5;Fp6;Fp7;Fp8;Fp9;Fp10;Fp11;Fp12;Fp13;Fp14;Fp15;Fp16];
%Rp = [0;0;((a^4)/24)*(1/Dstar-1/Dc);((a^3)/6)*(1/Dstar-1/Dc);0;0;((b^4)/24)*(1/Dc-
1/D);(b^4)/(24*Dc);((b^3)/6)*(1/Dc-1/D);(b^3)/(6*Dc);0;0;0;0;1/(24*D);1/(6*D)]; %clamped
Rp = [0;0;((a^4)/24)*(1/Dstar-1/Dc);((a^3)/6)*(1/Dstar-1/Dc);0;0;((b^4)/24)*(1/Dc-
1/D);(b^4)/(24*Dc);((b^3)/6)*(1/Dc-1/D);(b^3)/(6*Dc);0;0;0;0;1/(24*D);1/(2*D)]; %hinged
CONSTp = inv(Fptotal)*Rp; % Matrix composed of integration constants, P factored out


function[CONSTth] = CFIXsys_TH(a,b,Dstar,Dc,D,mstar)
% This function creates the system of 16 equations and solves it for the
% integration constants.

Ft1 = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0];
Ft2 = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
Ft3 = [0,(a^2)/2,0,1,0,-(a^2)/2,-a,-1,0,0,0,0,0,0,0,0];
Ft4 = [0,a,0,0,0,-a,-1,0,0,0,0,0,0,0,0,0];
Ft5 = [0,Dstar,0,0,0,-Dc,0,0,0,0,0,0,0,0,0,0];
Ft6 = [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0];
Ft7 = [0,0,0,0,0,(b^2)/2,b,1,0,-(b^2)/2,-b,-1,0,0,0,0];
Ft8 = [0,0,0,0,0,(b^2)/2,b,1,0,0,0,0,0,0,-b,-1];
Ft9 = [0,0,0,0,0,b,1,0,0,-b,-1,0,0,0,0,0];
Ft10 = [0,0,0,0,0,b,1,0,0,0,0,0,0,0,-1,0];
Ft11 = [0,0,0,0,0,Dc,0,0,0,-D,0,0,0,0,0,0];
Ft12 = [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0];
Ft13 = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0];
Ft14 = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0];
Ft15 = [0,0,0,0,0,0,0,0,0,0.5,1,1,0,0,0,0];
%Ft16 = [0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0]; %Clamped
Ft16 = [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0];  % Hinged
```

```
Fttotal = [Ft1;Ft2;Ft3;Ft4;Ft5;Ft6;Ft7;Ft8;Ft9;Ft10;Ft11;Ft12;Ft13;Ft14;Ft15;Ft16];
Rt = [0;0;0;0;mstar;0;0;0;0;0;0;0;0;0;0;0];
CONSTth = inv(Fttotal)*Rt; % Matrix composed of integration constants, thetatilda factored out


function[NNN] = N_HFIX(NOT,NOP,P_ch)
global TTstar
NNN = NOT*TTstar+NOP*P_ch;


function[NT] = N_ZEROHFIX(aa,CONST_2th,CONST_2N,CONST_2P,Dstar,alpha11,hh,rrho,Cstar,C)

% This function calculates N0T as a function of thetatilda

% An expression for w1starprime(a) may be expressed as
% f*thetatilda+g*N0+j*P
f_a = CONST_2th*aa; % times theta
g_a = CONST_2N*aa; % times N0
j_a = -(aa^3)/(6*Dstar)+CONST_2P*aa; %times P

% N0 may be expressed as a function of a, thetatilda, and P, N0 = NoT*thetatilda+NoP*P. This expression
comes from
% the integrability condition:
C_a = aa/Cstar-aa/C+1/C;
alpha_of_a = 1-aa+alpha11*aa;
rhat = (hh/2)+rrho;

NT = (rhat*f_a-alpha_of_a)/(C_a-rhat*g_a);


function[NP] = N_ZEROHFIX2(aa,CONST_2th,CONST_2N,CONST_2P,Dstar,alpha11,hh,rrho,Cstar,C)

% This function calculates N0P as a function of P

% An expression for w1starprime(a) may be expressed as
% f*thetatilda+g*N0+j*P
f_a = CONST_2th*aa; % times theta
g_a = CONST_2N*aa; % times N0
j_a = -(aa^3)/(6*Dstar)+CONST_2P*aa; %times P

% N0 may be expressed as a function of a, thetatilda, and P, N0 = NoT*thetatilda+NoP*P. This expression
comes from
% the integrability condition:
C_a = aa/Cstar-aa/C+1/C;
%alpha_of_a = 1-aa+alpha11*aa;
rhat = (hh/2)+rrho;

NP = (rhat*j_a)/(C_a-rhat*g_a);


function[Pout] = P_HorC_FIX(a,A2P,A2N,A2T,C2P,C2N,C2T,N0t,N0p,Dstar,D,Ce,alph1,etatilda)
```

```
% This function calculates P from the values obtained in the matrix
% solution, with a little help from Maple.
global theta gamma

% Trans. Con. is such that Q1P^2+Q2P+Q3=2gamma

Q1 = -1/2*a^2*C2P+1/2*D*C2N^2*N0p^2+1/2*a^2*A2P-
1/2*Dstar*A2N^2*N0p^2+1/8*a^4/D+1/2*D*C2P^2-1/8*a^4/Dstar-1/2*Dstar*A2P^2+1/2/Ce*N0p^2-
1/2*a^2*C2N*N0p+D*C2N*N0p*C2P+1/2*a^2*A2N*N0p-Dstar*A2N*N0p*A2P;
Q2 = -theta*N0p*alph1-1/2*a^2*C2T*theta-
1/2*a^2*C2N*N0t*theta+D*C2T*theta*C2P+D*C2T*theta*C2N*N0p+D*C2N^2*N0t*theta*N0p+D*C2
N*N0t*theta*C2P+1/2*a^2*A2T*theta+1/2*a^2*A2N*N0t*theta-Dstar*A2T*theta*A2P-
Dstar*A2T*theta*A2N*N0p-Dstar*A2N^2*N0t*theta*N0p-
Dstar*A2N*N0t*theta*A2P+1/Ce*N0t*theta*N0p+theta*N0p;
Q3 = 1/2*D*C2N^2*N0t^2*theta^2-
1/2*Dstar*A2N^2*N0t^2*theta^2+1/2*etatilda*theta^2+N0t*theta^2+1/2*D*C2T^2*theta^2-
1/2*Dstar*A2T^2*theta^2+1/2/Ce*N0t^2*theta^2+D*C2T*theta^2*C2N*N0t-
Dstar*A2T*theta^2*A2N*N0t-N0t*theta^2*alph1;
Q4 = Q3-2*gamma;

Pout = (-Q2+sqrt(Q2^2-4*Q1*Q4))/(2*Q1);
%Pout = (-Q2-sqrt(Q2^2-4*Q1*Q4))/(2*Q1);


function[P] = Pstar_CFIX(a,A2P,A2N,A2T,B2P,B2N,B2T,N0t,N0p,Dstar,Dc,Ce,alph1,etatilda)
% This function uses the constants determined from "therm_system1" to calculate
% value of pstar, the quadratic formula is used to find pstar from G.

%kappa2 and kappa1 may be expressed as: J2*thetatilda+Q2*P, and
%J1*thetatilda+q1*P.

J2 = B2T+B2N*N0t;
Q2 = -(a^2)/(2*Dc)+B2N*N0p+B2P;
J1 = A2T+A2N*N0t;
Q1 = -(a^2)/(2*Dstar)+A2N*N0p+A2P;

%Constants from transversality equation G = 2gamma
Z1 = (Dc/2)*(J2^2)-(Dstar/2)*(J1^2)+(N0t^2)/(2*Ce)+N0t*(1-alph1)+(etatilda/2);
Z2 = Dc*J2*Q2-Dstar*J2*Q2+(N0t*N0p)/Ce+N0p*(1-alph1);
%Z1 = 0; %tt=0
%Z2 = 0;  %tt=0
Z3 = (Dc/2)*(Q2^2)-(Dstar/2)*(Q1^2)+(N0p^2)/(2*Ce);

global TTstar
TTstar_sq = TTstar^2;

%more constants to obtain a quadratic polynomial
Z4 = Z1*TTstar_sq-1;
Z5 = Z2*TTstar;

P = (-Z5+sqrt((Z5^2)-4*Z3*Z4))/(2*Z3); % solution 1
%P = (-Z5-sqrt((Z5^2)-4*Z3*Z4))/(2*Z3); % solution 2, gives unrealistic
%result
```

```
function[del] = delta_HFIX(A4N,A4T,A4P,NT,NP,Pres)
%This function Calculates the deflection w1(0)
%Delta, NOT delta_star!
global TTstar theta
%del = A4N*(NT*TTstar+NP*Pres)+A4T*TTstar+A4P*Pres;
del = A4N*(NT*theta+NP*Pres)+A4T*theta+A4P*Pres;


function[GG] = OMEG_HFIX(a,CON2th,CON2n,N_0,Dstar,D,etatilda,Ce,alpha1)
% This function uses the constants determined from "therm_system3's" to calculate
% value of omega
w3_2prime_a =0; % times thetatilda
w1star_2prime_a = CON2th+CON2n*N_0; % times thetatilda
GG = 0.5*(D*(w3_2prime_a^2)-Dstar*(w1star_2prime_a^2)+(N_0^2)/Ce+2*N_0*(1-alpha1)+etatilda); %
times thetatilda^2
```

A.3 Maple Code for Fixed Ends – Nonlinear Model

```
> restart:
> # These functions F are defined in my notes
> #Transverse Pressure and Thermal Load, hinged-fixed
> #also works for clamped-fixed
> #No CZ
> #This is the "P" version!
> c2 := P*F1+F2+theta*F3:
> c1 := P*F4+F5+theta*F6:
> a1 := P*F7+F8+theta*F9:
> c3 := P*F10+F11+theta*F12:
> c4 := P*F13+F14+theta*F15:
> a4 := P*F16+F17+theta*F18:
> a2 := 0:
> a3 := 0:
> w1prime := a1*k1*sinh(k1*x)+a2*k1*cosh(k1*x)+a3+P*x/Nn:
> w1prime_at_a := subs(x=an,w1prime):
> type(w1prime_at_a,polynom(anything,P)):
> degree(w1prime_at_a,P):
> B1 := coeff(w1prime_at_a,P,1):
> B2 := coeff(w1prime_at_a,P,0):
> w3prime := c1*k3*sinh(k3*x)+c2*k3*cosh(k3*x)+c3+P*x/Nn:
> w1prime_sq := expand(w1prime^2):
> w3prime_sq := expand(w3prime^2):
> IC1 := int(w1prime_sq, x = 0..an):
> IC3 := int(w3prime_sq, x = an..1):
> type(IC1,polynom(anything,P)):
> degree(IC1,P):
```

```
> type(IC3,polynom(anything,P)):
> degree(IC3,P):
> B3 := coeff(IC1,P,2):
> B4 := coeff(IC1,P,1):
> B5 := coeff(IC1,P,0):
> B6 := coeff(IC3,P,2):
> B7 := coeff(IC3,P,1):
> B8 := coeff(IC3,P,0):
> #Ca = 1/C-a/C+a/C* and rt = h/2+rho*
> #I will once again use lower case 'b' to keep things tidy
> # This is the same as in the compression case (sort of)!
> B0 := (1-an)*1+an*alpha_1: #alpha=1
> int_con := Nn*Ca+b0*theta-rt*(P*b1+b2)-(1/2)*((P^2)*b3+P*b4+b5)-
(1/2)*((P^2)*b6+P*b7+b8):
> type(int_con, polynom(anything,P)):
> #Separate the coeffs of P in the integ con
> Q1 := coeff(int_con,P,2):
> Q2 := coeff(int_con,P,1):
> Q3 := coeff(int_con,P,0):
> #Transversality equation...
> kappa3 := c1*(k3^2)*cosh(k3*an)+c2*(k3^2)*sinh(k3*an)+P/Nn:
> kappa3_sq := expand(kappa3^2):
> kappa1 := a1*(k1^2)*cosh(k1*an)+a2*(k1^2)*sinh(k1*an)+P/Nn:
> kappa1_sq := expand(kappa1^2):
> Trans_eq := (1/2)*(D3*kappa3_sq-Ds*kappa1_sq+(Nn^2)/Ce+2*Nn*(1-
alpha_1)*theta+etatilda*theta^2):
> type(Trans_eq,polynom(anything,P)):
> degree(Trans_eq,P):
> Q4 := coeff(Trans_eq,P,2):
> Q5 := coeff(Trans_eq,P,1):
> Q6 := coeff(Trans_eq,P,0):


> restart:
> # This is the ThermPress CONTACT ZONE solution!
> a2:=0:
> a3:=0:
> b2:=F1*P+F2*theta+F3:
> b1:=F4*P+F5*theta+F6:
> c2:=F7*P+F8*theta+F9:
> c1:=F10*P+F11*theta+F12:
> a1:=F13*P+F14*theta+F15:
> c3:=F16*P+F17*theta+F18:
> c4:=F19*P+F20*theta+F21:
> b3:=F22*P+F23*theta+F24:
> b4:=F25*P+F26*theta+F27:
```

```
> a4:=F28*P+F29*theta+F30:
> w1_prime := a1*k1*sinh(k1*x)+a2*k1*cosh(k1*x)+a3+P*x/Nn:
> w2_prime := b1*k2*sinh(k2*x)+b2*k2*cosh(k2*x)+b3+P*x/Nn:
> w3_prime := c1*k3*sinh(k3*x)+c2*k3*cosh(k3*x)+c3+P*x/Nn:
> w1prime_at_a := subs(x=an,w1_prime):
> type(w1prime_at_a,polynom(anything,P)):
> B_1 := coeff(w1prime_at_a,P,1):
> B_2 := coeff(w1prime_at_a,P,0):
> w1prime_sq := expand(w1_prime^2):
> w2prime_sq := expand(w2_prime^2):
> w3prime_sq := expand(w3_prime^2):
> IC1 := int(w1prime_sq, x = 0..an):
> IC2 := int(w2prime_sq, x = an..b):
> IC3 := int(w3prime_sq, x = b..1):
> type(IC1,polynom(anything,P)):
> degree(IC1,P):
> type(IC2,polynom(anything,P)):
> degree(IC2,P):
> type(IC3,polynom(anything,P)):
> degree(IC3,P):
> B_3 := coeff(IC1,P,2):
> B_4 := coeff(IC1,P,1):
> B_5 := coeff(IC1,P,0):
> B_6 := coeff(IC2,P,2):
> B_7 := coeff(IC2,P,1):
> B_8 := coeff(IC2,P,0):
> B_9 := coeff(IC3,P,2):
> B_10 := coeff(IC3,P,1):
> B_11 := coeff(IC3,P,0):
> #Ca = 1/C-a/C+a/C* and rt = h/2+rho*
> B_0 := (1-an)*1+an*alpha_1: #alpha=1
> int_con := Nn*Ca+B0*theta-rt*(P*B1+B2)-(1/2)*((P^2)*B3+P*B4+B5)-
(1/2)*((P^2)*B6+P*B7+B8)-(1/2)*((P^2)*B9+P*B10+B11):
> type(int_con, polynom(anything,P)):
> Q1 := coeff(int_con,P,2):
> Q2 := coeff(int_con,P,1):
> Q3 := coeff(int_con,P,0):
> #Transversality equation...
> kappa2 := b1*(k2^2)*cosh(k2*an)+b2*(k2^2)*sinh(k2*an)+P/Nn:
> kappa2_sq := expand(kappa2^2):
> kappa1 := a1*(k1^2)*cosh(k1*an)+a2*(k1^2)*sinh(k1*an)+P/Nn:
> kappa1_sq := expand(kappa1^2):
> Trans_eq := (1/2)*(Dc*kappa2_sq-Ds*kappa1_sq+(Nn^2)/Ce+2*Nn*(1-
alpha_1)*theta+etatilda*theta^2):
> type(Trans_eq,polynom(anything,P)):
> degree(Trans_eq,P):
```

> **Q4 := coeff(Trans_eq,P,2):**
> **Q5 := coeff(Trans_eq,P,1):**
> **Q6 := coeff(Trans_eq,P,0):**

A.4 MATLAB Code for Fixed Ends – Nonlinear Model

```
% Separation of Patched Structure
% Patched Plate, thermal load,                HINGED-FIXED ends
% Non-Linear  TENSION!!!
% This is the "P" version
% This program uses the functions:
clear
global h C D3 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
D3 = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rhostar Ds
% stiffnesses of composite structure
Astar = D3+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rhostar = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rhostar*Bstar; %Dstar
global  Dc Cs Ce DI DII
Dc = D3+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
DI = (Dp*D3)/Dc;
DII = (D3^2)/Dc;
global mstar alpha_1 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-rhostar*nstar;
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;

% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure

m = 1; %initialize index value for result array
n=1;
```

```
q=1;
%Set 2*gamma
global gamma
gamma = 0.1/2;

global theta
theta = -0.01;

a = [0.01:0.01:0.9];
N = [0:.1:100];

for i = 1:length(a);
  global an;
   an = a(i);
   for j = 1:length(N);
      [Big(j,1),Big(j,2),Big(j,3),Big(j,4),Big(j,5),Big(j,6)] = BigFuntentherm2(N(j));
      %[value of big F, P ,Delta~w1(0),err1,err2,K]
   end
   for k = 1:length(N)-1;
      if Big(k,1)*Big(k+1,1)<0;
         Nroot(k) = fzero(@BigFuntentherm2_root,N(k));
         group(m,1) = a(i); %'a'
         group(m,2)= 1 - a(i); % a*
         group(m,3) = Nroot(k); % The value of N that's a root
         [blah,group(m,4),group(m,5),group(m,6),group(m,7),group(m,8)] = BigFuntentherm2(Nroot(k));
%[value of BF=0, P, delta,err1,err2,K]
         group(m,9) = group(m,7)+group(m,6); % Should be 1
         %To assemble a matix of nice results...
         if group(m,4)>0;
         Results(n,1)= 1 - a(i); %a*
         Results(n,2) = Nroot(k); %N
         Results(n,3) = group(m,4); %P
         Results(n,4) = group(m,5); %Delta
         Results(n,5) = group(m,8); %K
         n=n+1;
         end
         m = m+1;
      else
      end
   end
end
q_end=n-1; %or n-1
for q = 1:q_end;
   Xend=1-Results(q,1);
x1 = [0:0.01:Xend];
x3 = [Xend:0.01:1];
[a1,a4,c1,c2,c3,c4,K1,K3] = BFtenthermCONST(Xend,Results(q,2),Results(q,3),theta);
         for j1 = 1:length(x1);
            w1star(q,j1) = a1*cosh(K1*x1(j1))+a4+(Results(q,3)*(x1(j1))^2)/(2*Results(q,2));
            %kappa1(q,j1) =
a1*K1^2*cosh(K1*x1(j1))+a2*K1^2*sinh(K1*x1(j1))+Results(q,3)/Results(q,2);
         end
         for j3 = 1:length(x3);
            w3(q,j3) =
c1*cosh(K3*x3(j3))+c2*sinh(K3*x3(j3))+c3*x3(j3)+c4+(Results(q,3)*(x3(j3))^2)/(2*Results(q,2));
```

