ABSTRACT

Title of Thesis: A SIMPLIFIED MODEL FOR INTERCONNECT STRESSES INDUCED BY BENDING OF PRINTED WIRING BOARDS

Mark Mifsud, Master of Science, 2015

Thesis Directed By: Professor Abhijit Dasgupta
Department of Mechanical Engineering

A simplified model is developed for analysis of interconnect stresses induced by changes in the curvature of printed wiring boards. The model utilizes the Rayleigh-Ritz variational approach and can be used for rapid assessment and is well-suited for parametric studies because it does not need any numerical meshing. This simplified model represents the component as an equivalent shell and the interconnects as deformable beams. As a simplification, any initial warpage of the component has been neglected in this study. Finite element models are used to verify the simplified model: a simplified FEA model that utilizes the same shell idealization as the proposed Rayleigh-Ritz model and a more detailed 3D solid model. The proposed simplified model provides a faster, more versatile alternative to FEA and can be used to estimate the interconnect stresses caused by PWB warpage under a variety of thermomechanical, vibration, and shock/drop loading conditions.

This thesis focuses on demonstrating the use of this simplified modeling approach for area array surface mount components (e.g. stud-grid array, land-grid array, column grid array, and ball grid array). In particular, the example problem addressed in this thesis is the pre-stress induced in surface mount area-array interconnects during the solder reflow process used for attaching surface mount packages to printed wiring boards (PWBs). The possibility exists for the PWB

and component to warp during the reflow process and therefore exhibit some concave or convex curvature once the process has been completed. If the PWB is then straightened during the assembly process, the act of straightening the PWB can cause pre-stresses to develop in the interconnects between the PWB and the component package. It is important to understand these pre-stresses because unaccounted for interconnect pre-stresses can result in premature wear-out failures or unexpected overstress failures of the assembly.

A SIMPLIFIED MODEL FOR INTERCONNECT STRESSES INDUCED BY BENDING OF PRINTED WIRING BOARDS

By

Mark Paul Mifsud

Thesis submitted to the faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements of the degree of
Master of Science
2015

Thesis Committee:

Dr. Abhijit Dasgupta, Chair and Advisor

Dr. Patrick McCluskey

Dr. Michael Osterman

DEDICATION

To my parents, who have always supported me in everything I have done and who have always been there for me. And to my girlfriend, Nina Cooperman, who has been by my side offering encouragement throughout this entire process.

ACKNOWLEDGEMENT

I would like to thank Mr. Stephen Meschter at BAE for sponsoring the project and providing experimental specimens.

I would like to thank all of the professors and staff of the University of Maryland Mechanical Engineering Department and the Center for Advanced Life Cycle Engineering (CALCE) who supported me during this project.

I would like to thank all of my labmates, Joshua Akman, Ehsan Parsa, Hao Huang, Jaemi Herzberger , Elaine Lin, David Leslie, and Matt Ernst, as well as the other members of our group, Cholmin Choi, Subhasis Mukherjee, Jingshi Meng, Qian Jiang, Ed Habtour, Raman Sridharan, Jon Kordell, Abhishek Deshpande, and Neil Dalal for working with me during our time at the University of Maryland.

I would also like to thank my Thesis Defense Committee, Dr. Patrick McCluskey and Dr. Michael Osterman for taking the time to listen to and review this thesis and offer suggestions.

Most of all, I would like to thank Dr. Abhijit Dasgupta, my advisor, for all of the teaching and support he provided me throughout this entire process. I sincerely appreciate all of the help, encouragement, patience, and guidance he offered.

TABLE OF CONTENTS

Dedication ii

Acknowledgement iii

Table of Contents iv

Table of Tables vii

Table of Figures viii

1. Introduction 1
    - 1.1 Background 1
    - 1.2 Motivation 2
    - 1.3 Objective Statement 3
2. Literature Review 4
    - 2.1 PWB Warpage during Manufacturing 4
    - 2.2 Gaps in the Literature 6
3. Methodology 7
    - 3.1 Volume-Weighted Averaging 8
    - 3.2 Anticlastic Curvature 8
    - 3.3 Strain Energy of Rectangular Plates 9
    - 3.4 Rayleigh-Ritz Method 12
    - 3.5 Unidirectional and Bidirectional Curvature 13
4. Three-Dimensional Area Array Simplified Model 14
    - 4.1 Simplified Model Overview 15
    - 4.2 Initial Displacement Fields and Interconnect Heights 17
    - 4.3 Interconnect Stiffness Matrix 18
    - 4.4 Potential Energy Equation for the Assembly 20
        - 4.4.1 Strain Energy of the Package 20
        - 4.4.2 Strain Energy of the Interconnects 21
        - 4.4.3 Potential Energy of the Entire System 22

4.5 Final Displacement Field for the Package 22

4.6 Final Interconnect Heights, Forces, and Pre-Stresses 23

5. Three-Dimensional Area Array FEA Models 24

5.1 Three-Dimensional FEA Models Overview...25

5.2 Area Array FEA Model: Shell and Beam Model 25

5.3 Area Array FEA Model: 3D Solid Brick Element Model..26

5.3.1 Modeling of the PWB 27

5.3.2 Modeling of the Component Package and Leads 27

5.3.3 Modeling of the Solder Joints 28

6. Results from the Three-Dimensional Area Array Models 28

6.1 Simplified Model Results 28

6.2 Results from Shell and Beam FEA Area Array Model 31

6.3 Detailed Solid 3D Brick FEA Model Results 33

6.4 Comparison of Simplified and Shell and Beam FEA Model Results 35

6.5 Comparing Results of the Solid 3D Brick FEA Model
to the Shell and Beam FEA Model 36

6.6 Comparing Results of the Simplified and Solid 3D Brick FEA Models 38

7. Alternate Applications for Simplified Method 40

7.1 Alternate Applications for Simplified Method: Bending an Initially Straight PWB...40

8. Summary, Conclusions, Future Work, and Contributions 43

8.1 Summary of Work Completed 43

8.2 Conclusions 45

8.3 Future Work 46

8.4 Contributions 49

Appendices 50

Appendix 1: Perimeter-Leaded Component Models 50

Appendix 2: Kotlowitz Equations and Definitions of Variables and Constants................69

Appendix 3: Full Code for the Area Array Simplified Model 71
Appendix 4: Full Code for the Perimeter-Leaded Simplified Model 88
References 107

TABLE OF TABLES

Table 1: Interconnect Pre-Stresses for Unidirectional Radius of Curvature of 1208.7 mm 53

Table 2: Interconnect Pre-Stresses for a Bidirectional Radius of Curvature of 1208.7 mm 54

Table 3: Parametric Study Radii of Curvature and Corresponding Displacement Field Constants 54

Table 4: Maximum and Minimum Pre-Stresses for Various Unidirectional Radii of Curvature 56

Table 5: Maximum and Minimum Pre-Stresses for Various Bidirectional Radii of Curvature 57

Table 6: Pre-stresses for a 1208.7 mm Unidirectional Concave Up Radius of Curvature 58

Table 7: Pre-stresses for a 1208.7 mm Bidirectional Concave Up Radius of Curvature 59

Table 8: Parametric Study Results for 3D FEA Model 61

TABLE OF FIGURES

Figure 1: An Anticlastic Surface (Horrocks & Johnson, 1967) 9

Figure 2: Orientation of Plate for Strain Energy Calculation 10

Figure 3: Visualization of Unidirectional Curvature 13

Figure 4: Visualization of Bidirectional Curvature 14

Figure 5: 3D Shell and Beam FEA Model 26

Figure 6: 3D Area Array FEA Model 27

Figure 7: Interconnect Pre-Stresses for Unidirectional Curvature Area Array Simplified Model 29

Figure 8: Interconnect Pre-Stresses for Bidirectional Curvature Area Array Simplified Model..30

Figure 9: Interconnect Pre-Stresses for Unidirectional Curvature Shell and Beam FEA Model 31

Figure 10: Interconnect Pre-Stresses for Bidirectional Curvature Shell and Beam FEA Model..32

Figure 11: Interconnect Pre-Stresses for Unidirectional Curvature Detailed FEA Model 33

Figure 12: Interconnect Pre-Stresses for Bidirectional Curvature Detailed FEA Model 34

Figure 13: Normalized Pre-Stresses for Unidirectional Curvature..35

Figure 14: Normalized Pre-Stresses for Bidirectional Curvature...36

Figure 15: Normalized Pre-Stresses for Unidirectional Curvature 37

Figure 16: Normalized Pre-Stresses for Bidirectional Curvature 37

Figure 17: Normalized Pre-Stresses for Unidirectional Curvature 38

Figure 18: Normalized Pre-Stresses for Bidirectional Curvature 39

Figure 19: Interconnect Stresses for Unidirectional Curvature...41

Figure 20: Interconnect Stresses for Bidirectional Curvature..42

Figure 21: Three-dimensional FEA Model 51

Figure 22: Location of Curved Side and Straight Side of
Component for Unidirectional Curvature 60
Figure 23: Plot of Pre-Stress vs. Radius of Curvature of PWB for Unidirectional Curvature......62
Figure 24: Plot of Pre-Stress vs. Radius of Curvature of PWB for Bidirectional Curvature 62

Figure 25: Plot of Simplified Model Interconnect
Pre-Stresses for Unidirectional Curvature 64
Figure 26: Plot of Simplified Model Interconnect
Pre-Stresses for Bidirectional Curvature 64
Figure 27: Plot of FEA Model Interconnect Pre-Stresses for Unidirectional Curvature 65
Figure 28: Plot of FEA Model Interconnect Pre-Stresses for Bidirectional Curvature 65
Figure 29: Plot of Interconnect Pre-Stress vs. Unidirectional
PWB Curvature for Simplified Model 66
Figure 30: Plot of Interconnect Pre-Stress vs. Bidirectional
PWB Curvature for Simplified Model 67
Figure 31: Plot of Interconnect Pre-Stress vs. Unidirectional PWB Curvature for FEA Model ..67
Figure 32: Plot of Interconnect Pre-Stress vs. Bidirectional PWB Curvature for FEA Model 68
Figure 33: Gull-Wing Lead Geometry Variables (Kotlowitz and Taylor, 1991) 69

1. Introduction

Surface-mount devices in electronic assemblies present a mechanical design challenge because the interconnects perform not only electrical functions, but also provide thermal and mechanical interconnections to the printed wiring board (PWB). As a result, any mismatch in deformation between the component and the PWB generates mechanical stresses in the interconnect. In this thesis the focus is on stresses caused by changes in the PWB curvature. Such curvatures can be generated either by mechanical loading such as vibration and shock/drop, or by thermo-mechanical warpage caused during solder reflow to attach the components to the PWB using solder. The thermo-mechanical warpage situation is examined further in this thesis.

The soldering process requires the assembly to be exposed to very high temperatures which can result in differential material expansion and cause warpage of the PWB due to asymmetry about the PWB mid-surface. Correcting this curvature by clamping into fixtures during down-stream assembly steps can result in the development of pre-stresses in the interconnects of the surface mount component. These pre-stresses can then cause premature failure of the assembly and reduce its useful life.

1.1 Background

Surface-mount devices (SMDs) are components which can be connected directly to the surface of a board and do not require leads to be placed in through holes. SMDs can be either leaded or leadless. These devices have been used as a primary component on PWBs since the 1960’s. Surface-mount technology (SMT) has largely replaced through-hole technology (also called insertion mount technology – IMT) because SMT components are generally smaller, faster, cheaper, and allow for higher circuit densities on the PWB (SMTA, 2014). In addition to these benefits, SMT devices are also cheaper and easier to install because the process can be mechanized (Radio-Electronics, 2014). These components are typically attached to the PWB using only solder. This is done by applying a solder paste onto the matching copper pads of the PWB and placing the matching leads or pads of the SMT components in the correct location. The assembly is then passed through a reflow oven where the solder is heated to the point where it melts and then reflows around the component leads. Once the solder cools, it solidifies and holds the SMT component in place.

Many different SMT component architectures exist, and a few of the most common are leaded peripheral quad flat packages (QFPs), or leaded pin grid arrays (PGAs), and leadless ball grid arrays (BGAs). QFPs are square packages with a line of leads along each side face. PGAs have pin leads arranged in an area-array grid pattern on the bottom face of the package. BGAs are very similar to PGAs except that the pin leads are replaced by solder balls.

1.2 Motivation

As mentioned in the previous section, PWBs experience curvature due to a variety of loading conditions, both mechanical and thermo-mechanical. These flexural deformations generate stresses in the interconnects of the components mounted on the PWB. It is important to have simplified methods for assessing such stresses, to ensure reliable PWA designs.

An important example of thermo-mechanical loading occurs when the solder reflow process is used to attach the surface-mount component to the PWB. This entails heating the PWB and the component to very high temperatures, where the solder will melt and reflow in order to securely hold the component. Due to the fact that the PWB, solder, leads, and component package are all made of different materials, they will all have different coefficients of thermal expansion (CTE).

This CTE mismatch can cause differential expansion of the various parts of the assembly during the post-reflow cool-down process which can cause the assembly to warp or bend because of the asymmetry of the assembly about the PWB mid-surface. Therefore, it is possible for the PWB to be warped rather than straight at the end of the solder reflow process.

In many practical applications, a warped PWB is a hindrance to further down-stream assembly steps. Therefore, before the PWB can be integrated into a product it must be straightened by clamping it in a straightening fixture. The straightening force usually causes inelastic deformation of the solder interconnects. Furthermore, some of the stress used to straighten the PWB is transferred into the package and the package interconnects. This can cause the package to warp and can cause pre-stresses to develop in the interconnects. These pre-stresses can then cause unexpected failures if they are not accounted for in the original design.

Vibrational, thermomechanical, or shock/drop loading can also cause bending of a PWB. This bending leads to stresses developing in the interconnects between the bent PWB and the component package and these stresses can also cause failure of the assembly.

Therefore, it is very important to consider the stresses resulting from PWB bending, whether the bending is from straightening a warped board or loading induced, when assessing the reliability of a product. In order to do this, there must be a simplified way to estimate the magnitude of the stresses which develop during the bending process and to assess how these stresses impact the durability of the product.

1.3 Objective Statement

The primary objective of this study is to develop a simplified modeling method which requires no numerical meshing and can be used to provide sufficiently accurate estimates of the stresses which develop in package interconnects of SMT components during the bending of a PWB. In particular, the initial focus is on area array components like stud grid arrays, land grid arrays, column grid arrays, and ball grid arrays, solder to an initially warped board. Ideally, any method used to determine the stresses should be fast, accurate, versatile, and repeatable so it can be quickly performed to analyze any assembly with any initial or final PWB curvature. Therefore, the goal is to develop a method which has all of these qualities. For these reasons, simply using FEA to estimate the pre-stresses will not be sufficient. Using FEA would require an entirely new model to be constructed for each different situation which could potentially take a long time to create, run, and post-process. Therefore, a simplified model will be developed and coded in MATLAB, as an alternative to detailed FEA. A simplified model which does not require numerical meshing is not only much faster than a FEA model, but the model parameters can be easily varied to represent a multitude of different component geometries.

2. Literature Review

Current literature shows that warpage of the PWB throughout the manufacturing process, especially during solder reflow, is a fairly common occurrence which is known to have a negative effect on assembly reliability. Primarily, the literature focuses on determining what factors contribute to the PWB warpage and how different assemblies can be affected.

2.1 PWB Warpage during Manufacturing

Many different investigators have researched PWB warpage in a wide range of situations. In 2004, a study was conducted which focused on modeling PWB warpage during the solder reflow process (Halvi et. al., 2004). PWB warpage during reflow was examined using an ANSYS model for boards containing 4, 8, 12, and 24 layers. The effects of different reflow times and peak temperatures on the warpage of the PWB were also examined, because CTE mismatch is one of the main causes of PWB warpage and the effects of CTE mismatch are amplified by higher temperatures and longer exposure to these temperatures. Ume and co-authors studied PWB warpage throughout the manufacturing process (Ume, et. al., 1997). While the focus of their work was on warpage during the solder masking process, they explain that PWB warpage as a result of CTE mismatch can occur throughout the manufacturing process. Not only is the CTE mismatch between the PWB and the component an issue, but mismatch between the copper and dielectric layers of the PWB itself can also contribute to the warpage of the PWB unless the PWB stack-up is symmetric about the PWB mid-surface. The paper goes on to say that if a PWB warps during the manufacturing process it can lead to failure of the solder joints connecting the component to the board (Ume, et. al., 1997). Another study experimentally examined PWB warpage in flip-chip assemblies (Yang, et. al., 2005). In this study, flip-chips were soldered onto a substrate using a reflow module and resulting warpages were measured. Warpage of both the PWB and the flip-chip component were found after the reflow process. It was explained that the cause of this warpage was significant CTE mismatch between the board and the chip. Mechanically coupling the chip to the board using an underfill can help alleviate this issue, but cannot remove it entirely. The paper also says the effects of the solder reflow process are more severe for lead-free solders as compared to leaded solders (Yang, et. al., 2005). In 1996, Stiteler and co-authors began looking at how to measure PWB warpage because they recognized the importance of the issue. They knew that PWB warpage could occur during the soldering process and that, “residual stress resulting from transient warpage induced in the PWB/PWBA during soldering can still affect reliability” (Stiteler, et. al., 1997). In their study, they measured the warpage of a PWB during a simulated wave soldering process. They found significant warpage across the PWB throughout the process (Stiteler, et. al., 1997). Another study examined PWB warpage resulting from both wave soldering and infrared soldering. In this study, the warpages of two different types of PWBs were measured under wave soldering and infrared soldering (Polsky, et. al., 2000). In all cases, the soldering process caused warpage of the PWB. The study explains that CTE mismatch between the component and the PWB leads to board warpage which results in residual stresses which cause premature solder joint failure (Polsky, et. al., 2000). Mittal and co-authors also examined PWB warpage during infrared soldering. They used an ANSYS model to monitor the PWB warpage throughout the infrared soldering process. Not only did they find that the soldering process caused warpage of the PWB, but also that large stresses were generated in the assembly as a result of the CTE mismatch between the board and component (Mittal, et. al, 1996). All of this research shows that PWB warpage is a fairly common issue and that it can have a detrimental effect on the reliability and useful life of an electronic assembly.

2.2 Gaps in the Literature

As illustrated in the previous section, the literature focuses in large part on determining the causes of PWB warpage during manufacturing. Conversely, there is very little work concerning modeling a warped PWA and examining how straightening the board affects the solder joints. What research does exist on this topic involves using a FEA model to study a specific PWA, component, and curvature combination and reveals that PWB warpage can lead to solder joint failure. Despite this, a general model which can be applied across a wide range of PWAs, components, and curvatures does not exist. Without such a model, a new FEA model would need to be created and run for every new configuration. This is a less than ideal situation. A general model which is fast, accurate, versatile, and repeatable is a much better alternative. The fact that such a model does not exist, is a major gap in the literature. In order to properly understand the effect of straightening a warped PWB can have on solder joint life, such a model is needed.

3. Methodology

In order to establish an effective process for determining the pre-stresses which develop in the interconnects of a SMT component when an initially warped PWB is straightened, finite element and simplified methods must be used together. An accurate simplified model can be used to quickly determine the interconnect pre-stresses for a wide range of components, board curvatures, and materials. In order to develop such a simplified model, FEA must be used in order to guide and verify the model.

Throughout the process of developing the simplified model, FEA must be used in order to ensure the final model gives an accurate approximation of the interconnect pre-stresses. The simplified model is developed in particular for area array components. An area array component was selected as the initial component type because it is a commonly used component type. Details on the simplified and FEA models can be found in the following sections. Five important concepts, pertaining to model development, are addressed in this section. The first involves methods for spatial averaging to estimate average stresses in interconnects when the stress field is highly non-uniform. The second concept explains the importance of anticlastic curvature in flexural deformation of PWAs. The third important concept involves the methodology for estimating the flexural strain energy of PWAs and the fourth important concept is the Rayleigh Ritz variational method for numerical estimates of stresses caused by flexural deformation of the PWB. The final concept discussed below is the difference between unidirectional and bi-directional curvature due to warpage of the PWB.

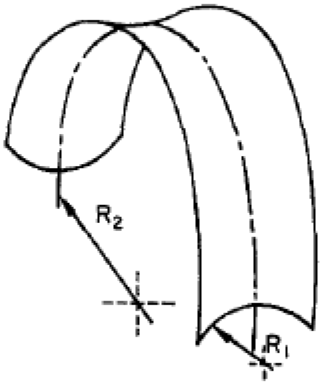
3.1 Volume-Weighted Averaging

Spatial averaging is often important for finding locally averaged values of nonuniform field variables. In particular, the volume-weighted averaging scheme is useful for post-processing results from all of the FEA models included in this study. This method is more complicated than an element averaging technique, but it is also more accurate. The volume-weighted averaging technique has four steps. First, the total volume of the elements within the averaging region is calculated by summing the volume of each of the elements in the region of interest. Once the total volume is known, the volume ratio for each element in the region can be determined. This is done by dividing the volume of each element by the total volume of all the elements in the region. Next, the volume-weighted average of the parameter in question can be determined for each element by multiplying the value of the parameter in each element by that element's volume ratio and summing the results over all the elements in the region of interest.

3.2 Anticlastic Curvature

Anticlastic curvature is the curvature which can occur orthogonal to the primary flexure direction in beams and plates. The sign of the anticlastic curvature is opposite to that of the primary curvature. Horrocks and Johnson define a surface exhibiting anticlastic curvature as a surface "which has two curvatures transverse to each other in opposite directions" (Horrocks & Johnson, 1967). They provide the image seen in Figure 1 to illustrate anticlastic curvature.

Figure 1: An Anticlastic Surface (Horrocks & Johnson, 1967)



It is explained that when a beam is bent to a certain radius, it develops longitudinal bending strains of opposite signs on the upper and lower surfaces. As a result, transverse strains of opposite signs develop because of Poisson's ratio. If the ratio of the beam width squared over the radius times the beam thickness is less than one, then these transverse strains will cause anticlastic curvature to develop. For larger ratios, the deformation seems to be restrained and more concentrated at the edges (Horrocks & Johnson, 1967). In 1970, Pomeroy also did some work involving anticlastic curvature of beams. Pomeroy showed that anticlastic curvature can be expected to occur in most beams, except in extremely wide beams (Pomeroy, 1970). Therefore, it is important to incorporate anticlastic curvature into the simplified models used to predict the interconnect pre-stresses.

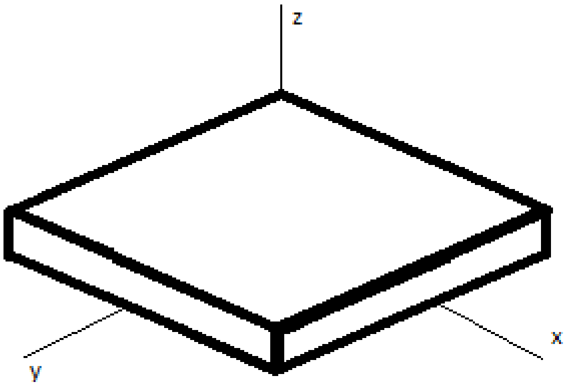
3.3 Strain Energy of Rectangular Plates

Calculating the strain energy of rectangular plates under bending is important for use of energy methods to estimate flexural stresses in PWAs. In 1850 G. R. Kirchhoff proposed equations which gave the energy of a plate of constant thickness in terms of the curvatures of the midplane of the plate. Kirchhoff also proposed that under small deflections and transverse loading, the midplane of the plate does not stretch, it only bends. Many other people have used Kirchhoff's works as a starting point for other studies into plate theory and the strain energy of plates. In the early part of the twentieth century, von Kármán developed a plate theory which allows for large deformations. The fundamental consequence of the Kirchhoff kinematic approximation is that: (i) there is a neutral plane in the center of the plate that experiences no in-plane strains; (ii) the strains above and below the neutral plane are of opposite signs and are linearly proportional to the distance from the neutral plane; (iii) there are no extensional or shear strains in the thickness direction. The resulting expression for the plate strain energy function is reproduced below as Equation 1.

$$U = \frac{1}{2}\int_V \left(\sigma_x \epsilon_x + \sigma_y \epsilon_y + \tau_{xy}\gamma_{xy}\right) dxdydz \qquad (Equation\ 1)$$

In this equation, it is assumed that the plate is positioned in the x-y plane and the z-direction is the out-of-plane direction, as shown in Figure 2.

Figure 2: Orientation of Plate for Strain Energy Calculation



For small deflections, this equation can be simplified. From classical small strain, small deflection theory, the strains in a Kirchhoff plate can be approximated, as shown in Equations 2-4.

$$\epsilon_x = -z\frac{d^2w}{dx^2} \qquad (Equation\ 2)$$

$$\epsilon_y = -z\frac{d^2w}{dy^2} \qquad (Equation\ 3)$$

$$\gamma_{xy} = -2z\frac{d^2w}{dxdy} \qquad (Equation\ 4)$$

Here, w(x,y) represents the vertical displacement field for the plate neutral plane and z is the distance from the neutral plane. Similarly, the stresses can also be approximated as shown in Equations 5-7.

$$\sigma_x = \frac{E}{1-v^2}\left(\epsilon_x + v\epsilon_y\right) = \frac{-zE}{1-v^2}\left(\frac{d^2w}{dx^2} + v\frac{d^2w}{dy^2}\right) \qquad (Equation\ 5)$$

$$\sigma_y = \frac{E}{1-v^2}\left(\epsilon_y + v\epsilon_x\right) = \frac{-zE}{1-v^2}\left(\frac{d^2w}{dy^2} + v\frac{d^2w}{dx^2}\right) \qquad (Equation\ 6)$$

$$\tau_{xy} = \frac{E}{2(1+v)}\left(\gamma_{xy}\right) = \frac{-zE}{(1+v)}\frac{d^2w}{dxdy} \qquad (Equation\ 7)$$

Plugging these stress and strain relations into Equation 1 gives a simplified version of the plate strain energy equation. This equation can be further simplified by integrating along the z-direction over the thickness of the plate, from -h/2 to h/2. The resulting strain energy formula is shown below in Equation 8.

$$U = \iint_R \left\{ \frac{D}{2} \left(\frac{d^2w}{dx^2} + \frac{d^2w}{dy^2} \right)^2 - D(1-v) \left[\left(\frac{d^2w}{dx^2} \frac{d^2w}{dy^2} \right) - \left(\frac{d^2w}{dxdy} \right)^2 \right] \right\} dxdy \qquad (Equation\ 8)$$

In this equation, D represents the flexural rigidity of the plate, given by the formula shown in Equation 9.

$$D = \frac{Eh^3}{12(1-v^2)} \qquad (Equation\ 9)$$

This strain energy equation can be used to solve for the strain energy of a wide array of plates (Haslach & Armstrong, 2004), and is used for an energy-based variational solution scheme described in the next section.

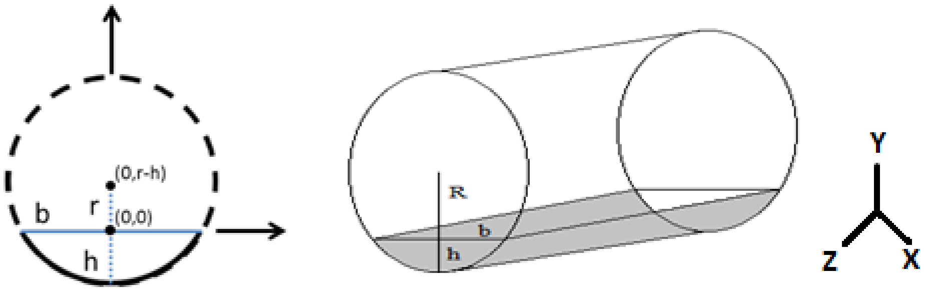
3.4 Rayleigh-Ritz Method

The Rayleigh-Ritz method is a commonly used energy-based variational method for analyzing the elastic solution for plates. This method has been studied and utilized extensively. The Rayleigh-Ritz method has been used for general plate analysis (Liew & Wang, 1993), vibrational analysis of Mindlin plates (Dawe & Roufaeil, 1980), studying flexural vibration of rectangular plates (Kim et. al., 1990), and studying natural frequencies of rectangular plates (Baht, 1985). The application of the Rayleigh-Ritz method is explained by Yuan and Dickinson. The total strain energy of a system can be found by summing the strain energies across the whole system. This is done by first assuming an admissible parametric displacement field in the plate, parameterized by a set of unknown coefficients and assumed functional forms. The resulting strain energy is then minimized by considering the variation of the strain energy equation in terms of these unknown coefficients in the equation. This yields a system of equations for solving for the unknown coefficients in the assumed displacement field (Yuan & Dickinson, 1992).

3.5 Unidirectional and Bidirectional Curvature

When looking at an initially warped board, two types of PWB curvature are possible. The first type is unidirectional curvature. Unidirectional curvature refers to curvature along only one axis of the PWB. This means the top face of the PWB resembles the curved surface of a cylinder. Unidirectional curvature is illustrated in Figure 3.

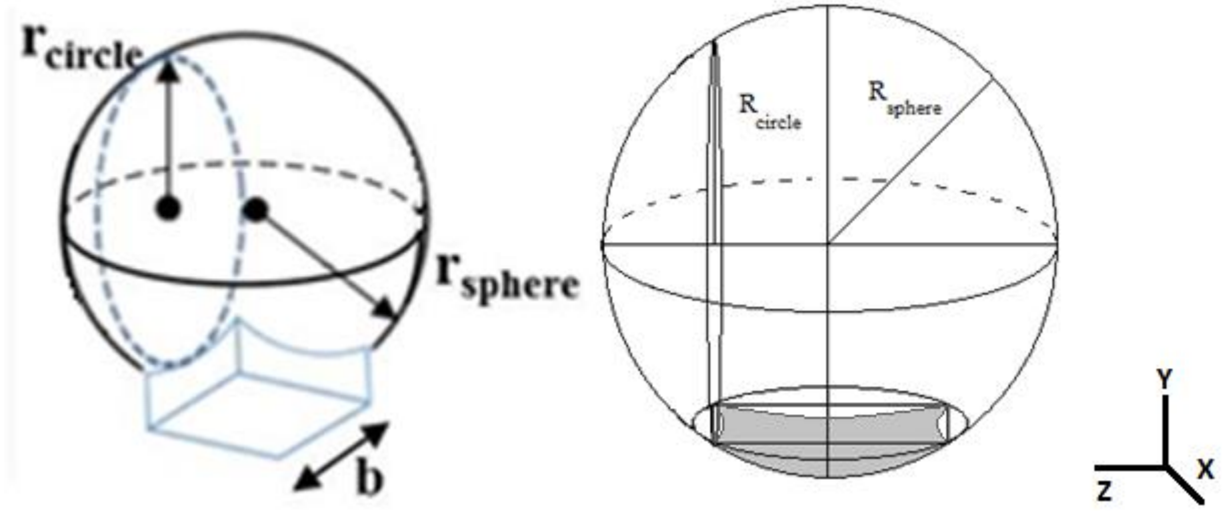
Figure 3: Visualization of Unidirectional Curvature



The grey area shown on the cylinder in the figure represents the top surface of the PWB. It can be seen that this surface is only curved along the x-direction (about the z-direction).

In addition to unidirectional curvature, bidirectional curvature is also possible. Bidirectional curvature refers to curvature about two axes of the PWB. In this case, the top face of the PWB resembles the surface of a sphere rather than the curved surface of a cylinder. This curvature is shown in Figure 4.

Figure 4: Visualization of Bidirectional Curvature



It can be clearly seen that the top face of the PWB, represented by the grey surface in the figure, is curved along both the z-axis and the x-axis.

4. Three-Dimensional Area Array Simplified Model

As stated previously, in order to quickly and accurately predict the pre-stresses which develop in the component interconnects when an initially warped board is straightened, a reliable simplified model is needed. Such a model would need to be able to capture the structural details and mechanics present in a realistic printed wiring assembly (PWA). In order to do this, the three-dimensional simplified model uses a beam approximation for the interconnects and a shell approximation for the component. The analysis method is based on the Raleigh-Ritz Method and utilizes the potential energies and strain energies of the assembly. This simplified model is first developed for area array components. Area array components can be leaded, like a stud grid array (SGA) or land grid array (LGA) component, or leadless, like a ball grid array (BGA) or column grid array (CGA) component. Analysis of a leaded component is slightly more complicated due to the fact that the interconnects are made up of the lead and solder, rather than just solder. This simplified model is capable of modeling either situation by including or excluding the input lead properties.

4.1 Simplified Model Overview

This three-dimensional simplified model treats each interconnect as an assembly of two beams. One beam represents the solder joint, and the other beam represents the component lead. A single beam of equivalent stiffness is then substituted to represent the entire interconnect instead of the series pair. This simplification significantly reduces the complexity of the interconnects. For this model, the PWB is not explicitly included, but rather its effect is simply modeled by imposing the PWB out-of-plane displacement field as boundary conditions at the base of the interconnects. This approximation simplifies the actual physical assembly by removing the PWB and only considering the effects that straightening the board has on the rest of the assembly. The board displacement field is estimated by considering the displacements needed to go from a known warped configuration before the straightening process, to the final flat configuration after the straightening process. This model further simplifies the real-world assembly by treating the package as a deformable shell rather than a fully 3D deformable body. In accordance with the Rayleigh-Ritz method, an approximate parameterized quadratic displacement field is assumed for the package, where it meets with the tops of the interconnects. The initial interconnect heights are calculated as the difference between the initial values for the flat package and known displacement field of the warped PWB at the location of each interconnect. With these initial values known, the strain energy of the package and of all the interconnects is determined to estimate the potential energy equation for the system. By minimizing the potential energy equation the final curvature of the package is determined and the final interconnect heights, forces, and pre-stresses are calculated.

4.2 Initial Displacement Fields and Interconnect Heights

The first step in the process of determining the interconnect pre-stresses is an estimation of the initial height of each solder joint. Before this can be done however, the initial displacement field for the PWB must be determined. The displacement fields give the vertical coordinate, y, as a function of the two horizontal coordinates, z and x. For this model, it is assumed that both the package and board deformation fields are symmetric about the x and z centerlines and can be approximated by a quadratic function and the higher order effects can be ignored. Equation 10 below shows the general form of the displacement field for the PWB, and Equation 11 shows the general form of the displacement field for the package.

$$y_{PWB} = C * z^2 + D * x^2 \qquad (Equation\ 10)$$

$$y_{package} = A_0 + A * z^2 + B * x^2 \qquad (Equation\ 11)$$

In these equations, the constants A, B, C, and D can be adjusted to represent different curvatures, and A_0 is a constant representing the vertical distance between the center of the PWB and the package. A_0 is always a positive value because it is assumed that the center of the PWB is located at a zero reference height and the package is somewhere above this in the positive y direction.

For this model, the focus is on a situation where initially the PWB is curved and the package is initially flat. Therefore, the constants for the package equation are fairly easy to determine. Both A and B are initially zero because there is no initial curvature and A_0 is simply the initial distance between the package and board. For the PWB, determining the values of the constants is slightly more complicated. For unidirectional curvature, the value for C is nonzero and the value for D is zero because there is no curvature in the second direction. For bidirectional curvature, the values for C and D are equal because it is assumed that the curvature is the same in both directions. Because the PWB curvatures in this study are all based on the difference between the heights of the tallest solder joint and shortest solder joint, the values for the nonzero constants can be solved for directly. The horizontal locations of the tallest and shortest solder joints are known fixed quantities, and the solder joint height difference is also known for a given curvature. Therefore by solving the system of equations shown in Equations 12 through 14, the value of the constant can be determined.

$$y_{tallest} = C * z_{tallest}^2 \qquad (Equation\ 12)$$

$$y_{shorstest} = C * z_{shortest}^2 \qquad (Equation\ 13)$$

$$y_{tallest} - y_{shorstest} = \Delta h_{solder\ joint} \qquad (Equation\ 14)$$

This system gives three equations with three unknowns, $y_{tallest}$, $y_{shortest}$, and C, so the final solution for C is given in Equation 15.

$$C = \frac{\Delta h_{solder\ joint}}{z_{tallest}^2 - z_{shortest}^2} \qquad (Equation\ 15)$$

This same equation can be used to find the value for D in the bidirectional curvature case.

With the initial displacement fields known, the initial solder joint heights can be calculated. This is done using the initial displacement field of the board and the initial offset height (A_0) of the package along with the lead height. The lead height is a known value dependent on the component being used. The total height of each interconnect, the lead and solder joint combination, can be calculated by subtracting the PWB displacement field from the initial package offset value (A_0) at the horizontal coordinates of each interconnect. The solder joint height can then be calculated by subtracting the nominal undeformed lead height from the total interconnect height.

4.3 Interconnect Stiffness Matrix

As mentioned previously, this model treats the interconnects as a combination of two beams. One beam represents the solder joint and the other represents the lead. In order to determine the stiffness of the entire interconnect, first the stiffness of the solder joint and lead must be calculated. The stiffness of each beam is represented using the transverse flexural stiffness matrix for a 3D Euler-Bernoulli beam and the extensional stiffness of an axial bar. Equation 16 below shows the beam-bar stiffness matrix used in this simplified model. The stiffness matrix is dependent on the area, modulus, length, and moment of inertia of the beam. The AE/L terms reflect the axial stiffness of the bar and the EI/L terms reflect the bending and transverse stiffness of the beam.

Equation 16: Interconnect Stiffness Matrix, k_i

$$\begin{bmatrix}
AE/L & 0 & 0 & 0 & 0 & -AE/L & 0 & 0 & 0 & 0 \\
0 & 12EI_x/L^3 & 0 & 0 & 6EI_x/L^2 & 0 & -12EI_x/L^3 & 0 & 0 & 6EI_x/L^2 \\
0 & 0 & 12EI_z/L^3 & -6EI_z/L^2 & 0 & 0 & 0 & -12EI_z/L^3 & 6EI_z/L^2 & 0 \\
0 & 0 & -6EI_z/L^2 & 4EI_z/L & 0 & 0 & 0 & 6EI_z/L^2 & 2EI_z/L & 0 \\
0 & 6EI_x/L^2 & 0 & 0 & 4EI_x/L & 0 & -6EI_x/L^2 & 0 & 0 & 2EI_x/L \\
-AE/L & 0 & 0 & 0 & 0 & AE/L & 0 & 0 & 0 & 0 \\
0 & -12EI_x/L^3 & 0 & 0 & -6EI_x/L^2 & 0 & 12EI_x/L^3 & 0 & 0 & -6EI_x/L^2 \\
0 & 0 & -12EI_z/L^3 & 6EI_z/L^2 & 0 & 0 & 0 & 12EI_z/L^3 & 6EI_z/L^2 & 0 \\
0 & 0 & -6EI_z/L^2 & 2EI_z/L & 0 & 0 & 0 & 6EI_z/L^2 & 4EI_z/L & 0 \\
0 & 6EI_x/L^2 & 0 & 0 & 2EI_x/L & 0 & -6EI_x/L^2 & 0 & 0 & 4EI_x/L
\end{bmatrix}$$

This matrix can be calculated for the lead and for the solder joint. The next step is to determine the equivalent matrix for the whole interconnect. The interconnect consists of two elements (the lead and the solder) and three nodes (top of the lead, lead/solder interface, bottom of the solder). There are 5 degrees of freedom at each node, three axial and two rotational. The lead element encompasses two nodes with five degrees of each which is why the matrix is a 10x10 matrix. The same is true for the solder stiffness matrix. The interconnect matrix encompasses 3 nodes, so it will be a 15 x 15 matrix. The local element matrices, which are the individual lead and solder stiffness matrices, can be used to assemble the global matrix for the interconnect. This is done using the finite element assembly process. In this process, the lead and solder matrices are arranged so the lower right quadrant of the lead stiffness matrix overlaps with the upper left quadrant of the solder stiffness matrix. Where the terms of the two matrices overlap, they are added together and zeros are used to fill in the rest of the 15 x 15 interconnect stiffness matrix.

4.4 Potential Energy of the Assembly

In order to solve for the final curvature of the package, the Rayleigh-Ritz method needs to be applied. In order to do this, the potential energy must be minimized. Since the displacement field of the PWB is imposed as essential boundary conditions, the potential energy of the system does not include any external work terms and is given by the sum of the strain energy of the package plus the strain energies of all of the interconnects.

4.4.1 Strain Energy of the Package

The strain energy of the package can be determined using the rectangular plate strain energy equation explained and presented in Section 3.3 and presented below in Equation 17. The equation has been adjusted to account for the fact that the package lies in the x-z plane and the y-direction is the vertical, out-of-plane direction.

$$U = \iint_R \left\{ \frac{D}{2}\left(\frac{d^2v}{dz^2} + \frac{d^2v}{dx^2}\right)^2 - D(1-\upsilon)\left[\left(\frac{d^2v}{dz^2}\frac{d^2v}{dx^2}\right) - \left(\frac{d^2v}{dzdx}\right)^2\right]\right\} dzdx \qquad (Equation\ 17)$$

Here, v(x,z) is the displacement field for the package in the y direction, υ is the package's effective Poisson's Ratio, and D represents the flexural rigidity of the effective plate. The equation for the flexural rigidity of a plate is given in Section 3.3. The package displacement field after the PWB has been straightened is therefore a function of A_0, A, and B. As explained in Section 3.3, the equation above is valid for rectangular plates of a uniform thickness which experience small deflections resulting from transverse loading, and is valid for the component package. The three-dimensional FEA model, which will be discussed in detail later, confirms that the vertical deflection of the package is very small relative to the package thickness, so the small deflection assumption is valid for this model.

4.4.2 Strain Energy of the Interconnects

The strain energy of the interconnects is determined using the strain energy equation presented in Equation18.

$$U = \frac{1}{2} u_i^T k_i u_i \qquad (Equation\ 18)$$

Here, the strain energy of each interconnect is given by the transpose of the interconnect displacement vector times the interconnect stiffness matrix times the interconnect displacement vector. The interconnect displacement vector contains both the vertical and horizontal displacements (v,u,w) of the top and bottom of the interconnect as well as the rotations about the two horizontal axes (θ_z, θ_x) at the top and bottom of the interconnect. Equation 19 provides a definition for the displacement vector, u_i.

$$u_i^T = \left[v_p\ u_p\ w_p\ \theta_{pz}\ \theta_{px}\ v_i\ u_i\ w_i\ \theta_{iz}\ \theta_{ix}\ v_b\ u_b\ w_b\ \theta_{bz}\ \theta_{bx} \right] \qquad (Equation\ 19)$$

In this definition, subscript p refers to the top of the interconnects (where they meet the package), subscript i refers to the interface between the lead and the solder, and subscript b refers to the bottom of the interconnects (where they meet the board). The displacements and rotations at both ends of the interconnects can be determined based on the final displacement fields for the package and board. The final PWB displacement field is known; however, the final package displacement field is unknown. Therefore, the board displacements are known values while the package displacements are functions of the coefficients from the final package displacement field, A_0, A, and B. The displacements at the lead/solder interface are unknown, and must be determined by applying the other displacements as essential boundary conditions and solving the force equation given in Equation 20.

$$Q = k_i u_i \qquad (Equation\ 20)$$

In this equation, Q represents the external applied forces which are zero in this case. By applying the known package and board displacements as essential boundary conditions in this equation, the unknown interface displacements can be determined. With all of the displacements known, the interconnect strain energy equation can be solved. Because some of the displacements are functions of A_0, A, and B, the interconnect strain energy will also be a function of A_0, A, and B.

4.4.3 Potential Energy of the Entire System

The potential energy of the entire system is given by the sum of the strain energies for each element of the system. This means the potential energy of the assembly is equal to the sum of the strain energy of the package plus the strain energy of all the interconnects. Both the package and interconnect strain energies are functions of the coefficients of the final package displacement field, A_0, A, and B. As discussed in the next section, the potential energy equation can be minimized using the Rayleigh-Ritz method in order to solve for these coefficients which define the final package displacement field.

4.5 Final Displacement Field for the Package

In order to determine the final displacement field for the package, the Rayleigh-Ritz method must be used to minimize the potential energy equation and solve for the displacement field coefficients. As explained in Section 3.4, the Rayleigh-Ritz method uses energy minimization to determine the unknown coefficients that were used to parametrize the displacement field.

Differentiating the potential energy equation in terms of A_0, A, and B and setting the resulting expressions equal to zero yields the system of equations shown in Equation 21.

$$\frac{dPE}{dA_0} = 0, \frac{dPE}{dA} = 0, \frac{dPE}{dB} = 0 \qquad (Equation\ 21)$$

This system contains three equations and three unknowns, so solving the system gives the values for the coefficients which minimize the potential energy of the system. Plugging these values into the final package displacement field equation gives the final curvature of the package after the PWB has been straightened.

4.6 Final Interconnect Heights, Forces, and Pre-Stresses

The final interconnect heights are calculated based on the final displacement field equations. This is done in a very similar manner to how the initial interconnect heights were calculated. Subtracting the value of the final package displacement field at the (z, x) coordinates of a specific interconnect from the value of the final board displacement field at the (z, x) coordinates of the interconnect gives the final height of that interconnect. With the final interconnect heights known, the forces and pre-stresses are determined. The force in each interconnect is found by multiplying the axial stiffness of each interconnect by the change in height of each interconnect. The change in height is given by the final interconnect height minus the initial interconnect height. The interconnect pre-stresses are then determined by dividing the force in each interconnect by the cross-sectional area of the solder joint. Appendix 3 contains the full MATLAB code for the three-dimensional area array simplified model.

5. Three-Dimensional Area Array FEA Models

In order to determine if the new area array simplified model is accurately capturing the physics of the real-world assembly, the results from this simplified model need to be compared to results from a finite element model. Therefore, an area array FEA model must be created. As mentioned earlier, area array components encompass a large variety of electronic components from leadless ball grid arrays to leaded stud grid arrays. For this area array component model, a stud grid array component is modeled. A SGA is selected because including the pin leads of the SGA component adds a level of complexity not included with a leadless component. Using a more complex component to verify the simplified model ensures the simplified model will be useful in the widest range of applications.

Two different FEA models are used to verify the simplified model. The first is a shell and beam FEA model and the second is a more detailed solid 3D brick FEA model. In order to directly compare the simplified and FEA models and determine the accuracy of the simplified model calculations, the FEA model should include the same simplifying approximations as the simplified model. The shell and beam FEA model is created for this purpose. Exactly how the simplifying assumptions are included in the model will be explained in the following sections. Once the simplified model has been compared to an FEA model which includes the same simplifying assumptions, the validity of the simplifying assumptions must be confirmed. This is done by comparing the simplified shell and beam FEA model with a more detailed solid 3D brick element FEA model. Details will be provided in later sections about this model as well.

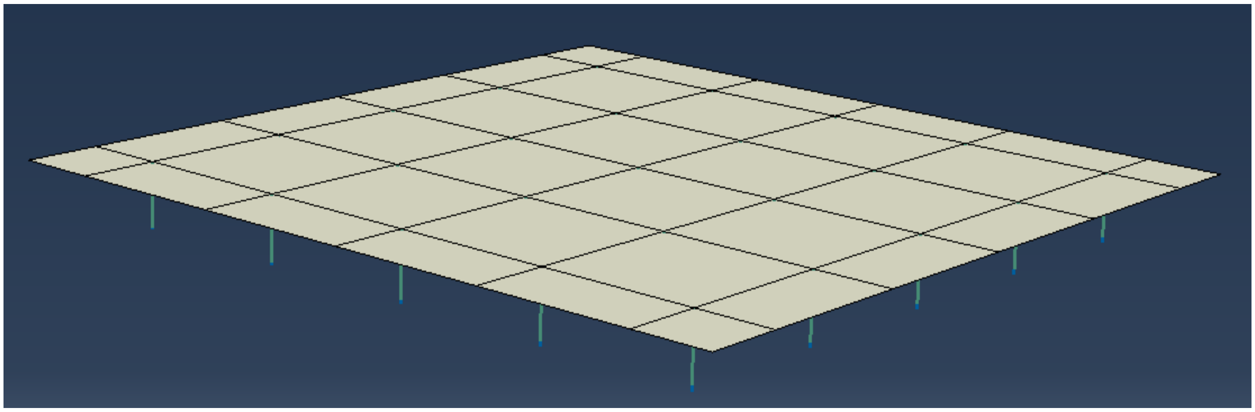
5.1 Three-Dimensional FEA Models Overview

The three-dimensional FEA models are created using ABAQUS finite element software. Due to the symmetric nature of the assembly, a quarter symmetry model of the PWB and component assembly is used. This means that only one fourth of the total assembly is modeled, and symmetry constraints are applied to the faces of the model representing the two major horizontal axes through the component. The component being modeled has 100 leads arranged in a ten x ten grid along the base of the component.

5.2 Area Array FEA Model: Shell and Beam Model

As mentioned previously, this FEA model is designed to include the same simplifying assumptions as the simplified model. The simplified model includes three main simplifying assumptions. The first is that the PWB is treated as a deformable shell. Therefore, the component package is modeled using shell elements in the FEA model. The second simplifying assumption in the simplified model is the interconnects are treated as deformable beams. This is incorporated into the FEA model by modeling the component leads and solder joints using beam elements. The third simplifying assumption made in the simplified model is the exclusion of the PWB. For this reason, the PWB is not included in the simplified FEA model. To include the effect of straightening the PWB in the model, displacement boundary conditions are applied directly to the bases of the solder joints. This mimics the simplified model process of using the PWB displacement field to govern the displacements of the bases of the solder joints. Figure 5 shows the shell and beam FEA model.

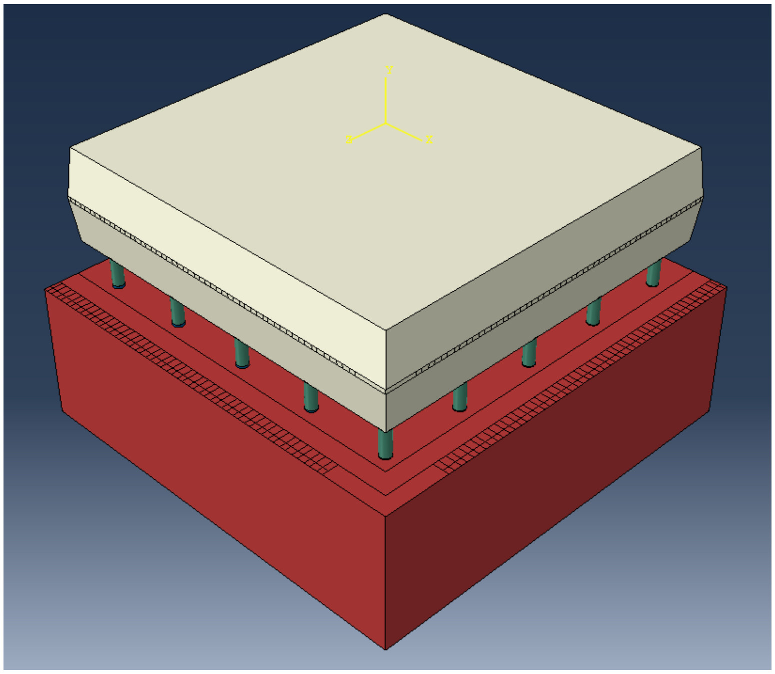
Figure 5: 3D Shell and Beam FEA Model



5.3 Area Array FEA Model: 3D Solid Brick Element Model

The detailed area array FEA model is setup like the shell and beam area array FEA model. In this detailed FEA model however, the component package, component leads, and solder joints are modeled using 3D brick elements. Also, the PWB is included in the model and boundary conditions are applied to the edges of the board to straighten it. Figure 6 shows the FEA model.

Figure 6: 3D Area Array FEA Model



5.3.1 Modeling of the PWB

Two separate PWB models are used for the detailed three-dimensional model. They are exactly the same, except one is for unidirectional curvature and the second is for bidirectional curvature. Both are modeled as deformable solid bodies made of FR4 with a curved top face. Displacement boundary conditions are applied across the external edges of the top PWB face to straighten the board. The curvature of the PWB is defined by the radius of curvature rather than the solder height difference. This method ensures that the PWB curvature will be the same across both models.

5.3.2 Modeling of the Component Package and Leads

In the three-dimensional model each lead is simply modeled as a solid bar with the material properties of copper. The component package itself is modeled as a solid deformable body made of epoxy molding compound. The leads are mated to the package using tie constraints which ensure the package and leads move together.

5.3.3 Modeling of the Solder Joints

The solder joints are all partitioned from one part where the bottom face has the same curvature as the top face of PWB. Starting with parts like this with the same size and curvatures as the PWB and partitioning them to create the solder joints ensures that the board and solder joints all fit together properly. Once the partitioning is completed, the top of the PWB is tied to the base of the solder joints and the top of the solder joints are tied to the base of the leads.

6. Results from Three-Dimensional Area Array Models

Comparing the simplified model results with the FEA model results can reveal the accuracy of the pre-stresses predicted by the simplified model. Comparing the results can also show if the simplifying assumptions made in the simplified model are valid. For this study, all of the pre-stresses to be presented are for the straightening of an initially warped PWB with a radius of curvature of 1208.7 mm. This radius of curvature corresponds to a 90 micron height difference between the corner and middle of the edge of the PWB and assumes that PWB is warped in such a way that the corners of the board are above the middle of the board.

6.1 Simplified Model Results

Figure 7 shows the interconnect pre-stress predictions for unidirectional curvature and Figure 8 shows the predictions for bidirectional curvature. In these plots, the pre-stresses are normalized with respect to the absolute value of the minimum (most compressive) pre-stress in the unidirectional curvature case.