```
            %kappa3(q,j3) =
c1*K3^2*cosh(K3*x3(j3))+c2*K3^2*sinh(K3*x3(j3))+Results(q,3)/Results(q,2);
        end
end

ii=51;
x1plot = [0:0.01:1-Results(ii,1)];
x3plot = [1-Results(ii,1):0.01:1];
figure
plot(x1plot,-w1star(ii,1:length(x1plot)),'b');
hold on
plot(x3plot,-w3(ii,1:length(x3plot)),'r');

%Plot of deflection of Chosen root
%atest =0.8 ;
%Ntest = 39.135;
%TTtest = -0.005;
%Ptest = 4.0056;
%[a1,a4,c1,c2,c3,c4] = BFtenthermCONST(atest,Ntest,Ptest,TTtest);
%k1test = sqrt(Ntest/Ds);
%k3test = sqrt(Ntest/D3);
%x1 = [0:0.01:atest];
%for j1 = 1:length(x1);
%    w1(j1) = -1*(a1*cosh(k1test*x1(j1))+a4+Ptest*(x1(j1)^2)/(2*Ntest));
%end
%x3 = [atest:0.01:1];
%for j3 = 1:length(x3);
%    w3(j3) = -
1*(c1*cosh(k3test*x3(j3))+c2*sinh(k3test*x3(j3))+c3*x3(j3)+c4+Ptest*(x3(j3)^2)/(2*Ntest));
%end
%figure
%plot(x1,w1,'r')
%hold on
%plot(x3,w3,'b')
%title('Deflection' )
%xlabel('x')
%ylabel('deflection')

qq=1;
global an Pin
an=0.5;
Pin_test=[-0.4:0.1:0.2];
Nin=[-1:.001:3];
for ib = 1:length(Pin_test);
   Pin = Pin_test(ib);
   for jb = 1:length(Nin);
      I_C(jb) = DEF_HFIX(Nin(jb));
   end
   for kb = 1:length(Nin)-1;
      if I_C(kb)*I_C(kb+1)<0;
         Nroot2(kb) = fzero(@DEF_HFIX,Nin(kb));
         group2(q,1)= Nroot2(kb); % N root value
         group2(q,2) = Pin_test(ib); %"P"
         [group2(q,3),group2(q,4)] = DEF_HFIX_A4(group2(q,1),group2(q,2)); %deflection, IC=0
         q = q+1;
```

```
      else
      end
   end
end

function[BF,PBF,DelBF,MODE1,MODE2,Kstiff] = BigFuntentherm2(Nn)
   %Ds,D3,rhostar,h,C,Cstar,Ce,gamma,P,mstar,alpha_1,etatilda
%Using my hand solution, this function solves the non linear IC, and TC for P
%This is the "P" version
global an Ds D3 Dc DI DII rhostar h C Cstar Ce gamma theta mstar alpha_1 etatilda

k1 = sqrt(Nn/Ds);
k3 = sqrt(Nn/D3);
y = k1*an;
x = k3*an;
%Some general stuff to get started
GC1 = D3*(k3^2)*(k3*sinh(x)*cosh(y)-k1*cosh(x)*sinh(y));
GC2 = D3*(k3^2)*(k3*cosh(x)*cosh(y)-k1*sinh(x)*sinh(y));
C2den = GC2/GC1-sinh(k3)/cosh(k3);
A1den = Ds*(k1^3)*sinh(y);
%Now, the F's
% c2
F1 = (-Ds+D3)*(k1*sinh(y))/(Nn*GC1*C2den)+1/(Nn*(k3^2)*cosh(k3)*C2den);
F2 = ((k1*sinh(y))/(GC1*C2den))*(-h*Nn/2-rhostar*Nn);
F3 = ((k1*sinh(y))/(GC1*C2den))*mstar;
% c1
F4 = (k1*sinh(y)*(-Ds+D3))/(Nn*GC1)-F1*GC2/GC1;
F5 = (k1*sinh(y)*(-h*Nn/2-rhostar*Nn))/GC1-F2*GC2/GC1;
F6 = (k1*sinh(y)*mstar)/GC1-F3*GC2/GC1;
% a1
F7 = D3*(k3^3)*sinh(x)*F4/A1den+D3*(k3^3)*cosh(x)*F1/A1den;
F8 = D3*(k3^3)*sinh(x)*F5/A1den+D3*(k3^3)*cosh(x)*F2/A1den;
F9 = D3*(k3^3)*sinh(x)*F6/A1den+D3*(k3^3)*cosh(x)*F3/A1den;
% c3
F10 = F7*k1*sinh(y)-F4*k3*sinh(x)-F1*k3*cosh(x);
F11 = F8*k1*sinh(y)-F5*k3*sinh(x)-F2*k3*cosh(x);
F12 = F9*k1*sinh(y)-F6*k3*sinh(x)-F3*k3*cosh(x);
%c4
F13 = -1/(2*Nn)-F10-F1*sinh(k3)-F4*cosh(k3);
F14 = -F11-F2*sinh(k3)-F5*cosh(k3);
F15 = -F12-F3*sinh(k3)-F6*cosh(k3);
%a4
F16 = F4*cosh(x)+F1*sinh(x)+F10*an+F13-F7*cosh(y);
F17 = F5*cosh(x)+F2*sinh(x)+F11*an+F14-F8*cosh(y);
F18 = F6*cosh(x)+F3*sinh(x)+F12*an+F15-F9*cosh(y);

%Now the B's
B0 = 1-an+an*alpha_1;
B1 = F7*k1*sinh(k1*an)+an/Nn;
B2 = (F8+theta*F9)*k1*sinh(k1*an);

B3 = -1/6*(-3*k1^2*F7^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*F7^2*Nn^2*an-
12*F7*Nn*k1*an*cosh(k1*an)+12*F7*Nn*sinh(k1*an)-2*an^3*k1)/k1/Nn^2;
B4 = -1/6*(-6*k1^2*F7*F8*Nn^2*cosh(k1*an)*sinh(k1*an)+6*k1^3*F7*F8*Nn^2*an-
6*k1^2*F7*theta*F9*Nn^2*cosh(k1*an)*sinh(k1*an)+...
```

```
   6*k1^3*F7*theta*F9*Nn^2*an-12*F8*Nn*k1*an*cosh(k1*an)+12*F8*Nn*sinh(k1*an)-
12*theta*F9*Nn*k1*an*cosh(k1*an)+12*theta*F9*Nn*sinh(k1*an))/k1/Nn^2;
B5 = -1/6*(-3*k1^2*F8^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*F8^2*Nn^2*an-
6*k1^2*F8*theta*F9*Nn^2*cosh(k1*an)*sinh(k1*an)+...
   6*k1^3*F8*theta*F9*Nn^2*an-
3*k1^2*theta^2*F9^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*theta^2*F9^2*Nn^2*an)/k1/Nn^2;
B6 = -1/6*(-2*k3+2*an^3*k3+6*k3^2*F4*F1*sinh(k3*an)^2*Nn^2-
12*F1*Nn*k3*sinh(k3)+3*k3^3*F4^2*Nn^2-3*k3^3*F1^2*Nn^2+12*F4*Nn*sinh(k3)+...
   12*F1*Nn*cosh(k3)-6*F10^2*Nn^2*k3-6*F10*Nn*k3-12*F4*Nn*sinh(k3*an)-
12*F1*Nn*cosh(k3*an)+6*F10*an^2*Nn*k3-12*F4*F10*cosh(k3)*Nn^2*k3-...
   12*F1*F10*sinh(k3)*Nn^2*k3-12*F4*Nn*k3*cosh(k3)+6*F10^2*an*Nn^2*k3-
3*k3^3*F4^2*Nn^2*an-6*k3^2*F4*F1*sinh(k3)^2*Nn^2-...
   3*k3^2*F4^2*Nn^2*sinh(k3)*cosh(k3)-
3*k3^2*F1^2*Nn^2*sinh(k3)*cosh(k3)+12*F4*F10*cosh(k3*an)*Nn^2*k3+12*F1*F10*sinh(k3*an)*Nn
^2*k3+...

3*k3^3*F1^2*Nn^2*an+3*k3^2*F4^2*Nn^2*sinh(k3*an)*cosh(k3*an)+3*k3^2*F1^2*Nn^2*sinh(k3*an)
*cosh(k3*an)+12*F4*Nn*k3*an*cosh(k3*an)+...
   12*F1*Nn*k3*an*sinh(k3*an))/Nn^2/k3;
B7 = -1/6*(-12*F2*F10*sinh(k3)*Nn^2*k3-6*theta*F12*Nn*k3-
6*k3^3*F1*F2*Nn^2+6*k3^3*F4*F5*Nn^2-12*F10*F11*Nn^2*k3-12*F5*Nn*k3*cosh(k3)+...
   12*theta*F3*Nn*cosh(k3)-12*theta*F6*F10*cosh(k3)*Nn^2*k3-
12*F4*theta*F12*cosh(k3)*Nn^2*k3+12*theta*F6*Nn*sinh(k3)-...
   12*theta*F3*F10*sinh(k3)*Nn^2*k3-12*F1*theta*F12*sinh(k3)*Nn^2*k3-
6*k3^2*F4*theta*F3*sinh(k3)^2*Nn^2-6*k3^2*theta*F6*F1*sinh(k3)^2*Nn^2-...
   6*k3^2*F1*F2*Nn^2*sinh(k3)*cosh(k3)-6*k3^2*F4*F5*Nn^2*sinh(k3)*cosh(k3)-
12*F2*Nn*k3*sinh(k3)+12*F10*theta*F12*an*Nn^2*k3-...
   6*k3^3*F4*theta*F6*Nn^2*an+6*F11*an^2*Nn*k3+12*F2*Nn*cosh(k3)-
6*k3^3*F1*theta*F3*Nn^2+6*k3^3*F4*theta*F6*Nn^2-6*F11*Nn*k3+...
   12*F5*Nn*sinh(k3)-12*F4*F11*cosh(k3)*Nn^2*k3-12*F10*theta*F12*Nn^2*k3-
6*k3^2*F4*theta*F6*Nn^2*sinh(k3)*cosh(k3)+...
   6*k3^3*F1*theta*F3*Nn^2*an+6*k3^2*F4*F5*Nn^2*sinh(k3*an)*cosh(k3*an)-
12*theta*F6*Nn*k3*cosh(k3)+6*k3^2*F1*F2*Nn^2*sinh(k3*an)*cosh(k3*an)-...
   12*theta*F6*Nn*sinh(k3*an)-12*F5*F10*cosh(k3)*Nn^2*k3+12*theta*F6*Nn*k3*an*cosh(k3*an)-
12*theta*F3*Nn*cosh(k3*an)+...

12*theta*F3*Nn*k3*an*sinh(k3*an)+12*theta*F3*F10*sinh(k3*an)*Nn^2*k3+12*F1*theta*F12*sinh(k3
*an)*Nn^2*k3+...

12*theta*F6*F10*cosh(k3*an)*Nn^2*k3+6*k3^2*theta*F6*F1*sinh(k3*an)^2*Nn^2+12*F4*theta*F12*
cosh(k3*an)*Nn^2*k3+...
   6*k3^2*F4*theta*F3*sinh(k3*an)^2*Nn^2-12*F5*Nn*sinh(k3*an)-12*F2*Nn*cosh(k3*an)-
12*theta*F3*Nn*k3*sinh(k3)-6*k3^2*F5*F1*sinh(k3)^2*Nn^2-...
   6*k3^2*F4*F2*sinh(k3)^2*Nn^2-
12*F1*F11*sinh(k3)*Nn^2*k3+6*k3^3*F1*F2*Nn^2*an+6*theta*F12*an^2*Nn*k3+12*F10*F11*an*N
n^2*k3-...

6*k3^2*F1*theta*F3*Nn^2*sinh(k3)*cosh(k3)+12*F4*F11*cosh(k3*an)*Nn^2*k3+6*k3^2*F5*F1*sinh(
k3*an)^2*Nn^2-6*k3^3*F4*F5*Nn^2*an+...

12*F2*F10*sinh(k3*an)*Nn^2*k3+12*F1*F11*sinh(k3*an)*Nn^2*k3+12*F5*F10*cosh(k3*an)*Nn^2*k
3+6*k3^2*F4*theta*F6*Nn^2*sinh(k3*an)*cosh(k3*an)+...

6*k3^2*F4*F2*sinh(k3*an)^2*Nn^2+12*F5*Nn*k3*an*cosh(k3*an)+12*F2*Nn*k3*an*sinh(k3*an)+6*
k3^2*F1*theta*F3*Nn^2*sinh(k3*an)*cosh(k3*an))/Nn^2/k3;
```

```
B8 = -1/6*(3*k3^3*F5^2*Nn^2-3*k3^3*F2^2*Nn^2-12*F2*theta*F12*sinh(k3)*Nn^2*k3-
6*F11^2*Nn^2*k3+3*k3^3*theta^2*F6^2*Nn^2-3*k3^3*theta^2*F3^2*Nn^2-...
   6*theta^2*F12^2*Nn^2*k3-12*F11*theta*F12*Nn^2*k3+6*k3^3*F5*theta*F6*Nn^2-
6*k3^3*F2*theta*F3*Nn^2-12*F5*F11*cosh(k3)*Nn^2*k3-...
   12*F5*theta*F12*cosh(k3)*Nn^2*k3-12*theta*F6*F11*cosh(k3)*Nn^2*k3-
12*theta^2*F6*F12*cosh(k3)*Nn^2*k3-12*F2*F11*sinh(k3)*Nn^2*k3-...
   12*theta^2*F3*F12*sinh(k3)*Nn^2*k3-6*k3^2*F5*F2*sinh(k3)^2*Nn^2-
12*theta*F3*F11*sinh(k3)*Nn^2*k3-6*k3^2*theta^2*F6*F3*sinh(k3)^2*Nn^2-...
   6*k3^2*F5*theta*F3*sinh(k3)^2*Nn^2-6*k3^2*theta*F6*F2*sinh(k3)^2*Nn^2-
3*k3^2*F5^2*Nn^2*sinh(k3)*cosh(k3)-3*k3^2*F2^2*Nn^2*sinh(k3)*cosh(k3)-...
   3*k3^2*theta^2*F6^2*Nn^2*sinh(k3)*cosh(k3)-3*k3^2*theta^2*F3^2*Nn^2*sinh(k3)*cosh(k3)-
6*k3^2*F5*theta*F6*Nn^2*sinh(k3)*cosh(k3)-...
   6*k3^2*F2*theta*F3*Nn^2*sinh(k3)*cosh(k3)+6*F11^2*an*Nn^2*k3+3*k3^3*F2^2*Nn^2*an-
3*k3^3*F5^2*Nn^2*an+6*theta^2*F12^2*an*Nn^2*k3+...
   12*F11*theta*F12*an*Nn^2*k3-6*k3^3*F5*theta*F6*Nn^2*an-
3*k3^3*theta^2*F6^2*Nn^2*an+3*k3^3*theta^2*F3^2*Nn^2*an+6*k3^3*F2*theta*F3*Nn^2*an+...

12*F5*F11*cosh(k3*an)*Nn^2*k3+6*k3^2*theta^2*F6*F3*sinh(k3*an)^2*Nn^2+12*F2*F11*sinh(k3*a
n)*Nn^2*k3+6*k3^2*theta*F6*F2*sinh(k3*an)^2*Nn^2+...

12*F5*theta*F12*cosh(k3*an)*Nn^2*k3+12*theta*F6*F11*cosh(k3*an)*Nn^2*k3+12*theta^2*F6*F12*
cosh(k3*an)*Nn^2*k3+...

12*F2*theta*F12*sinh(k3*an)*Nn^2*k3+12*theta*F3*F11*sinh(k3*an)*Nn^2*k3+12*theta^2*F3*F12*s
inh(k3*an)*Nn^2*k3+...

6*k3^2*F5*F2*sinh(k3*an)^2*Nn^2+6*k3^2*F5*theta*F3*sinh(k3*an)^2*Nn^2+3*k3^2*F2^2*Nn^2*si
nh(k3*an)*cosh(k3*an)+...

3*k3^2*F5^2*Nn^2*sinh(k3*an)*cosh(k3*an)+6*k3^2*F5*theta*F6*Nn^2*sinh(k3*an)*cosh(k3*an)+3*
k3^2*theta^2*F6^2*Nn^2*sinh(k3*an)*cosh(k3*an)+...