Figure 7: Interconnect Pre-stresses for Unidirectional Curvature

Area Array Simplified Model

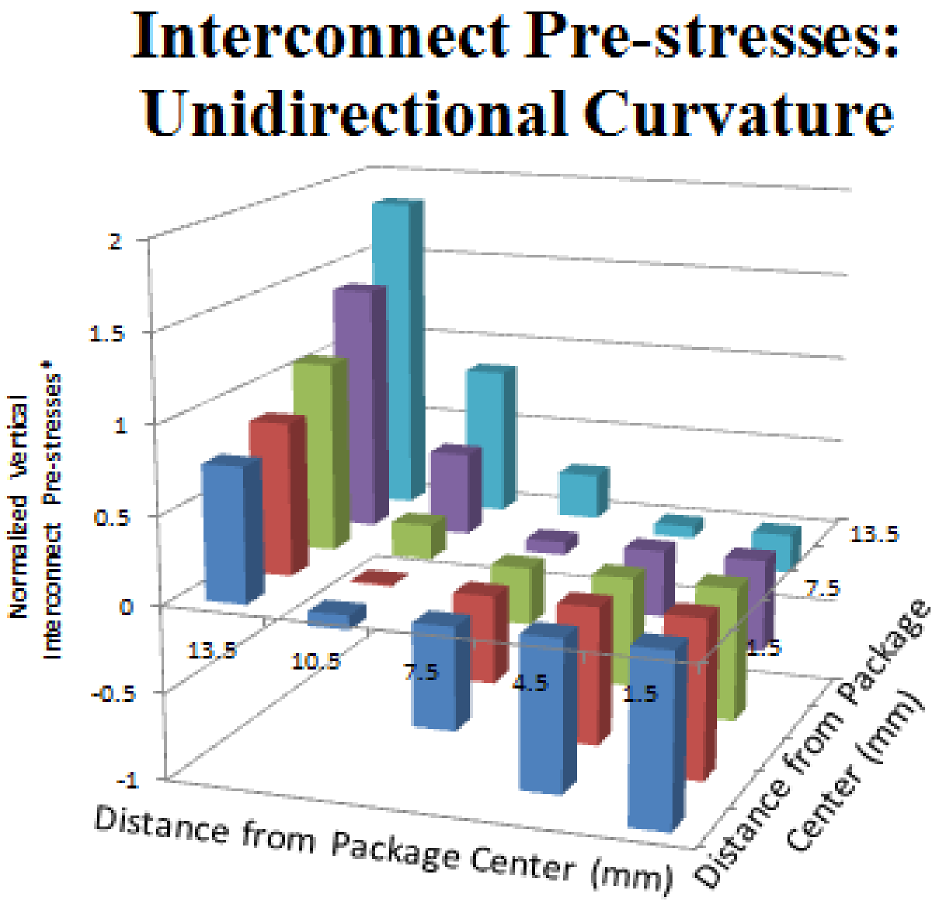
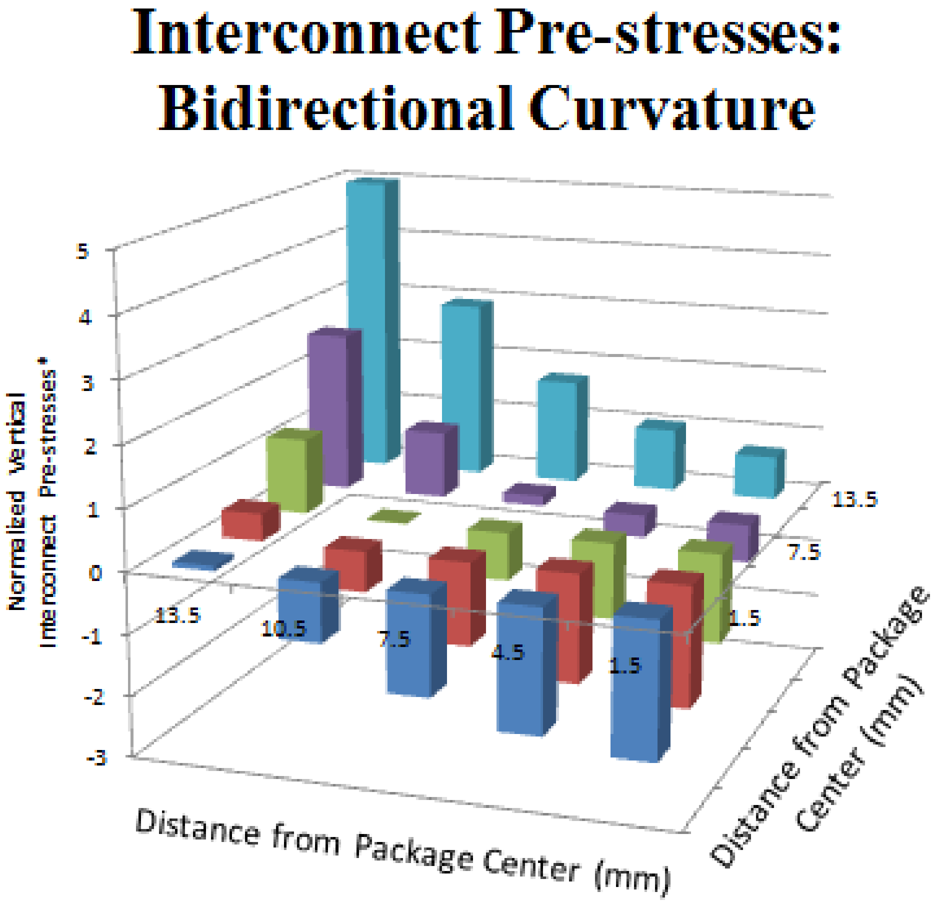


Figure 8: Interconnect Pre-stresses for Bidirectional Curvature

Area Array Simplified Model



These results show two main trends. First, there is a monotonic increase in the interconnect pre-stresses from the center of the package to the corner of the package. Second, the magnitudes of the pre-stresses are generally higher in the bidirectional curvature case than in the unidirectional curvature case. These results also show which solder joints are most susceptible to failure. For concave up curvature, which is presented above and refers to curvature where the PWB corners are initially above the center of the package, the solder joint closest to the corner of the package is the most likely to fail. This joint experiences the highest tensile (positive) pre-stress which is why it is the most likely to fail. For concave down curvature, the reverse of concave up curvature, the signs will flip in the results shown above. This means the solder joint closest to the center of the package will be most likely to fail because it experiences the highest tensile pre-stress.

6.2 Results from Shell and Beam FEA Area Array Model

Figure 9 shows the interconnect pre-stress predictions for unidirectional curvature and Figure 10 shows the predictions for bidirectional curvature. Once again, these results are normalized with respect to the absolute value of the minimum (most compressive) pre-stress in the unidirectional curvature case.

Figure 9: Interconnect Pre-stresses for Unidirectional Curvature Shell and Beam FEA Model

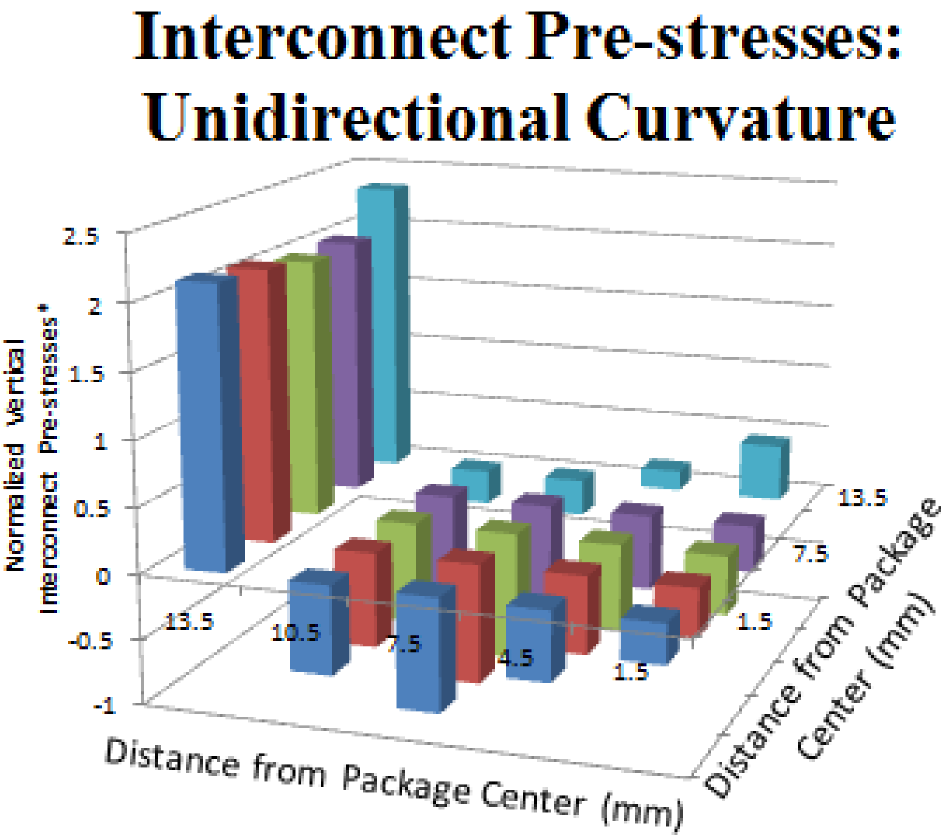
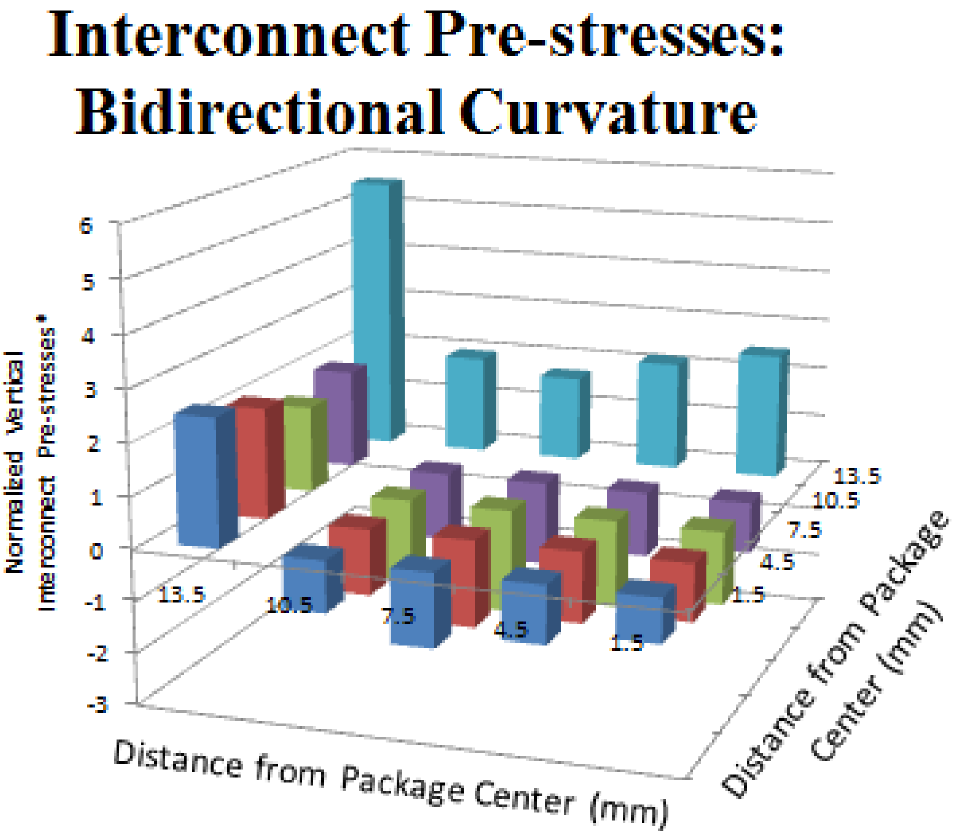


Figure 10: Interconnect Pre-stresses for Bidirectional Curvature Shell and Beam FEA Model



Qualitatively, these results are similar to the results from the simplified model. Again, the interconnect pre-stresses increase from the center of the package to the corner. However, in this FEA model, the increase is not monotonic. The pre-stresses decrease, or become more compressive, before they begin to increase. Just like in the simplified model, the pre-stress predictions are generally higher for the bidirectional curvature case. In the FEA model for concave up curvature, the highest tensile pre-stress occurs in the solder joint closest to the package corner. This is same location as in the simplified model. The location of the largest tensile stress for concave down curvature is different. Because of the non-monotonic pre-stress increase present in the shell and beam FEA, the maximum tensile pre-stress occurs in the joint in the center of the quadrant modeled.

6.3 Detailed Solid 3D Brick FEA Model Results

Figure 11 shows the interconnect pre-stress predictions for unidirectional curvature and Figure 12 shows the predictions for bidirectional curvature. Again, the results are normalized with respect to the absolute value of the minimum (most compressive) pre-stress in the unidirectional case.

Figure 11: Interconnect Pre-Stresses for Unidirectional Curvature Detailed FEA Model

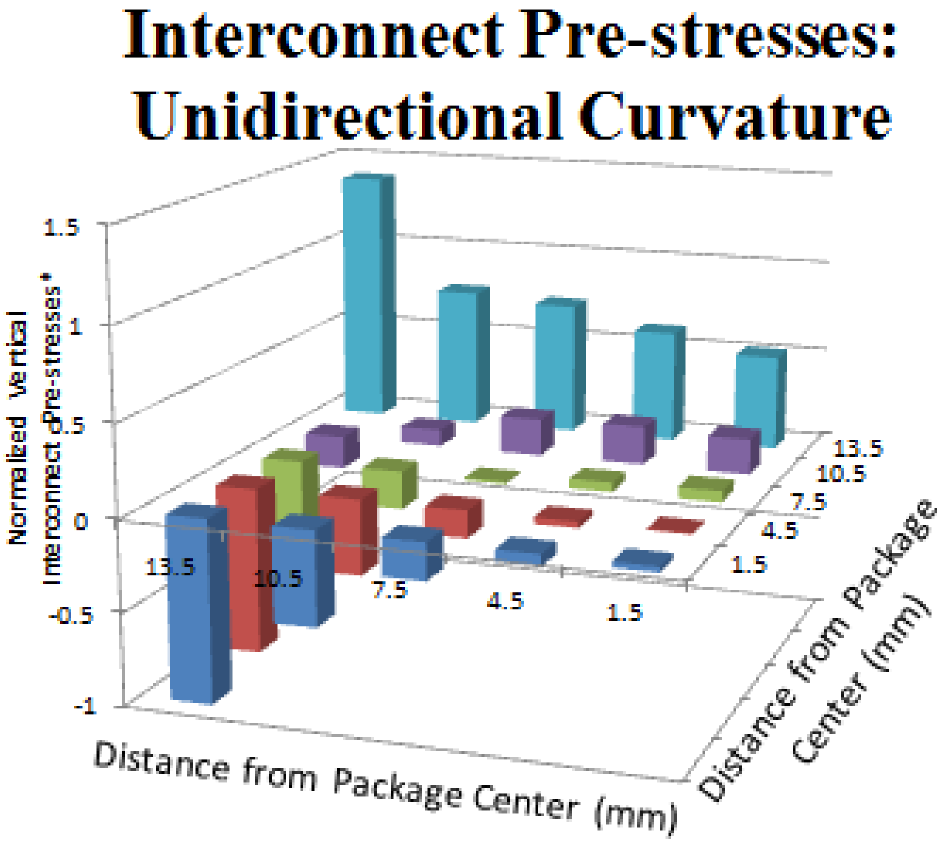
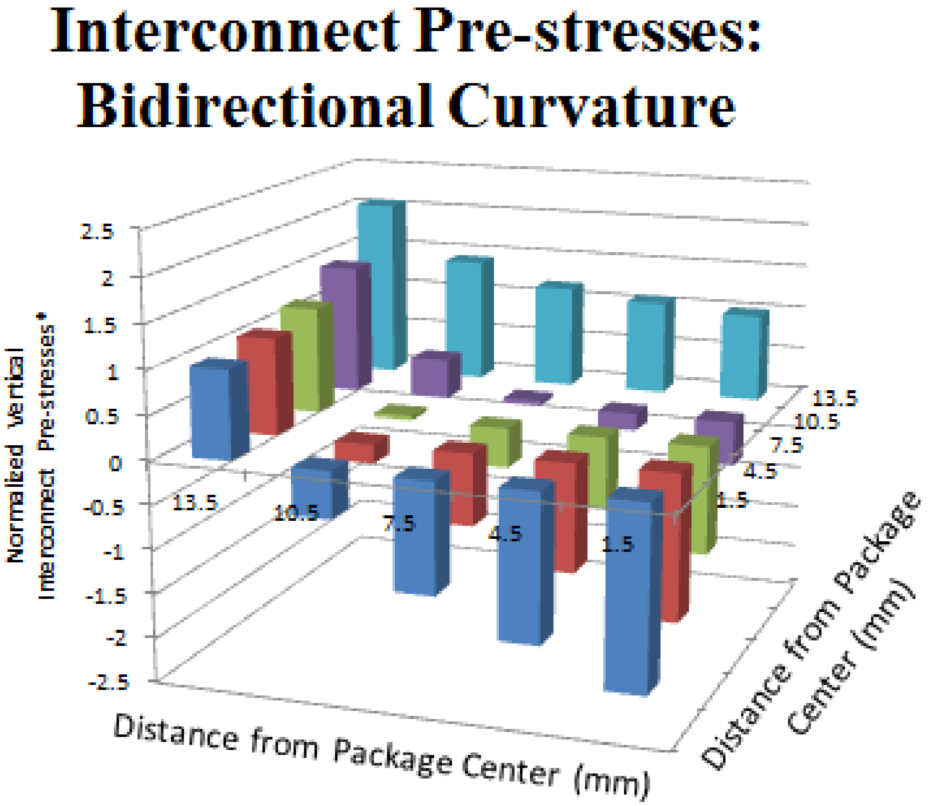


Figure 12: Interconnect Pre-Stresses for Bidirectional Curvature Detailed FEA Model



Once again, the pre-stresses change monotonically. In this model however, there is a pre-stress increase along one axis and a decrease along the other in the unidirectional case. This is due to the larger effect anticlastic curvature has in the detailed FEA model. Overall, the pre-stresses still increase from the center of the package to the corner. Again, just like in the previous area array models, the pre-stresses are generally higher for the bidirectional curvature case.

This detailed FEA model shows that corner solder joint is most susceptible to failure for concave up curvature in both the unidirectional and bidirectional curvature cases. This joint has the highest tensile stress in this model, just like in the previous area array models. For concave down curvature, the results are slightly different. In the unidirectional curvature case, the joint most likely to fail is the one in the center of edge that undergoes anticlastic curvature. In the bidirectional curvature case, the joint closest to the center of the package is most likely to fail.

6.4 Comparison of Simplified and Shell and Beam FEA Model Results

Figure 13 gives a selection of the results presented in Figures 7 and 9. The plot shows the maximum and minimum pre-stresses for unidirectional curvature from the simplified and FEA models normalized with respect to the absolute value of the minimum (most compressive) pre-stress from the FEA model. Figure 14 gives the same plot but for bidirectional curvature.

Figure 13: Normalized Pre-stresses for Unidirectional Curvature

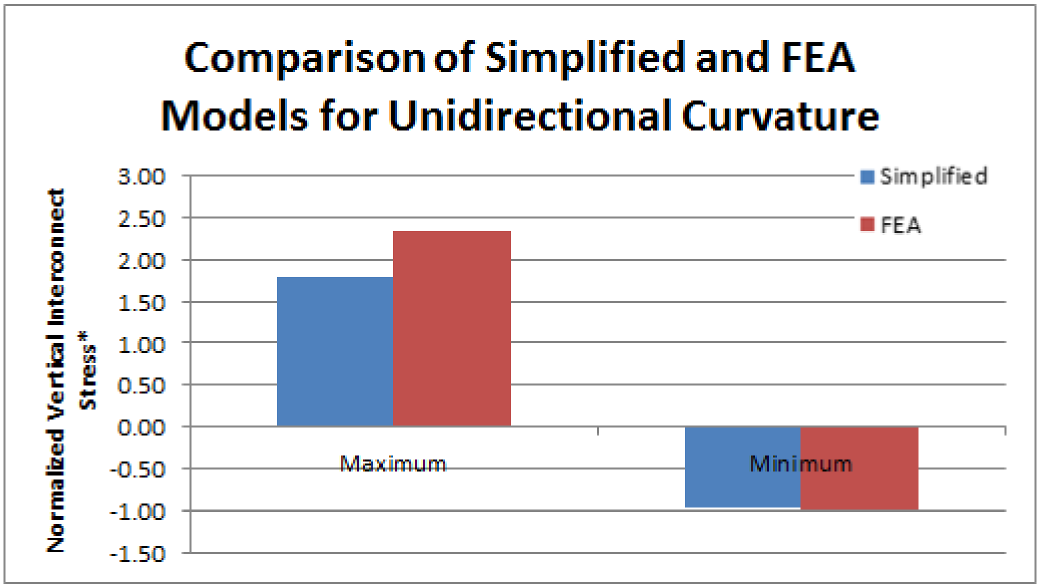
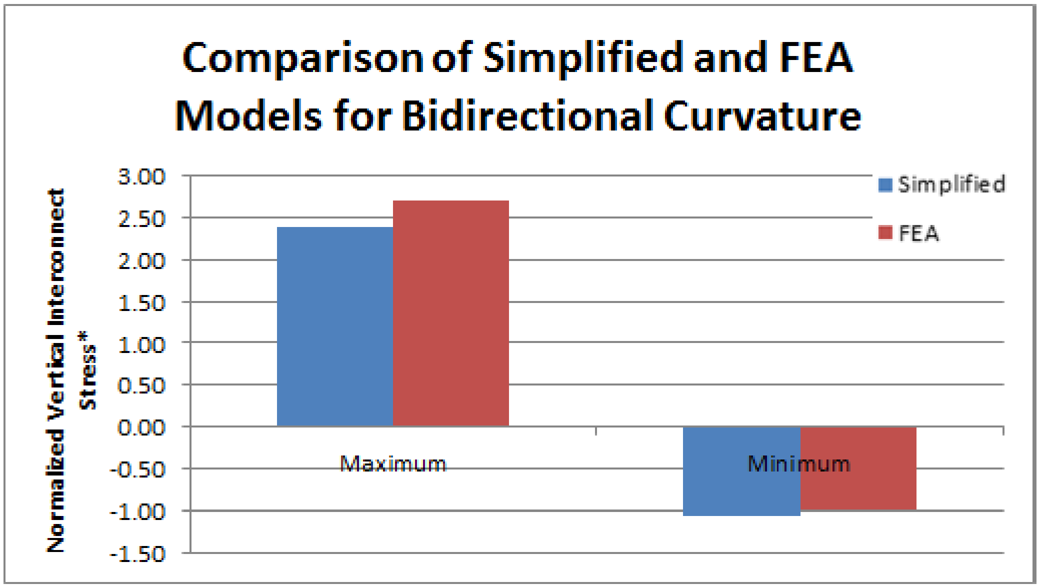


Figure 14: Normalized Pre-stresses for Bidirectional Curvature



These figures present some exciting results. They show that there is close agreement between the simplified and FEA models. There are some discrepancies between the two models, like non-monotonic changes in the FEA model, but these are likely due to the geometric complexities present in the FEA model. For example, in the simplified model, the PWB is constrained to deform as a quadratic, but it is free to deform in any manner in the FEA model. Overall, the agreement between the models is a good sign which means that the simplified model is accurately capturing the general physics of the problem and is functioning as intended.

6.5 Comparing Results of the Solid 3D Brick FEA Model to the Shell and Beam FEA Model

Figure 15 shows the maximum and minimum pre-stresses for unidirectional curvature from the two FEA models normalized with respect to the absolute value of the minimum (most compressive) pre-stress from the detailed FEA model. Figure 16 gives the same plot but for bidirectional curvature.

Figure 15: Normalized Pre-Stresses for Unidirectional Curvature

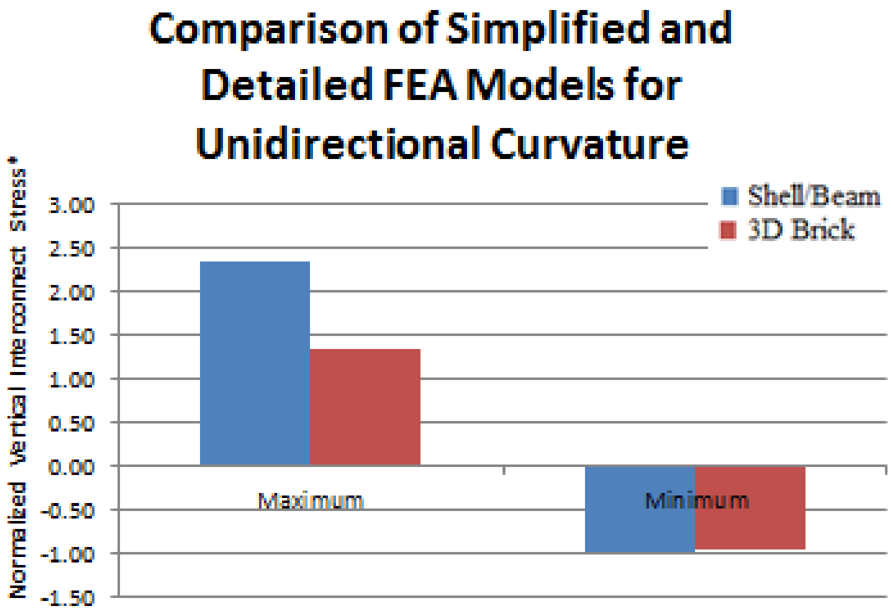
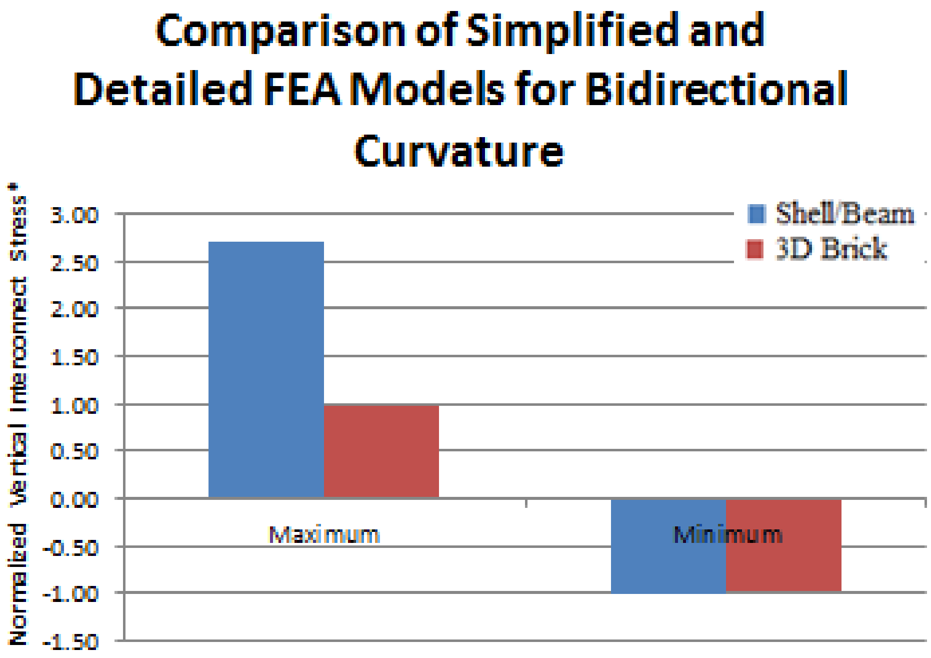


Figure 16: Normalized Pre-stresses for Bidirectional Curvature



These results show good agreement between the FEA models in terms of the minimum pre-stresses predicted, but some difference between the maximum pre-stresses predicted. The differences in the results of the two models can be attributed to the simplifying assumptions which do not allow the shell and beam FEA model to include all of the 3D geometric non-uniformities present in the detailed 3D brick FEA model. Still, the inclusion of the simplifying assumptions in the shell and beam model does not result in drastically different results from the detailed FEA model.

6.6 Comparing Results of the Simplified and Solid 3D Brick FEA Models

Figure 17 shows the maximum and minimum pre-stresses for unidirectional curvature from the simplified model and the detailed solid 3D FEA model normalized with respect to the minimum (most compressive) pre-stress from the detailed FEA model. Figure 18 gives the same plot but for bidirectional curvature.

Figure 17: Normalized Pre-Stresses for Unidirectional Curvature

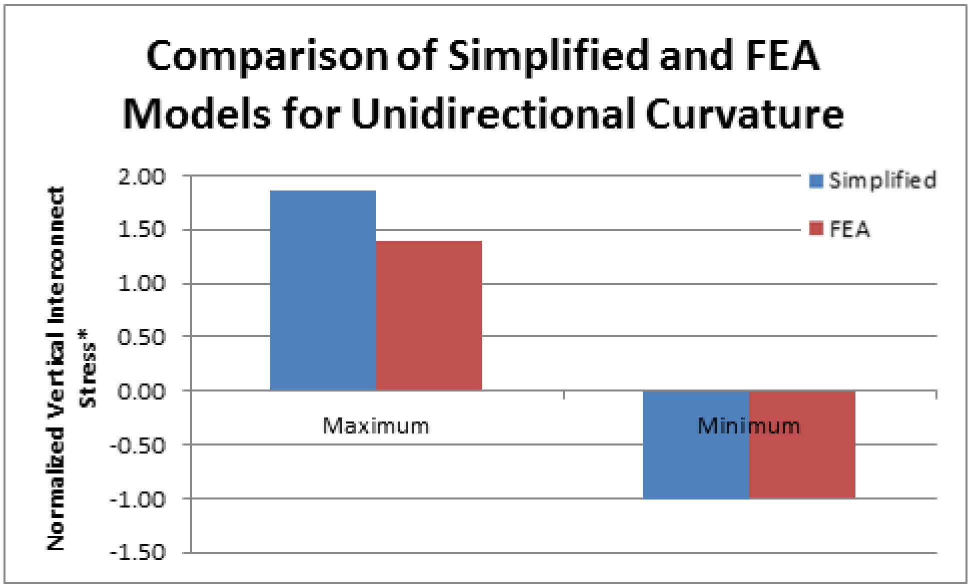
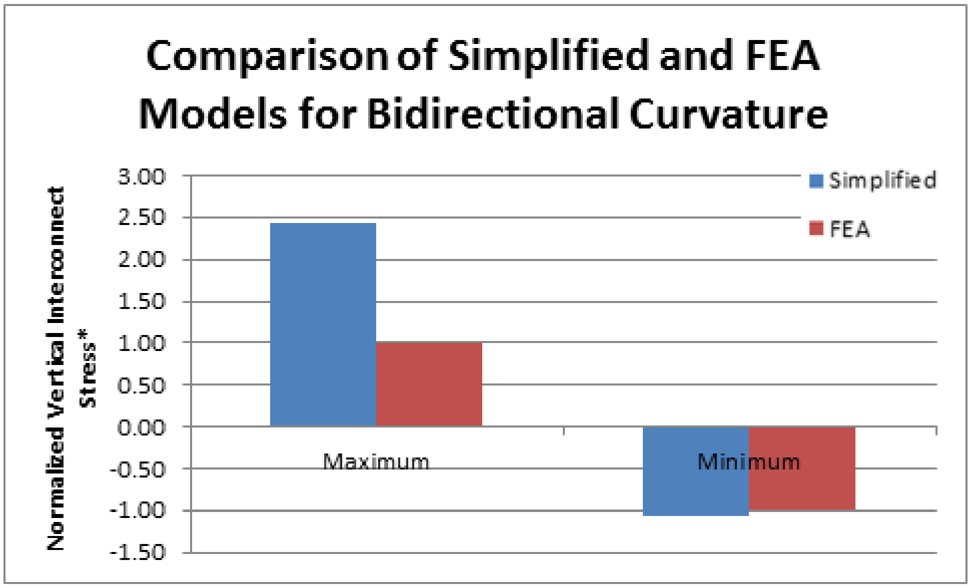


Figure 18: Normalized Pre-stresses for Bidirectional Curvature



These results backup the conclusions the FEA model comparison revealed about the simplifying assumptions. The plots show very good agreement between the simplified model and the detailed FEA model in terms of the unidirectional curvature results and the minimum pre-stresses predicted for bidirectional curvature. There is a larger difference between the maximum pre-stresses predicted for bidirectional curvature. Once again, the differences in the results can be attributed to the simplifying assumptions which do not capture all of the 3D geometric non-uniformities present in the detailed 3D brick FEA model. However, the simplifying assumptions do not drastically change the pre-stress predictions for the simplified model. This means that the area array simplified model is a fairly accurate representation of the real-world assembly and makes good approximations of the pre-stresses resulting from straightening an initially warped PWB.

7. Alternate Applications for Simplified Model

This simplified model can be applied in situations other than just determining the pre-stresses which develop from straightening an initially warped board. The model is set up to be able to consider any initial PWB curvature as well as any final PWB curvature. This means it can be used to study an initially warped board which is straightened, as presented previously, or an initially flat board which is bent to some final curvature, or a board with some initial curvature which is bent to some other final curvature. This capability makes the simplified model both versatile and useful. Not only can it be used to predict pre-stresses which develop in component interconnects, it can also be used to predict the stresses which develop in component interconnects as a result of some mechanical bending. As long as the final bend curvature is known, it doesn't matter if the PWB bending is caused by vibrational, thermomechanical, or shock/drop loading. The simplified model can be used to predict the bending stresses resulting from all three of these loading conditions. Although this capability has not yet been verified with FEA, an example will be presented below.

7.1 Alternate Application for Simplified Model: Bending an Initially Straight PWB

This example is very similar to the case studied in the previous sections of this paper with one major difference. Instead of starting with a board with an initial curvature of 1208.7 mm and straightening it so it becomes flat, the starting point is a flat board and it is bent to a final curvature of 1208.7 mm. In this case, because the board is initially flat, there is no variation in the initial height or stiffness of the solder joints. This is the main difference between bending an initially flat board and straightening an initially bent board. The stresses resulting from bending the initially flat PWB in this fashion are presented in Figures 19 and 20. Figure 19 shows the stresses if the final board curvature is unidirectional and Figure 20 shows the stresses if the final board curvature is bidirectional. The stresses presented in these plots are normalized with respect to the absolute value of the minimum (most compressive) stress in the unidirectional case.

Figure 19: Interconnect Stresses for Unidirectional Curvature

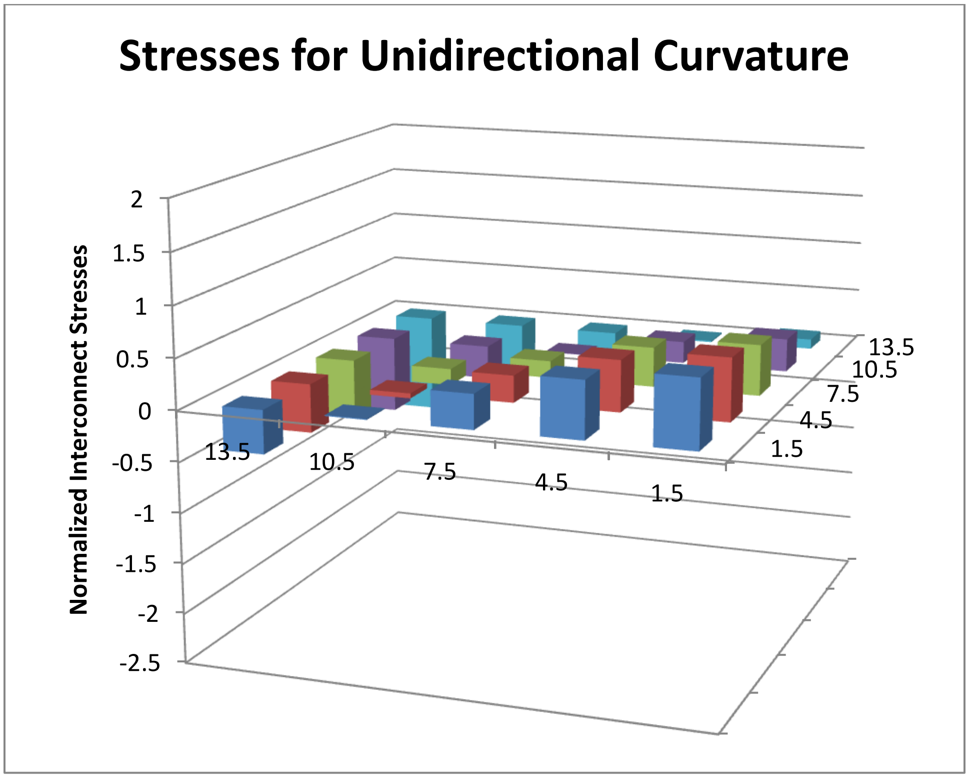
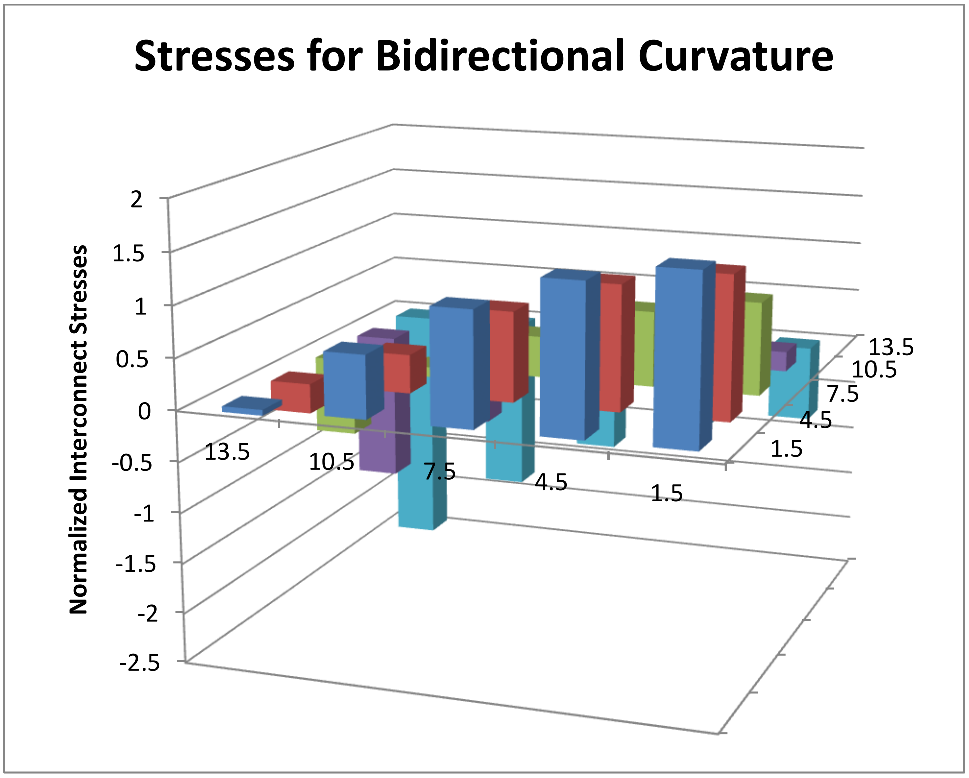


Figure 20: Interconnect Stresses for Bidirectional Curvature



These plots illustrate the simplified model's ability to make stress predictions in situations other than just straightening an initially warped PWB. While these stress predictions have not yet been verified by FEA, the process and model have been verified. The model is unchanged from the one presented previously in Section 4 which was verified for straightening an initially warped board and the same process can be used. The only difference is the inputs are changed to reflect an initially flat board and a final board which is bent. Therefore, little to no adjustment should be needed to get the stresses predicted by the simplified model to agree with predictions from an FEA model.

8. Summary, Conclusions, Future Work, and Contributions

The study has provided a rapid and simple modeling capability to estimate the stresses caused in SMT interconnects by PWB flexure. As an example, the model has been verified with more detailed modeling techniques like FEA, for pre-stresses developing in the interconnects as a result of straightening of a PWB that warped during the soldering process. Future applications of this model include assessment of interconnects stresses due to vibration-induced PWB curvature.

8.1 Summary of Work Completed

The objective of this study was to develop a method for quickly and accurately predicting the pre-stresses which develop in the interconnects of a component which is part of an electronic assembly. If the component interconnects are soldered to a PWB which warps during the soldering process, they will deform when the PWB is straightened during the assembly process. This deformation causes pre-stresses to develop in the interconnects which reduce the overall life and reliability of the assembly.

FEA models, while accurate, are not fast enough or versatile enough to be used to predict the pre-stresses. Creating, running, and analyzing an FEA model can take a long time, and a new model is needed for each new situation. Conversely, a simplified model which does not require numerical meshing can be run quickly and allows for many different geometries to be analyzed with the same model; with minor changes in the inputs. Therefore, if an accurate simplified model can be created, it will provide the quick and versatile method for predicting interconnect stresses caused by PWB curvature (including the pre-stresses from straightening initially warped PWAs). In order to verify the accuracy of the simplified model, its results are verified here with more detailed FEA models.

The simplified model is developed in this study using the Rayleigh-Ritz principle. The model is first created for area array components. Each interconnect is modeled as two deformable beams in series, one representing the solder joint and one representing the component lead. The simplified model calculates the initial height and stiffness of the each solder joint as well as the stiffness of the component leads. The height of each solder joint depends on the initial-curvature of the PWB. The component is modeled as an equivalent shell. Next, the strain energies of the package and interconnects are determined due to PWB and package curvatures that are parameterized with a quadratic curve. These strain energies are used to develop the potential energy equation of the system. Minimizing the potential energy equation yields the final package curvature. The final package curvature is then used to determine the final interconnect heights, forces, and pre-stresses.

This simplified Rayleigh-Ritz model is coded in MATLAB and the implementation is verified by comparing to the results from a similar shell and beam FEA model. This simplified shell and beam FEA model includes the same simplifying assumptions made in the Rayleigh-Ritz model. These assumptions are that the PWB compliance is excluded, the component package is modeled as a deformable shell, and the interconnects are modeled as deformable beams. The pre-stress predictions from the area array simplified model agree well with the predictions from the shell and beam FEA model.

In order to determine if the simplifying assumptions made in these models still provide a good approximation of the real-world assembly a more detailed FEA model is needed. This detailed model does not include the simplifying assumptions of the shell and beam FEA model and uses full 3D brick elements and also includes the compliance of the PWB. Comparing the results from this detailed FEA model with the results from the shell and beam FEA model reveal that the simplifying assumptions do not capture all of the 3D geometric non-uniformities, but do a generally good job of approximating the overall trends of the detailed model. This is backed up by comparing the simplified model results directly to the detailed FEA model results and suggests that the area array Rayleigh-Ritz model is a reasonable tool for predicting the pre-stresses which occur in the real-world assembly.

8.2 Conclusions

There are many significant conclusions to be drawn from this project. First, the basic modeling technique of treating the solder joints and leads as beams in series has been shown to be an effective method for representing the component interconnects. Second, utilizing the Rayleigh-Ritz method in the simplified model provides the capability to determine the final package curvature after the straightening process is completed. The results from the area array model confirm that the general process used in the simplified model is a good approximation of the mechanics included in more detailed FEA models.

Finally, and most importantly, the area array simplified model can be used as a tool to predict the pre-stresses which result from straightening an initially warped PWB. The area array simplified model closely agrees with the shell and beam FEA model, showing the model functions properly and includes all of the basic physics of the problem. The shell and beam FEA model shows general agreement with the detailed 3D FEA model, which means the area array simplified model does not capture all of the 3D geometric non-uniformities, but still is a good overall approximation of the real-world assembly.

8.3 Future Work

This study provides a very solid foundation for future work in the same area. The area array simplified model can be used as a basis for further model development. Expansion of the simplified model to include perimeter-leaded components was started, but due to time restrictions, further work into developing the simplified model could not be done at this time. The completed perimeter-leaded model work can be found in Appendix 1.

First, the perimeter-leaded component model provided in Appendix 1 can be used as a starting point for improving the versatility of the simplified model. The simplified model can also be expanded to include other component types in addition to area array and perimeter-leaded components. The final result of this work would be a simplified model which can then be used to quickly and accurately predict the stresses in a wide variety of circuit assemblies. These stresses could be the result of straightening an initially warped board or from exposing a board to some external loading conditions (vibration, thermomechanical, shock/drop) which cause bending.

After adequate calibration and verification, this simplified model can then be improved upon and expanded. The current model is an elastic model, meaning it does not take the plasticity or viscoplasticity of the materials into account. One potential avenue for future work is to expand the simplified model to include material plasticity and viscoplasticity. Doing this will further improve the accuracy of the model and make it a more realistic representation of the real-world assembly.

As mentioned in Section 7, this simplified model can be used in general to predict the stresses that result from bending a board of some known initial curvature to some other known final curvature. This provides another opportunity for future work. First, an FEA model must be developed to verify the results presented in Section 7.1. These results are for bending an initially flat board. Another FEA model could be developed to verify results for bending an initially bent board to a different final curvature.

Finally, a completed simplified model can be a vital part of a larger model focused on determining the expected life of an electronic assembly. When pre-stresses develop in the interconnects of an electronic component, the overall life of the component is reduced. A simplified rapid-assessment model that can accurately predict the pre-stresses which develop in the interconnects is necessary before a model which can predict how various types of PWB bending affects the life of the component or assembly can be created. This PWB bending could be the result of vibrational, thermomechanical, or shock/drop loading or from straightening an initially warped board.

8.4 Contributions

This study offers many benefits to the electronic packaging community. The primary benefit of this model is that it provides a method to quickly and accurately predict the stresses which develop in the component interconnects as a result of bending a PWB onto which the component is mounted. The simplified model can be used to predict these pre-stresses for a wide variety of electronic assemblies. This provides a reliable avenue to predict the interconnect pre-stresses without relying on finite element analysis. FEA does not offer the speed or versatility provided by this simplified model. Creating an FEA model is a very time-consuming process, and a new model will be needed for each different component configuration. The simplified model only requires the initial conditions to be input, and results are produced within a few minutes. No meshing is required in this model. Also, a different configuration can be analyzed by simply changing the initial conditions and running the model again.

Another benefit this study provides is that it offers insight into how the initial curvature of the PWB affects the interconnect pre-stresses which develop when the PWB is straightened. The studies performed with the perimeter-leaded component FEA models which are included in Appendix 1 offer this insight. The first study, which examined the pre-stresses along the length of the component, shows how the interconnect pre-stresses vary along the component. It can be seen how the pre-stresses decrease from the corner interconnect to the middle interconnect. The results for the unidirectional curvature case also show the large variation in the interconnect pre-stresses along the straight side of the component. In fact, the overall variation in the interconnect pre-stresses along the straight side of the component is larger than the variation in the interconnect pre-stresses along the curved side of the component. The second study, the parametric study, shows how the maximum and minimum pre-stresses change as the initial curvature of the PWB changes. Not only do these results show how the pre-stresses decrease as the initial radius of curvature of the PWB increases, but they also show that the magnitudes of the maximum and minimum pre-stresses are larger for the bidirectional curvature case.

A third contribution from this study is the simplified model itself. This model can predict the stresses resulting from bending a PWB and can act as a critical building block for many other models. As explained in the future work section, this model can be used as jumping-off point for a model which predicts the life of the electronic component or assembly. Having models which can quickly and accurately predict how bending of a PWB affects the interconnect stresses and overall assembly life will be of great help to the designers, manufacturers, and producers of these electronic assemblies, and this simplified model is the first step in the process of developing these models.

Appendix 1: Perimeter-Leaded Component Models

One way to improve the versatility of the simplified model is to make the model applicable for a wider range of components. This can be done by expanding the simplified model to include perimeter-leaded components in addition to area array components. The same process used to develop the area array simplified model can be applied to develop the perimeter-leaded model. First the simplified model is developed and then an FEA model is used to verify the model.

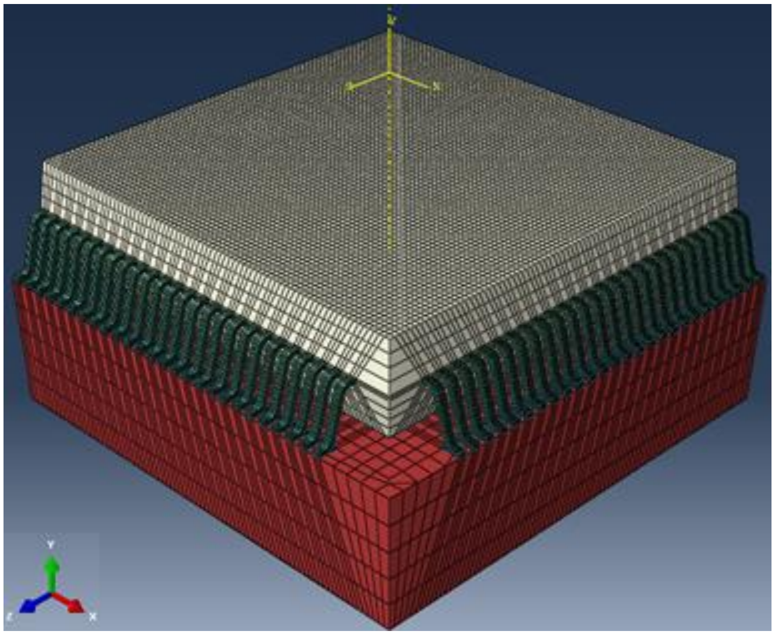
Perimeter-Leaded Simplified Model

Like the area array model, the perimeter-leaded simplified model is based on the Raleigh-Ritz Method and utilizes the potential energies and strain energies of the assembly. The perimeter-leaded component model follows the same basic process as the area array model. First solder joint heights and interconnect stiffnesses are calculated. This model assumes the leads are gull-wing leads, and therefore the equations developed by Kotlowitz to approximate gull-wing lead stiffness are used. The Kotlowitz equations are presented in Appendix 2. Then the strain energies of the package and interconnects are determined and the potential energy of the entire system is calculated. Next, the potential energy equation is minimized to determine the final package displacement field and this displacement field is used to determine the final interconnect heights, stresses, and forces. The full perimeter-leaded simplified model code is provided in Appendix 4.

Perimeter-Leaded FEA Model

Like the area array FEA models, the perimeter-leaded FEA model is created using ABAQUS finite element software. The model is created using a quad flat pack (QFP) component with 52 gull-wing leads on each side. For the perimeter-leaded model, only a detailed solid 3D brick model is created, not a shell and beam FEA model. This FEA model contains the PWB, the component package, the component leads, and the solder joints. Due to the symmetric nature of the assembly, a quarter symmetry model of the PWB and component assembly is used. This means that only one fourth of the total assembly is modeled, and symmetry constraints are applied to the faces of the model representing the two major horizontal axes through the component. Figure 21 below shows the full three-dimensional FEA model assembly. In this image, the symmetry constraints are applied along the two rear faces.

Figure 21: Three-dimensional FEA Model



Results from the Perimeter-Leaded Simplified Model

Two studies are performed with perimeter-leaded models. The first is a parametric study of various initial PWB radii of curvature looking at the maximum and minimum pre-stresses in each case. The second study examines the pre-stresses in each interconnect along the length of the component for a radius of curvature of 1208.7 mm.

Table 1 shows the pre-stresses in each interconnect along the component for a unidirectional convex radius of curvature of 1208.7 mm. This unidirectional curvature can be represented in the PWB displacement field by setting C to .0004 and D to 0. Table 2 shows the pre-stresses in each interconnect along the component for a bidirectional radius of curvature of 1208.7 mm. For bidirectional curvature, both constants are set to .0004. Both of these tables show results for concave up curvature.

Table 1: Interconnect Pre-Stresses for Unidirectional Radius of Curvature of 1208.7 mm

Interconnect	Curved	Straight
1	0.29	1.05
2	0.14	1.04
3	-0.01	1.04
4	-0.15	1.03
5	-0.28	1.03
6	-0.41	1.02
7	-0.53	1.02
8	-0.65	1.01
9	-0.76	1.01
10	-0.86	1.01
11	-0.96	1.00
12	-1.05	1.00
13	-1.13	1.00
14	-1.21	0.99
15	-1.28	0.99
16	-1.35	0.99
17	-1.41	0.99
18	-1.46	0.98
19	-1.51	0.98
20	-1.56	0.98
21	-1.59	0.98
22	-1.62	0.98
23	-1.65	0.98
24	-1.67	0.98
25	-1.68	0.98
26	-1.68	0.98

Table 2: Interconnect Pre-Stresses for a Bidirectional Radius of Curvature of 1208.7 mm

Interconnect	Pre-Stress (MPa)
1	1.33
2	1.18
3	1.03
4	0.88
5	0.74
6	0.61
7	0.49
8	0.37
9	0.25
10	0.15
11	0.05
12	-0.05
13	-0.13
14	-0.22
15	-0.29
16	-0.36
17	-0.42
18	-0.48
19	-0.53
20	-0.57
21	-0.61
22	-0.64
23	-0.67
24	-0.69
25	-0.70
26	-0.71

These results show that for unidirectional curvature there is some variation in the pre-stresses on the initially straight side of the component in addition the greater variation along the initially curved side of the component. They also show that the results are identical on both sides of the component for the bidirectional curvature case which is why only one column of data is provided.

This purpose of the parametric study is to determine the effect changing the initial curvature of the PWB has on the pre-stresses which develop in the interconnects. Table 3 shows the curvatures used in the parametric study and the corresponding values for the constants of the initial PWB displacement field in the simplified model.