3*k3^2*theta^2*F3^2*Nn^2*sinh(k3*an)*cosh(k3*an)+6*k3^2*F2*theta*F3*Nn^2*sinh(k3*an)*cosh(k3
*an))/Nn^2/k3;

Ca = 1/C-an/C+an/Cstar;
rt = h/2 + rhostar;
%Now the Q's, such that the integrability condition is 0= Q1P^2+Q2P+Q3
Q1 = -1/2*B3-1/2*B6;
Q2 = -rt*B1-1/2*B4-1/2*B7;
Q3 = Nn*Ca+B0*theta-rt*B2-1/2*B5-1/2*B8;

%And the transversality equation is 0=Q4P^2+Q5P+Q6bar
Q4 =
1/2*D3*(1/(Nn^2)+2*k3^2*cosh(k3*an)*F4/Nn+2*k3^2*sinh(k3*an)*F1/Nn+2*k3^4*cosh(k3*an)*F4*si
nh(k3*an)*F1+k3^4*sinh(k3*an)^2*F1^2+...
   k3^4*cosh(k3*an)^2*F4^2)-
1/2*Ds*(k1^4*cosh(k1*an)^2*F7^2+2*k1^2*cosh(k1*an)*F7/Nn+1/(Nn^2));
Q5 =
1/2*D3*(2*k3^4*cosh(k3*an)*theta*F6*sinh(k3*an)*F1+2*k3^2*cosh(k3*an)*F5/Nn+2*k3^4*cosh(k3*
an)*F4*sinh(k3*an)*theta*F3+...

2*k3^4*cosh(k3*an)^2*F4*F5+2*k3^2*sinh(k3*an)*F2/Nn+2*k3^4*sinh(k3*an)^2*F1*F2+2*k3^4*cosh
(k3*an)*F4*sinh(k3*an)*F2+...
```

```
2*k3^4*cosh(k3*an)^2*F4*theta*F6+2*k3^4*cosh(k3*an)*F5*sinh(k3*an)*F1+2*k3^4*sinh(k3*an)^2*F
1*theta*F3+2*k3^2*cosh(k3*an)*theta*F6/Nn+...
   2*k3^2*sinh(k3*an)*theta*F3/Nn)-
1/2*Ds*(2*k1^4*cosh(k1*an)^2*F7*F8+2*k1^4*cosh(k1*an)^2*F7*theta*F9+2*k1^2*cosh(k1*an)*F8/
Nn+...
   2*k1^2*cosh(k1*an)*theta*F9/Nn);
Q6 =
1/2*D3*(k3^4*sinh(k3*an)^2*F2^2+k3^4*cosh(k3*an)^2*F5^2+2*k3^4*cosh(k3*an)*F5*sinh(k3*an)*F
2+2*k3^4*cosh(k3*an)^2*F5*theta*F6+...

2*k3^4*sinh(k3*an)^2*F2*theta*F3+2*k3^4*cosh(k3*an)*theta*F6*sinh(k3*an)*F2+2*k3^4*cosh(k3*a
n)*F5*sinh(k3*an)*theta*F3+...

k3^4*sinh(k3*an)^2*theta^2*F3^2+k3^4*cosh(k3*an)^2*theta^2*F6^2+2*k3^4*cosh(k3*an)*theta^2*F6
*sinh(k3*an)*F3)-1/2*Ds*(k1^4*cosh(k1*an)^2*F8^2+...
   2*k1^4*cosh(k1*an)^2*F8*theta*F9+k1^4*cosh(k1*an)^2*theta^2*F9^2)+1/2*Nn^2/Ce+Nn*(1-
alpha_1)*theta+1/2*etatilda*theta^2;
Q6bar = Q6-2*gamma;

%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6bar*Q1-Q3*Q4;
PBF = -Q8/Q7;
BF = Q4*(PBF^2)+Q5*PBF+Q6bar; %Sub solution into TE

%delta = a1+a4, loading param. is opposite of w1(0)
DelBF = -(PBF*F7+F8+theta*F9+PBF*F16+F17+theta*F18);
Kstiff = PBF/DelBF;



function[a_1,a_4,c_1,c_2,c_3,c_4,k1,k3] = BFtenthermCONST(ac,Nc,Pc,TTc)
% Uses the hand soltuion to find the const. of integ.
global an Ds D3 rhostar h C Cstar Ce gamma P mstar alpha_1 etatilda

k1 = sqrt(Nc/Ds);
k3 = sqrt(Nc/D3);
y = k1*ac;
x = k3*ac;
%Some general stuff to get started
GC1 = D3*(k3^2)*(k3*sinh(x)*cosh(y)-k1*cosh(x)*sinh(y));
GC2 = D3*(k3^2)*(k3*cosh(x)*cosh(y)-k1*sinh(x)*sinh(y));
C2den = GC2/GC1-sinh(k3)/cosh(k3);
A1den = Ds*(k1^3)*sinh(y);
%Now, the F's
% c2
F1 = (-Ds+D3)*(k1*sinh(y))/(Nc*GC1*C2den)+1/(Nc*(k3^2)*cosh(k3)*C2den);
F2 = ((k1*sinh(y))/(GC1*C2den))*(-h*Nc/2-rhostar*Nc);
F3 = ((k1*sinh(y))/(GC1*C2den))*mstar;
% c1
F4 = (k1*sinh(y)*(-Ds+D3))/(Nc*GC1)-F1*GC2/GC1;
F5 = (k1*sinh(y)*(-h*Nc/2-rhostar*Nc))/GC1-F2*GC2/GC1;
F6 = (k1*sinh(y)*mstar)/GC1-F3*GC2/GC1;
% a1
F7 = D3*(k3^3)*sinh(x)*F4/A1den+D3*(k3^3)*cosh(x)*F1/A1den;
```

```
F8 = D3*(k3^3)*sinh(x)*F5/A1den+D3*(k3^3)*cosh(x)*F2/A1den;
F9 = D3*(k3^3)*sinh(x)*F6/A1den+D3*(k3^3)*cosh(x)*F3/A1den;
% c3
F10 = F7*k1*sinh(y)-F4*k3*sinh(x)-F1*k3*cosh(x);
F11 = F8*k1*sinh(y)-F5*k3*sinh(x)-F2*k3*cosh(x);
F12 = F9*k1*sinh(y)-F6*k3*sinh(x)-F3*k3*cosh(x);
%c4
F13 = -1/(2*Nc)-F10-F1*sinh(k3)-F4*cosh(k3);
F14 = -F11-F2*sinh(k3)-F5*cosh(k3);
F15 = -F12-F3*sinh(k3)-F6*cosh(k3);
%a4
F16 = F4*cosh(x)+F1*sinh(x)+F10*ac+F13-F7*cosh(y);
F17 = F5*cosh(x)+F2*sinh(x)+F11*ac+F14-F8*cosh(y);
F18 = F6*cosh(x)+F3*sinh(x)+F12*ac+F15-F9*cosh(y);

a_1 = F7*Pc+F8+F9*TTc;
a_4 = F16*Pc+F17+F18*TTc;
c_1 = F4*Pc+F5+F6*TTc;
c_2 = F1*Pc+F2+F3*TTc;
c_3 = F10*Pc+F11+F12*TTc;
c_4 = F13*Pc+F14+F15*TTc;

% Separation of Patched Structure
% Patched Plate, thermal load, CLAMPED-fixed ends
% Non-Linear  TENSION!!!
%                                NO CONTACT ZONE!!!!!!
% This is the "P" version
% This program uses the functions:
clear
global h C D3 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
D3 = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rhostar Ds
% stiffnesses of composite structure
Astar = D3+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rhostar = Bstar/Cstar; % location of centroid of composite structure wrt ref. surface
Ds = Astar-rhostar*Bstar; %Dstar
global Cs Ce
% Dc = D+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar alpha_1 etatilda
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 2; %ratio alphaP/alpha (patch to baseplate)
nstar = C+alphaP*Cp;
mustar = -(h/2)*C+(hp/2)*Cp*alphaP;
mstar = mustar-rhostar*nstar;
```

```
alpha_1 = nstar/Cstar;
etatilda = C+(alphaP^2)*Cp-(alpha_1^2)*Cstar;

% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure

m = 1; %initialize index value for result array
n = 1;
%Set 2*gamma
global gamma
gamma = 0.1/2;

global theta
theta = -0.0045;

a = [0.69:0.001:0.70];
N = [50:.1:70];

for i = 1:length(a);
  global an;
   an = a(i);
   for j = 1:length(N);
      [Big(j,1),Big(j,2),Big(j,3)] = BigFuntenthermCFIX(N(j));
      %[value of big F, P ,Delta~w1(0)]
   end
   for k = 1:length(N)-1;
      if Big(k,1)*Big(k+1,1)<0;
         Nroot(k) = fzero(@BigFuntenthermCFIXroot,N(k));
         group(m,1) = a(i); %'a'
         group(m,2)= 1-a(i); %a*
         group(m,3) = Nroot(k); % The value of N that's a root
         [blah,group(m,4),group(m,5),group(m,6)] = BigFuntenthermCFIX(Nroot(k)); %[value of BF=0, P,
delta,K]
         %if group(m,4)>0; % & group(m,5)<1);
         Results(n,1)= 1 - a(i); %a*
         Results(n,2) = Nroot(k); %N
         Results(n,3) = group(m,4); %P
         Results(n,4) = group(m,5); %Delta
         Results(n,5) = group(m,6); %K
         n=n+1;
         %end
         m = m+1;
      else
      end
   end
end
q_end=n-1; %or n-1
for q = 1:q_end;
   Xend=1-Results(q,1);
x1 = [0:0.01:Xend];
x3 = [Xend:0.01:1];
[a1,a4,c1,c2,c3,c4,K1,K3] = BFtenthermCONST(Xend,Results(q,2),Results(q,3),theta);
```

```
        for j1 = 1:length(x1);
            w1star(q,j1) = a1*cosh(K1*x1(j1))+a4+(Results(q,3)*(x1(j1))^2)/(2*Results(q,2));
            %kappa1(q,j1) =
a1*K1^2*cosh(K1*x1(j1))+a2*K1^2*sinh(K1*x1(j1))+Results(q,3)/Results(q,2);
        end
        for j3 = 1:length(x3);
            w3(q,j3) =
c1*cosh(K3*x3(j3))+c2*sinh(K3*x3(j3))+c3*x3(j3)+c4+(Results(q,3)*(x3(j3))^2)/(2*Results(q,2));
            %kappa3(q,j3) =
c1*K3^2*cosh(K3*x3(j3))+c2*K3^2*sinh(K3*x3(j3))+Results(q,3)/Results(q,2);
        end
end

ii=3;
x1plot = [0:0.01:1-Results(ii,1)];
x3plot = [1-Results(ii,1):0.01:1];
figure
plot(x1plot,-w1star(ii,1:length(x1plot)),'b');
hold on
plot(x3plot,-w3(ii,1:length(x3plot)),'r');

an = 0.90;
for n = 1:length(N);
    [Big2(n,1),Big2(n,2),Big2(n,3)] = BigFuntenthermCFIX(N(n));
    %[value of big F, P ,Delta~w1(0),energy rel. rate]
end
figure
plot(N,Big2(:,1),'b')

q=1;
qq=1;
global an Pin
an=0.5;
Pin_test=[-1.4:0.1:0.2];
Nin=[-50:1:0];
for ib = 1:length(Pin_test);
   Pin = Pin_test(ib);
   for jb = 1:length(Nin);
      I_C(jb) = DEF_CFIX(Nin(jb));
   end
   for kb = 1:length(Nin)-1;
      if I_C(kb)*I_C(kb+1)<0;
         Nroot2(kb) = fzero(@DEF_CFIX,Nin(kb));
         group2(q,1)= Nroot2(kb); % N root value
         group2(q,2) = Pin_test(ib); %"P"
         [group2(q,3),group2(q,4)] = DEF_CFIX_A4(group2(q,1),group2(q,2)); %deflection, IC=0
         q = q+1;
      else
      end
   end
end

function[BF,PBF,DelBF, Kstiff] = BigFuntenthermCFIX(Nn)
   %Ds,D3,rhostar,h,C,Cstar,Ce,gamma,P,mstar,alpha_1,etatilda
%Using my hand solution, this function solves the non linear IC, and TC for P
```

```
%This is the "P" version
global an Ds D3 rhostar h C Cstar Ce gamma theta mstar alpha_1 etatilda

k1 = sqrt(Nn/Ds);
k3 = sqrt(Nn/D3);

%Now, the F's
% c2
F1 = -(k1^2*Ds^2*sinh(k3)*sinh(k1*an)-k3^2*sinh(k1*an)*D3^2*sinh(k3*an)-
k1*Ds*D3*k3^2*sinh(k3*an)*cosh(k1*an)+k1^2*Ds*D3*k3*sinh(k1*an)*cosh(k3*an)+sinh(k3*an)*k1^
2*Ds*sinh(k1*an)*D3+k3^2*sinh(k1*an)*D3*sinh(k3*an)*Ds-sinh(k3*an)*k1^2*Ds^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*D3)/k3^2/Nn/D3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
F2 = -
1/2*sinh(k1*an)*Nn*(2*k1^2*Ds*sinh(k3)*rhostar+k1^2*Ds*sinh(k3)*h+k3^2*D3*sinh(k3*an)*h+2*k3
^2*D3*sinh(k3*an)*rhostar-2*sinh(k3*an)*k1^2*Ds*rhostar-
sinh(k3*an)*k1^2*Ds*h)/k3^2/D3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
F3 = sinh(k1*an)*mstar*(k3^2*D3*sinh(k3*an)-
sinh(k3*an)*k1^2*Ds+k1^2*Ds*sinh(k3))/k3^2/D3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
% c1
F4 = -(-
k1^2*sinh(k1*an)*cosh(k3*an)*Ds^2+k1^2*Ds*D3*sinh(k1*an)*cosh(k3*an)+k1^2*sinh(k1*an)*sinh(k3
*an)*Ds*D3*k3-k1^2*sinh(k1*an)*D3*Ds*cosh(k3)+k1^2*sinh(k1*an)*Ds^2*cosh(k3)-
cosh(k1*an)*k3^2*Ds*D3*cosh(k3*an)*k1-
sinh(k1*an)*cosh(k3*an)*D3^2*k3^2+sinh(k1*an)*cosh(k3*an)*Ds*D3*k3^2)/k3^2/Nn/D3/(-
D3*k3^2*sinh(k1*an)+k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)-
k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)-
k1^2*Ds*cosh(k3)*sinh(k1*an)*cosh(k3*an)+k1^2*Ds*sinh(k1*an)+k3*k1*Ds*cosh(k3)*sinh(k3*an)*co
sh(k1*an));
F5 = -1/2*Nn*(k1^2*cosh(k3*an)*h*Ds+2*k1^2*cosh(k3*an)*rhostar*Ds-k1^2*h*Ds*cosh(k3)-
2*k1^2*rhostar*Ds*cosh(k3)-cosh(k3*an)*h*D3*k3^2-
2*cosh(k3*an)*rhostar*D3*k3^2)*sinh(k1*an)/k3^2/D3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
F6 = mstar*(k1^2*cosh(k3*an)*Ds-k1^2*Ds*cosh(k3)-
cosh(k3*an)*D3*k3^2)*sinh(k1*an)/k3^2/D3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
% a1
F7 = -(cosh(k3*an)*Ds*sinh(k3)-cosh(k3*an)*sinh(k3)*D3+D3*k3+sinh(k3*an)*D3*cosh(k3)-
sinh(k3*an)*Ds*cosh(k3))*k3/Nn/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-
k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an))/k1;
F8 = -1/2*Nn*(2*cosh(k3*an)*sinh(k3)*rhostar+cosh(k3*an)*sinh(k3)*h-sinh(k3*an)*h*cosh(k3)-
2*sinh(k3*an)*rhostar*cosh(k3))*k3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-
k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an))/k1;
```

```
F9 = mstar*(-sinh(k3*an)*cosh(k3)+sinh(k3)*cosh(k3*an))*k3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-
k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an))/k1;
% c3
F10 = -(-k1^2*cosh(k3*an)*sinh(k3)*D3*Ds+k1^2*cosh(k3*an)*sinh(k3)*Ds^2+k1^2*D3*k3*Ds-
k1^2*cosh(k3)*sinh(k3*an)*Ds^2+k1^2*cosh(k3)*sinh(k3*an)*Ds*D3-
cosh(k3*an)*sinh(k3)*Ds*D3*k3^2+cosh(k3*an)*sinh(k3)*D3^2*k3^2+cosh(k3)*k3^2*D3*sinh(k3*an)*
Ds-cosh(k3)*k3^2*D3^2*sinh(k3*an)-D3^2*k3^3)*sinh(k1*an)/k3/D3/Nn/(-
D3*k3^2*sinh(k1*an)+k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)-
k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)-
k1^2*Ds*cosh(k3)*sinh(k1*an)*cosh(k3*an)+k1^2*Ds*sinh(k1*an)+k3*k1*Ds*cosh(k3)*sinh(k3*an)*co
sh(k1*an));
F11 = -1/2*Nn*(k1^2*cosh(k3*an)*Ds*sinh(k3)*h+2*k1^2*cosh(k3*an)*Ds*sinh(k3)*rhostar-
2*k1^2*cosh(k3)*sinh(k3*an)*Ds*rhostar-k1^2*cosh(k3)*sinh(k3*an)*Ds*h-
2*cosh(k3*an)*sinh(k3)*rhostar*D3*k3^2-
cosh(k3*an)*sinh(k3)*h*D3*k3^2+cosh(k3)*k3^2*D3*sinh(k3*an)*h+2*cosh(k3)*k3^2*D3*sinh(k3*an)
*rhostar)*sinh(k1*an)/k3/D3/(-D3*k3^2*sinh(k1*an)+k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)-
k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)-
k1^2*Ds*cosh(k3)*sinh(k1*an)*cosh(k3*an)+k1^2*Ds*sinh(k1*an)+k3*k1*Ds*cosh(k3)*sinh(k3*an)*co
sh(k1*an));
F12 = mstar*(k1^2*cosh(k3*an)*Ds*sinh(k3)-k1^2*cosh(k3)*sinh(k3*an)*Ds-
cosh(k3*an)*sinh(k3)*D3*k3^2+cosh(k3)*k3^2*D3*sinh(k3*an))*sinh(k1*an)/k3/D3/(-
D3*k3^2*sinh(k1*an)+k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)-
k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)-
k1^2*Ds*cosh(k3)*sinh(k1*an)*cosh(k3*an)+k1^2*Ds*sinh(k1*an)+k3*k1*Ds*cosh(k3)*sinh(k3*an)*co
sh(k1*an));
%c4
F13 = -
1/2*(k1^2*sinh(k1*an)*D3*k3^2*Ds+2*k1^2*sinh(k1*an)*sinh(k3)*sinh(k3*an)*Ds^2+k1*cosh(k3*an)*
sinh(k3)*cosh(k1*an)*k3^3*Ds*D3-
2*k1*cosh(k3*an)*cosh(k3)*cosh(k1*an)*k3^2*Ds*D3+4*k1^2*sinh(k1*an)*cosh(k3)*sinh(k3*an)*Ds*
D3*k3-2*k1^2*sinh(k1*an)*k3*cosh(k3)*sinh(k3*an)*Ds^2-
k1^2*sinh(k1*an)*sinh(k3)*k3^2*sinh(k3*an)*Ds*D3+2*k1*k3^2*sinh(k3)*sinh(k3*an)*cosh(k1*an)*D
s*D3-
k1*cosh(k3)*cosh(k1*an)*k3^3*D3*sinh(k3*an)*Ds+2*sinh(k1*an)*cosh(k3*an)*k3^2*cosh(k3)*Ds*D3
-2*sinh(k1*an)*sinh(k3)*k3^2*sinh(k3*an)*Ds*D3-
2*sinh(k1*an)*cosh(k3*an)*cosh(k3)*D3^2*k3^2+2*sinh(k1*an)*cosh(k3*an)*k3^3*sinh(k3)*D3^2-
2*sinh(k1*an)*cosh(k3*an)*k3^3*sinh(k3)*Ds*D3+2*sinh(k1*an)*sinh(k3)*k3^2*D3^2*sinh(k3*an)+2*
sinh(k1*an)*k3^3*cosh(k3)*D3*sinh(k3*an)*Ds-2*sinh(k1*an)*k3^3*cosh(k3)*D3^2*sinh(k3*an)-
2*k1^2*sinh(k1*an)*cosh(k3*an)*cosh(k3)*Ds^2+2*k1^2*sinh(k1*an)*cosh(k3*an)*k3*sinh(k3)*Ds^2-
2*k1^2*sinh(k1*an)*sinh(k3)*sinh(k3*an)*Ds*D3+2*k1^2*sinh(k1*an)*cosh(k3*an)*cosh(k3)*D3*Ds+
k1^2*sinh(k1*an)*cosh(k3*an)*k3^2*cosh(k3)*Ds*D3-
4*k1^2*sinh(k1*an)*cosh(k3*an)*sinh(k3)*Ds*D3*k3+2*k1^2*sinh(k1*an)*Ds^2-
sinh(k1*an)*D3^2*k3^4-2*k1^2*sinh(k1*an)*Ds*D3)/k3^2/D3/Nn/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
F14 = -1/2*Nn*(2*k1^2*cosh(k3*an)*k3*sinh(k3)*rhostar*Ds+k1^2*cosh(k3*an)*k3*sinh(k3)*h*Ds-
2*k1^2*cosh(k3*an)*cosh(k3)*rhostar*Ds-
2*k1^2*k3*cosh(k3)*sinh(k3*an)*Ds*rhostar+2*k1^2*Ds*rhostar-
k1^2*k3*cosh(k3)*sinh(k3*an)*Ds*h+k1^2*sinh(k3)*sinh(k3*an)*Ds*h+2*k1^2*sinh(k3)*sinh(k3*an)*
Ds*rhostar+k3^3*cosh(k3)*D3*sinh(k3*an)*h+2*k3^3*cosh(k3)*D3*sinh(k3*an)*rhostar-
sinh(k3)*k3^2*D3*sinh(k3*an)*h-2*sinh(k3)*k3^2*D3*sinh(k3*an)*rhostar-
cosh(k3*an)*k3^3*sinh(k3)*h*D3+2*cosh(k3*an)*cosh(k3)*rhostar*D3*k3^2+cosh(k3*an)*cosh(k3)*h*
D3*k3^2-2*cosh(k3*an)*k3^3*sinh(k3)*rhostar*D3+k1^2*Ds*h-
k1^2*cosh(k3*an)*cosh(k3)*h*Ds)*sinh(k1*an)/k3^2/D3/(D3*k3^2*sinh(k1*an)-
```

```
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
F15 = mstar*(k1^2*cosh(k3*an)*k3*sinh(k3)*Ds-
k1^2*k3*cosh(k3)*sinh(k3*an)*Ds+k1^2*Ds*sinh(k3)*sinh(k3*an)+k3^3*cosh(k3)*D3*sinh(k3*an)-
sinh(k3)*k3^2*D3*sinh(k3*an)-
cosh(k3*an)*k3^3*sinh(k3)*D3+cosh(k3*an)*cosh(k3)*D3*k3^2+k1^2*Ds-
k1^2*Ds*cosh(k3)*cosh(k3*an))*sinh(k1*an)/k3^2/D3/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an));
%a4
F16 = 1/2*(-4*k1^3*sinh(k1*an)*Ds^2+2*k1^3*sinh(k1*an)*k3*cosh(k3)*sinh(k3*an)*Ds^2-
2*k3^3*cosh(k3)*sinh(k3*an)*cosh(k1*an)*Ds*D3+2*cosh(k3*an)*k3^3*cosh(k1*an)*D3*Ds*sinh(k3)+
2*k1*sinh(k1*an)*an*k3^3*cosh(k3)*D3*sinh(k3*an)*Ds-
2*k1*sinh(k1*an)*an*k3^3*cosh(k3)*D3^2*sinh(k3*an)+2*k1*sinh(k1*an)*sinh(k3)*k3^2*sinh(k3*an)*
Ds*D3-
2*k1*cosh(k3*an)*sinh(k1*an)*k3^3*sinh(k3)*D3^2+2*k1*cosh(k3*an)*sinh(k1*an)*cosh(k3)*D3^2*k3
^2-
4*k1^3*sinh(k1*an)*cosh(k3)*sinh(k3*an)*Ds*D3*k3+k1^3*sinh(k1*an)*sinh(k3)*k3^2*sinh(k3*an)*D
s*D3+2*k1*sinh(k1*an)*k3^3*cosh(k3)*D3^2*sinh(k3*an)-
2*k1*cosh(k3*an)*sinh(k1*an)*k3^2*cosh(k3)*Ds*D3+2*k3^3*sinh(k3*an)*cosh(k1*an)*cosh(k3)*D3^
2-4*k1^3*sinh(k1*an)*sinh(k3)*sinh(k3*an)*Ds^2+4*k1^3*sinh(k1*an)*D3*Ds-
2*k1^2*Ds*D3*k3^2*cosh(k1*an)-2*k1*sinh(k1*an)*k3^2*D3^2+k1*sinh(k1*an)*D3^2*k3^4-
k1^3*sinh(k1*an)*D3*k3^2*Ds-
2*k1*sinh(k1*an)*an*k3^4*D3^2+2*k3^4*cosh(k1*an)*D3^2+2*k1*sinh(k1*an)*D3*k3^2*Ds+2*k1^3
*sinh(k1*an)*an*k3^2*D3*Ds+2*k1^3*sinh(k1*an)*an*k3*cosh(k3)*sinh(k3*an)*Ds*D3-
2*k1^3*sinh(k1*an)*an*k3*cosh(k3)*sinh(k3*an)*Ds^2+2*k1*cosh(k3*an)*sinh(k1*an)*k3^3*sinh(k3)*
Ds*D3+2*k1*cosh(k3*an)*sinh(k1*an)*an*k3^3*sinh(k3)*D3^2+4*k1^3*cosh(k3*an)*sinh(k1*an)*sinh
(k3)*Ds*D3*k3-
4*k1^3*cosh(k3*an)*sinh(k1*an)*cosh(k3)*D3*Ds+4*k1^3*sinh(k1*an)*sinh(k3)*sinh(k3*an)*Ds*D3-
2*k1*sinh(k1*an)*k3^3*cosh(k3)*D3*sinh(k3*an)*Ds-
2*k1*cosh(k3*an)*sinh(k1*an)*an*k3^3*sinh(k3)*Ds*D3-
2*k1*sinh(k1*an)*sinh(k3)*k3^2*D3^2*sinh(k3*an)-
k1^2*cosh(k3*an)*k3^3*cosh(k1*an)*D3*Ds*sinh(k3)+2*k1^2*cosh(k3*an)*cosh(k3)*cosh(k1*an)*k3^
2*Ds*D3-
2*k1^2*k3^2*sinh(k3)*sinh(k3*an)*cosh(k1*an)*Ds*D3+k1^2*k3^3*cosh(k3)*sinh(k3*an)*cosh(k1*an)
*Ds*D3-
2*k1^3*cosh(k3*an)*sinh(k1*an)*an*k3*sinh(k3)*D3*Ds+4*k1^3*cosh(k3*an)*sinh(k1*an)*cosh(k3)*
Ds^2-2*cosh(k3*an)*k3^3*cosh(k1*an)*D3^2*sinh(k3)-
k1^3*cosh(k3*an)*sinh(k1*an)*k3^2*cosh(k3)*Ds*D3+2*k1^3*cosh(k3*an)*sinh(k1*an)*an*k3*sinh(k3
)*Ds^2-2*k1^3*cosh(k3*an)*sinh(k1*an)*k3*sinh(k3)*Ds^2)/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-
k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an))/k1/Nn/D3/k3^2;
F17 = 1/2*Nn*(-
4*k1^3*sinh(k1*an)*sinh(k3)*sinh(k3*an)*Ds*rhostar+2*k1^3*cosh(k3*an)*sinh(k1*an)*cosh(k3)*h*Ds
+2*k1^3*sinh(k1*an)*k3*cosh(k3)*sinh(k3*an)*Ds*rhostar+k1^3*cosh(k3*an)*sinh(k1*an)*an*k3*sinh(
k3)*h*Ds+2*k1*cosh(k3*an)*sinh(k1*an)*k3^3*sinh(k3)*rhostar*D3+k1*sinh(k1*an)*sinh(k3)*k3^2*D
3*sinh(k3*an)*h+2*cosh(k3*an)*k3^3*cosh(k1*an)*D3*sinh(k3)*rhostar-
k1^3*sinh(k1*an)*an*k3*cosh(k3)*sinh(k3*an)*Ds*h-k3^3*sinh(k3*an)*cosh(k1*an)*cosh(k3)*h*D3-
2*k1^3*sinh(k1*an)*sinh(k3)*sinh(k3*an)*Ds*h+k1*cosh(k3*an)*sinh(k1*an)*k3^3*sinh(k3)*h*D3-
2*k1^3*sinh(k1*an)*h*Ds-
2*k1*cosh(k3*an)*sinh(k1*an)*an*k3^3*sinh(k3)*rhostar*D3+2*k1*sinh(k1*an)*k3^2*D3*rhostar-
k1^3*cosh(k3*an)*sinh(k1*an)*k3*sinh(k3)*h*Ds-k1*cosh(k3*an)*sinh(k1*an)*cosh(k3)*h*D3*k3^2-
k1*sinh(k1*an)*k3^3*cosh(k3)*D3*sinh(k3*an)*h+k1*sinh(k1*an)*k3^2*D3*h-
2*k1*sinh(k1*an)*k3^3*cosh(k3)*D3*sinh(k3*an)*rhostar-
2*k3^3*sinh(k3*an)*cosh(k1*an)*cosh(k3)*rhostar*D3-
```

```
2*k1^3*sinh(k1*an)*an*k3*cosh(k3)*sinh(k3*an)*Ds*rhostar+cosh(k3*an)*k3^3*cosh(k1*an)*D3*sinh(
k3)*h+4*k1^3*cosh(k3*an)*sinh(k1*an)*cosh(k3)*rhostar*Ds-
k1*cosh(k3*an)*sinh(k1*an)*an*k3^3*sinh(k3)*h*D3+2*k1^3*cosh(k3*an)*sinh(k1*an)*an*k3*sinh(k3
)*rhostar*Ds-2*k1^3*cosh(k3*an)*sinh(k1*an)*k3*sinh(k3)*rhostar*Ds-
2*k1*cosh(k3*an)*sinh(k1*an)*cosh(k3)*rhostar*D3*k3^2-
4*k1^3*sinh(k1*an)*rhostar*Ds+k1^3*sinh(k1*an)*k3*cosh(k3)*sinh(k3*an)*Ds*h+2*k1*sinh(k1*an)*s
inh(k3)*k3^2*D3*sinh(k3*an)*rhostar+k1*sinh(k1*an)*an*k3^3*cosh(k3)*D3*sinh(k3*an)*h+2*k1*sinh
(k1*an)*an*k3^3*cosh(k3)*D3*sinh(k3*an)*rhostar)/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-
k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an))/k1/D3/k3^2;
F18 = -mstar*(-k1^3*sinh(k1*an)*an*k3*cosh(k3)*sinh(k3*an)*Ds-
k3^3*sinh(k3*an)*cosh(k1*an)*cosh(k3)*D3+k1*sinh(k1*an)*k3^2*D3-
2*k1^3*sinh(k1*an)*sinh(k3)*sinh(k3*an)*Ds+k1*cosh(k3*an)*sinh(k1*an)*k3^3*sinh(k3)*D3-
2*k1^3*sinh(k1*an)*Ds-k1^3*cosh(k3*an)*sinh(k1*an)*k3*sinh(k3)*Ds-
k1*cosh(k3*an)*sinh(k1*an)*cosh(k3)*D3*k3^2-k1*sinh(k1*an)*k3^3*cosh(k3)*D3*sinh(k3*an)-
k1*cosh(k3*an)*sinh(k1*an)*an*k3^3*sinh(k3)*D3+cosh(k3*an)*k3^3*cosh(k1*an)*D3*sinh(k3)+k1^3*
sinh(k1*an)*k3*cosh(k3)*sinh(k3*an)*Ds+k1*sinh(k1*an)*an*k3^3*cosh(k3)*D3*sinh(k3*an)+2*k1^3*
cosh(k3*an)*sinh(k1*an)*cosh(k3)*Ds+k1^3*cosh(k3*an)*sinh(k1*an)*an*k3*sinh(k3)*Ds+k1*sinh(k1*
an)*sinh(k3)*k3^2*D3*sinh(k3*an))/(D3*k3^2*sinh(k1*an)-
k1^2*Ds*sinh(k3)*sinh(k1*an)*sinh(k3*an)+k3*k1*Ds*sinh(k3)*cosh(k1*an)*cosh(k3*an)+k1^2*Ds*co
sh(k3)*sinh(k1*an)*cosh(k3*an)-k1^2*Ds*sinh(k1*an)-
k3*k1*Ds*cosh(k3)*sinh(k3*an)*cosh(k1*an))/k1/D3/k3^2;

%Now the B's
B0 = 1-an+an*alpha_1;
B1 = F7*k1*sinh(k1*an)+an/Nn;
B2 = (F8+theta*F9)*k1*sinh(k1*an);

B3 = -1/6*(-3*k1^2*F7^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*F7^2*Nn^2*an-
12*F7*Nn*k1*an*cosh(k1*an)+12*F7*Nn*sinh(k1*an)-2*an^3*k1)/k1/Nn^2;
B4 = -1/6*(-6*k1^2*F7*F8*Nn^2*cosh(k1*an)*sinh(k1*an)+6*k1^3*F7*F8*Nn^2*an-
6*k1^2*F7*theta*F9*Nn^2*cosh(k1*an)*sinh(k1*an)+...
   6*k1^3*F7*theta*F9*Nn^2*an-12*F8*Nn*k1*an*cosh(k1*an)+12*F8*Nn*sinh(k1*an)-
12*theta*F9*Nn*k1*an*cosh(k1*an)+12*theta*F9*Nn*sinh(k1*an))/k1/Nn^2;
B5 = -1/6*(-3*k1^2*F8^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*F8^2*Nn^2*an-
6*k1^2*F8*theta*F9*Nn^2*cosh(k1*an)*sinh(k1*an)+...
   6*k1^3*F8*theta*F9*Nn^2*an-
3*k1^2*theta^2*F9^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*theta^2*F9^2*Nn^2*an)/k1/Nn^2;
B6 = -1/6*(-2*k3+2*an^3*k3+6*k3^2*F4*F1*sinh(k3*an)^2*Nn^2-
12*F1*Nn*k3*sinh(k3)+3*k3^3*F4^2*Nn^2-3*k3^3*F1^2*Nn^2+12*F4*Nn*sinh(k3)+...
   12*F1*Nn*cosh(k3)-6*F10^2*Nn^2*k3-6*F10*Nn*k3-12*F4*Nn*sinh(k3*an)-
12*F1*Nn*cosh(k3*an)+6*F10*an^2*Nn*k3-12*F4*F10*cosh(k3)*Nn^2*k3-...
   12*F1*F10*sinh(k3)*Nn^2*k3-12*F4*Nn*k3*cosh(k3)+6*F10^2*an*Nn^2*k3-
3*k3^3*F4^2*Nn^2*an-6*k3^2*F4*F1*sinh(k3)^2*Nn^2-...
   3*k3^2*F4^2*Nn^2*sinh(k3)*cosh(k3)-
3*k3^2*F1^2*Nn^2*sinh(k3)*cosh(k3)+12*F4*F10*cosh(k3*an)*Nn^2*k3+12*F1*F10*sinh(k3*an)*Nn
^2*k3+...