Table 3: Parametric Study Radii of Curvature and Corresponding Displacement Field Constants

	Unidirectional		Bidirectional	
Radius of Curvature	C	D	C	D
1208.7	0.0004	0	0.0004	0.0004
1359.8	0.00036	0	0.00036	0.00036
1554.1	0.00031	0	0.00031	0.00031
1813.1	0.00027	0	0.00027	0.00027
2175.7	0.00022	0	0.00022	0.00022
2719.6	0.00018	0	0.00018	0.00018
3626.1	0.00013	0	0.00013	0.00013
5439.1	0.00009	0	0.00009	0.00009
10878.1	0.00004	0	0.00004	0.00004

Tables 4 and 5 below show the results of the parametric study. Table 4 shows the maximum and minimum pre-stresses for unidirectional curvatures. Table 5 shows the maximum and minimum pre-stresses for bidirectional curvatures.

Table 4: Maximum and Minimum Pre-Stresses for Various Unidirectional Radii of Curvature

Radius of Curvature	Curved	Straight
1208.7	-1.7	1.0
1359.8	-1.5	0.9
1554.1	-1.3	0.8
1813.1	-1.1	0.7
2175.7	-0.9	0.6
2719.6	-0.8	0.5
3626.1	-0.5	0.3
5439.1	-0.4	0.2
10878.1	-0.2	0.1
-10878.1	0.2	-0.1
-5439.1	0.3	-0.2
-3626.1	0.5	-0.3
-2719.6	0.8	-0.5
-2175.7	0.9	-0.6
-1813.1	1.1	-0.7
-1554.1	1.3	-0.8
-1359.8	1.5	-0.9
-1208.7	1.7	-1.0

Table 5: Maximum and Minimum Pre-Stresses for Various Bidirectional Radii of Curvature

Radius of Curvature	Middle	Corner
1208.7	-0.7	1.3
1359.8	-0.6	1.2
1554.1	-0.5	1.0
1813.1	-0.5	0.9
2175.7	-0.4	0.7
2719.6	-0.3	0.6
3626.1	-0.2	0.4
5439.1	-0.2	0.3
10878.1	-0.1	0.1
-10878.1	0.1	-0.1
-5439.1	0.2	-0.3
-3626.1	0.2	-0.4
-2719.6	0.3	-0.6
-2175.7	0.4	-0.7
-1813.1	0.5	-0.9
-1554.1	0.5	-1.0
-1359.8	0.6	-1.2
-1208.7	0.7	-1.3

These tables show how the interconnect pre-stresses change as a result of changes in the initial PWB curvature.

Results from Three-Dimensional FEA Model

The same studies performed for the simplified models are performed with the FEA Model. The pre-stresses which develop in each interconnect from straightening a PWB with an initial concave up curvature of 1208.7 millimeters are presented in Table 6 and Table 7. Table 6 shows the pre-stresses for unidirectional curvature and Table 7 shows the pre-stresses for bidirectional curvature.

Table 6: Pre-stresses for a 1208.7 mm Unidirectional Concave Up Radius of Curvature

	Pre-stress (MPa)	
Interconnect #	Curved Side	Straight Side
1	8.5	26.9
2	4.2	22.8
3	1.5	18.9
4	-0.7	15.4
5	-2.7	14.6
6	-4.2	12.5
7	-5.7	9.8
8	-6.8	9.2
9	-7.9	7.8
10	-8.8	5.6
11	-9.9	4.5
12	-10.7	3.3
13	-11.4	2.4
14	-11.9	1.0
15	-12.6	0.4
16	-13.2	-0.7
17	-13.7	-1.3
18	-14.0	-2.3
19	-14.6	-2.6
20	-15.0	-2.5
21	-15.2	-3.0
22	-15.5	-3.0
23	-15.7	-3.3
24	-15.8	-3.5
25	-16.3	-3.3
26	-15.3	-3.4

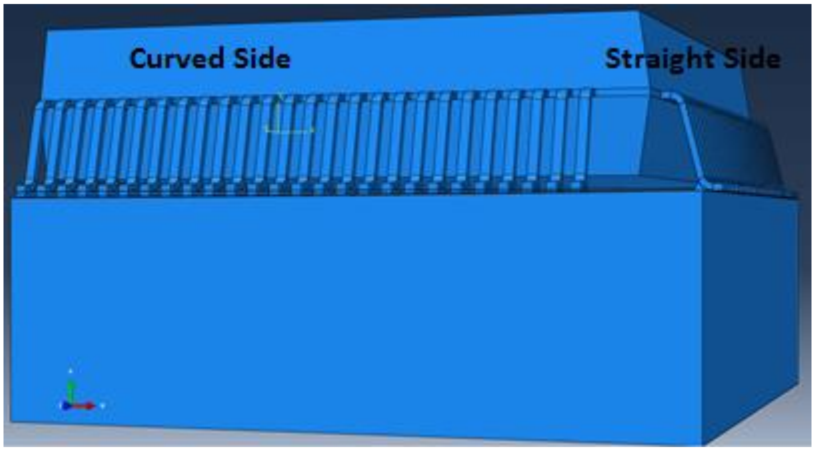
Table 7: Pre-stresses for a 1208.7 mm Bidirectional Concave Up Radius of Curvature

Interconnect #	Pre-stress (MPa)
1	16.7
2	14.9
3	12.4
4	10.0
5	7.6
6	5.2
7	2.9
8	0.7
9	-1.5
10	-3.3
11	-5.1
12	-6.7
13	-8.2
14	-9.5
15	-10.7
16	-11.8
17	-12.8
18	-13.6
19	-14.3
20	-15.0
21	-15.5
22	-15.9
23	-16.2
24	-16.5
25	-16.7
26	-16.0

In Table 6, the curved side refers to the interconnects on the curved edge of the PWB, and the straight side refers to the interconnects located on the straight edge of the PWB. These are labeled in Figure 22 below. Due to the symmetry of the bidirectional curvature model, the pre-stresses are the same on both of sides. This is why only one column of data is presented.

Figure 22: Location of Curved Side and Straight Side

of Component for Unidirectional Curvature



The results in Tables 6 and 7 above show how the pre-stresses vary from the middle interconnects to the edge interconnects. Table 6 also shows that the interconnect pre-stresses vary along the straight side of the package in addition to the curved side.

The results for the parametric study are presented in Table 8. As explained earlier, the radius of curvature of the PWB is varied in this study as opposed to the difference in solder joint heights. The radius of curvature of 1208.7 millimeters is the most curved case and the radius of curvature of 10878.1 is the least curved case.

Table 8: Parametric Study Results for 3D FEA Model

	Vertical Stress (in MPa)			
	Unidirectional		Bidirectional	
RoC (mm)	Curved	Straight	Middle	Corner
1208.7	-15.3	26.9	-18.8	44.9
1359.8	-13.5	25.2	-16.7	40.2
1554.1	-11.7	21.5	-14.8	35.8
1813.1	-10.0	19.2	-12.7	30.9
2175.7	-8.3	16.1	-10.7	26.2
2719.6	-6.6	12.9	-8.5	21.4
3626.1	-4.9	9.7	-6.5	16.1
5439.1	-3.2	6.6	-4.3	11.0
10878.1	-1.6	3.3	-2.3	5.3
-10878.1	1.5	-3.0	2.3	-4.6
-5439.1	3.2	-5.4	4.2	-8.8
-3626.1	4.7	-7.4	6.4	-12.0
-2719.6	6.2	-9.5	8.4	-15.2
-2175.7	8.0	-11.4	10.6	-17.9
-1813.1	9.5	-13.0	12.4	-20.5
-1554.1	11.0	-15.1	14.6	-23.1
-1359.8	12.3	-16.8	16.4	-25.6
-1208.7	14.5	-18.7	19.5	-28.3

For the unidirectional case, these extreme pre-stresses occur in the middle interconnect on the curved side of the package and the corner interconnect on the straight side of the package. Due to the symmetric nature of the bidirectional curvature model, the pre-stresses are the same on all sides, and the extreme pre-stresses occur in the middle and corner interconnects. The results from Table 8 are presented visually in Figures 23 and 24. Figure 23 shows the unidirectional curvature results, and Figure 24 shows the bidirectional curvature results.

Figure 23: Plot of Pre-Stress versus Radius of Curvature of PWB for Unidirectional Curvature

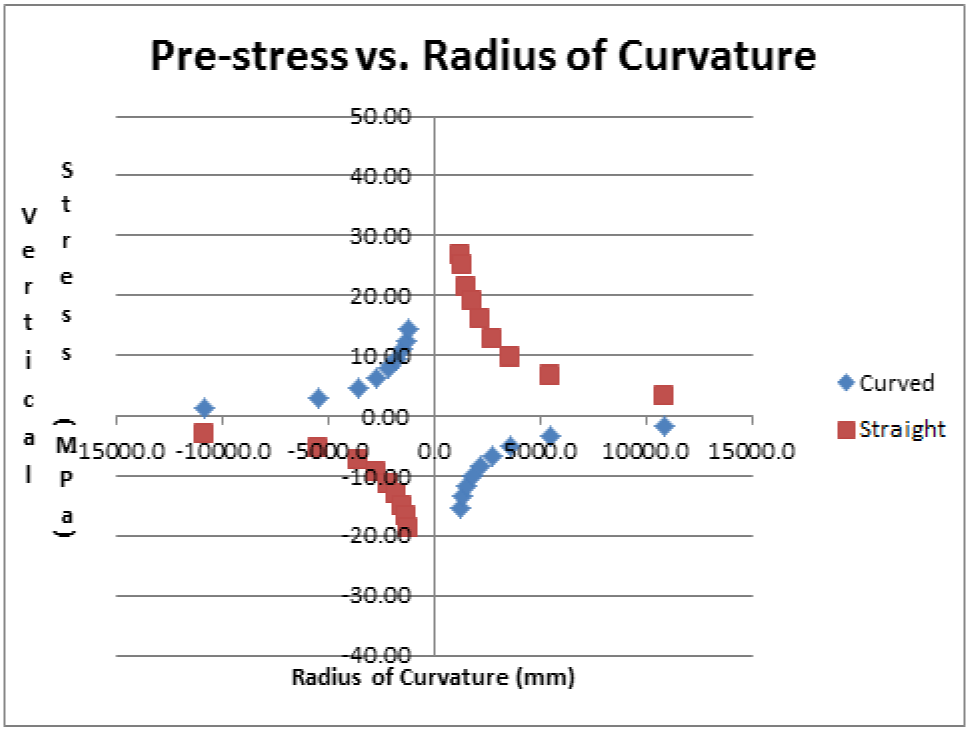
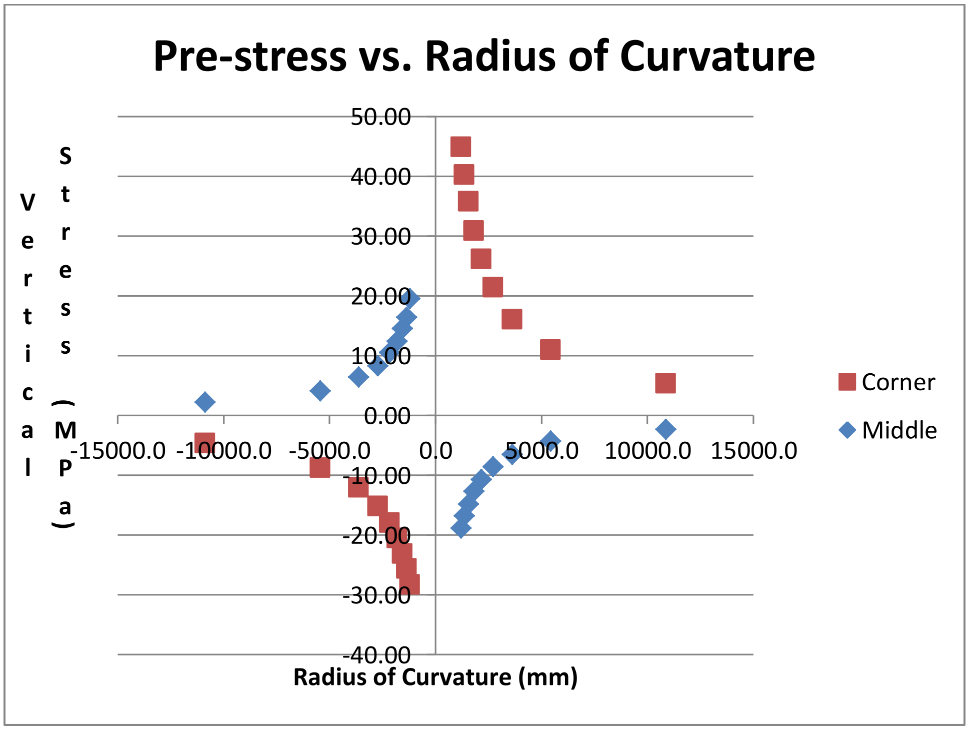


Figure 24: Plot of Pre-Stress versus Radius of Curvature of PWB for Bidirectional Curvature



By looking at these graphs, two features become clear. The first is that in both cases, the corner interconnect has a higher magnitude pre-stress than the middle interconnect. The second is that the bidirectional curvature case results in higher overall pre-stresses than the unidirectional curvature case.

Comparison of Perimeter-Leaded Simplified and FEA Model Results

In order to determine if the three-dimensional simplified model is accurate, its results must be compared to the results of the three-dimensional FEA model. Figure 25 shows a plot of the interconnect pre-stresses along the component for unidirectional curvature as predicted by the simplified model. Figure 26 shows a similar plot for the bidirectional curvature case. Figure 27 is a plot of the interconnect pre-stresses under unidirectional curvature for the FEA model. Figure 28 is a plot for the FEA model for the bidirectional curvature case. All of these plots are for concave up curvature.

Figure 25: Plot of Simplified Model

Interconnect Pre-Stresses for Unidirectional Curvature

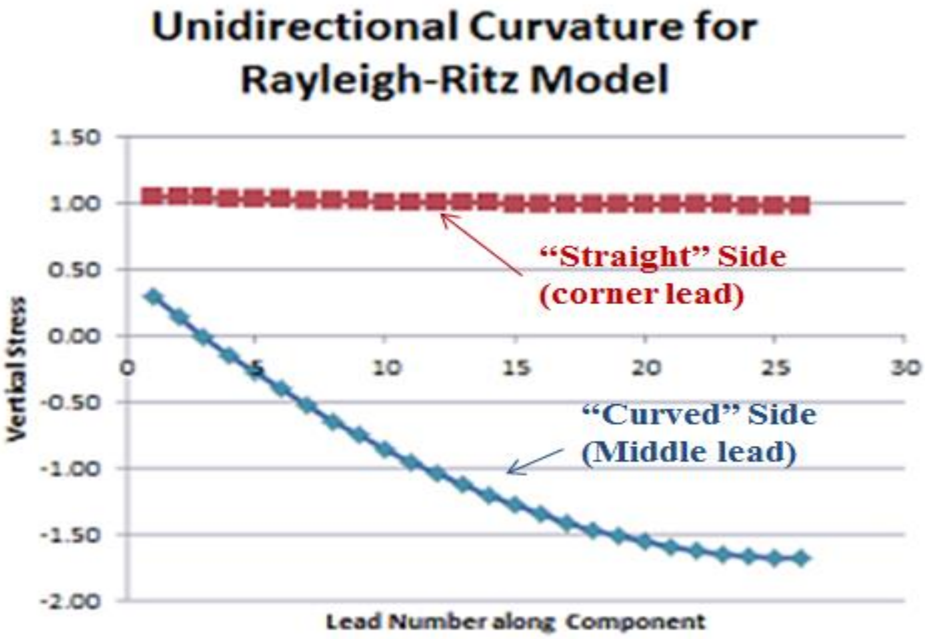


Figure 26: Plot of Simplified Model

Interconnect Pre-Stresses for Bidirectional Curvature

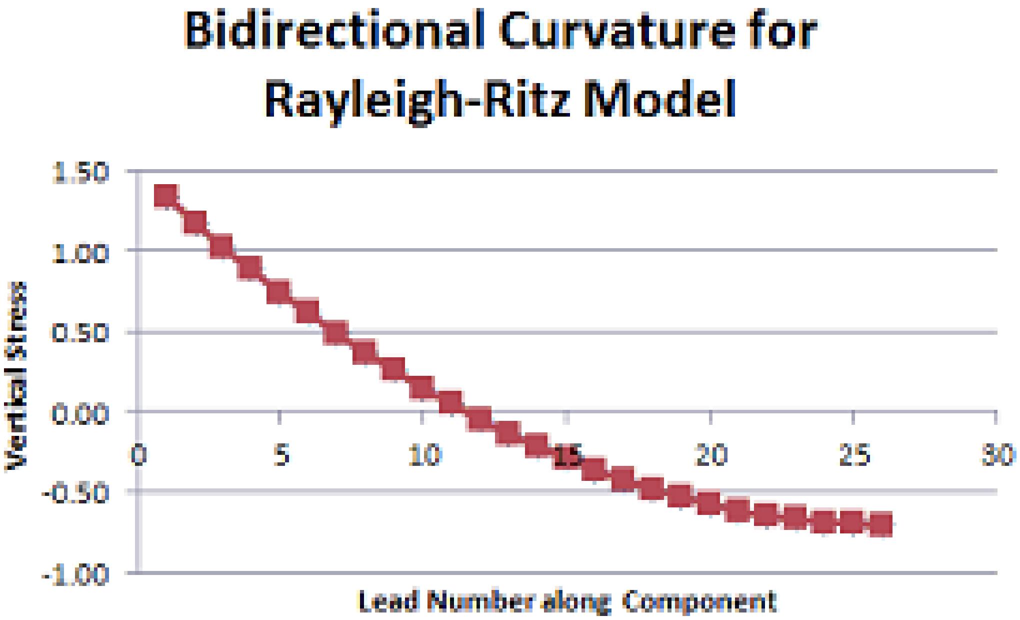


Figure 27: Plot of FEA Model Interconnect Pre-Stresses for Unidirectional Curvature

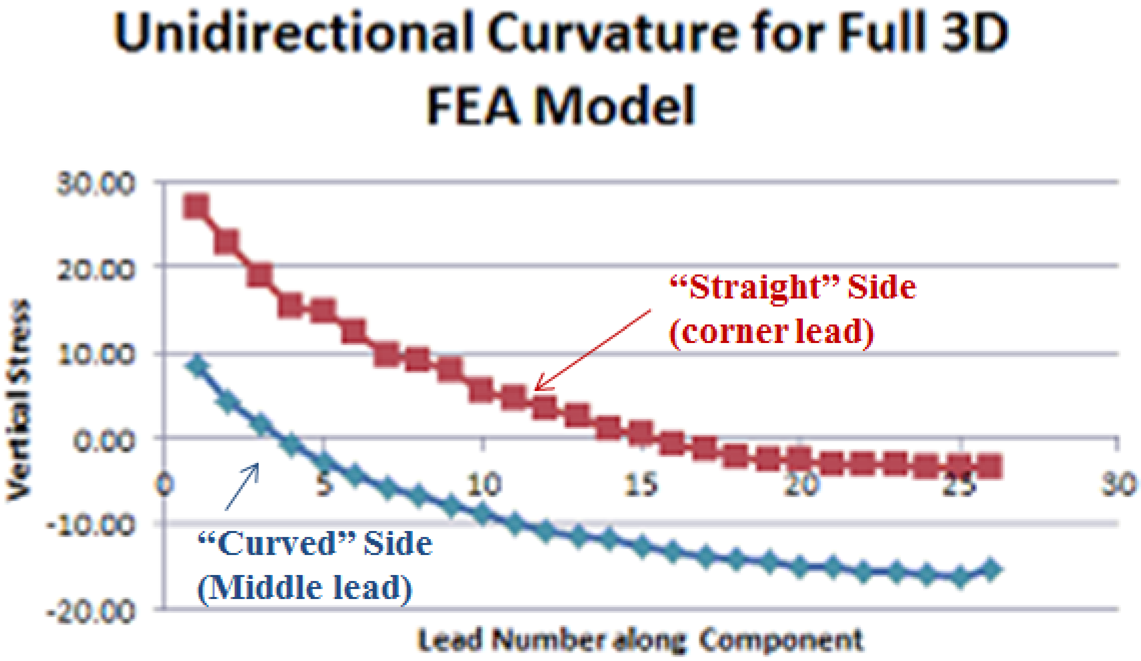
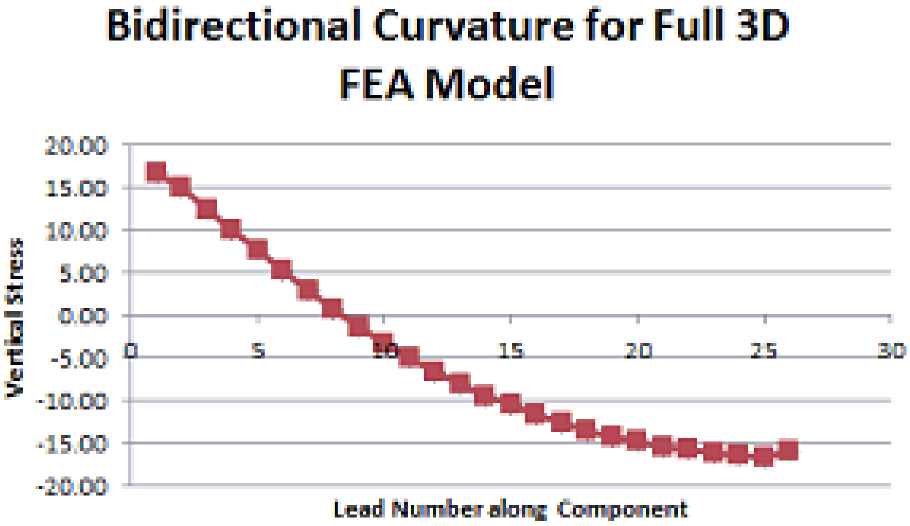


Figure 28: Plot of FEA Model Interconnect Pre-Stresses for Bidirectional Curvature



These plots show some similarities and some differences between the simplified and FEA models. Both of these three-dimensional models show variability in the interconnect pre-stresses along the straight side of the component in the unidirectional curvature case. In addition to this, the interconnect pre-stresses exhibit similar trends along the component for the bidirectional curvature case. However, there are major differences between the two models. First, the magnitude of the pre-stress variability in the interconnects along the straight side of the component is much greater for the FEA model. This means the anticlastic curvature is more significant in the FEA model. Also, the overall magnitudes of the interconnect pre-stresses are much greater for the FEA model.

Figure 29 shows the maximum and minimum pre-stresses predicted by the simplified model versus initial PWB curvature for unidirectional curvature, while Figure 30 shows the same predictions for bidirectional curvature. Figure 31 shows the maximum and minimum pre-stresses from the FEA model versus initial PWB curvature for unidirectional curvature, and a similar plot is shown in Figure 32 for bidirectional curvature.