3*k3^3*F1^2*Nn^2*an+3*k3^2*F4^2*Nn^2*sinh(k3*an)*cosh(k3*an)+3*k3^2*F1^2*Nn^2*sinh(k3*an)
*cosh(k3*an)+12*F4*Nn*k3*an*cosh(k3*an)+...
   12*F1*Nn*k3*an*sinh(k3*an))/Nn^2/k3;
B7 = -1/6*(-12*F2*F10*sinh(k3)*Nn^2*k3-6*theta*F12*Nn*k3-
6*k3^3*F1*F2*Nn^2+6*k3^3*F4*F5*Nn^2-12*F10*F11*Nn^2*k3-12*F5*Nn*k3*cosh(k3)+...
```

```
   12*theta*F3*Nn*cosh(k3)-12*theta*F6*F10*cosh(k3)*Nn^2*k3-
12*F4*theta*F12*cosh(k3)*Nn^2*k3+12*theta*F6*Nn*sinh(k3)-...
   12*theta*F3*F10*sinh(k3)*Nn^2*k3-12*F1*theta*F12*sinh(k3)*Nn^2*k3-
6*k3^2*F4*theta*F3*sinh(k3)^2*Nn^2-6*k3^2*theta*F6*F1*sinh(k3)^2*Nn^2-...
   6*k3^2*F1*F2*Nn^2*sinh(k3)*cosh(k3)-6*k3^2*F4*F5*Nn^2*sinh(k3)*cosh(k3)-
12*F2*Nn*k3*sinh(k3)+12*F10*theta*F12*an*Nn^2*k3-...
   6*k3^3*F4*theta*F6*Nn^2*an+6*F11*an^2*Nn*k3+12*F2*Nn*cosh(k3)-
6*k3^3*F1*theta*F3*Nn^2+6*k3^3*F4*theta*F6*Nn^2-6*F11*Nn*k3+...
   12*F5*Nn*sinh(k3)-12*F4*F11*cosh(k3)*Nn^2*k3-12*F10*theta*F12*Nn^2*k3-
6*k3^2*F4*theta*F6*Nn^2*sinh(k3)*cosh(k3)+...
   6*k3^3*F1*theta*F3*Nn^2*an+6*k3^2*F4*F5*Nn^2*sinh(k3*an)*cosh(k3*an)-
12*theta*F6*Nn*k3*cosh(k3)+6*k3^2*F1*F2*Nn^2*sinh(k3*an)*cosh(k3*an)-...
   12*theta*F6*Nn*sinh(k3*an)-12*F5*F10*cosh(k3)*Nn^2*k3+12*theta*F6*Nn*k3*an*cosh(k3*an)-
12*theta*F3*Nn*cosh(k3*an)+...

12*theta*F3*Nn*k3*an*sinh(k3*an)+12*theta*F3*F10*sinh(k3*an)*Nn^2*k3+12*F1*theta*F12*sinh(k3
*an)*Nn^2*k3+...

12*theta*F6*F10*cosh(k3*an)*Nn^2*k3+6*k3^2*theta*F6*F1*sinh(k3*an)^2*Nn^2+12*F4*theta*F12*
cosh(k3*an)*Nn^2*k3+...
   6*k3^2*F4*theta*F3*sinh(k3*an)^2*Nn^2-12*F5*Nn*sinh(k3*an)-12*F2*Nn*cosh(k3*an)-
12*theta*F3*Nn*k3*sinh(k3)-6*k3^2*F5*F1*sinh(k3)^2*Nn^2-...
   6*k3^2*F4*F2*sinh(k3)^2*Nn^2-
12*F1*F11*sinh(k3)*Nn^2*k3+6*k3^3*F1*F2*Nn^2*an+6*theta*F12*an^2*Nn*k3+12*F10*F11*an*N
n^2*k3-...

6*k3^2*F1*theta*F3*Nn^2*sinh(k3)*cosh(k3)+12*F4*F11*cosh(k3*an)*Nn^2*k3+6*k3^2*F5*F1*sinh(
k3*an)^2*Nn^2-6*k3^3*F4*F5*Nn^2*an+...

12*F2*F10*sinh(k3*an)*Nn^2*k3+12*F1*F11*sinh(k3*an)*Nn^2*k3+12*F5*F10*cosh(k3*an)*Nn^2*k
3+6*k3^2*F4*theta*F6*Nn^2*sinh(k3*an)*cosh(k3*an)+...

6*k3^2*F4*F2*sinh(k3*an)^2*Nn^2+12*F5*Nn*k3*an*cosh(k3*an)+12*F2*Nn*k3*an*sinh(k3*an)+6*
k3^2*F1*theta*F3*Nn^2*sinh(k3*an)*cosh(k3*an))/Nn^2/k3;
B8 = -1/6*(3*k3^3*F5^2*Nn^2-3*k3^3*F2^2*Nn^2-12*F2*theta*F12*sinh(k3)*Nn^2*k3-
6*F11^2*Nn^2*k3+3*k3^3*theta^2*F6^2*Nn^2-3*k3^3*theta^2*F3^2*Nn^2-...
   6*theta^2*F12^2*Nn^2*k3-12*F11*theta*F12*Nn^2*k3+6*k3^3*F5*theta*F6*Nn^2-
6*k3^3*F2*theta*F3*Nn^2-12*F5*F11*cosh(k3)*Nn^2*k3-...
   12*F5*theta*F12*cosh(k3)*Nn^2*k3-12*theta*F6*F11*cosh(k3)*Nn^2*k3-
12*theta^2*F6*F12*cosh(k3)*Nn^2*k3-12*F2*F11*sinh(k3)*Nn^2*k3-...
   12*theta^2*F3*F12*sinh(k3)*Nn^2*k3-6*k3^2*F5*F2*sinh(k3)^2*Nn^2-
12*theta*F3*F11*sinh(k3)*Nn^2*k3-6*k3^2*theta^2*F6*F3*sinh(k3)^2*Nn^2-...
   6*k3^2*F5*theta*F3*sinh(k3)^2*Nn^2-6*k3^2*theta*F6*F2*sinh(k3)^2*Nn^2-
3*k3^2*F5^2*Nn^2*sinh(k3)*cosh(k3)-3*k3^2*F2^2*Nn^2*sinh(k3)*cosh(k3)-...
   3*k3^2*theta^2*F6^2*Nn^2*sinh(k3)*cosh(k3)-3*k3^2*theta^2*F3^2*Nn^2*sinh(k3)*cosh(k3)-
6*k3^2*F5*theta*F6*Nn^2*sinh(k3)*cosh(k3)-...
   6*k3^2*F2*theta*F3*Nn^2*sinh(k3)*cosh(k3)+6*F11^2*an*Nn^2*k3+3*k3^3*F2^2*Nn^2*an-
3*k3^3*F5^2*Nn^2*an+6*theta^2*F12^2*an*Nn^2*k3+...
   12*F11*theta*F12*an*Nn^2*k3-6*k3^3*F5*theta*F6*Nn^2*an-
3*k3^3*theta^2*F6^2*Nn^2*an+3*k3^3*theta^2*F3^2*Nn^2*an+6*k3^3*F2*theta*F3*Nn^2*an+...

12*F5*F11*cosh(k3*an)*Nn^2*k3+6*k3^2*theta^2*F6*F3*sinh(k3*an)^2*Nn^2+12*F2*F11*sinh(k3*a
n)*Nn^2*k3+6*k3^2*theta*F6*F2*sinh(k3*an)^2*Nn^2+...

12*F5*theta*F12*cosh(k3*an)*Nn^2*k3+12*theta*F6*F11*cosh(k3*an)*Nn^2*k3+12*theta^2*F6*F12*
cosh(k3*an)*Nn^2*k3+...
```

```
12*F2*theta*F12*sinh(k3*an)*Nn^2*k3+12*theta*F3*F11*sinh(k3*an)*Nn^2*k3+12*theta^2*F3*F12*s
inh(k3*an)*Nn^2*k3+...

6*k3^2*F5*F2*sinh(k3*an)^2*Nn^2+6*k3^2*F5*theta*F3*sinh(k3*an)^2*Nn^2+3*k3^2*F2^2*Nn^2*si
nh(k3*an)*cosh(k3*an)+...

3*k3^2*F5^2*Nn^2*sinh(k3*an)*cosh(k3*an)+6*k3^2*F5*theta*F6*Nn^2*sinh(k3*an)*cosh(k3*an)+3*
k3^2*theta^2*F6^2*Nn^2*sinh(k3*an)*cosh(k3*an)+...

3*k3^2*theta^2*F3^2*Nn^2*sinh(k3*an)*cosh(k3*an)+6*k3^2*F2*theta*F3*Nn^2*sinh(k3*an)*cosh(k3
*an))/Nn^2/k3;

Ca = 1/C-an/C+an/Cstar;
rt = h/2 + rhostar;
%Now the Q's, such that the integrability condition is 0= Q1P^2+Q2P+Q3
Q1 = -1/2*B3-1/2*B6;
Q2 = -rt*B1-1/2*B4-1/2*B7;
Q3 = Nn*Ca+B0*theta-rt*B2-1/2*B5-1/2*B8;

%And the transversality equation is 0=Q4P^2+Q5P+Q6bar
Q4 =
1/2*D3*(1/(Nn^2)+2*k3^2*cosh(k3*an)*F4/Nn+2*k3^2*sinh(k3*an)*F1/Nn+2*k3^4*cosh(k3*an)*F4*si
nh(k3*an)*F1+k3^4*sinh(k3*an)^2*F1^2+...
   k3^4*cosh(k3*an)^2*F4^2)-
1/2*Ds*(k1^4*cosh(k1*an)^2*F7^2+2*k1^2*cosh(k1*an)*F7/Nn+1/(Nn^2));
Q5 =
1/2*D3*(2*k3^4*cosh(k3*an)*theta*F6*sinh(k3*an)*F1+2*k3^2*cosh(k3*an)*F5/Nn+2*k3^4*cosh(k3*
an)*F4*sinh(k3*an)*theta*F3+...

2*k3^4*cosh(k3*an)^2*F4*F5+2*k3^2*sinh(k3*an)*F2/Nn+2*k3^4*sinh(k3*an)^2*F1*F2+2*k3^4*cosh
(k3*an)*F4*sinh(k3*an)*F2+...

2*k3^4*cosh(k3*an)^2*F4*theta*F6+2*k3^4*cosh(k3*an)*F5*sinh(k3*an)*F1+2*k3^4*sinh(k3*an)^2*F
1*theta*F3+2*k3^2*cosh(k3*an)*theta*F6/Nn+...
   2*k3^2*sinh(k3*an)*theta*F3/Nn)-
1/2*Ds*(2*k1^4*cosh(k1*an)^2*F7*F8+2*k1^4*cosh(k1*an)^2*F7*theta*F9+2*k1^2*cosh(k1*an)*F8/
Nn+...
   2*k1^2*cosh(k1*an)*theta*F9/Nn);
Q6 =
1/2*D3*(k3^4*sinh(k3*an)^2*F2^2+k3^4*cosh(k3*an)^2*F5^2+2*k3^4*cosh(k3*an)*F5*sinh(k3*an)*F
2+2*k3^4*cosh(k3*an)^2*F5*theta*F6+...

2*k3^4*sinh(k3*an)^2*F2*theta*F3+2*k3^4*cosh(k3*an)*theta*F6*sinh(k3*an)*F2+2*k3^4*cosh(k3*a
n)*F5*sinh(k3*an)*theta*F3+...

k3^4*sinh(k3*an)^2*theta^2*F3^2+k3^4*cosh(k3*an)^2*theta^2*F6^2+2*k3^4*cosh(k3*an)*theta^2*F6
*sinh(k3*an)*F3)-1/2*Ds*(k1^4*cosh(k1*an)^2*F8^2+...
   2*k1^4*cosh(k1*an)^2*F8*theta*F9+k1^4*cosh(k1*an)^2*theta^2*F9^2)+1/2*Nn^2/Ce+Nn*(1-
alpha_1)*theta+1/2*etatilda*theta^2;
Q6bar = Q6-2*gamma;

%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6bar*Q1-Q3*Q4;
```

```
PBF = -Q8/Q7;
BF = Q4*(PBF^2)+Q5*PBF+Q6bar; %Sub solution into TE
%delta = -(a1+a4) opposite of deflection
DelBF = -(PBF*F7+F8+theta*F9+PBF*F16+F17+theta*F18);
Kstiff = PBF/DelBF;
%Energy Release Rate...
err = Q4+Q5+Q6;

% Separation of Patched Structure
% Patched Plate, thermal load, CLAMPED-fixed ends
% Non-Linear  TENSION!!!

%                          CONTACT ZONE!!!!!!

% This is the "P" version
% This program uses the functions:
clear
global h C D3 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
D3 = 1; % bending stiffness of baseplate
Dp = E0*(h0^3); %bending stiffness of patch
global Cstar rhostar Ds rho
% stiffnesses of composite structure
Astar = D3+Dp+((h/2)^2)*C+((hp/2)^2)*Cp;
Bstar = -(h/2)*C+(hp/2)*Cp;
Cstar = C+Cp;
rho = Bstar/Cstar; % rhostar, location of centroid of composite structure wrt ref. surface
rhostar = rho;
Ds = Astar-rho*Bstar; %Dstar
global Cs Ce Dc
Dc = D3+Dp; % bending stiffness of debonded segment (NA)
Cs = (C*Cp)/Cstar;
Ce = Cstar/(Cp/C);
global mstar alpha_1 etatilda
%Using the Normalization ThetaTilda = alpha*Theta
alpha = 1; %ratio alpha/alpha (baseplate to baseplate)
alphaP = 2; %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;

% a is the length of the bonded segment
% b is the length of the bonded segment plus contact zone (NA)
% Lp is the length of the patch
% L is the length of the baseplate (normalized to 1)
% p is the applied pressure

m = 1; %initialize index value for result array
```

```
n=1;
%Set 2*gamma
global gamma
gamma =0.1/2;

global theta b
theta = -0.005;
b=0.9;
a = [0.63:0.01:b];
N = [115:.1:150];

for i = 1:length(a);
  global an;
   an = a(i);
   for j = 1:length(N);
      N1=N(j);
      [Big(j,1),Big(j,2),Big(j,3)] = BigFuntenthermCFIXCZ_2(N1);
      %[value of big F, P ,Delta~w1(0)]
   end
   for k = 1:length(N)-1;
      if Big(k,1)*Big(k+1,1)<0;
         Nroot(k) = fzero(@BigFuntenthermCFIXCZroot_2,N(k));
         group(m,1) = a(i); %'a'
         group(m,2)= 1-a(i); %a*
         group(m,3) = Nroot(k); % The value of N that's a root
         [blah,group(m,4),group(m,5),group(m,6)] = BigFuntenthermCFIXCZ_2(Nroot(k)); %[value of
BF=0, P, delta,K]
         %if group(m,4)>0; % & group(m,5)>0);
         Results(n,1)= 1 - a(i); %a*
         Results(n,2) = Nroot(k); %N
         Results(n,3) = group(m,4); %P
         Results(n,4) = group(m,5); %Delta
         Results(n,5) = group(m,6); %K
         n=n+1;
         %end
         m = m+1;
      else
      end
   end
end
% Now to check the validity...
for q = 1:n-1;
   a_temp=1-Results(q,1);
x1 = [0:0.01:a_temp];
x2 = [a_temp:0.01:b];
x3 = [b:0.01:1];
[a1,a2,a3,a4,b1,b2,b3,b4,c1,c2,c3,c4,K1,K2,K3]=
BFtenthermCONST_CFIXCZ_2(Results(q,2),Results(q,3),a_temp);
         for j1 = 1:length(x1);
            w1star(q,j1) =
a1*cosh(K1*x1(j1))+a2*sinh(K1*x1(j1))+a3*x1(j1)+a4+(Results(q,3)*(x1(j1))^2)/(2*Results(q,2));
            kappa1(q,j1) = a1*K1^2*cosh(K1*x1(j1))+a2*K1^2*sinh(K1*x1(j1))+Results(q,3)/Results(q,2);
         end
         for j2 = 1:length(x2);
```

```
            w2star(q,j2) =
b1*cosh(K2*x2(j2))+b2*sinh(K2*x2(j2))+b3*x2(j2)+b4+(Results(q,3)*(x2(j2))^2)/(2*Results(q,2));
            kappa2(q,j2) = b1*K2^2*cosh(K2*x2(j2))+b2*K2^2*sinh(K2*x2(j2))+Results(q,3)/Results(q,2);
         end
         for j3 = 1:length(x3);
            w3(q,j3) =
c1*cosh(K3*x3(j3))+c2*sinh(K3*x3(j3))+c3*x3(j3)+c4+(Results(q,3)*(x3(j3))^2)/(2*Results(q,2));
            kappa3(q,j3) = c1*K3^2*cosh(K3*x3(j3))+c2*K3^2*sinh(K3*x3(j3))+Results(q,3)/Results(q,2);
         end
         for ja=1:length(x2);
            if  kappa2(q,ja)<=0;
            ValidCZ(q)=Results(q,1);
            else
            ValidCZ(q)=0;
            break
            end
         end
end
New_ValidCZ=ValidCZ';

ii=5;
x1plot = [0:0.01:1-Results(ii,1)];
x2plot = [1-Results(ii,1):0.01:b];
x3plot = [b:0.01:1];
figure
plot(x1plot,-w1star(ii,1:length(x1plot)),'b');
hold on
plot(x2plot,-w2star(ii,1:length(x2plot)),'g');
hold on
plot(x3plot,-w3(ii,1:length(x3plot)),'r');

function[BF,PBF,DelBF,stiff] = BigFuntenthermCFIXCZ_2(Nn)
   %Ds,D3,rhostar,h,C,Cstar,Ce,gamma,P,mstar,alpha_1,etatilda
%Using my hand solution, this function solves the non linear IC, and TC for P
%This is the "P" version
global an b Ds Dc D3 rho h C Cstar Ce gamma theta mstar alpha_1 etatilda

k1 = sqrt(Nn/Ds);
k2 = sqrt(Nn/Dc);
k3 = sqrt(Nn/D3);

%Now, the F's
%b2
F1 = -1/2*(-
2*Ds*k1^2*Dc^2*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)+2*Ds*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k
3*D3*sinh(k1*an)*cosh(k2*an)+2*Ds^2*k1^2*Dc*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*si
nh(k1*an)-2*Ds*k1^2*Dc^2*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)-
2*Ds*k1*Dc*k2^2*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)+2*Ds*k1^2*Dc*k2*D3*k3^2*sinh(k
3*b)*sinh(k1*an)*cosh(k2*an)-
2*sinh(k2*an)*Ds^2*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)+2*sinh(k2*b)*Ds^2*k1^2*D3*k3^2*sinh(
k3*b)*sinh(k1*an)-
2*Ds^2*k1^2*Dc*k2^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)+2*Ds^2*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*
b)*sinh(k1*an)+2*k2^2*sinh(k1*an)*Dc*D3*k3^2*sinh(k3*b)*Ds*sinh(k2*an)-
2*k2^2*sinh(k1*an)*Dc^2*D3*k3^2*sinh(k3*b)*sinh(k2*an)-
2*Ds^2*k1^2*Dc*k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)+2*sinh(k2*an)*Ds*k1^2*D3*k
```

```
3^2*sinh(k3*b)*sinh(k1*an)*Dc-
2*sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*Dc+2*Ds*k1^2*Dc^2*k2^2*sinh(k3*b)*sinh(
k2*b)*sinh(k1*an)-
2*Ds*k1*Dc^2*k2^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)+2*Ds*k1^2*Dc^2*
k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)+2*Ds^2*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sin
h(k3*b)*cosh(k2*b)*sinh(k1*an)-
2*Ds*k1^2*Dc^2*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)+2*Ds*k1*Dc^2*k2^
2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*an)*cosh(k1*an)+2*Ds*k1^2*Dc^2*k2*sinh(k3)*cosh(k3*b)*k3*s
inh(k3*b)*sinh(k1*an)*cosh(k2*an)+2*Ds*k1*Dc*k2^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*D3*sinh(k
2*an)*cosh(k1*an)-
2*Ds*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*D3*sinh(k1*an)*cosh(k2*an)-
2*Ds^2*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)+2*Ds*k1^2*Dc^2*k2*cosh(k
3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)-
2*Ds*k1^2*Dc^2*k2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k1*an)*cosh(k2*an)-
2*Ds*k1*Dc*k2^2*cosh(k3)*sinh(k3*b)^2*k3*D3*sinh(k2*an)*cosh(k1*an))/Nn/k2^2/Dc/(-
Dc*k2*Ds*k1^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^3*Ds*k1*sinh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+cosh(k2*b)*Ds*k1^2*D3*k3^2*sinh(
k3*b)*sinh(k1*an)*cosh(k2*an)-Dc*k2^2*Ds*k1^2*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^2*sinh(k1*an)*D3*k3^2*sinh(k3*b)*sinh(k2*an)^2+Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*cos
h(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*sinh(
k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*D
s*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*sinh(k
3)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+k2*sinh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*cosh(k1*an)*cosh(
k2*an)-k2*cosh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*cos
h(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*sin
h(k1*an)*D3*k3^2*sinh(k3*b)*cosh(k2*an)^2+Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*
sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*
Ds*k1*sinh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2*Ds*k1^2*sinh(k3
)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b
)*cosh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+sinh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(
k1*an)-cosh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)-
sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b
)^2*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k
2*an));
F2 = -1/2*(-
2*k2^2*sinh(k1*an)*Dc*D3*k3^2*sinh(k3*b)*mstar*Nn*sinh(k2*an)+2*sinh(k2*an)*Ds*k1^2*D3*k3^
2*sinh(k3*b)*sinh(k1*an)*mstar*Nn-
2*sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*mstar*Nn+2*Ds*k1^2*Dc*k2^2*sinh(k3*b)*s
inh(k2*b)*sinh(k1*an)*mstar*Nn-
2*Ds*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*mstar*Nn+2*Ds*k1^2*Dc*k2^2*sinh(k3)*cosh(
k3*b)^2*sinh(k2*b)*sinh(k1*an)*mstar*Nn-
2*Ds*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*mstar*Nn+2*Ds*k1^2
*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)*mstar*Nn-
2*Ds*k1^2*Dc*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*mstar*Nn)/Nn/k2^2/Dc/
(-Dc*k2*Ds*k1^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^3*Ds*k1*sinh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+cosh(k2*b)*Ds*k1^2*D3*k3^2*sinh(
```

k3*b)*sinh(k1*an)*cosh(k2*an)-Dc*k2^2*Ds*k1^2*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^2*sinh(k1*an)*D3*k3^2*sinh(k3*b)*sinh(k2*an)^2+Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*sinh(k3)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+k2*sinh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*cosh(k1*an)*cosh(k2*an)-k2*cosh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*sinh(k1*an)*D3*k3^2*sinh(k3*b)*cosh(k2*an)^2+Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*sinh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2*Ds*k1^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+sinh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)-cosh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)-
sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an));
F3 = -
1/2*(2*k2^2*sinh(k1*an)*Dc*D3*k3^2*sinh(k3*b)*rho*Nn^2*sinh(k2*an)+k2^2*sinh(k1*an)*Dc*D3*k3^2*sinh(k3*b)*h*Nn^2*sinh(k2*an)-
2*sinh(k2*an)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*rho*Nn^2-
sinh(k2*an)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*h*Nn^2+2*sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*rho*Nn^2+sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*h*Nn^2-
2*Ds*k1^2*Dc*k2^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*rho*Nn^2-
Ds*k1^2*Dc*k2^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*h*Nn^2+2*Ds*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*rho*Nn^2+2*Ds*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*rho*Nn^2+Ds*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*h*Nn^2-
2*Ds*k1^2*Dc*k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)*rho*Nn^2-
Ds*k1^2*Dc*k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)*h*Nn^2+Ds*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*h*Nn^2-
2*Ds*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)*rho*Nn^2-
Ds*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)*h*Nn^2+2*Ds*k1^2*Dc*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*rho*Nn^2+Ds*k1^2*Dc*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*h*Nn^2)/Nn/k2^2/Dc/(-
Dc*k2*Ds*k1^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^3*Ds*k1*sinh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+cosh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*cosh(k2*an)-Dc*k2^2*Ds*k1^2*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^2*sinh(k1*an)*D3*k3^2*sinh(k3*b)*sinh(k2*an)^2+Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*sinh(k3)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+k2*sinh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*cosh(k1*an)*cosh(k2*an)-k2*cosh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*cos

```
h(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*sin
h(k1*an)*D3*k3^2*sinh(k3*b)*cosh(k2*an)^2+Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*
sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*
Ds*k1*sinh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2*Ds*k1^2*sinh(k3
)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b
)*cosh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+sinh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(
k1*an)-cosh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)-
sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b
)^2*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k
2*an));
%b1
F4 = -1/2*(-
2*k1*sinh(k1*an)*Ds+2*k1*sinh(k1*an)*Dc*k2^2*sinh(k2*an)*Nn*F1+2*k1*sinh(k1*an)*Dc-
2*cosh(k1*an)*Nn*Dc*k2^3*cosh(k2*an)*F1)/Dc/k2^2/Nn/(-
sinh(k2*an)*cosh(k1*an)*k2+k1*sinh(k1*an)*cosh(k2*an));
F5 = -1/2*(2*k1*sinh(k1*an)*mstar*Nn+2*k1*sinh(k1*an)*Dc*k2^2*sinh(k2*an)*Nn*F2-
2*cosh(k1*an)*Nn*Dc*k2^3*cosh(k2*an)*F2)/Dc/k2^2/Nn/(-
sinh(k2*an)*cosh(k1*an)*k2+k1*sinh(k1*an)*cosh(k2*an));
F6 = -1/2*(-2*k1*sinh(k1*an)*rho*Nn^2+2*k1*sinh(k1*an)*Dc*k2^2*sinh(k2*an)*Nn*F3-
k1*sinh(k1*an)*h*Nn^2-2*cosh(k1*an)*Nn*Dc*k2^3*cosh(k2*an)*F3)/Dc/k2^2/Nn/(-
sinh(k2*an)*cosh(k1*an)*k2+k1*sinh(k1*an)*cosh(k2*an));
%c2
F7 = (-k3*sinh(k3*b)*Dc*k2^2*cosh(k2*b)*Nn*F4-k3*sinh(k3*b)*Dc*k2^2*sinh(k2*b)*Nn*F1-
k3*sinh(k3*b)*Dc+k3*sinh(k3*b)*D3+cosh(k3*b)*Nn*Dc*k2^3*sinh(k2*b)*F4+cosh(k3*b)*Nn*Dc*k2
^3*cosh(k2*b)*F1)/Nn/k3^3/D3/(-sinh(k3*b)^2+cosh(k3*b)^2);
F8 = (-k3*sinh(k3*b)*Dc*k2^2*cosh(k2*b)*Nn*F5-
k3*sinh(k3*b)*Dc*k2^2*sinh(k2*b)*Nn*F2+cosh(k3*b)*Nn*Dc*k2^3*sinh(k2*b)*F5+cosh(k3*b)*Nn*
Dc*k2^3*cosh(k2*b)*F2)/Nn/k3^3/D3/(-sinh(k3*b)^2+cosh(k3*b)^2);
F9 = (-k3*sinh(k3*b)*Dc*k2^2*cosh(k2*b)*Nn*F6-
k3*sinh(k3*b)*Dc*k2^2*sinh(k2*b)*Nn*F3+cosh(k3*b)*Nn*Dc*k2^3*sinh(k2*b)*F6+cosh(k3*b)*Nn*
Dc*k2^3*cosh(k2*b)*F3)/Nn/k3^3/D3/(-sinh(k3*b)^2+cosh(k3*b)^2);
%c1
F10 = (Dc*k2^3*sinh(k2*b)*F4+Dc*k2^3*cosh(k2*b)*F1-
D3*k3^3*cosh(k3*b)*F7)/D3/k3^3/sinh(k3*b);
F11 = (Dc*k2^3*sinh(k2*b)*F5+Dc*k2^3*cosh(k2*b)*F2-
D3*k3^3*cosh(k3*b)*F8)/D3/k3^3/sinh(k3*b);
F12 = (Dc*k2^3*sinh(k2*b)*F6+Dc*k2^3*cosh(k2*b)*F3-
D3*k3^3*cosh(k3*b)*F9)/D3/k3^3/sinh(k3*b);
%a1
F13 = 1/2*(-
2*Ds+2*Dc*k2^2*cosh(k2*an)*Nn*F4+2*Dc*k2^2*sinh(k2*an)*Nn*F1+2*Dc)/Ds/k1^2/cosh(k1*an)/N
n;
F14 =
1/2*(2*mstar*Nn+2*Dc*k2^2*cosh(k2*an)*Nn*F5+2*Dc*k2^2*sinh(k2*an)*Nn*F2)/Ds/k1^2/cosh(k1*a
n)/Nn;
F15 = 1/2*(-2*rho*Nn^2+2*Dc*k2^2*cosh(k2*an)*Nn*F6+2*Dc*k2^2*sinh(k2*an)*Nn*F3-
h*Nn^2)/Ds/k1^2/cosh(k1*an)/Nn;
%c3
F16 = -(k3*sinh(k3)*Nn*F10+k3*cosh(k3)*Nn*F7+1)/Nn;
F17 = -(k3*sinh(k3)*Nn*F11+k3*cosh(k3)*Nn*F8)/Nn;
F18 = -(k3*sinh(k3)*Nn*F12+k3*cosh(k3)*Nn*F9)/Nn;
%c4
```

```
F19 = -1/2*(2*cosh(k3)*Nn*F10+2*sinh(k3)*Nn*F7+2*F16*Nn+1)/Nn;
F20 = -1/2*(2*cosh(k3)*Nn*F11+2*sinh(k3)*Nn*F8+2*F17*Nn)/Nn;
F21 = -1/2*(2*cosh(k3)*Nn*F12+2*sinh(k3)*Nn*F9+2*F18*Nn)/Nn;
%b3
F22 = -k2*sinh(k2*b)*F4-k2*cosh(k2*b)*F1+k3*sinh(k3*b)*F10+k3*cosh(k3*b)*F7+F16;
F23 = -k2*sinh(k2*b)*F5-k2*cosh(k2*b)*F2+k3*sinh(k3*b)*F11+k3*cosh(k3*b)*F8+F17;
F24 = -k2*sinh(k2*b)*F6-k2*cosh(k2*b)*F3+k3*sinh(k3*b)*F12+k3*cosh(k3*b)*F9+F18;
%b4
F25 = -cosh(k2*b)*F4-sinh(k2*b)*F1-b*F22+cosh(k3*b)*F10+sinh(k3*b)*F7+b*F16+F19;
F26 = -cosh(k2*b)*F5-sinh(k2*b)*F2-b*F23+cosh(k3*b)*F11+sinh(k3*b)*F8+b*F17+F20;
F27 = -cosh(k2*b)*F6-sinh(k2*b)*F3-b*F24+cosh(k3*b)*F12+sinh(k3*b)*F9+b*F18+F21;
%a4
F28 = -cosh(k1*an)*F13+cosh(k2*an)*F4+sinh(k2*an)*F1+an*F22+F25;
F29 = -cosh(k1*an)*F14+cosh(k2*an)*F5+sinh(k2*an)*F2+an*F23+F26;
F30 = -cosh(k1*an)*F15+cosh(k2*an)*F6+sinh(k2*an)*F3+an*F24+F27;

%Now the B's
B0 = 1-an+an*alpha_1;
B1 = F13*k1*sinh(k1*an)+an/Nn;
B2 = (F14*theta+F15)*k1*sinh(k1*an);
B3 = -1/6*(-3*k1^2*F13^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*F13^2*Nn^2*an-
12*F13*Nn*k1*an*cosh(k1*an)+12*F13*Nn*sinh(k1*an)-2*an^3*k1)/k1/Nn^2;
B4 = -1/6*(-6*k1^2*F13*F14*theta*Nn^2*cosh(k1*an)*sinh(k1*an)+6*k1^3*F13*F14*theta*Nn^2*an-
6*k1^2*F13*F15*Nn^2*cosh(k1*an)*sinh(k1*an)+6*k1^3*F13*F15*Nn^2*an-
12*F14*theta*Nn*k1*an*cosh(k1*an)+12*F14*theta*Nn*sinh(k1*an)-
12*F15*Nn*k1*an*cosh(k1*an)+12*F15*Nn*sinh(k1*an))/k1/Nn^2;
B5 = -1/6*(-3*k1^2*F14^2*theta^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*F14^2*theta^2*Nn^2*an-
6*k1^2*F14*theta*F15*Nn^2*cosh(k1*an)*sinh(k1*an)+6*k1^3*F14*theta*F15*Nn^2*an-
3*k1^2*F15^2*Nn^2*cosh(k1*an)*sinh(k1*an)+3*k1^3*F15^2*Nn^2*an)/k1/Nn^2;
B6 = -1/6*(6*F22*an^2*Nn*k2-2*b^3*k2+2*an^3*k2+12*F1*Nn*cosh(k2*b)+12*F4*Nn*sinh(k2*b)-
12*F1*Nn*cosh(k2*an)-12*F4*Nn*sinh(k2*an)+3*k2^3*F1^2*Nn^2*an-
3*k2^3*F4^2*Nn^2*an+3*k2^3*F4^2*Nn^2*b-3*k2^3*F1^2*Nn^2*b-12*F1*Nn*k2*b*sinh(k2*b)-
12*F4*Nn*k2*b*cosh(k2*b)+12*F1*Nn*k2*an*sinh(k2*an)+3*k2^2*F4^2*Nn^2*cosh(k2*an)*sinh(k2*
an)-
3*k2^2*F4^2*Nn^2*cosh(k2*b)*sinh(k2*b)+12*F4*Nn*k2*an*cosh(k2*an)+3*k2^2*F1^2*Nn^2*cosh(k
2*an)*sinh(k2*an)-
12*F4*F22*cosh(k2*b)*Nn^2*k2+12*F1*F22*sinh(k2*an)*Nn^2*k2+6*F22^2*an*Nn^2*k2-
3*k2^2*F1^2*Nn^2*cosh(k2*b)*sinh(k2*b)+6*k2^2*F4*F1*cosh(k2*an)^2*Nn^2-
6*k2^2*F4*F1*cosh(k2*b)^2*Nn^2-
12*F1*F22*sinh(k2*b)*Nn^2*k2+12*F4*F22*cosh(k2*an)*Nn^2*k2-6*F22^2*b*Nn^2*k2-
6*F22*b^2*Nn*k2)/Nn^2/k2;
B7 = -1/6*(-
6*k2^2*F1*F2*theta*Nn^2*cosh(k2*b)*sinh(k2*b)+12*F5*theta*F22*cosh(k2*an)*Nn^2*k2+6*k2^2*F
4*F2*theta*cosh(k2*an)^2*Nn^2+12*F4*F23*theta*cosh(k2*an)*Nn^2*k2+12*F3*Nn*k2*an*sinh(k2*a
n)-
12*F5*theta*Nn*k2*b*cosh(k2*b)+12*F5*theta*Nn*sinh(k2*b)+6*k2^2*F6*F1*cosh(k2*an)^2*Nn^2+6
*k2^2*F4*F3*cosh(k2*an)^2*Nn^2-6*k2^2*F6*F1*cosh(k2*b)^2*Nn^2-
6*k2^2*F5*theta*F1*cosh(k2*b)^2*Nn^2-12*F5*theta*F22*cosh(k2*b)*Nn^2*k2-
12*F4*F23*theta*cosh(k2*b)*Nn^2*k2-
12*F22*F23*theta*b*Nn^2*k2+12*F22*F23*theta*an*Nn^2*k2-
12*F22*F24*b*Nn^2*k2+12*F22*F24*an*Nn^2*k2-6*F24*b^2*Nn*k2-
6*k2^2*F4*F3*cosh(k2*b)^2*Nn^2+12*F1*F23*theta*sinh(k2*an)*Nn^2*k2-
12*F2*theta*F22*sinh(k2*b)*Nn^2*k2+12*F2*theta*F22*sinh(k2*an)*Nn^2*k2+6*k2^2*F5*theta*F1*
cosh(k2*an)^2*Nn^2+12*F6*Nn*k2*an*cosh(k2*an)+12*F2*theta*Nn*k2*an*sinh(k2*an)+12*F5*theta
*Nn*k2*an*cosh(k2*an)-12*F6*Nn*k2*b*cosh(k2*b)-
12*F2*theta*Nn*k2*b*sinh(k2*b)+12*F2*theta*Nn*cosh(k2*b)-12*F3*Nn*k2*b*sinh(k2*b)-
```