Figure 29: Plot of Interconnect Pre-Stress vs. Unidirectional PWB Curvature for Simplified Model

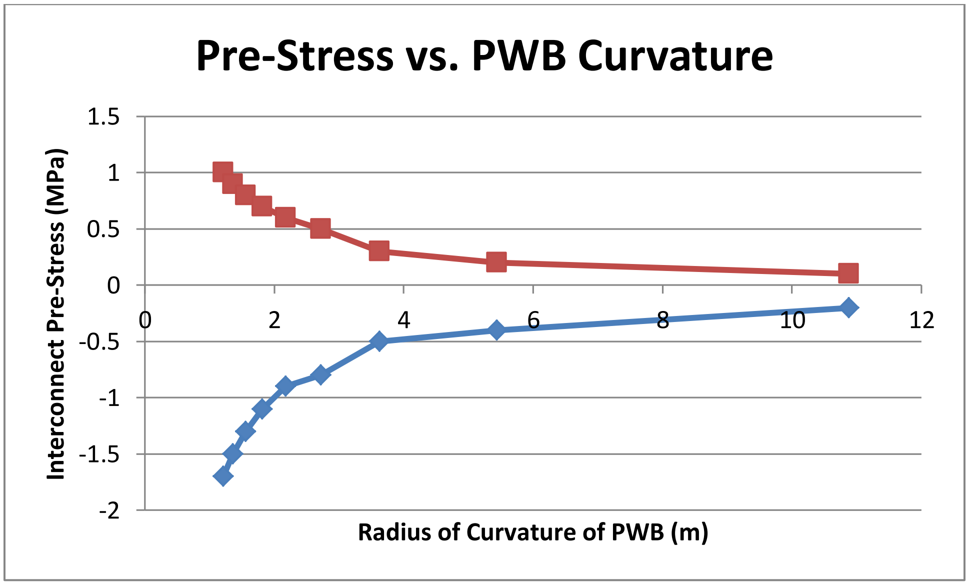


Figure 30: Plot of Interconnect Pre-Stress vs. Bidirectional PWB Curvature for Simplified Model

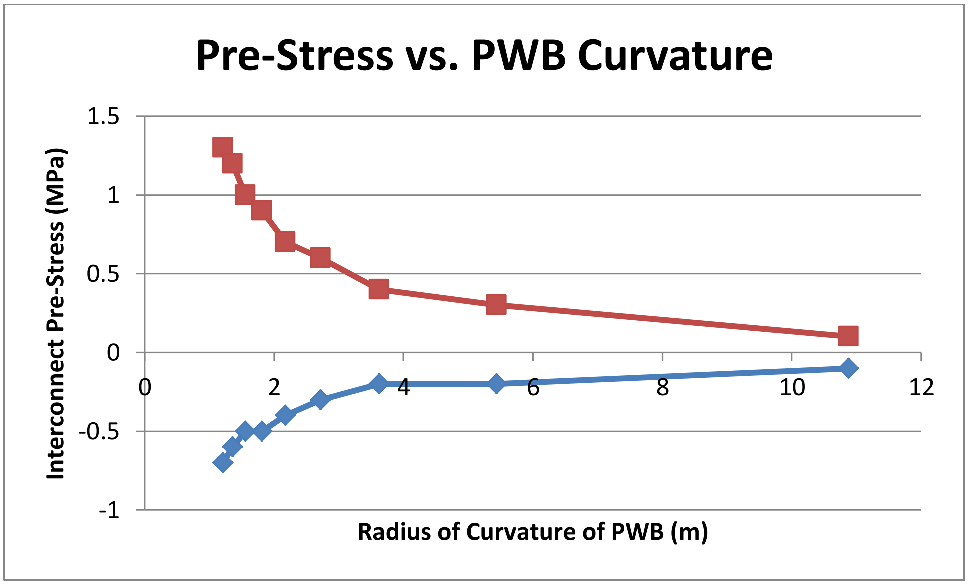


Figure 31: Plot of Interconnect Pre-Stress vs. Unidirectional PWB Curvature for FEA Model

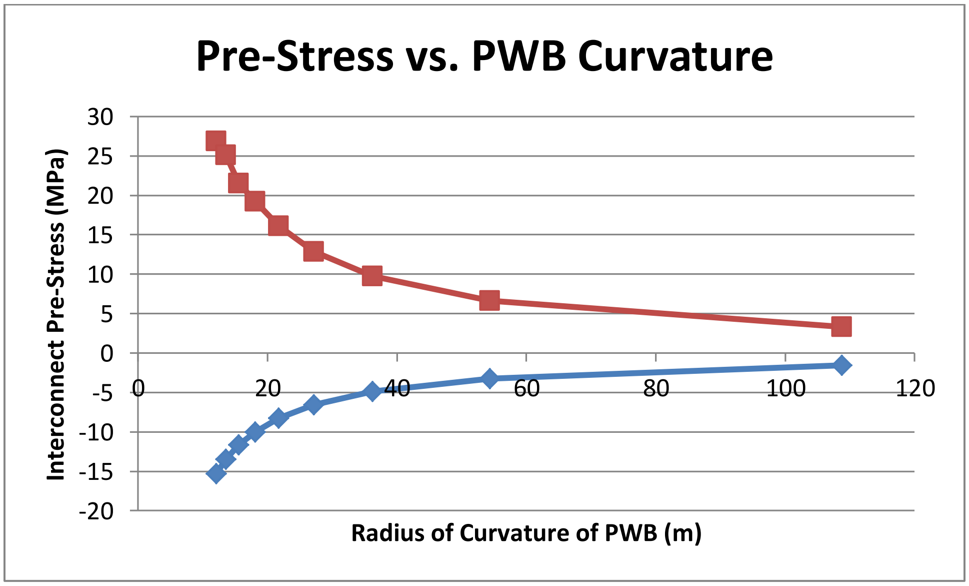
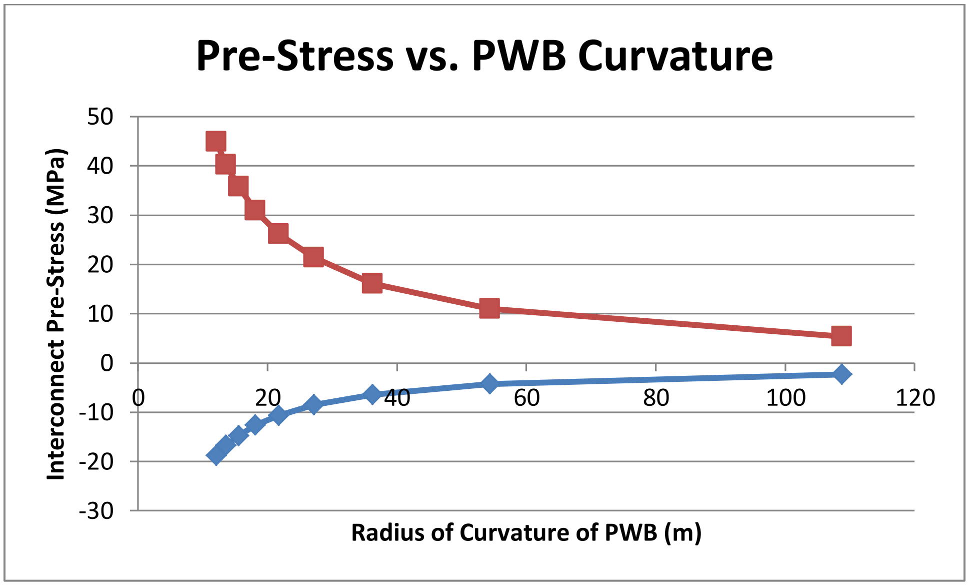


Figure 32: Plot of Interconnect Pre-Stress vs. Bidirectional PWB Curvature for FEA Model



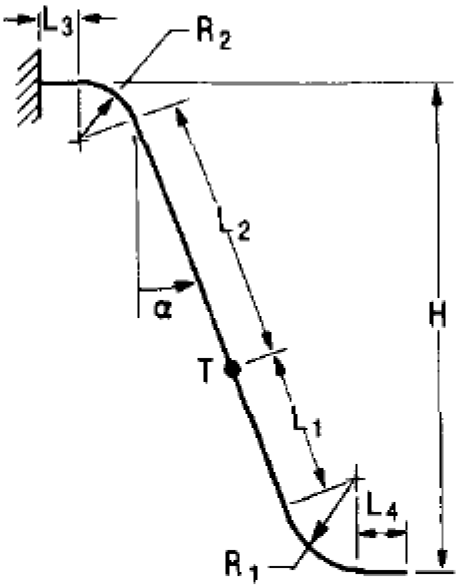
These plots reinforce the conclusions from the previous results. Once again, the pre-stresses from both models exhibit similar general trends, but the magnitudes of the pre-stresses are much greater for the FEA model.

The pre-stresses the simplified model predicts do not come close to those of the FEA model. This means that there is still some work to be done to get the simplified model to agree with FEA for perimeter-leaded components. Future work on this project can be dedicated to working on determining what exactly is causing the discrepancy between the simplified and FEA models and adjusting the simplified model to include it. This will improve the versatility of the general simplified model.

Appendix 2: Kotlowitz Equations and Definitions of Variables and Constants

Before the equations, variables, and constants can be presented, the geometric variables represented by the different dimensions of the gull-wing lead must be defined. Figure 33 shows a gull-wing lead and labels the important dimensions.

Figure 33: Gull-Wing Lead Geometry Variables (Kotlowitz and Taylor, 1991)



For the gull-wing leads in this study, the point T is in the center of the lead. This means that L_1 and L_2 are equal. The constants used throughout Kotlowitz's equations are presented in Equations 22-27.

$$C_1 = H - R_2 \qquad (Equation\ 22)$$

$$C_2 = .7854 - \frac{\alpha}{2} + \frac{1}{4}\sin(2\alpha) \qquad (Equation\ 23)$$

$$C_3 = .7854 - \frac{\alpha}{2} - \frac{1}{4}\sin(2\alpha) \qquad (Equation\ 24)$$

$$C_4 = L_4 \sin(\alpha) + R_1 \cos(\alpha) \qquad (Equation\ 25)$$

$$C_5 = L_4 \cos(\alpha) + R_1(1 - \sin(\alpha)) \qquad (Equation\ 26)$$

$$C_6 = (L_1 + L_2) \sin(\alpha) + (R_1 + R_2) \cos(\alpha) + L_4 \qquad (Equation\ 27)$$

The flexural spring constant in the vertical direction is now presented. For this model, the vertical direction is the y-direction, but in Kotlowitz's models the vertical direction is the z-direction. The vertical flexural spring constant for the gull wing lead is presented in Equation 28, and the equations for the variables in the flexural spring constant equation are presented in Equations 29-32.

$$\frac{1}{K_z} = \frac{1}{E}\left[\frac{S_5}{I_1} + \frac{S_6}{I_2}\right] + \frac{1}{E}\left[\frac{S_{15}}{A_1} + \frac{S_{16}}{A_2}\right] \qquad (Equation\ 28)$$

$$S_5 = \frac{1}{3}[L_4^3 + L_1^3 sin^2(\alpha)] + R_1[L_4^2(1.5708 - \alpha) + 2L_4R_1(1 - \sin(\alpha)) + R_1^2C_3]$$
$$+ L_1[(L_4 + R_1 \cos(\alpha))^2 + L_1 \sin(\alpha)(L_4 + R_1 \cos(\alpha))] \qquad (Equation\ 29)$$

$$S_6 = \frac{1}{3}L_2^3 sin^2(\alpha) + R_2^3C_3 + R_2C_6^2(1.5708 - \alpha) - 2R_2^2C_6(1 - \sin(\alpha)) + \frac{1}{3}[(C_6 + L_3)^3 - C_6^3]$$
$$+ L_2(L_4 + R_1 \cos(\alpha) + L_1 \sin(\alpha))^2$$
$$+ L_2^2 \sin(\alpha)(L_4 + R_1 \cos(\alpha) + L_1 \sin(\alpha)) \qquad (Equation\ 30)$$

$$S_{15} = L_1 cos^2(\alpha) + R_1C_3 + f\frac{E}{G}(L_1 sin^2(\alpha) + L_4 + R_1C_2) \qquad (Equation\ 31)$$

$$S_{16} = L_2 cos^2(\alpha) + C_3R_2 + f\frac{E}{G}(L_2 sin^2(\alpha) + L_3 + C_2R_2) \qquad (Equation\ 32)$$

These are the only equations from Kotlowitz's work used in this study. The full set of equations, constants, and variables can be found in his paper (Kotlowitz and Taylor, 1991).

Appendix 3: Full Code for the Area Array Simplified Model

The following MATLAB functions comprise the three-dimensional area array simplified model. The model consists of a series of smaller programs which each perform a specific function and are referenced by a master program. Comments have been included to help explain exactly what each part of the code is doing.

The first program presented is the master program, called fullmodel.m. To run this program, the user calls this program and enters all of the required inputs. Each input is explained in the code's comments. This program calls each subprogram in order to fully run the simplified model and outputs the final results.

```
function [sigma,varvalues_pf,U] =
fullmodel(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_bf,varvalues_p,h_min
,h_lead,ll,el_b,el_p,A_s,E_s,E_p,h_p,v_p,E_lead,A_lead,p_s,p_lead,r,h_b)

% This function is an simplified model which determines the pre-stresses
% present in the solder joints after a straightening process has occurred
% Inputs:
% Eq_b = equation of board displacement field
% Eq_p = equation of package displacement field
% vars_b = list of variables in board equation (symbolic)
% vars_p = list of variables in package equation (symbolic)
% varvalues_b0 = initial values for variables in board equation
% varvalues_bf = final values for variables in board equation
% varvalues_p = initial values for variables in package equation
% h_min = minimum solder joint height
% h_lead = lead height
% ll = coordinates of leads along package base
% el_b = horizontal distance of ends of board
% el_p = horizontal distance of ends of package face
% *Note: For horizontal distances, (0,0) is taken as center of package*
% A_s = crosssectional area of the solder
% E_s = Elastic Modulus of the solder
% E_p = Elastic Modulus of package
% h_p = height of package
% v_p = Poisson's Ratio of package
% E_lead = Elastic Modulus of leads
% A_lead = cross sectional area of leads
% r = radius of interconnect
% h_p = height of board
% *Note: Lead dimensions are measrued on lead centerline*

syms x y z A1 B1
```

```
% Step 0: Solve for A0
varvalues_p0 =
computeA0(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_p,h_min,h_lead,ll);

% Place package variables in alphabetical order (necessary when solving for
% final values)
pvars = [vars_p;varvalues_p0];
pvars = transpose(pvars);
pvars = sortrows(pvars);
pvars = transpose(pvars);
vars_p = pvars(1,:);
varvalues_p0 = pvars(2,:);

% Step 1: Calculate Interconnect Lengths
hi = calcjointheights(Eq_b,vars_b,varvalues_b0,Eq_p,vars_p,varvalues_p0,ll);

% Step 2: Calculate Interconnect Stiffness
kl = calcstiffnessmatrix(h_lead,A_lead,E_lead,r);
ks = calcstiffnessmatrix(hi-h_lead,A_s,E_s,r);
k_axial = getaxialstiffness(kl,ks);
klarge = combinestiffnessmatrix(kl,ks);

% Step 3: Calculate Flexural Rigidity of Package
D_p = (E_p*h_p^3)/(12*(1-v_p^2));
% Step 4: Compute Final Strain Energies
Upf = calcpackagestrainenergy(Eq_p,vars_p,varvalues_p0,D_p,v_p,el_p,2);
Ulf =
calcinterconnectstrainenergy4(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_
bf,varvalues_p0,klarge,ll,h_b,h_p);

% Step 5: Develop PE Equation
PE = (Upf+Ulf);

% Step 6: Minimize PE Equation
varvalues_pf = minimizePE2(PE,vars_p);
U = solvestrainenergy(PE,vars_p,varvalues_pf); % Calc. final strain energy

% Step 7: Plot Inital & Final Curvatures of Board & Package
plotcurvature(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varvalues_p0,
varvalues_pf,el_b,el_p);

% Step 8: Determine Final Interconnect Lengths
hf =
calcfinaljointheights(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varva
lues_p0,varvalues_pf,ll,hi);

% Step 9: Determine Interconnect Forces
F = calcforces(hf,hi,k_axial);

% Step 10: Calculate Stresses in Solder Joints
sigma = calcstresses(F,A_s);
sigmap = sigma(1:50);
sigman = sigma(51:end);
sigmapm =
[sigmap(1:5),sigmap(6:10),sigmap(11:15),sigmap(16:20),sigmap(21:25),sigmap(26
:30),sigmap(31:35),sigmap(36:40),sigmap(41:45),sigmap(46:50)];
```

```
sigmanm =
[sigman(1:5),sigman(6:10),sigman(11:15),sigman(16:20),sigman(21:25),sigman(26
:30),sigman(31:35),sigman(36:40),sigman(41:45),sigman(46:50)];
sigma = [sigmapm;sigmanm];
```

The first step in the process of predicting the interconnect pre-stresses is to compute the value of A0 for the initial displacement field governing the component package. This value can be provided as an input by the user or determined by the model. The program computeA0.m determines if the value is provided and calculates A0 if it is not provided.

```
function varvalues_p =
computeA0(Eq_b,Eq_p,vars_b,vars_p,varvalue_b,varvalue_p,h_min,h_lead,ll)

% This function computes A0, the separation between the package and the
% board for given vales of the other variables

% Inputs:
% Eq_b = equation of board displacement field
% Eq_p = equation of package displacement field
% vars_b = list of variables in board equation (symbolic)
% vars_p = list of variables in package equation (symbolic)
% varvalue_b = values for variables in board equation
% varvalue_p = values for variables in package equation
% h_min = minimum solder joint height
% h_lead = lead height
% ll = lead locations (coordinates) on package base
% *Note: For horizontal distances, (0,0) is taken as center of package*

% Outputs:
% A0 = value for A0 (constant in package displacement field equation)
% varvalues_p = values for variables in package equation

syms x y z
[n,~] = size(transpose(vars_b)); % n = # of board variables
[m,~] = size(transpose(vars_p)); % m = # of package variables
[o,~] = size(transpose(varvalue_b)); % o = # of board variables provided
[p,~] = size(transpose(varvalue_p)); % p = # of package variables provided

if n==o
else
    disp('Number of board variables provided does not match number of
variables listed');
end

if m==p % A0 is predefined
    varvalues_p = varvalue_p;
elseif m==p+1 % A0 must be calculated
    % Replace symbolic variables with actual values
    for i=1:n
        Eq_b = subs(Eq_b,vars_b(i),varvalue_b(i));
    end

    for i=1:p
        Eq_p = subs(Eq_p,vars_p(i+1),varvalue_p(i));
    end

    Eq_p = subs(Eq_p,vars_p(1),0); % Removes A0 from equation so equations
    % will not be symbolic

    [n1,n2] = size(ll);
    d = zeros(n1,n2/2); % Creates a dummy matrix of the correct size which
    % will be filled in with the distances between the board and package at
    % the lead locations

    % Substitute lead locations (z,x) into displcament field equations to
    % find distance between the board and package
    m=1;
    counter=1;
    llz = zeros(n1,n2/2);
    llx = zeros(n1,n2/2);
    while m<=n2
        llz(:,counter) = ll(:,m);
        llx(:,counter) = ll(:,m+1);
        m=m+2;
        counter=counter+1;
    end

    for i=1:n1
        for j=1:n2/2
            EQ1 = Eq_b;
            EQ1 = subs(EQ1,z,llz(i,j));
            EQ1 = subs(EQ1,x,llx(i,j));
            EQ2 = Eq_p;
            EQ2 = subs(EQ2,z,llz(i));
            EQ2 = subs(EQ2,x,llx(i));
            d(i,j) = EQ2-EQ1;
        end
    end

    minval = min(min(d)); % Determines the minimum distance between the
    % package and board (most negative or least positive distance)

    A0=-minval+h_min+h_lead; % A0 is set so the smallest interconnect
    % length is fixed as the minimum joint height plus the lead height

    varvalues_p = [A0,varvalue_p]; % Adds value of A0 to list of variable
    % values for package displacement field equation
else
    disp('Number of package variables provided does not match number of
variables listed.  Only the constant coefficient can be undefined.');
end
```

After A0 is determined, the list of variables and variable titles for the component package are put into alphabetical order. This is done so these variables are in the same order as the outputs of the potential energy minimization program which will be explained later. The next step is to calculate the equivalent heights of the interconnects. This is done by the program calcjointheights.m.

```
function
h=calcjointheights(Eq_b,vars_b,varvalues_b,Eq_p,vars_p,varvalues_p,ll)

% This function is used to calculate the interconnect heights along the
% component based on the curvature of the board and package

% Inputs:
% Eq_b = governing equation for displacement filed of board
% vars_b = variables in board equation
% varvalues_b = values for variables in board equation
% Eq_p = governing equation for displacement filed of package
% vars_p = variables in package equation
% varvalues_p = values for variables in package equation
% ll = horizontal distance of leads along package face
% el_b = horizontal distance of ends of board
% el_p = horizontal distance of ends of package face
% *Note: For horizontal distances, (0,0) is taken as center of package*

% Outputs:
% h = interconnect lengths

syms x y z
[n ~] = size(transpose(vars_b)); % n = # of variables in board equation
[m ~] = size(transpose(vars_p)); % m = # of variables in package equation
Eqb = Eq_b;
Eqp = Eq_p;

% Replace symbolic variables with actual variable values
for i=1:n
    Eqb = subs(Eqb,vars_b(i),varvalues_b(i));
end

for i=1:m
    Eqp = subs(Eqp,vars_p(i),varvalues_p(i));
end

[n1 n2] = size(ll);
h = zeros(n1,n2/2); % Creates dummy height matrix of the correct size to be
% filled in later (first row is front z face, second row is rear z face,
% thrid row is right x face, fourth row is left x face

% Determine height by substituting lead locations (z,x) into displacement
% fields and subtracting the height of the package from the height of the
% board
m=1;
counter=1;
llz = zeros(n1,n2/2);
llx = zeros(n1,n2/2);
while m<=n2
    llz(:,counter) = ll(:,m);
    llx(:,counter) = ll(:,m+1);
    m=m+2;
    counter=counter+1;
end
```

```
for i=1:n1
    for j=1:n2/2
        Eq1 = Eqb;
        Eq1 = subs(Eq1,z,llz(i,j));
        Eq1 = subs(Eq1,x,llx(i,j));
        Eq2 = Eqp;
        Eq2 = subs(Eq2,z,llz(i,j));
        Eq2 = subs(Eq2,x,llx(i,j));
        h(i,j) = Eq2-Eq1;
    end
end
```

After calculating the heights of the joints, the next step is to determine stiffness of interconnect. This process has a few parts. The first is to determine the stiffness matrix of the lead and solder joints. This is done by the program calcstiffnessmatrix.m.

```
function k = calcstiffnessmatrix(l,A,E,r)

% This function calculates the stiffness matrix for the interconnect based
% on the equivalent interconnect stiffness

% Inputs:
% A = Cross sectional area
% E = Elastic modulus
% r = Radius
% l = Length

[n1 n2] = size(l);

for i=1:n1
    for j=1:n2
        Ia(i,j) = pi*.25*r^4;
    end
end

k = zeros(10,10,n1*n2);

for i=1:n1
    for j=1:n2
        num = n2*(i-1)+j;
        k(1,1,num) = A*E/l(i,j);
        k(1,6,num) = -A*E/l(i,j);
        k(2,2,num) = 12*E*Ia(i,j)/l(i,j)^3;
        k(2,5,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(2,7,num) = -12*E*Ia(i,j)/l(i,j)^3;
        k(2,10,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(3,3,num) = 12*E*Ia(i,j)/l(i,j)^3;
        k(3,4,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(3,8,num) = -12*E*Ia(i,j)/l(i,j)^3;
        k(3,9,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(4,3,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(4,4,num) = 4*E*Ia(i,j)/l(i,j);
        k(4,8,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(4,9,num) = 2*E*Ia(i,j)/l(i,j);
        k(5,2,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(5,5,num) = 4*E*Ia(i,j)/l(i,j);
        k(5,7,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(5,10,num) = 2*E*Ia(i,j)/l(i,j);
        k(6,1,num) = -A*E/l(i,j);
        k(6,6,num) = A*E/l(i,j);
        k(7,2,num) = -12*E*Ia(i,j)/l(i,j)^3;
        k(7,5,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(7,7,num) = 12*E*Ia(i,j)/l(i,j)^3;
        k(7,10,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(8,3,num) = -12*E*Ia(i,j)/l(i,j)^3;
        k(8,4,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(8,8,num) = 12*E*Ia(i,j)/l(i,j)^3;
        k(8,9,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(9,3,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(9,4,num) = 2*E*Ia(i,j)/l(i,j);
        k(9,8,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(9,9,num) = 4*E*Ia(i,j)/l(i,j);
        k(10,2,num) = 6*E*Ia(i,j)/l(i,j)^2;
        k(10,5,num) = 2*E*Ia(i,j)/l(i,j);
        k(10,7,num) = -6*E*Ia(i,j)/l(i,j)^2;
        k(10,10,num) = 4*E*Ia(i,j)/l(i,j);
    end
end
```

The next part of the process is to determine the equivalent axial stiffness of the interconnects. This is done using the series spring combination equation to combine the axial stiffness terms for the lead and solder joint. This is done by the function getaxialstiffness.m.

```
function k_axial = getaxialstiffness(kl,ks)

% kl = stifness matrix for lead
% ks = stifness matrix for solder

[i,j,l] = size(ks);

% Axial stiffness is first entry of matrix
kl_axial = kl(1,1);
ks_axial = ks(1,1,:);

k_axial = zeros(l,1);

% Series spring combination: Kt = 1/((1/k1)+(1/K2))
for a=1:l
    k_axial(a) = ((1/kl_axial)+(1/ks_axial(a)))^(-1);
end
```

The final part of the process of determining the interconnect stiffness is to get the equivalent interconnect matrix from the lead and solder stiffness matrices. This is done by the program combinestiffnessmatrix.m.

```
function klarge = combinestiffnessmatrix(kl,ks)

% kl = stifness matrix for lead
% ks = stifness matrix for solder
```

```
[a b c] = size(ks);
klarge = zeros(15,15,c);

% Assemble global interconnect stiffness matrix from local element matrices
for i=1:10
    for j=1:10
        for l=1:c
            klarge(i,j,l) = klarge(i,j,l)+kl(i,j,1);
        end
    end
end

for i=6:15
    for j=6:15
        for l=1:c
            klarge(i,j,l) = klarge(i,j,l)+ks(i-5,j-5,l);
        end
    end
end
```

With the interconnect stiffness determined, the next step in the analytic model process if to calculate the flexural rigidity of the component package. This is done directly in the fullmodel.m code using the flexural rigidity formula. After this is done, the next step in the process is to calculate the strain energies of the package and interconnects. The strain energy of the package is determined by the program calcpackagestrainenergy.m .

```
function Up = calcpackagestrainenergy(wp,vars_p,varvalues_p,Dp,vp,el,step)

% This function calculates the strain energy of the package

% Inputs:
% wp = displacemnt field function
% vars_p = list of variables in displacement field function (excludes A0)
% varvalues = list of values for variables
% Dp = flexural rigidity of package
% vp = package's Poisson ratio
% el = horizontal locatios of edges of package face
% step = 1 for initial strain energy calc, 2 for final strain energy calc

% Outputs:
% Up = package strain energy

syms x y z
[n ~] = size(transpose(vars_p)); % n gives number of variables

% Define Up equation to be integrated
Eq = (Dp/2)*(diff(wp,z,2)+diff(wp,x,2))^2-Dp*(1-
vp)*(diff(wp,z,2)*diff(wp,x,2)-diff(diff(wp,z),x)^2);

% Substitute actual variable values into equation for symbolic variables if
% they are known
if step == 1
```

```
    for i=1:n
        Eq = subs(Eq,vars_p(i),varvalues_p(i));
    end
end

% Integrate Up equation
Up = int(int(Eq,z,el(1),el(2)),x,el(1),el(2));

if step == 1
    Up = double(Up); % Convert symbolic var to number if all vars are known
end
```

The strain energy of the interconnects is determined by the program calcinterconnectstrainenergy4.m.

```
function Ul =
calcinterconnectstrainenergy4(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_
bf,varvalues_p,klarge,ll,h_b,h_p)

% This functon calculates the strain energy present in the interconnects
% after deformation

% Inputs:
% Eq_b = displacement field for board
% Eq_p = displacement field for package
% vars_b = variables in board equation
% vars_p = variables in packgae equation
% varvalues_bf = values for variables in board equation
% ki = interconnect stiffness
% ll = horizontal lead location along package face
% h_p = height of package
% h_p = height of board

% Outputs:
% Ul = lead strain energy

syms x y z
[n1 n2] = size(ll);
[m ~] = size(transpose(vars_b)); % m = # variables in board equation
[o ~] = size(transpose(vars_p)); % o = # variables in package equation
Eqb0 = Eq_b; % Initial board equation
Eqbf = Eq_b; % Final board equation
Eqp0 = Eq_p; % Initial package equation
Eqpf = Eq_p; % Final package equation

% Replace symbolic variables with actual variable values
for i=1:m
    Eqb0 = subs(Eqb0,vars_b(i),varvalues_b0(i));
    Eqbf = subs(Eqbf,vars_b(i),varvalues_bf(i));
end

for i=1:o
    Eqp0 = subs(Eqp0,vars_p(i),varvalues_p(i));
end
```

```
u1 = sym('t',[15 n1*n2/2]); % Creates dummy strain energy matrix of correct
% size to be filled in later

u2 = sym('t',[15 n1*n2/2]); % Creates dummy strain energy matrix of correct
% size to be filled in later

u3 = sym('t',[15 n1*n2/2]); % Creates dummy strain energy matrix of correct
% size to be filled in later

% Determine lead displacement by solving all 4 equations at each lead
% location (z,x)
m=1;
counter=1;
llz = zeros(n1,n2/2);
llx = zeros(n1,n2/2);
while m<=n2
    llz(:,counter) = ll(:,m);
    llx(:,counter) = ll(:,m+1);
    m=m+2;
    counter=counter+1;
end

counter=1;
for i=1:n1
    for j=1:n2/2
        Eq1 = Eqb0;
        Eq1 = subs(Eq1,z,llz(i,j));
        Eq1 = subs(Eq1,x,llx(i,j));
        Eq1x = diff(Eqb0,x);
        Eq1x = subs(Eq1x,x,llx(i,j));
        Eq1x = subs(Eq1x,z,llz(i,j));
        Eq1z = diff(Eqb0,z);
        Eq1z = subs(Eq1z,z,llz(i,j));
        Eq1z = subs(Eq1z,x,llx(i,j));
        Eq2 = Eqbf;
        Eq2 = subs(Eq2,z,llz(i,j));
        Eq2 = subs(Eq2,x,llx(i,j));
        Eq2x = diff(Eqbf,x);
        Eq2x = subs(Eq2x,x,llx(i,j));
        Eq2x = subs(Eq2x,z,llz(i,j));
        Eq2z = diff(Eqbf,z);
        Eq2z = subs(Eq2z,z,llz(i,j));
        Eq2z = subs(Eq2z,x,llx(i,j));
        Eq3 = Eqp0;
        Eq3 = subs(Eq3,z,llz(i,j));
        Eq3 = subs(Eq3,x,llx(i,j));
        Eq3x = diff(Eqp0,x);
        Eq3x = subs(Eq3x,x,llx(i,j));
        Eq3x = subs(Eq3x,z,llz(i,j));
        Eq3z = diff(Eqp0,z);
        Eq3z = subs(Eq3z,z,llz(i,j));
        Eq3z = subs(Eq3z,x,llx(i,j));
        Eq4 = Eqpf;
        Eq4 = subs(Eq4,z,llz(i,j));
        Eq4 = subs(Eq4,x,llx(i,j));
        Eq4x = diff(Eqpf,x);
        Eq4x = subs(Eq4x,x,llx(i,j));
        Eq4x = subs(Eq4x,z,llz(i,j));
        Eq4z = diff(Eqpf,z);
        Eq4z = subs(Eq4z,z,llz(i,j));
        Eq4z = subs(Eq4z,x,llx(i,j));

u1(:,counter)=[Eq4;Eq4x*h_p/2;Eq4z*h_p/2;Eq4z;Eq4x;0;0;0;0;0;Eq2;Eq2x*h_b/2;E
q2z*h_b/2;Eq2z;Eq2x];

u2(:,counter)=[Eq3;Eq3x*h_p/2;Eq3z*h_p/2;Eq3z;Eq3x;0;0;0;0;0;Eq1;Eq1x*h_b/2;E
q1z*h_b/2;Eq1z;Eq1x];
        u3(:,counter) = u1(:,counter)-u2(:,counter);
        counter=counter+1;
    end
end

% Solve Q=ku equation using essential boundary conditions
% This gives the displacements ustar for the solder/lead interface
kstar = klarge;
Q = zeros(15,n1*(n2/2));
rhs = sym(Q);

for i=1:5
    for l = 1:(n1*(n2/2))
        rhs(:,l) = rhs(:,l)-kstar(:,1,l)*u3(i,l);
    end
    rhs = rhs(2:end,:);
    kstar = kstar(2:end,2:end,:);
end
for i=1:5
    for l = 1:(n1*(n2/2))
        rhs(:,l) = rhs(:,l)-kstar(:,end,l)*u3(16-i,l);
    end
    rhs = rhs(1:end-1,:);
    kstar = kstar(1:end-1,1:end-1,:);
end

ustar = zeros(5,n1*(n2/2));
ustar = sym(ustar);

for l = 1:(n1*(n2/2))
    ustar(:,l) = inv(kstar(:,:,l))*rhs(:,l);
end

for i = 1:(n1*(n2/2))
    for j= 1:5
        u3(5+j,i) = ustar(j,i);
    end
end

% determine strain energy in each lead
U = zeros(n1*n2/2,1); % Creates dummy strain energy matrix
U = sym(U);
```

```
for i=1:(n1*(n2/2))
    U(i) = .5*transpose(u3(:,i))*klarge(:,:,i)*u3(:,i);
end


% Develop final strain energy equation by summing all leads together
Ul = sum(sum(U));
```

Both the package and interconnect strain energies will be in terms of the unknown variables, A, B, and A0, which govern the final package curvature. After all of the strain energies are calculated, the next step is to determine the potential energy equation of the entire system. This is done by summing the strain energies of the package and interconnects in the fullmodel.m code. With the potential energy equation known, the next step is to minimize the potential energy equation to determine the values of the final package values. This is done by the function minimizePE2.m.

```
function [vars_pf,vars_p] = minimizePE2(PE,vars_p)

% This function solves for the unknown coefficients of the package and
% interconnect displacement fields by minimizing the PE equation

% Inputs:
% PE = potential energy equation
% vars_p = list of variables in package equation (excludes A0)

% Output:
% vars_pf = final values for package displacement field variables

[n ~] = size(transpose(vars_p)); % n gives total number of variables

dPE = sym('f',[n,1]); % Creates dummy differential vector

% Differentiate PE in terms of each unknown variable
for i=1:n
    dPE(i,1) = simplify(diff(PE,vars_p(i)));
end

% Solve the system ogf equations
vars_pf = solve(dPE==0);
% Convert results from structures to numbers
vars_pf = transpose(structfun(@subs,vars_pf));
```

After the final package variables are determined, they can be plugged back into the potential energy equation to determine the final potential energy. This is done by the function solvestrainenergy.m.

```
function U = solvestrainenergy(PE,vars_p,varvalues_pf)

% This function solves for the strain energy once all of the variables are
% known
% Inputs:
% PE = PE equation
% varvalues_pf = fibnal values for unknown variables in PE equation

% Outputs:
% U = final value for strain energy

[n ~] = size(transpose(vars_p));
U=PE;

for i=1:n
    U = subs(U,vars_p(i),varvalues_pf(i));
end
```

With the final package curvature known, the initial and final curvatures of the board, package, and entire system can be plotted. This is done by the function plotcurvature.m.

```
function
plotcurvature(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varvalues_p,v
arvalues_pf,el_b,el_p)

% This functions plots the initial and final curvature of the board and
% package

% Inputs:
% Eq_b = governing equation for displacement filed of board
% vars_b = variables in board equation
% varvalues_b0 = initial values for variables in board equation
% varvalues_bf = final values for variables in board equation
% Eq_p = governing equation for displacement filed of package
% vars_p = variables in package equation
% varvalues_p = initial values for variables in package equation
% varvalues_pf = final values for variables in package equation
% el_b = edge locations for board
% el_p = edge locations for package

syms x y z
z1 = (el_b(1):.5:el_b(2));
x1 = (el_b(1):.5:el_b(2));
z2 = (el_p(1):.5:el_p(2));
x2 = (el_p(1):.5:el_p(2));
[n ~] = size(transpose(vars_b)); % n = # of variables in board equation
[m ~] = size(transpose(vars_p)); % m = # of variables in package equation
[o ~] = size(transpose(z1)); % o = # of data points for board
[p ~] = size(transpose(z2)); % p = # of data points for package
Eqb0 = Eq_b; % Initial board equation
Eqbf = Eq_b; % Final board equation
Eqp0 = Eq_p; % Initial package equation
Eqpf = Eq_p; % Final package equation
% Replace symbolic variables with actual variable values
for i=1:n
    Eqb0 = subs(Eqb0,vars_b(i),varvalues_b0(i));
    Eqbf = subs(Eqbf,vars_b(i),varvalues_bf(i));
end
```

```
for i=1:m
    Eqp0 = subs(Eqp0,vars_p(i),varvalues_p(i));
    Eqpf = subs(Eqpf,vars_p(i),varvalues_pf(i));
end

Eqb0m = ones(o,o); % Matrix of intial board equation
Eqbfm = ones(o,o); % Matrix of final board equation
Eqp0m = ones(p,p); % Matrix of initial package equation
Eqpfm = ones(p,p); % Matrix of final package equation

for i=1:o
    for j=1:o
        Eqb0m(i,j) = subs(subs(Eqb0,z,z1(i)),x,x1(j));
        Eqbfm(i,j) = subs(subs(Eqbf,z,z1(i)),x,x1(j));
    end
end

for i=1:p
    for j=1:p
        Eqp0m(i,j) = subs(subs(Eqp0,z,z2(i)),x,x2(j));
        Eqpfm(i,j) = subs(subs(Eqpf,z,z2(i)),x,x2(j));
    end
end

% Plot
surf(z1,x1,Eqb0m)
figure
surf(z2,x2,Eqp0m)
figure
surf(z1,x1,Eqbfm)
figure
surf(z2,x2,Eqpfm)
figure
surf(z1,x1,Eqb0m)
hold on
surf(z2,x2,Eqp0m)
figure
surf(z1,x1,Eqbfm)
hold on
surf(z2,x2,Eqpfm)
```

The next step in the analytic model process is to determine the final interconnect heights after the straightening process has been completed. This is done by the program calcfinaljointheights.m.

```
function hf =
calcfinaljointheights(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varva
lues_p0,varvalues_pf,ll,hi)

% This function calculates the final interconnect lengths given the initla
% and final displacment fields for both the board and package

% Inputs:
% Eq_b = displacement field for board
% vars_b = variables in displacement field for board
% varvalues_b0 = initial values for board variables
% varvalues_bf = final values for board variables
% Eq_p = displacement field for package
% vars_p = variables in displacement field for package
% varvalues_p0 = initial values for package variables
% varvalues_pf = final values for package variables
% ll = horizontal lead locations along face of package
% hi = initial interconnect lengths

% Outputs:
% hf = final interconnect lengths


syms x y z
[n1 n2] = size(ll); % n gives number of leads
[m ~] = size(transpose(vars_b)); % m = # variables in board equation
[o ~] = size(transpose(vars_p)); % o = # variables in package equation
Eqb0 = Eq_b; % Initial board equation
Eqbf = Eq_b; % Final board equation
Eqp0 = Eq_p; % Initial package equation
Eqpf = Eq_p; % Final package equation

% Replace symbolic variables with actual variable values
for i=1:m
    Eqb0 = subs(Eqb0,vars_b(i),varvalues_b0(i));
    Eqbf = subs(Eqbf,vars_b(i),varvalues_bf(i));
end

for i=1:o
    Eqp0 = subs(Eqp0,vars_p(i),varvalues_p0(i));
    Eqpf = subs(Eqpf,vars_p(i),varvalues_pf(i));
end

delta = zeros(n1,n2/2); % Creates dummy  matrix of the correct
% size to be filled in later

hf = zeros(n1,n2/2); % Creates dummy  matrix of the correct
% size to be filled in later

% Solve each equation at each lead location (z,x) and determine length
m=1;
counter=1;
llz = zeros(n1,n2/2);
llx = zeros(n1,n2/2);
while m<=n2
    llz(:,counter) = ll(:,m);
    llx(:,counter) = ll(:,m+1);
    m=m+2;
    counter=counter+1;
end

for i=1:n1
    for j=1:n2/2
        Eq2 = Eqbf;
        Eq2 = subs(Eq2,z,llz(i,j));
        Eq2 = subs(Eq2,x,llx(i,j));
        Eq4 = Eqpf;
        Eq4 = subs(Eq4,z,llz(i,j));
        Eq4 = subs(Eq4,x,llx(i,j));
        hf(i,j) = Eq4-Eq2;
    end
end
```

With the final interconnect height known, the next step is to calculate the final interconnect forces. This is done by the function calcforces.m.

```
function F = calcforces(hf,hi,ki)

% This function calculates the forces present in the interconnects after
% the straightening process has taken place

% Inputs:
% hf = final interconnect lengths
% hi = initial interconnect lengths
% ki = interconnect stiffnesses

% Outputs:
% F = forces in interconnects

[n1 n2] = size(ki);
F = ki; % Creates dummy force matrix

% Solve for force in each lead F=k*(hf-hi)
for i=1:n1
    for j=1:n2
        F(i,j) = ki(i,j)*(hf(i,j)-hi(i,j));
    end
end
```

Finally, the last step in the analytic model process is to calculate the final pre-stresses in the interconnects. This is done by the function calcstresses.m.

```
function sigma = calcstresses(F,A_s)

% This function determines the prestresses present in the solder joint
% after the straightening process has taken place

% Inputs:
% F = interconnect forces
% A_s - solder joint cross sectional area

% Outputs:
% sigma = prestresses in solder joint

[n1 n2] = size(F);
sigma = F; % Creates dummy force matrix

% Solve for prestresses in each solder joint (sigma = F/A)
```

```
for i=1:n1
    for j=1:n2
        sigma(i,j) = F(i,j)/A_s;
    end
end
```

The inputs to this model used to generate the results presented in this paper are given below.

```
>> syms x y z A0 A1 B1 C D
>> Eq_b = C*z^2+D*x^2;
>> Eq_p = A0+A1*z^2+B1*x^2;
>> vars_b = [C D];
>> vars_p = [A0 A1 B1];
>> varvaluesb0 = [.0004 0]; % Becomes varvaluesb0 = [.0004 .0004]; for bidirectional curvature
>> varvaluesbf = [0 0];
>> varvaluesp = [0 0];
>> h_min = .01;
>> h_lead = 1;
>> ll1 = [1.5;1.5;1.5;1.5;1.5];
>> ll2 = [4.5;4.5;4.5;4.5;4.5];
>> ll3 = [7.5;7.5;7.5;7.5;7.5];
>> ll4 = [10.5;10.5;10.5;10.5;10.5];
>> ll5 = [13.5;13.5;13.5;13.5;13.5];
>> ll6 = [13.5;10.5;7.5;4.5;1.5];
>> ll7 = [1.5;4.5;7.5;10.5;13.5];
>> l1 = [ll1,ll6;ll2,ll6;ll3,ll6;ll4,ll6;ll5,ll6];
>> l2 = [ll1,-ll7;ll2,-ll7;ll3,-ll7;ll4,-ll7;ll5,-ll7];
>> l3 = [-ll5,ll6;-ll4,ll6;-ll3,ll6;-ll2,ll6;-ll1,ll6];
>> l4 = [-ll5,-ll7;-ll4,-ll7;-ll3,-ll7;-ll2,-ll7;-ll1,-ll7];
>> ll = [l3;l1;l4;l2];
>> el_b = [-15,15];
>> el_p = [-14,14];
>> A_s = .1257;
>> E_s = 18135;
>> E_p = 20000;
>> h_p = 3.4;
>> v_p = .35;
>> E_lead = 102200;
>> A_lead = .1257;
>> p_lead = .00000896;
>> p_s = .00000737;
>> r = .2;
>> h_b = 5;
```

Appendix 4: Full Code for the Perimeter-Leaded Simplified Model

The following MATLAB functions comprise the three-dimensional simplified model. The model consists of a series of smaller programs which each perform a specific function and are referenced by a master program. Comments have been included to help explain exactly what each part of the code is doing.

The first program presented is the master program, called fullmodel.m. To run this program, the user calls this program and enters all of the required inputs. Each input is explained in the code's comments. This program calls each subprogram in order to fully run the simplified model and outputs the final results.

```
function [sigma,varvalues_pf,U] =
fullmodel(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_bf,varvalues_p,h_min,h_lead,ll,el_
b,el_p,A_s,E_s,E_p,h_p,v_p,E_lead,v_lead,w_lead,t_lead,L1,L2,L3,L4,R1,R2,alpha)
% This function is  a simplified model which determines the prestresses
% present in the solder joints after a straightening process has occurred
% Inputs:
% Eq_b = equation of board displacement field
% Eq_p = equation of package displacement field
% vars_b = list of variables in board equation (symbolic)
% vars_p = list of variables in package equation (symbolic)
% varvalues_b0 = initial values for variables in board equation
% varvalues_bf = final values for variables in board equation
% varvalues_p = initial values for variables in package equation
% h_min = minimum solder joint height
% h_lead = lead height
% ll = horizontal distance of leads along package face
% el_b = horizontal distance of ends of board
% el_p = horizontal distance of ends of package face
% *Note: For horizontal distances, (0,0) is taken as center of package*
% A_s = crosssectional area of the solder
% E_s = Elastic Modulus of the solder
% E_p = Elastic Modulus of package
% h_p = height of package
% v_p = Poisson's Ratio of package
% E_lead = Elastic Modulus of leads
% v_lead = Poisson's Ratio of leads
% w_lead = lead width (width of base)
% t_lead = lead thickness (height of base)
% L1 = Length of upper half of lead body (.405/2);
% L2 = Length of lower half of lead body (.405/2);
% L3 = Length of top of lead (.21);
% L4 = Length of base of lead (.21);
% R1 = Radius of top curvature (.275);
% R2 = Radius of bottom curvature (.275);
% alpha = angle between top/base of lead and lead body (.253 rad);
% *Note: Lead dimensions are measured on lead centerline*
syms x y z
% Step 0: Solve for A0
varvalues_p0 =
computeA0(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_p,h_min,h_lead,ll,el_b,el_p);
% Place package variables in alphabetical order (necessary when solving for
% final values)
pvars = [vars_p;varvalues_p0];
pvars = transpose(pvars);
pvars = sortrows(pvars);
pvars = transpose(pvars);
vars_p = pvars(1,:);
varvalues_p0 = pvars(2,:);
% Step 1: Calculate Interconnect Lengths
hi = calcjointheights(Eq_b,vars_b,varvalues_b0,Eq_p,vars_p,varvalues_p0,ll,el_b,el_p);
% Step 2: Calculate Interconnect Stiffness
ki = calcjointstiffness(hi,h_lead,A_s,E_s,E_lead,v_lead,w_lead,t_lead,L1,L2,L3,L4,R1,R2,alpha);
% Step 3: Calculate Flexural Rigidity of Package
D_p = (E_p*h_p^3)/(12*(1-v_p^2));
% Step 4: Compute Final Strain Energies
Upf = calcpackagestrainenergy(Eq_p,vars_p,varvalues_p0,D_p,v_p,el_p);
Ulf =
calcinterconnectstrainenergy(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_bf,varvalues_p
0,ki,ll,el_b,el_p);
% Step 5: Develop PE Equation
PE = (Upf+Ulf);
% Step 6: Minimize PE Equation
varvalues_pf = minimizePE2(PE,vars_p);
U = solvestrainenergy(PE,vars_p,varvalues_pf); % Calc. final strain energy
% Step 7: Plot Inital & Final Curvatures of Board & Package
plotcurvature(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varvalues_p0,varvalues_pf,e
l_b,el_p);
% Step 8: Determine Final Interconnect Lengths
```

```
hf =
calcfinaljointheights(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varvalues_p0,varvalu
es_pf,ll,el_b,el_p,hi);
% Step 9: Determine Interconnect Forces
F = calcforces(hf,hi,ki);
% Step 10: Calculate Stresses in Solder Joints
sigma = calcstresses(F,A_s);
```

The first step in this model is to solve for the initial value of A0 in the package displacement field. This value can be entered by the user, but it is not required. The next program, computeA0.m, determines if the user has input the initial value for A0 and, if not, calculates the value.

```
function varvalues_p =
computeA0(Eq_b,Eq_p,vars_b,vars_p,varvalue_b,varvalue_p,h_min,h_lead,ll,el_b,el_p)
% This function computes A0, the separation between the package and the
% board for given values of the other variables
% Inputs:
% Eq_b = equation of board displacement field
% Eq_p = equation of package displacement field
% vars_b = list of variables in board equation (symbolic)
% vars_p = list of variables in package equation (symbolic)
% varvalue_b = values for variables in board equation
% varvalue_p = values for variables in package equation
% h_min = minimum solder joint height
% h_lead = lead height
% ll = horizontal distance of leads along package face
% el_b = horizontal distance of ends of board
% el_p = horizontal distance of ends of package face
% *Note: For horizontal distances, (0,0) is taken as center of package*
% Outputs:
% A0 = value for A0 (constant in package displacement field equation)
% varvalues_p = values for variables in package equation
syms x y z
[n,~] = size(transpose(vars_b)); % n = # of board variables
[m,~] = size(transpose(vars_p)); % m = # of package variables
[o,~] = size(transpose(varvalue_b)); % o = # of board variables provided
[p,~] = size(transpose(varvalue_p)); % p = # of package variables provided
if n==o
else
```

```
   disp('Number of board variables provided does not match number of variables listed');
end
if m==p % A0 is predefined
   varvalues_p = varvalue_p;
elseif m==p+1 % A0 must be calculated
   % Replace symbolic variables with actual values
   for i=1:n
      Eq_b = subs(Eq_b,vars_b(i),varvalue_b(i));
   end
   for i=1:p
      Eq_p = subs(Eq_p,vars_p(i+1),varvalue_p(i));
   end
   Eq_2 = subs(Eq_p,vars_p(1),0); % Removes A0 from equation so equations
   % will not be symbolic
   [n,~] = size(transpose(ll)); % n gives number of leads
   d = [ll;ll;ll;ll]; % Creates a dummy matrix of the correct size which
   % will be filled in with the distances between the board and package at the lead locations
   % Substitute lead locations (z,x) into displacement field equations to
   % find distance between the board and package
   for i=1:n
      EQ1 = Eq_b;
      EQ1 = subs(EQ1,z,ll(i));
      EQ1 = subs(EQ1,x,el_b(2));
      EQ2 = Eq_2;
      EQ2 = subs(EQ2,z,ll(i));
      EQ2 = subs(EQ2,x,el_p(2));
      d(1,i) = EQ2-EQ1;
      EQ1 = Eq_b;
      EQ1 = subs(EQ1,z,ll(i));
      EQ1 = subs(EQ1,x,el_b(1));
      EQ2 = Eq_2;
      EQ2 = subs(EQ2,z,ll(i));
      EQ2 = subs(EQ2,x,el_p(1));
      d(2,i) = EQ2-EQ1;
      EQ1 = Eq_b;
      EQ1 = subs(EQ1,x,ll(i));
      EQ1 = subs(EQ1,z,el_b(2));
      EQ2 = Eq_2;
      EQ2 = subs(EQ2,x,ll(i));
      EQ2 = subs(EQ2,z,el_p(2));
      d(3,i) = EQ2-EQ1;
      EQ1 = Eq_b;
      EQ1 = subs(EQ1,x,ll(i));
      EQ1 = subs(EQ1,z,el_b(1));
```

```
        EQ2 = Eq_2;
        EQ2 = subs(EQ2,x,ll(i));
        EQ2 = subs(EQ2,z,el_p(1));
        d(4,i) = EQ2-EQ1;
    end
    minval = min(min(d)); % Determines the minimum distance between the
    % package and board (most negative or least positive distance)
    A0=-minval+h_min+h_lead; % A0 is set so the smallest interconnect
    % length which is fixed as the minimum joint height plus the lead height
    varvalues_p = [A0,varvalue_p]; % Adds value of A0 to list of variable
    % values for package displacement field equation
else
    disp('Number of package variables provided does not match number of variables listed.  Only
the constant coefficient can be undefined.');
end
```

After A0 is calculated, the master program rearranges the list of package variables and package variable names so they are in alphabetical order. This is done so that the variables are in the same order as the outputs of the potential energy minimization program which will be explained later. The next program called by the master program calculates the initial heights of the interconnects. This is program is named calcjointheights.m.

```
function h=calcjointheights(Eq_b,vars_b,varvalues_b,Eq_p,vars_p,varvalues_p,ll,el_b,el_p)
% This function is used to calculate the interconnect heights along the
% component based on the curvature of the board and package
% Inputs:
% Eq_b = governing equation for displacement field of board
% vars_b = variables in board equation
% varvalues_b = values for variables in board equation
% Eq_p = governing equation for displacement field of package
% vars_p = variables in package equation
% varvalues_p = values for variables in package equation
% ll = horizontal distance of leads along package face
% el_b = horizontal distance of ends of board
% el_p = horizontal distance of ends of package face
% *Note: For horizontal distances, (0,0) is taken as center of package*
% Outputs:
% h = interconnect lengths
syms x y z
```

```
[n ~] = size(transpose(vars_b)); % n = # of variables in board equation
[m ~] = size(transpose(vars_p)); % m = # of variables in package equation
Eqb = Eq_b;
Eqp = Eq_p;
% Replace symbolic variables with actual variable values
for i=1:n
   Eqb = subs(Eqb,vars_b(i),varvalues_b(i));
end
for i=1:m
   Eqp = subs(Eqp,vars_p(i),varvalues_p(i));
end
[n ~] = size(transpose(ll)); % n gives number of leads
h = [ll;ll;ll;ll]; % Creates dummy height matrix of the correct size to be
% filled in later (first row is front z face, second row is rear z face,
% third row is right x face, fourth row is left x face
% Determine height by substituting lead locations (z,x) into displacement
% fields and subtracting the height of the package from the height of the board
for i=1:n
   Eq1 = Eqb;
   Eq1 = subs(Eq1,z,ll(i));
   Eq1 = subs(Eq1,x,el_b(2));
   Eq2 = Eqp;
   Eq2 = subs(Eq2,z,ll(i));
   Eq2 = subs(Eq2,x,el_p(2));
   h(1,i) = Eq2-Eq1;
   Eq1 = Eqb;
   Eq1 = subs(Eq1,z,ll(i));
   Eq1 = subs(Eq1,x,el_b(1));
   Eq2 = Eqp;
   Eq2 = subs(Eq2,z,ll(i));
   Eq2 = subs(Eq2,x,el_p(1));
   h(2,i) = Eq2-Eq1;
   Eq1 = Eqb;
   Eq1 = subs(Eq1,x,ll(i));
   Eq1 = subs(Eq1,z,el_b(2));
   Eq2 = Eqp;
   Eq2 = subs(Eq2,x,ll(i));
   Eq2 = subs(Eq2,z,el_p(2));
   h(3,i) = Eq2-Eq1;
   Eq1 = Eqb;
   Eq1 = subs(Eq1,x,ll(i));
   Eq1 = subs(Eq1,z,el_b(1));
   Eq2 = Eqp;
   Eq2 = subs(Eq2,x,ll(i));
   Eq2 = subs(Eq2,z,el_p(1));
   h(4,i) = Eq2-Eq1;
end
```

Next, the interconnect stiffnesses are calculated. This is done by the program calcjointstiffness.m.

```
function ki =
calcjointstiffness(hi,h_lead,A_solder,E_solder,E_lead,v_lead,w_lead,t_lead,L1,L2,L3,L4,R1,R2,
alpha)
% This function calculates the stiffness of the interconnect by determining
% the stiffness of the solder joint and stiffness of the lead and using the
% series spring stiffness equation to find the overall interconnect
% stiffness
% Inputs:
% hi = initial interconnect heights
% h_lead = initial lead height
% A_solder = crosssectional area of the solder
% E_solder = elastic modulus of the solder
% E_lead = Elastic Modulus of leads
% v_lead = Poisson's Ratio of leads
% w_lead = lead width (width of base)
% t_lead = lead thickness (height of base)
% L1 = Length of upper half of lead body (.405/2);
% L2 = Length of lower half of lead body (.405/2);
% L3 = Length of top of lead (.21);
% L4 = Length of base of lead (.21);
% R1 = Radius of top curvature (.275);
% R2 = Radius of bottom curvature (.275);
% alpha = angle between top/base of lead and lead body (.253 rad);
% Outputs:
% ki = interconnect stiffnesses
% Step 1: Compute stiffness of solder joints (use k=AE/L approximation)
kj = hi; % Dummy joint stiffness matrix to be filled in later
[m,n] = size(hi); % m = # of faces, n= # leads/face
for i=1:m
   for j=1:n
      kj(i,j) = (A_solder*E_solder)/(hi(i,j)-h_lead); % K=AE/L
   end
end
% Step 2: Compute stiffness of gull wing leads (use Kotlowitz equations)
kl = kotlowitz(E_lead,v_lead,w_lead,t_lead,L1,L2,L3,L4,R1,R2,alpha);
```

```
% Step 3: Compute equivalent interconnect stiffnesses (use series spring
% stiffness equation k = 1/((1/k1)+(1/k2))
ki = hi; % Dummy interconnect stiffness matrix to be filled in later
for i=1:m
   for j=1:n
      ki(i,j) = 1/((1/kj(i,j))+(1/kl));
   end
end
```