```
6*k2^2*F1*F3*Nn^2*cosh(k2*b)*sinh(k2*b)+6*k2^2*F1*F2*theta*Nn^2*cosh(k2*an)*sinh(k2*an)+6*k
2^2*F1*F3*Nn^2*cosh(k2*an)*sinh(k2*an)-12*F2*theta*Nn*cosh(k2*an)-
12*F1*F24*sinh(k2*b)*Nn^2*k2-12*F3*F22*sinh(k2*b)*Nn^2*k2-
12*F4*F24*cosh(k2*b)*Nn^2*k2+12*F3*F22*sinh(k2*an)*Nn^2*k2+12*F4*F24*cosh(k2*an)*Nn^2*k
2-
12*F6*F22*cosh(k2*b)*Nn^2*k2+12*F6*F22*cosh(k2*an)*Nn^2*k2+12*F1*F24*sinh(k2*an)*Nn^2*k
2-12*F1*F23*theta*sinh(k2*b)*Nn^2*k2-6*k2^2*F4*F2*theta*cosh(k2*b)^2*Nn^2-
12*F5*theta*Nn*sinh(k2*an)+6*F23*theta*an^2*Nn*k2+6*F24*an^2*Nn*k2-
6*F23*theta*b^2*Nn*k2+6*k2^3*F4*F5*theta*Nn^2*b+6*k2^3*F4*F6*Nn^2*b-
6*k2^3*F1*F3*Nn^2*b-6*k2^3*F4*F5*theta*Nn^2*an-
6*k2^3*F4*F6*Nn^2*an+6*k2^3*F1*F3*Nn^2*an-
6*k2^3*F1*F2*theta*Nn^2*b+6*k2^3*F1*F2*theta*Nn^2*an+12*F6*Nn*sinh(k2*b)+12*F3*Nn*cosh(k
2*b)-12*F6*Nn*sinh(k2*an)-12*F3*Nn*cosh(k2*an)-
6*k2^2*F4*F5*theta*Nn^2*cosh(k2*b)*sinh(k2*b)-
6*k2^2*F4*F6*Nn^2*cosh(k2*b)*sinh(k2*b)+6*k2^2*F4*F5*theta*Nn^2*cosh(k2*an)*sinh(k2*an)+6*k
2^2*F4*F6*Nn^2*cosh(k2*an)*sinh(k2*an))/Nn^2/k2;
B8 = -
1/6*(12*F2*theta*F24*sinh(k2*an)*Nn^2*k2+6*k2^2*F6*F2*theta*cosh(k2*an)^2*Nn^2+6*k2^2*F6*F
3*cosh(k2*an)^2*Nn^2+6*F24^2*an*Nn^2*k2-
6*F24^2*b*Nn^2*k2+3*k2^3*F3^2*Nn^2*an+3*k2^3*F6^2*Nn^2*b-3*k2^3*F3^2*Nn^2*b-
3*k2^3*F6^2*Nn^2*an+6*k2^3*F2*theta*F3*Nn^2*an+6*k2^3*F5*theta*F6*Nn^2*b-
6*k2^3*F5*theta*F6*Nn^2*an-6*k2^3*F2*theta*F3*Nn^2*b-
3*k2^3*F5^2*theta^2*Nn^2*an+3*k2^3*F2^2*theta^2*Nn^2*an+3*k2^3*F5^2*theta^2*Nn^2*b-
3*k2^3*F2^2*theta^2*Nn^2*b+3*k2^2*F3^2*Nn^2*cosh(k2*an)*sinh(k2*an)-
3*k2^2*F6^2*Nn^2*cosh(k2*b)*sinh(k2*b)-
3*k2^2*F3^2*Nn^2*cosh(k2*b)*sinh(k2*b)+3*k2^2*F6^2*Nn^2*cosh(k2*an)*sinh(k2*an)+6*k2^2*F2*
theta*F3*Nn^2*cosh(k2*an)*sinh(k2*an)-
6*k2^2*F5*theta*F6*Nn^2*cosh(k2*b)*sinh(k2*b)+6*k2^2*F5*theta*F6*Nn^2*cosh(k2*an)*sinh(k2*a
n)-
6*k2^2*F2*theta*F3*Nn^2*cosh(k2*b)*sinh(k2*b)+3*k2^2*F5^2*theta^2*Nn^2*cosh(k2*an)*sinh(k2*a
n)+3*k2^2*F2^2*theta^2*Nn^2*cosh(k2*an)*sinh(k2*an)-
3*k2^2*F5^2*theta^2*Nn^2*cosh(k2*b)*sinh(k2*b)-
3*k2^2*F2^2*theta^2*Nn^2*cosh(k2*b)*sinh(k2*b)-12*F3*F24*sinh(k2*b)*Nn^2*k2-
12*F6*F24*cosh(k2*b)*Nn^2*k2+12*F6*F24*cosh(k2*an)*Nn^2*k2-
6*k2^2*F6*F2*theta*cosh(k2*b)^2*Nn^2-
6*k2^2*F5*theta^2*F2*cosh(k2*b)^2*Nn^2+6*k2^2*F5*theta*F3*cosh(k2*an)^2*Nn^2+6*k2^2*F5*the
ta^2*F2*cosh(k2*an)^2*Nn^2-
6*k2^2*F5*theta*F3*cosh(k2*b)^2*Nn^2+12*F3*F24*sinh(k2*an)*Nn^2*k2-
12*F3*F23*theta*sinh(k2*b)*Nn^2*k2-12*F2*theta^2*F23*sinh(k2*b)*Nn^2*k2-
12*F6*F23*theta*cosh(k2*b)*Nn^2*k2-
12*F2*theta*F24*sinh(k2*b)*Nn^2*k2+12*F3*F23*theta*sinh(k2*an)*Nn^2*k2+12*F2*theta^2*F23*si
nh(k2*an)*Nn^2*k2+12*F6*F23*theta*cosh(k2*an)*Nn^2*k2-
12*F5*theta^2*F23*cosh(k2*b)*Nn^2*k2-
12*F5*theta*F24*cosh(k2*b)*Nn^2*k2+12*F5*theta^2*F23*cosh(k2*an)*Nn^2*k2+12*F5*theta*F24*c
osh(k2*an)*Nn^2*k2-6*k2^2*F6*F3*cosh(k2*b)^2*Nn^2+6*F23^2*theta^2*an*Nn^2*k2-
12*F23*theta*F24*b*Nn^2*k2-
6*F23^2*theta^2*b*Nn^2*k2+12*F23*theta*F24*an*Nn^2*k2)/Nn^2/k2;
B9 = -1/6*(-2*k3-6*F16*Nn*k3+3*k3^2*F10^2*Nn^2*sinh(k3*b)*cosh(k3*b)-
12*F10*Nn*k3*cosh(k3)+3*k3^3*F7^2*Nn^2*b+12*F7*Nn*k3*b*sinh(k3*b)+12*F7*F16*sinh(k3*b)*
Nn^2*k3+3*k3^2*F7^2*Nn^2*sinh(k3*b)*cosh(k3*b)+12*F10*Nn*k3*b*cosh(k3*b)+6*k3^2*F10*F7*
cosh(k3*b)^2*Nn^2+12*F10*F16*cosh(k3*b)*Nn^2*k3+6*F16^2*b*Nn^2*k3+6*F16*b^2*Nn*k3-
6*F16^2*Nn^2*k3-12*F10*F16*cosh(k3)*Nn^2*k3-12*F7*F16*sinh(k3)*Nn^2*k3-
12*F7*Nn*k3*sinh(k3)-3*k3^3*F10^2*Nn^2*b-3*k3^2*F10^2*Nn^2*sinh(k3)*cosh(k3)-
6*k3^2*F10*F7*cosh(k3)^2*Nn^2-12*F7*Nn*cosh(k3*b)-12*F10*Nn*sinh(k3*b)-
3*k3^2*F7^2*Nn^2*sinh(k3)*cosh(k3)-
```

```
3*k3^3*F7^2*Nn^2+12*F7*Nn*cosh(k3)+3*k3^3*F10^2*Nn^2+12*F10*Nn*sinh(k3)+2*b^3*k3)/Nn^2/
k3;
B10 = -1/6*(-6*k3^3*F7*F8*theta*Nn^2-6*F18*Nn*k3+12*F9*Nn*cosh(k3)+12*F12*Nn*sinh(k3)-
12*F12*Nn*sinh(k3*b)-12*F9*Nn*cosh(k3*b)+6*k3^3*F10*F11*theta*Nn^2-
12*F16*F17*theta*Nn^2*k3-12*F8*theta*F16*sinh(k3)*Nn^2*k3-
6*k3^2*F10*F12*Nn^2*sinh(k3)*cosh(k3)-6*k3^2*F12*F7*cosh(k3)^2*Nn^2-
12*F7*F17*theta*sinh(k3)*Nn^2*k3-12*F10*F18*cosh(k3)*Nn^2*k3-
12*F12*Nn*k3*cosh(k3)+12*F8*theta*Nn*cosh(k3)-6*k3^2*F7*F9*Nn^2*sinh(k3)*cosh(k3)-
6*k3^2*F7*F8*theta*Nn^2*sinh(k3)*cosh(k3)-
12*F12*F16*cosh(k3)*Nn^2*k3+12*F8*theta*F16*sinh(k3*b)*Nn^2*k3+12*F9*Nn*k3*b*sinh(k3*b)-
6*k3^3*F10*F11*theta*Nn^2*b-6*k3^3*F10*F12*Nn^2*b-12*F10*F17*theta*cosh(k3)*Nn^2*k3-
12*F11*theta*F16*cosh(k3)*Nn^2*k3-6*k3^2*F10*F9*cosh(k3)^2*Nn^2-
12*F11*theta*Nn*sinh(k3*b)+12*F12*Nn*k3*b*cosh(k3*b)+6*k3^2*F12*F7*cosh(k3*b)^2*Nn^2+6*k
3^2*F7*F8*theta*Nn^2*sinh(k3*b)*cosh(k3*b)+6*k3^2*F7*F9*Nn^2*sinh(k3*b)*cosh(k3*b)-
12*F8*theta*Nn*cosh(k3*b)+12*F8*theta*Nn*k3*b*sinh(k3*b)+12*F11*theta*Nn*k3*b*cosh(k3*b)+12
*F11*theta*F16*cosh(k3*b)*Nn^2*k3+12*F9*F16*sinh(k3*b)*Nn^2*k3+6*k3^2*F10*F8*theta*cosh(k
3*b)^2*Nn^2+12*F10*F18*cosh(k3*b)*Nn^2*k3+12*F12*F16*cosh(k3*b)*Nn^2*k3+12*F7*F18*sinh(
k3*b)*Nn^2*k3+6*F18*b^2*Nn*k3+6*F17*theta*b^2*Nn*k3-
6*k3^3*F7*F9*Nn^2+12*F16*F18*b*Nn^2*k3+12*F16*F17*theta*b*Nn^2*k3-
6*k3^2*F10*F11*theta*Nn^2*sinh(k3)*cosh(k3)-12*F9*Nn*k3*sinh(k3)-
12*F8*theta*Nn*k3*sinh(k3)+6*k3^3*F7*F8*theta*Nn^2*b+6*k3^3*F7*F9*Nn^2*b+6*k3^3*F10*F12
*Nn^2+6*k3^2*F10*F11*theta*Nn^2*sinh(k3*b)*cosh(k3*b)+6*k3^2*F10*F12*Nn^2*sinh(k3*b)*cosh(
k3*b)+6*k3^2*F10*F9*cosh(k3*b)^2*Nn^2+12*F10*F17*theta*cosh(k3*b)*Nn^2*k3+6*k3^2*F11*thet
a*F7*cosh(k3*b)^2*Nn^2+12*F7*F17*theta*sinh(k3*b)*Nn^2*k3-6*F17*theta*Nn*k3-
12*F16*F18*Nn^2*k3-12*F7*F18*sinh(k3)*Nn^2*k3-12*F9*F16*sinh(k3)*Nn^2*k3-
12*F11*theta*Nn*k3*cosh(k3)-6*k3^2*F10*F8*theta*cosh(k3)^2*Nn^2-
6*k3^2*F11*theta*F7*cosh(k3)^2*Nn^2+12*F11*theta*Nn*sinh(k3))/Nn^2/k3;
B11 = -1/6*(-3*k3^3*F9^2*Nn^2+3*k3^3*F12^2*Nn^2-6*k3^3*F8*theta*F9*Nn^2-
3*k3^2*F8^2*theta^2*Nn^2*sinh(k3)*cosh(k3)-3*k3^2*F11^2*theta^2*Nn^2*sinh(k3)*cosh(k3)-
3*k3^2*F12^2*Nn^2*sinh(k3)*cosh(k3)-6*k3^2*F11*theta*F9*cosh(k3)^2*Nn^2-
12*F11*theta*F18*cosh(k3)*Nn^2*k3-12*F11*theta^2*F17*cosh(k3)*Nn^2*k3-6*F18^2*Nn^2*k3-
3*k3^3*F8^2*theta^2*Nn^2+3*k3^3*F11^2*theta^2*Nn^2-
12*F12*F17*theta*cosh(k3)*Nn^2*k3+6*k3^3*F8*theta*F9*Nn^2*b+3*k3^3*F8^2*theta^2*Nn^2*b-
12*F12*F18*cosh(k3)*Nn^2*k3-
6*k3^2*F11*theta^2*F8*cosh(k3)^2*Nn^2+6*k3^2*F11*theta^2*F8*cosh(k3*b)^2*Nn^2+6*k3^2*F11*
theta*F12*Nn^2*sinh(k3*b)*cosh(k3*b)+3*k3^2*F11^2*theta^2*Nn^2*sinh(k3*b)*cosh(k3*b)+6*k3^2*
F11*theta*F9*cosh(k3*b)^2*Nn^2+6*k3^2*F12*F8*theta*cosh(k3*b)^2*Nn^2+12*F12*F18*cosh(k3*b)
*Nn^2*k3+12*F9*F18*sinh(k3*b)*Nn^2*k3+12*F9*F17*theta*sinh(k3*b)*Nn^2*k3+12*F11*theta^2*F
17*cosh(k3*b)*Nn^2*k3+12*F11*theta*F18*cosh(k3*b)*Nn^2*k3+12*F12*F17*theta*cosh(k3*b)*Nn^
2*k3+6*k3^2*F12*F9*cosh(k3*b)^2*Nn^2+6*F17^2*theta^2*b*Nn^2*k3-
6*k3^3*F11*theta*F12*Nn^2*b+6*k3^2*F8*theta*F9*Nn^2*sinh(k3*b)*cosh(k3*b)+3*k3^2*F8^2*thet
a^2*Nn^2*sinh(k3*b)*cosh(k3*b)+3*k3^2*F9^2*Nn^2*sinh(k3*b)*cosh(k3*b)+3*k3^2*F12^2*Nn^2*si
nh(k3*b)*cosh(k3*b)+12*F8*theta^2*F17*sinh(k3*b)*Nn^2*k3+12*F8*theta*F18*sinh(k3*b)*Nn^2*k3
-12*F8*theta^2*F17*sinh(k3)*Nn^2*k3-6*F17^2*theta^2*Nn^2*k3-12*F17*theta*F18*Nn^2*k3-
3*k3^2*F9^2*Nn^2*sinh(k3)*cosh(k3)-12*F8*theta*F18*sinh(k3)*Nn^2*k3-
12*F9*F17*theta*sinh(k3)*Nn^2*k3-3*k3^3*F11^2*theta^2*Nn^2*b-
6*k3^2*F12*F8*theta*cosh(k3)^2*Nn^2-
6*k3^2*F12*F9*cosh(k3)^2*Nn^2+6*k3^3*F11*theta*F12*Nn^2+6*F18^2*b*Nn^2*k3+12*F17*theta*
F18*b*Nn^2*k3+3*k3^3*F9^2*Nn^2*b-3*k3^3*F12^2*Nn^2*b-
6*k3^2*F8*theta*F9*Nn^2*sinh(k3)*cosh(k3)-12*F9*F18*sinh(k3)*Nn^2*k3-
6*k3^2*F11*theta*F12*Nn^2*sinh(k3)*cosh(k3))/Nn^2/k3;

Ca = 1/C-an/C+an/Cstar;
rt = h/2 + rho;
%Now the Q's, such that the integrability condition is 0= Q1P^2+Q2P+Q3
Q1 = -1/2*B3-1/2*B6-1/2*B9;
```

```
Q2 = -rt*B1-1/2*B4-1/2*B7-1/2*B10;
Q3 = Nn*Ca+B0*theta-rt*B2-1/2*B5-1/2*B8-1/2*B11;

%And the transversality equation is 0=Q4P^2+Q5P+Q6bar
Q4 =
1/2*Dc*(1/(Nn^2)+k2^4*cosh(k2*an)^2*F4^2+k2^4*sinh(k2*an)^2*F1^2+2*k2^2*sinh(k2*an)*F1/Nn+2
*k2^4*cosh(k2*an)*F4*sinh(k2*an)*F1+2*k2^2*cosh(k2*an)*F4/Nn)-
1/2*Ds*(k1^4*cosh(k1*an)^2*F13^2+2*k1^2*cosh(k1*an)*F13/Nn+1/(Nn^2));
Q5 =
1/2*Dc*(2*k2^4*cosh(k2*an)*F4*sinh(k2*an)*F3+2*k2^4*sinh(k2*an)^2*F1*F2*theta+2*k2^2*sinh(k2
*an)*F2*theta/Nn+2*k2^2*sinh(k2*an)*F3/Nn+2*k2^4*cosh(k2*an)^2*F4*F5*theta+2*k2^4*cosh(k2*a
n)^2*F4*F6+2*k2^4*cosh(k2*an)*F4*sinh(k2*an)*F2*theta+2*k2^4*cosh(k2*an)*F5*theta*sinh(k2*an)
*F1+2*k2^2*cosh(k2*an)*F5*theta/Nn+2*k2^4*cosh(k2*an)*F6*sinh(k2*an)*F1+2*k2^2*cosh(k2*an)*
F6/Nn+2*k2^4*sinh(k2*an)^2*F1*F3)-
1/2*Ds*(2*k1^4*cosh(k1*an)^2*F13*F14*theta+2*k1^4*cosh(k1*an)^2*F13*F15+2*k1^2*cosh(k1*an)*
F14*theta/Nn+2*k1^2*cosh(k1*an)*F15/Nn);
Q6 =
1/2*Dc*(k2^4*sinh(k2*an)^2*F3^2+k2^4*cosh(k2*an)^2*F5^2*theta^2+k2^4*sinh(k2*an)^2*F2^2*theta
^2+2*k2^4*sinh(k2*an)^2*F2*theta*F3+k2^4*cosh(k2*an)^2*F6^2+2*k2^4*cosh(k2*an)^2*F5*theta*F
6+2*k2^4*cosh(k2*an)*F5*theta^2*sinh(k2*an)*F2+2*k2^4*cosh(k2*an)*F5*theta*sinh(k2*an)*F3+2*k
2^4*cosh(k2*an)*F6*sinh(k2*an)*F2*theta+2*k2^4*cosh(k2*an)*F6*sinh(k2*an)*F3)-
1/2*Ds*(k1^4*cosh(k1*an)^2*F14^2*theta^2+2*k1^4*cosh(k1*an)^2*F14*theta*F15+k1^4*cosh(k1*an)
^2*F15^2)+1/2*Nn^2/Ce+Nn*(1-alpha_1)*theta+1/2*etatilda*theta^2;
Q6bar = Q6-2*gamma;