This equation references another subprogram called kotlowitz.m. The purpose of this program is to determine the stiffness of a gull-wing lead using Kotlowitz's equations.

```
function kl = kotlowitz(E,v,w,t,L1,L2,L3,L4,R1,R2,alpha)
% This function determines the vertical stiffness of a gull wing lead using
% the kotlowitz analytical model equations
% Inputs:
% E = Elastic Modulus
% v = Poisson's Ratio
% w = lead width (width of base)
% t = lead thickness (height of base)
% Outputs:
% kl = lead stiffness
I = w*t^3/12; % Moment of inertia
f = 1.2; % Shear correction factor
A = t*w; % Cross-sectional area
G = E/(2*(1+v)); % Shear Modulus
C2 = .7854-alpha/2+.25*sin(2*alpha);
C3 = .7854-alpha/2-.25*sin(2*alpha);
C6 = (L1+L2)*sin(alpha)+(R1+R2)*cos(alpha)+L4;
S5 = (1/3)*(L4^3+L1^3*sin(alpha)^2)+R1*(L4^2*(1.5708-alpha)+2*L4*R1*(1-
sin(alpha))+R1^2*C3)+L1*((L4+R1*cos(alpha))^2+L1*sin(alpha)*(L4+R1*cos(alpha)));
S6 = (1/3)*L2^3*sin(alpha)^2+R2^3*C3+R2*C6^2*(1.5708-alpha)-2*R2^2*C6*(1-
sin(alpha))+(1/3)*((C6+L3)^3-
C6^3)+L2*(L4+R1*cos(alpha)+L1*sin(alpha))^2+L2^2*sin(alpha)*(L4+R1*cos(alpha)+L1*sin(alph
a));
S15 = L1*cos(alpha)^2+R1*C3+f*(E/G)*(L1*sin(alpha)^2+L4+R1*C2);
S16 = L2*cos(alpha)^2+C3*R2+f*(E/G)*(L2*sin(alpha)^2+L3+C2*R2);
kz = (1/E)*((S5/I)+(S6/I))+(1/E)*((S15/A)+(S16/A));
kl = 1/kz;
```

After the initial interconnect heights and stiffnesses have been calculated, the master program calculates the flexural rigidity of the package. With this done, the stain energy of the package is calculated using calcpackagestrainenergy.m.

```
function Up = calcpackagestrainenergy(wp,vars_p,varvalues_p,Dp,vp,el)
% This function calculates the strain energy of the package
% Inputs:
% wp = displacement field function
% vars_p = list of variables in displacement field function (excludes A0)
% varvalues = list of values for variables
% Dp = flexural rigidity of package
% vp = package's Poisson ratio
% el = horizontal locations of edges of package face
% Outputs:
% Up = package strain energy
syms x y z
[n ~] = size(transpose(vars_p)); % n gives number of variables
% Define Up equation to be integrated
Eq = (Dp/2)*(diff(wp,z,2)+diff(wp,x,2))^2-Dp*(1-vp)*(diff(wp,z,2)*diff(wp,x,2)-
diff(diff(wp,z),x)^2);
% Substitute actual variable values into equation for symbolic variables if
% they are known
if step == 1
   for i=1:n
      Eq = subs(Eq,vars_p(i),varvalues_p(i));
   end
end
% Integrate Up equation
Up = int(int(Eq,z,el(1),el(2)),x,el(1),el(2));
```

Next, the total strain energy of all of the interconnects is determined. This is done by calcinterconnectstrainenergy.m.

```
function UI =
calcinterconnectstrainenergy(Eq_b,Eq_p,vars_b,vars_p,varvalues_b0,varvalues_bf,varvalues_p,
ki,ll,el_b,el_p)
% This functon calculates the strain energy present in the interconnects
% after deformation
% Inputs:
% Eq_b = displacement field for board
% Eq_p = displacement field for package
% vars_b = variables in board equation
% vars_p = variables in package equation
% varvalues_bf = values for variables in board equation
% ki = interconnect stiffness
% ll = horizontal lead location along package face
% el_b = location of edges of board
% el_p = location of edges of package face
% Outputs:
% Ul = lead strain energy
syms x y z
[n ~] = size(transpose(ll)); % n gives number of leads
[m ~] = size(transpose(vars_b)); % m = # variables in board equation
[o ~] = size(transpose(vars_p)); % o = # variables in package equation
Eqb0 = Eq_b; % Initial board equation
Eqbf = Eq_b; % Final board equation
Eqp0 = Eq_p; % Initial package equation
Eqpf = Eq_p; % Final package equation
% Replace symbolic variables with actual variable values
for i=1:m
   Eqb0 = subs(Eqb0,vars_b(i),varvalues_b0(i));
   Eqbf = subs(Eqbf,vars_b(i),varvalues_bf(i));
end
for i=1:o
   Eqp0 = subs(Eqp0,vars_p(i),varvalues_p(i));
end
delta = sym('t',[4 n]); % Creates dummy strain energy matrix of the correct
% size to be filled in later (first row is front z face, second row is
% rear z face, third row is right x face, fourth row is left x face
% Determine lead displacement by solving all 4 equations at each lead
% location (z,x)
for i=1:n
   Eq1 = Eqb0;
   Eq1 = subs(Eq1,z,ll(i));
   Eq1 = subs(Eq1,x,el_b(2));
   Eq2 = Eqbf;
   Eq2 = subs(Eq2,z,ll(i));
   Eq2 = subs(Eq2,x,el_b(2));
   Eq3 = Eqp0;
   Eq3 = subs(Eq3,z,ll(i));
   Eq3 = subs(Eq3,x,el_p(2));
   Eq4 = Eqpf;
   Eq4 = subs(Eq4,z,ll(i));
   Eq4 = subs(Eq4,x,el_p(2));
```

```
    delta(1,i) =((Eq4-Eq3)-(Eq2-Eq1))^2;
    Eq1 = Eqb0;
    Eq1 = subs(Eq1,z,ll(i));
    Eq1 = subs(Eq1,x,el_b(1));
    Eq2 = Eqbf;
    Eq2 = subs(Eq2,z,ll(i));
    Eq2 = subs(Eq2,x,el_b(1));
    Eq3 = Eqp0;
    Eq3 = subs(Eq3,z,ll(i));
    Eq3 = subs(Eq3,x,el_p(1));
    Eq4 = Eqpf;
    Eq4 = subs(Eq4,z,ll(i));
    Eq4 = subs(Eq4,x,el_p(1));
    delta(2,i) = ((Eq4-Eq3)-(Eq2-Eq1))^2;
    Eq1 = Eqb0;
    Eq1 = subs(Eq1,x,ll(i));
    Eq1 = subs(Eq1,z,el_b(2));
    Eq2 = Eqbf;
    Eq2 = subs(Eq2,x,ll(i));
    Eq2 = subs(Eq2,z,el_b(2));
    Eq3 = Eqp0;
    Eq3 = subs(Eq3,x,ll(i));
    Eq3 = subs(Eq3,z,el_p(2));
    Eq4 = Eqpf;
    Eq4 = subs(Eq4,x,ll(i));
    Eq4 = subs(Eq4,z,el_p(2));
    delta(3,i) = ((Eq4-Eq3)-(Eq2-Eq1))^2;
    Eq1 = Eqb0;
    Eq1 = subs(Eq1,x,ll(i));
    Eq1 = subs(Eq1,z,el_b(1));
    Eq2 = Eqbf;
    Eq2 = subs(Eq2,x,ll(i));
    Eq2 = subs(Eq2,z,el_b(1));
    Eq3 = Eqp0;
    Eq3 = subs(Eq3,x,ll(i));
    Eq3 = subs(Eq3,z,el_p(1));
    Eq4 = Eqpf;
    Eq4 = subs(Eq4,x,ll(i));
    Eq4 = subs(Eq4,z,el_p(1));
    delta(4,i) = ((Eq4-Eq3)-(Eq2-Eq1))^2;
end
% determine strain energy in each lead
U = delta; % Creates dummy strain energy matrix
for i=1:n
```

```
   U(1,i) = .5*ki(1,i)*delta(1,i);
   U(2,i) = .5*ki(2,i)*delta(2,i);
   U(3,i) = .5*ki(3,i)*delta(3,i);
   U(4,i) = .5*ki(4,i)*delta(4,i);
end
% Develop final strain energy equation by summing all leads together
Ul = sum(U(1,:))+sum(U(2,:))+sum(U(3,:))+sum(U(4,:));
```

With all of the strain energies known, the potential energy equation of the system can be determined. This is done by the master program which sums the package strain energy and interconnect strain energy. The next step is to minimize this potential energy equation and solve for A0, A, and B for the final package displacement field. This is done by the program minimizePE2.m.

```
function [vars_pf,vars_p] = minimizePE2(PE,vars_p)
% This function solves for the unknown coefficients of the package and
% interconnect displacement fields by minimizing the PE equation
% Inputs:
% PE = potential energy equation
% vars_p = list of variables in package equation (excludes A0)
% Output:
% vars_pf = final values for package displacement field variables
[n ~] = size(transpose(vars_p)); % n gives total number of variables
dPE = sym('f',[n,1]); % Creates dummy differential vector
% Differentiate PE in terms of each unknown variable
for i=1:n
   dPE(i,1) = simplify(diff(PE,vars_p(i)));
end
% Solve the system of equations
vars_pf = solve(dPE==0);
% Convert results from structures to numbers
vars_pf = transpose(structfun(@subs,vars_pf));
```

Here, when the results are converted from structures to numbers, the output gives the variables in alphabetical order. This is why the initial user inputs had to be put into alphabetical order at the beginning of the master program. After the potential energy has been minimized, the value for the final strain energy can be determined using the solvestrainenergy.m function.

```
function U = solvestrainenergy(PE,vars_p,varvalues_pf)
% This function solves for the strain energy once all of the variables are
% known
% Inputs:
% PE = PE equation
% varvalues_pf = final values for unknown variables in PE equation
% Outputs:
% U = final value for strain energy
[n ~] = size(transpose(vars_p));
U=PE;
for i=1:n
   U = subs(U,vars_p(i),varvalues_pf(i));
end
```

Next, the curvatures are all plotted. This is done by the program plotcurvatures.m. This function plots surfaces showing the initial board curvature, initial package curvature, final board curvature, final package curvature, initial board and package curvature together, and final board and package curvature together.

```
function
plotcurvature(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varvalues_p,varvalues_pf,el
_b,el_p)
% This function plots the initial and final curvature of the board and
% package
% Inputs:
% Eq_b = governing equation for displacement field of board
% vars_b = variables in board equation
% varvalues_b0 = initial values for variables in board equation
% varvalues_bf = final values for variables in board equation
% Eq_p = governing equation for displacement field of package
% vars_p = variables in package equation
% varvalues_p = initial values for variables in package equation
% varvalues_pf = final values for variables in package equation
% el_b = edge locations for board
% el_p = edge locations for package
syms x y z
z1 = (el_b(1):.5:el_b(2));
x1 = (el_b(1):.5:el_b(2));
z2 = (el_p(1):.5:el_p(2));
x2 = (el_p(1):.5:el_p(2));
[n ~] = size(transpose(vars_b)); % n = # of variables in board equation
```

```
[m ~] = size(transpose(vars_p)); % m = # of variables in package equation
[o ~] = size(transpose(z1)); % o = # of data points for board
[p ~] = size(transpose(z2)); % p = # of data points for package
Eqb0 = Eq_b; % Initial board equation
Eqbf = Eq_b; % Final board equation
Eqp0 = Eq_p; % Initial package equation
Eqpf = Eq_p; % Final package equation
% Replace symbolic variables with actual variable values
for i=1:n
   Eqb0 = subs(Eqb0,vars_b(i),varvalues_b0(i));
   Eqbf = subs(Eqbf,vars_b(i),varvalues_bf(i));
end
for i=1:m
   Eqp0 = subs(Eqp0,vars_p(i),varvalues_p(i));
   Eqpf = subs(Eqpf,vars_p(i),varvalues_pf(i));
end
Eqb0m = ones(o,o); % Matrix of initial board equation
Eqbfm = ones(o,o); % Matrix of final board equation
Eqp0m = ones(p,p); % Matrix of initial package equation
Eqpfm = ones(p,p); % Matrix of final package equation
for i=1:o
   for j=1:o
      Eqb0m(i,j) = subs(subs(Eqb0,z,z1(i)),x,x1(j));
      Eqbfm(i,j) = subs(subs(Eqbf,z,z1(i)),x,x1(j));
   end
end
for i=1:p
   for j=1:p
      Eqp0m(i,j) = subs(subs(Eqp0,z,z2(i)),x,x2(j));
      Eqpfm(i,j) = subs(subs(Eqpf,z,z2(i)),x,x2(j));
   end
end
% Plot
surf(z1,x1,Eqb0m) % Initial Board
figure
surf(z2,x2,Eqp0m) % Initial Package
figure
surf(z1,x1,Eqbfm) % Final Board
figure
surf(z2,x2,Eqpfm) % Final Package
figure
surf(z1,x1,Eqb0m) % Initial Board and Package
hold on
surf(z2,x2,Eqp0m)
```

```
figure
surf(z1,x1,Eqbfm) % Final Board and Package
hold on
surf(z2,x2,Eqpfm)
```

After plotting the curvature, the final interconnect heights are calculated by the function calcfinaljointheights.m.

```
function hf =
calcfinaljointheights(Eq_b,vars_b,varvalues_b0,varvalues_bf,Eq_p,vars_p,varvalues_p0,varvalu
es_pf,ll,el_b,el_p,hi)
% This function calculates the final interconnect lengths given the initial
% and final displacement fields for both the board and package
% Inputs:
% Eq_b = displacement field for board
% vars_b = variables in displacement field for board
% varvalues_b0 = initial values for board variables
% varvalues_bf = final values for board variables
% Eq_p = displacement field for package
% vars_p = variables in displacement field for package
% varvalues_p0 = initial values for package variables
% varvalues_pf = final values for package variables
% ll = horizontal lead locations along face of package
% el_b= horizontal locations of edges of board
% el_p = horizontal locations of package face edges
% hi = initial interconnect lengths
% Outputs:
% hf = final interconnect lengths
syms x y z
[n ~] = size(transpose(ll)); % n gives number of leads
[m ~] = size(transpose(vars_b)); % m = # variables in board equation
[o ~] = size(transpose(vars_p)); % o = # variables in package equation
Eqb0 = Eq_b; % Initial board equation
Eqbf = Eq_b; % Final board equation
Eqp0 = Eq_p; % Initial package equation
Eqpf = Eq_p; % Final package equation
% Replace symbolic variables with actual variable values
for i=1:m
   Eqb0 = subs(Eqb0,vars_b(i),varvalues_b0(i));
   Eqbf = subs(Eqbf,vars_b(i),varvalues_bf(i));
end
for i=1:o
```

```
    Eqp0 = subs(Eqp0,vars_p(i),varvalues_p0(i));
    Eqpf = subs(Eqpf,vars_p(i),varvalues_pf(i));
end
delta = [ll;ll;ll;ll]; % Creates dummy  matrix of the correct
% size to be filled in later (first row is front z face, second row is
% rear z face, third row is right x face, fourth row is left x face
hf = [ll;ll;ll;ll]; % Creates dummy  matrix of the correct
% size to be filled in later (first row is front z face, second row is
% rear z face, third row is right x face, fourth row is left x face
% Solve each equation at each lead location (z,x) and determine length
for i=1:n
    Eq1 = Eqb0;
    Eq1 = subs(Eq1,z,ll(i));
    Eq1 = subs(Eq1,x,el_b(2));
    Eq2 = Eqbf;
    Eq2 = subs(Eq2,z,ll(i));
    Eq2 = subs(Eq2,x,el_b(2));
    Eq3 = Eqp0;
    Eq3 = subs(Eq3,z,ll(i));
    Eq3 = subs(Eq3,x,el_p(2));
    Eq4 = Eqpf;
    Eq4 = subs(Eq4,z,ll(i));
    Eq4 = subs(Eq4,x,el_p(2));
    delta(1,i) = (Eq4-Eq3)-(Eq2-Eq1);
    hf(1,i) = hi(1,i)+delta(1,i);
    Eq1 = Eqb0;
    Eq1 = subs(Eq1,z,ll(i));
    Eq1 = subs(Eq1,x,el_b(1));
    Eq2 = Eqbf;
    Eq2 = subs(Eq2,z,ll(i));
    Eq2 = subs(Eq2,x,el_b(1));
    Eq3 = Eqp0;
    Eq3 = subs(Eq3,z,ll(i));
    Eq3 = subs(Eq3,x,el_p(1));
    Eq4 = Eqpf;
    Eq4 = subs(Eq4,z,ll(i));
    Eq4 = subs(Eq4,x,el_p(1));
    delta(2,i) = (Eq4-Eq3)-(Eq2-Eq1);
    hf(2,i) = hi(2,i)+delta(2,i);
    Eq1 = Eqb0;
    Eq1 = subs(Eq1,x,ll(i));
    Eq1 = subs(Eq1,z,el_b(2));
    Eq2 = Eqbf;
    Eq2 = subs(Eq2,x,ll(i));
    Eq2 = subs(Eq2,z,el_b(2));
    Eq3 = Eqp0;
    Eq3 = subs(Eq3,x,ll(i));
    Eq3 = subs(Eq3,z,el_p(2));
    Eq4 = Eqpf;
    Eq4 = subs(Eq4,x,ll(i));
    Eq4 = subs(Eq4,z,el_p(2));
    delta(3,i) = (Eq4-Eq3)-(Eq2-Eq1);
    hf(3,i) = hi(3,i)+delta(3,i);
    Eq1 = Eqb0;
    Eq1 = subs(Eq1,x,ll(i));
    Eq1 = subs(Eq1,z,el_b(1));
    Eq2 = Eqbf;
    Eq2 = subs(Eq2,x,ll(i));
    Eq2 = subs(Eq2,z,el_b(1));
    Eq3 = Eqp0;
    Eq3 = subs(Eq3,x,ll(i));
    Eq3 = subs(Eq3,z,el_p(1));
    Eq4 = Eqpf;
    Eq4 = subs(Eq4,x,ll(i));
    Eq4 = subs(Eq4,z,el_p(1));
    delta(4,i) = (Eq4-Eq3)-(Eq2-Eq1);
    hf(4,i) = hi(4,i)+delta(4,i);
end
```

The next step is to calculate the forces which develop in the interconnects with the program calcforces.m.

```
function F = calcforces(hf,hi,ki)
% This function calculates the forces present in the interconnects after
% the straightening process has taken place
% Inputs:
% hf = final interconnect lengths
% hi = initial interconnect lengths
% ki = interconnect stiffnesses
% Outputs:
% F = forces in interconnects
[n m] = size(transpose(ki)); % n = # of leads per face, m = # of faces
F = ki; % Creates dummy force matrix
% Solve for force in each lead F=k*(hf-hi)
for i=1:n
    for j=1:m
```

```
        F(j,i) = ki(j,i)*(hf(j,i)-hi(j,i));
    end
end
```

The final step is to calculate the interconnect pre-stresses. This is done by the function calcstresses.m.

```
function sigma = calcstresses(F,A_s)
% This function determines the prestresses present in the solder joint
% after the straightening process has taken place
% Inputs:
% F = interconnect forces
% A_s - solder joint cross sectional area
% Outputs:
% sigma = prestresses in solder joint
[n m] = size(transpose(F)); % n = # of leads per face, m = # of faces
sigma = F; % Creates dummy force matrix
% Solve for prestresses in each solder joint (sigma = F/A)
for i=1:n
    for j=1:m
        sigma(j,i) = F(j,i)/A_s;
    end
end
```

The inputs to this model used to generate the results presented in Appendix 1 are given below.

```
>> syms x y z A0 A B C D
>> Eq_b = C*z^2+D*x^2;
>> Eq_p = A0+A*z^2+B*x^2;
>> vars_b = [C D];
>> vars_p = [A0 A B];
>> varvaluesb0 = [.0004 0]; % Becomes varvaluesb0 = [.0004 .0004]; for bidirectional curvature
>> varvaluesbf = [0 0];
>> varvaluesp = [0 0];
>> h_min = .01;
>> h_lead = 2.075;
>> ll = [-12.75:.5:12.75];
>> el_b = [-15,15];
>> el_p = [-14,14];
>> A_s = .132;
>> E_s = 18135;
>> E_p = 20000;
>> h_p = 3.4;
>> v_p = .35;
>> E_lead = 102200;
>> v_lead = .343;
>> w_lead = .22;
>> t_lead = .15;
>> p_lead = .00000896;
>> w_s = .6;
>> t_s = .22;
>> p_s = .00000737;
>> h_b = 5;
>> L1 = .81;
>> L2 = .81;
>> L3 = .21;
>> L4 = .21;
>> R1 = .275;
>> R2 = .275;
>> alpha = .253;
```

References

Bhat, R. B. 1985. “Natural Frequencies of Rectangular Plates Using Characteristic Orthogonal Polynomials in Rayleigh-Ritz Method.” *Journal of Sound and Vibration.* 102(4): 493-499. Accessed December 5, 2014. doi:10.1016/S0022-460X(85)80109-7.

Dawe, D. J., and O. L. Roufaeil. 1980. “Rayleigh-Ritz Vibration Analysis of Mindlin Plates.” *Journal of Sound and Vibration*. 69(3): 345-359. Accessed December 4, 2014. doi:10.1016/0022-460X(80)90477-0.

Halvi, Ananth Srivatsav, Wonkee Ahn, Dereje Agonafer, and Shlomo Novotny. 2004. “Simulation of PWB Warpage During Fabrication and Due to Reflow.” *Thermal and Thermomechanical Phenomena in Electronic Systems*. 2: 674-678. Accessed November 14, 2014. doi:10.1109/ITHERM.2004.1318352.

Haslach, Jr., Henry W., and Ronald Armstrong. 2004. *Deformable Bodies and Their Material Behavior*. Hoboken: John Wiley & Sons, Inc.

Horrocks, D., and W. Johnson. 1967. “On Anticlastic Curvature with Special Reference to Plastic Bending: A Literature Survey and Some Experimental Investigations.” *International Journal of Mechanical Sciences*. 9(12): 835-844. Accessed November 23, 2014. doi:10.1016/0020-7403(67)90011-2.

Kim, C. S., P. G. Young, and S. M. Dickinson. 1990. “On the Flexural Vibration of Rectangular Plates Approached by Using Simple Polynomials in the Rayleigh-Ritz Method.” *Journal of Sound and Vibration*. 143(3): 379-394. Accesses December 4 2014. doi:10.1016/0022-460X(90)90730-N.

Kotlowitz, Robert W., and Lisa Taylor. 1991. “Compliance Metrics for the Inclined Gull Wing.Spider J-Bend, and Spider Gull-Wing Lead Designs for Surface Mount Components.” *IEEE Transactions on Components, Hybrids, and Manufacturing Technology* 14(4):771-779.

Liew, K. M., and C. M. Wang. 1993. “pb-2 Rayleigh-Ritz method for general plate analysis.” *Engineering Structures*. 15(1): 55-60. Accessed December 3, 2014. doi:10.1016/0141-0296(93)90017-X.

Mittal, Sandeep, Glenn Masada, and Theodore Bergman. 1996. “Mechanical Response of PCB Assemblies During Infrared Reflow Soldering.” *IEEE Transactions on Components, Packaging, and Manufacturing Technology*. 19(1): 127-133. Accessed November 14, 2014. doi:10.1109/95.486624.

Polsky, Yarom, Walter Sutherlin, and Charles Ume. 2000. “A Comparison of PWB Warpage Due to Simulated Infrared and Wave Soldering Processes.” *IEEE Transactions on Electronics Packaging Manufacturing*. 23(3): 191-199. Accessed November 15. 2015. doi:10.1109/6104.873247.

Pomeroy, R. J., 1970. “The Effect of Anticlastic Bending on the Curvature of Beams.” *International Journal of Solids and Structures*. 6(2): 277-285. Accessed November 23, 2014. doi:10.1016/0020-7683(70)90024-7.

Radio-Electronics. 2014. “What is SMT Surface Mount Technology – Tutorial.” Accessed October 23. http://www.radio-electronics.com/info/data/smt/what-is-surface-mount-technology-tutorial.php.

Stiteler, Michael R., Charles Ume, and Brian Leutz. 1996. "In-Process Board Warpage Measurement in a Lab Scale Wave Soldering Oven." *IEEE Transactions on Components, Packaging, and Manufacturing Technology*. 19(4): 562-569. Accessed November 14, 2015. doi:10.1109/95.554938.

Surface Mount Technology Association. 2014. "Surface Mount Technology A Historical Perspective." Accessed October 23. http://www.emsnow.com/cnt/files/White%20Papers/smta_history_of_smt.pdf.

Ume, Charles, Timothy Martin, and Jeffrey Gatro. 1997. "Finite Element Analysis of PWB Warpage Due to the Solder Masking Process." *IEEE Transactions on Components, Packaging, and Manufacturing Technology*. 20(3): 295-306. Accessed November 14, 2014. doi:10.1109/95.623024.

Wolfram MathWorld. 2014. "Sagitta." Accessed October 27. http://mathworld.wolfram.com/Sagitta.html.

Yang, Se Young, Young-Doo Jeon, Soon-Bok Lee, and Kyung-Wook Paik. 2006. "Solder reflow process induced residual warpage measurement and its influence on reliability of flip-chip electronic package." *Microelectronics Reliability*. 46(2-4): 512-522. Accessed November 12, 2014. doi:10.1016/j.microrel.2005.06.007.

Yuan, J., and S. M. Dickinson. 1992. "On the Use of Artificial Springs in the Study of the Free Vibrations of Systems Comprised of Straight and Curved Beams." *Journal of Sound and Vibration*. 153(2): 203-216. Accessed December 3, 2014. doi:10.1016/S0022-460X(05)80002-1.

Yuan, J., and S. M. Dickinson. 1992. “The Flexural Vibration of Rectangular Plate Systems Approached by Using Artificial Springs in the Rayleigh-Ritz Method.” *Journal of Sound and Vibration*. 159(1): 39-55. Accessed December 3, 2014. doi:10.1016/0022-460X(92)90450-C.