%Now we solve simultaneously the IC and TC
Q7 = Q5*Q1-Q2*Q4;
Q8 = Q6bar*Q1-Q3*Q4;
PBF = -Q8/Q7;
BF = Q4*(PBF^2)+Q5*PBF+Q6bar; %Sub solution into TE
%delta = -(a1+a4)
DelBF = -(PBF*F13+F15+theta*F14+PBF*F28+F30+theta*F29);
stiff = PBF/DelBF;

function[a_1,a_2,a_3,a_4,b_1,b_2,b_3,b_4,c_1,c_2,c_3,c_4,k1,k2,k3] =
BFtenthermCONST_CFIXCZ_2(Nn,Pc,new_a)
% Uses the hand solution to find the const. of integ.
global an b Ds Dc D3 rho h C Cstar Ce gamma theta mstar alpha_1 etatilda theta
an=new_a;
k1 = sqrt(Nn/Ds);
k2 = sqrt(Nn/Dc);
k3 = sqrt(Nn/D3);
%Now, the F's
%b2
F1 = -1/2*(-
2*Ds*k1^2*Dc^2*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)+2*Ds*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k
3*D3*sinh(k1*an)*cosh(k2*an)+2*Ds^2*k1^2*Dc*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*si
nh(k1*an)-2*Ds*k1^2*Dc^2*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)-
2*Ds*k1*Dc*k2^2*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)+2*Ds*k1^2*Dc*k2*D3*k3^2*sinh(k
3*b)*sinh(k1*an)*cosh(k2*an)-
2*sinh(k2*an)*Ds^2*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)+2*sinh(k2*b)*Ds^2*k1^2*D3*k3^2*sinh(
k3*b)*sinh(k1*an)-
2*Ds^2*k1^2*Dc*k2^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)+2*Ds^2*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*
b)*sinh(k1*an)+2*k2^2*sinh(k1*an)*Dc*D3*k3^2*sinh(k3*b)*Ds*sinh(k2*an)-
2*k2^2*sinh(k1*an)*Dc^2*D3*k3^2*sinh(k3*b)*sinh(k2*an)-
```

```
2*Ds^2*k1^2*Dc*k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)+2*sinh(k2*an)*Ds*k1^2*D3*k
3^2*sinh(k3*b)*sinh(k1*an)*Dc-
2*sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*Dc+2*Ds*k1^2*Dc^2*k2^2*sinh(k3*b)*sinh(
k2*b)*sinh(k1*an)-
2*Ds*k1*Dc^2*k2^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)+2*Ds*k1^2*Dc^2*
k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)+2*Ds^2*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sin
h(k3*b)*cosh(k2*b)*sinh(k1*an)-
2*Ds*k1^2*Dc^2*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)+2*Ds*k1*Dc^2*k2^
2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*an)*cosh(k1*an)+2*Ds*k1^2*Dc^2*k2*sinh(k3)*cosh(k3*b)*k3*s
inh(k3*b)*sinh(k1*an)*cosh(k2*an)+2*Ds*k1*Dc*k2^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*D3*sinh(k
2*an)*cosh(k1*an)-
2*Ds*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*D3*sinh(k1*an)*cosh(k2*an)-
2*Ds^2*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)+2*Ds*k1^2*Dc^2*k2*cosh(k
3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)-
2*Ds*k1^2*Dc^2*k2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k1*an)*cosh(k2*an)-
2*Ds*k1*Dc*k2^2*cosh(k3)*sinh(k3*b)^2*k3*D3*sinh(k2*an)*cosh(k1*an))/Nn/k2^2/Dc/(-
Dc*k2*Ds*k1^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^3*Ds*k1*sinh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+cosh(k2*b)*Ds*k1^2*D3*k3^2*sinh(
k3*b)*sinh(k1*an)*cosh(k2*an)-Dc*k2^2*Ds*k1^2*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^2*sinh(k1*an)*D3*k3^2*sinh(k3*b)*sinh(k2*an)^2+Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*cos
h(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*sinh(
k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*D
s*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*sinh(k
3)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+k2*sinh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*cosh(k1*an)*cosh(
k2*an)-k2*cosh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*cos
h(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*sin
h(k1*an)*D3*k3^2*sinh(k3*b)*cosh(k2*an)^2+Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*
sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*
Ds*k1*sinh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2*Ds*k1^2*sinh(k3
)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b
)*cosh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+sinh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(
k1*an)-cosh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)-
sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b
)^2*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k
2*an));
F2 = -1/2*(-
2*k2^2*sinh(k1*an)*Dc*D3*k3^2*sinh(k3*b)*mstar*Nn*sinh(k2*an)+2*sinh(k2*an)*Ds*k1^2*D3*k3^
2*sinh(k3*b)*sinh(k1*an)*mstar*Nn-
2*sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*mstar*Nn+2*Ds*k1^2*Dc*k2^2*sinh(k3*b)*s
inh(k2*b)*sinh(k1*an)*mstar*Nn-
2*Ds*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*mstar*Nn+2*Ds*k1^2*Dc*k2^2*sinh(k3)*cosh(
k3*b)^2*sinh(k2*b)*sinh(k1*an)*mstar*Nn-
2*Ds*k1^2*Dc*k2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*mstar*Nn+2*Ds*k1^2
*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)*mstar*Nn-
2*Ds*k1^2*Dc*k2^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*mstar*Nn)/Nn/k2^2/Dc/
(-Dc*k2*Ds*k1^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
```

```
Dc*k2^3*Ds*k1*sinh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+cosh(k2*b)*Ds*k1^2*D3*k3^2*sinh(
k3*b)*sinh(k1*an)*cosh(k2*an)-Dc*k2^2*Ds*k1^2*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^2*sinh(k1*an)*D3*k3^2*sinh(k3*b)*sinh(k2*an)^2+Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*cos
h(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*sinh(
k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*D
s*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*sinh(k
3)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+k2*sinh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*cosh(k1*an)*cosh(
k2*an)-k2*cosh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*cos
h(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*sin
h(k1*an)*D3*k3^2*sinh(k3*b)*cosh(k2*an)^2+Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*
sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*
Ds*k1*sinh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2*Ds*k1^2*sinh(k3
)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b
)*cosh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+sinh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(
k1*an)-cosh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)-
sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b
)^2*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k
2*an));
F3 = -
1/2*(2*k2^2*sinh(k1*an)*Dc*D3*k3^2*sinh(k3*b)*rho*Nn^2*sinh(k2*an)+k2^2*sinh(k1*an)*Dc*D3*k
3^2*sinh(k3*b)*h*Nn^2*sinh(k2*an)-
2*sinh(k2*an)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*rho*Nn^2-
sinh(k2*an)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*h*Nn^2+2*sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh
(k3*b)*sinh(k1*an)*rho*Nn^2+sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*h*Nn^2-
2*Ds*k1^2*Dc*k2^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*rho*Nn^2-
Ds*k1^2*Dc*k2^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*h*Nn^2+2*Ds*k1^2*Dc*k2*sinh(k3)*cosh(k3*b
)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*rho*Nn^2+2*Ds*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1
*an)*rho*Nn^2+Ds*k1^2*Dc*k2^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*h*Nn^2-
2*Ds*k1^2*Dc*k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)*rho*Nn^2-
Ds*k1^2*Dc*k2^2*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*sinh(k1*an)*h*Nn^2+Ds*k1^2*Dc*k2*sinh(k3)*
cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*h*Nn^2-
2*Ds*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)*rho*Nn^2-
Ds*k1^2*Dc*k2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*sinh(k1*an)*h*Nn^2+2*Ds*k1^2*Dc*k2^2*cos
h(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*rho*Nn^2+Ds*k1^2*Dc*k2^2*cosh(k3)*sinh(k3*
b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*h*Nn^2)/Nn/k2^2/Dc/(-
Dc*k2*Ds*k1^2*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^3*Ds*k1*sinh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+cosh(k2*b)*Ds*k1^2*D3*k3^2*sinh(
k3*b)*sinh(k1*an)*cosh(k2*an)-Dc*k2^2*Ds*k1^2*sinh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^2*sinh(k1*an)*D3*k3^2*sinh(k3*b)*sinh(k2*an)^2+Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*cos
h(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b)^2*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*sinh(
k3)*cosh(k3*b)*k3*sinh(k3*b)*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)-
Dc*k2^2*Ds*k1*sinh(k3)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*D
s*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*sinh(k
3)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+k2*sinh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*cosh(k1*an)*cosh(
k2*an)-k2*cosh(k2*b)*Ds*k1*D3*k3^2*sinh(k3*b)*sinh(k2*an)*cosh(k1*an)-
```

```
Dc*k2^2*Ds*k1*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2^2*Ds*k1*cos
h(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2^2*Ds*k1^2*cosh(k3)*sinh(k3*b)*cosh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*sin
h(k1*an)*D3*k3^2*sinh(k3*b)*cosh(k2*an)^2+Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*cosh(k2*b)*
sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*
Ds*k1*sinh(k3*b)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)-
Dc*k2*Ds*k1^2*cosh(k3)*sinh(k3*b)^2*k3*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k3*b)^2*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+Dc*k2*Ds*k1^2*sinh(k3
)*cosh(k3*b)*k3*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*cosh(k2*an)+Dc*k2^3*Ds*k1*cosh(k3)*sinh(k3*b
)*cosh(k3*b)*sinh(k2*b)*cosh(k1*an)*cosh(k2*an)+sinh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(
k1*an)-cosh(k2*an)^2*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)-
sinh(k2*b)*Ds*k1^2*D3*k3^2*sinh(k3*b)*sinh(k1*an)*sinh(k2*an)-
Dc*k2^3*Ds*k1*sinh(k3)*cosh(k2*b)*sinh(k2*an)*cosh(k1*an)+Dc*k2^2*Ds*k1^2*sinh(k3)*cosh(k3*b
)^2*sinh(k2*b)*sinh(k1*an)*sinh(k2*an)+Dc*k2^2*Ds*k1^2*sinh(k3*b)*sinh(k2*b)*sinh(k1*an)*sinh(k
2*an));
%b1
F4 = -1/2*(-
2*k1*sinh(k1*an)*Ds+2*k1*sinh(k1*an)*Dc*k2^2*sinh(k2*an)*Nn*F1+2*k1*sinh(k1*an)*Dc-
2*cosh(k1*an)*Nn*Dc*k2^3*cosh(k2*an)*F1)/Dc/k2^2/Nn/(-
sinh(k2*an)*cosh(k1*an)*k2+k1*sinh(k1*an)*cosh(k2*an));
F5 = -1/2*(2*k1*sinh(k1*an)*mstar*Nn+2*k1*sinh(k1*an)*Dc*k2^2*sinh(k2*an)*Nn*F2-
2*cosh(k1*an)*Nn*Dc*k2^3*cosh(k2*an)*F2)/Dc/k2^2/Nn/(-
sinh(k2*an)*cosh(k1*an)*k2+k1*sinh(k1*an)*cosh(k2*an));
F6 = -1/2*(-2*k1*sinh(k1*an)*rho*Nn^2+2*k1*sinh(k1*an)*Dc*k2^2*sinh(k2*an)*Nn*F3-
k1*sinh(k1*an)*h*Nn^2-2*cosh(k1*an)*Nn*Dc*k2^3*cosh(k2*an)*F3)/Dc/k2^2/Nn/(-
sinh(k2*an)*cosh(k1*an)*k2+k1*sinh(k1*an)*cosh(k2*an));
%c2
F7 = (-k3*sinh(k3*b)*Dc*k2^2*cosh(k2*b)*Nn*F4-k3*sinh(k3*b)*Dc*k2^2*sinh(k2*b)*Nn*F1-
k3*sinh(k3*b)*Dc+k3*sinh(k3*b)*D3+cosh(k3*b)*Nn*Dc*k2^3*sinh(k2*b)*F4+cosh(k3*b)*Nn*Dc*k2
^3*cosh(k2*b)*F1)/Nn/k3^3/D3/(-sinh(k3*b)^2+cosh(k3*b)^2);
F8 = (-k3*sinh(k3*b)*Dc*k2^2*cosh(k2*b)*Nn*F5-
k3*sinh(k3*b)*Dc*k2^2*sinh(k2*b)*Nn*F2+cosh(k3*b)*Nn*Dc*k2^3*sinh(k2*b)*F5+cosh(k3*b)*Nn*
Dc*k2^3*cosh(k2*b)*F2)/Nn/k3^3/D3/(-sinh(k3*b)^2+cosh(k3*b)^2);
F9 = (-k3*sinh(k3*b)*Dc*k2^2*cosh(k2*b)*Nn*F6-
k3*sinh(k3*b)*Dc*k2^2*sinh(k2*b)*Nn*F3+cosh(k3*b)*Nn*Dc*k2^3*sinh(k2*b)*F6+cosh(k3*b)*Nn*
Dc*k2^3*cosh(k2*b)*F3)/Nn/k3^3/D3/(-sinh(k3*b)^2+cosh(k3*b)^2);
%c1
F10 = (Dc*k2^3*sinh(k2*b)*F4+Dc*k2^3*cosh(k2*b)*F1-
D3*k3^3*cosh(k3*b)*F7)/D3/k3^3/sinh(k3*b);
F11 = (Dc*k2^3*sinh(k2*b)*F5+Dc*k2^3*cosh(k2*b)*F2-
D3*k3^3*cosh(k3*b)*F8)/D3/k3^3/sinh(k3*b);
F12 = (Dc*k2^3*sinh(k2*b)*F6+Dc*k2^3*cosh(k2*b)*F3-
D3*k3^3*cosh(k3*b)*F9)/D3/k3^3/sinh(k3*b);
%a1
F13 = 1/2*(-
2*Ds+2*Dc*k2^2*cosh(k2*an)*Nn*F4+2*Dc*k2^2*sinh(k2*an)*Nn*F1+2*Dc)/Ds/k1^2/cosh(k1*an)/N
n;
F14 =
1/2*(2*mstar*Nn+2*Dc*k2^2*cosh(k2*an)*Nn*F5+2*Dc*k2^2*sinh(k2*an)*Nn*F2)/Ds/k1^2/cosh(k1*a
n)/Nn;
F15 = 1/2*(-2*rho*Nn^2+2*Dc*k2^2*cosh(k2*an)*Nn*F6+2*Dc*k2^2*sinh(k2*an)*Nn*F3-
h*Nn^2)/Ds/k1^2/cosh(k1*an)/Nn;
%c3
F16 = -(k3*sinh(k3)*Nn*F10+k3*cosh(k3)*Nn*F7+1)/Nn;
F17 = -(k3*sinh(k3)*Nn*F11+k3*cosh(k3)*Nn*F8)/Nn;
F18 = -(k3*sinh(k3)*Nn*F12+k3*cosh(k3)*Nn*F9)/Nn;
```

```
%c4
F19 = -1/2*(2*cosh(k3)*Nn*F10+2*sinh(k3)*Nn*F7+2*F16*Nn+1)/Nn;
F20 = -1/2*(2*cosh(k3)*Nn*F11+2*sinh(k3)*Nn*F8+2*F17*Nn)/Nn;
F21 = -1/2*(2*cosh(k3)*Nn*F12+2*sinh(k3)*Nn*F9+2*F18*Nn)/Nn;
%b3
F22 = -k2*sinh(k2*b)*F4-k2*cosh(k2*b)*F1+k3*sinh(k3*b)*F10+k3*cosh(k3*b)*F7+F16;
F23 = -k2*sinh(k2*b)*F5-k2*cosh(k2*b)*F2+k3*sinh(k3*b)*F11+k3*cosh(k3*b)*F8+F17;
F24 = -k2*sinh(k2*b)*F6-k2*cosh(k2*b)*F3+k3*sinh(k3*b)*F12+k3*cosh(k3*b)*F9+F18;
%b4
F25 = -cosh(k2*b)*F4-sinh(k2*b)*F1-b*F22+cosh(k3*b)*F10+sinh(k3*b)*F7+b*F16+F19;
F26 = -cosh(k2*b)*F5-sinh(k2*b)*F2-b*F23+cosh(k3*b)*F11+sinh(k3*b)*F8+b*F17+F20;
F27 = -cosh(k2*b)*F6-sinh(k2*b)*F3-b*F24+cosh(k3*b)*F12+sinh(k3*b)*F9+b*F18+F21;
%a4
F28 = -cosh(k1*an)*F13+cosh(k2*an)*F4+sinh(k2*an)*F1+an*F22+F25;
F29 = -cosh(k1*an)*F14+cosh(k2*an)*F5+sinh(k2*an)*F2+an*F23+F26;
F30 = -cosh(k1*an)*F15+cosh(k2*an)*F6+sinh(k2*an)*F3+an*F24+F27;

%So the constants are:
a_1 = F13*Pc+F14*theta+F15;
a_2 = 0;
a_3 = 0;
a_4 = F28*Pc+F29*theta+F30;
b_1 = F4*Pc+F5*theta+F6;
b_2 = F1*Pc+F2*theta+F3;
b_3 = F22*Pc+F23*theta+F24;
b_4 = F25*Pc+F26*theta+F27;
c_1 = F10*Pc+F11*theta+F12;
c_2 = F7*Pc+F8*theta+F9;
c_3 = F16*Pc+F17*theta+F18;
c_4 = F19*Pc+F20*theta+F21;
```