

January 2015

A Planar Pseudo-Rigid-Body Model for Cantilevers Experiencing Combined Endpoint Forces and Uniformly Distributed Loads Acting in Parallel

Philip James Logan

University of South Florida, pjlogan@mail.usf.edu

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>

 Part of the [Other Mechanical Engineering Commons](#)

Scholar Commons Citation

Logan, Philip James, "A Planar Pseudo-Rigid-Body Model for Cantilevers Experiencing Combined Endpoint Forces and Uniformly Distributed Loads Acting in Parallel" (2015). *Graduate Theses and Dissertations*.
<http://scholarcommons.usf.edu/etd/5730>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

A Planar Pseudo-Rigid-Body Model for Cantilevers Experiencing Combined Endpoint Forces
and Uniformly Distributed Loads Acting in Parallel

by

Philip James Logan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering
Department of Mechanical Engineering
College of Engineering
University of South Florida

Co-Major Professor: Craig Lusk, Ph.D.
Co-Major Professor: Alex Volinsky, Ph.D.
Yu Sun, Ph.D.
Kyle Reed, Ph.D.
Mark Weston, M.Arch.

Date of Approval:
June 16, 2015

Keywords: Compliant Mechanisms, Heavy Beams, Phase Space, Large Deflections, Kinematics

Copyright © 2015, Philip James Logan

DEDICATION

To my parents, Janice Alfonso and Jim Logan. Their love and support has been unwavering.

ACKNOWLEDGMENTS

The author of this work gratefully acknowledges the support and contributions from the people and organizations which made this dissertation possible. At the focus of my gratitude is Dr. Craig Lusk. It was with his near limitless patience and technical expertise that I was able to educate myself to the extent that the prospect of contribution was even possible.

My thanks extends to Dr. Autar Kaw who inspired me to understand programming language, not to merely memorize the commands. His example of teaching style also helped me in my struggles as a teaching assistant. I would like to thank Dr. Kyle Reed for always having time to talk to students with an open door and an open mind. And I would also like to thank Dr. John Dixon and Dr. Alex Volinsky for having confidence in my teaching abilities. Leading help sessions and occasionally filling in for lectures were very rewarding experiences.

My thanks also extends to the compliant mechanisms group, for educating me through sharing their efforts. I would particularly like to thank Ahmad Alqasimi for his help with CAD software.

My gratitude also extends to the committee members. The time and effort put into reviewing this work is deeply appreciated.

I would also like to acknowledge the National Science Foundation for their support through Grant # CMMI-1000138.

TABLE OF CONTENTS

LIST OF TABLES	iii
LIST OF FIGURES	iv
ABSTRACT	vii
CHAPTER 1: INTRODUCTION	1
1.1 Objective	1
1.2 Motivation	1
1.3 Scope	2
1.4 Overview	2
CHAPTER 2: BACKGROUND	4
2.1 Compliant Mechanisms	4
2.2 The Elastica	8
2.3 The Pseudo-Rigid-Body Model	11
2.3.1 Parameterized Kinematics	12
2.3.2 Parameterized Load-Deflection Relationships	15
2.4 Cantilever Beam Differential Equations	17
CHAPTER 3: NON-DIMENSIONAL FORM OF DIFFERENTIAL EQUATIONS	20
CHAPTER 4: PHASE SPACE AND INITIAL VALIDATIONS	24
CHAPTER 5: THE CONTOUR METHOD	32
5.1 The Data Structure	32
5.2 Finding $\gamma(\phi, \mu)$	35
5.3 Finding $c_\theta(\phi, \mu)$	47
5.4 Finding $K_\theta(\phi, \mu)$	51
5.5 Non-dimensional Strain Energy	52
5.6 Correction Factors	59
CHAPTER 6: RESULTS	73
6.1 Error Evaluations for $\mu = 1$ and $\mu = 0$	74
6.2 Example Problem	80
6.3 Error Analysis of Intermediate μ Values	83
CHAPTER 7: CONCLUSION	87

7.1 Recommendations for Use	87
7.2 Future Work	89
REFERENCES	91
APPENDIX A: NOMENCLATURE.....	95
APPENDIX B: MATLAB CODE	99
B.1 Function File	99
B.2 Elastica Example	99
B.3 Surface of μ	100
B.4 Distributed Phase Trajectory	103
B.5 ANSYS Comparisons.....	104
B.6 Phase Space of γ	112
B.7 γ Fit.....	117
B.8 Finding c_θ	122
B.9 Fitting c_θ	127
B.10 Strain Energy K_θ Comparisons	130
B.11 Finding K_θ Roots	135
B.12 Fitting $K_{\theta,init}$	144
B.13 Finding Correction Factors.....	147
B.14 Fitting Correction Factors	155
B.15 Finding the Final K_θ	160
B.16 Fitting the Final K_θ	166
B.17 Effects of Correction Factors	169
B.18 Error Analysis for $\mu = 0$ and $\mu = 1$	177
B.19 Example Problem	186
B.20 Error Analysis for ANSYS Results of Intermediate μ Value	188

LIST OF TABLES

Table 1 Equations recommended for use by designers.....	88
---	----

LIST OF FIGURES

Figure 1 The use of a compliant segment is visible on the left side of this MEMS device.....	5
Figure 2 An example of the wear associated with pin joints created on the MEMS scale.....	6
Figure 3 The ribbon-like curve of the elastica and its corresponding curvature	10
Figure 4 The Pseudo-Rigid-Body Model adapted from Howell.....	12
Figure 5 Cantilever beam with uniformly distributed load w and end force F pointing in the same direction	18
Figure 6 The distributed contribution factor versus w and F	23
Figure 7 Phase portraits for the force only case.....	25
Figure 8 Depiction of a single trajectory originating at $\pi/4$ for the distributed load only case.....	27
Figure 9 The results from a single trial in ANSYS with $\phi = 105^\circ$	28
Figure 10 Phase space trajectory comparison with FEA for $\mu = 1$ and $\mu = 0$	30
Figure 11 Phase space trajectory comparison with FEA for $\mu = 0.40973$ and $\mu = 0.60817$	31
Figure 12 Here, γ is populating the phase space, thereby replacing \tilde{s}	37
Figure 13 This represents the surface from which the contours, or levels curves, of γ were extracted, with $\mu = 1$	38
Figure 14 The contour of γ , at $\phi = 90^\circ$, as taken from the surface shown in Figure (13).....	39
Figure 15 The relative error surface surrounding the contour of γ , when $\mu = 1$ and $\phi = 90^\circ$	41
Figure 16 Observed values of γ and $\Theta_{max,\gamma}$ found by using the contour method, with $\phi = 90^\circ$ and $\mu = 1$	42

Figure 17 Surface of the $\gamma(\phi, \mu)$ fit to the contour data	44
Figure 18 Surface of the $\Theta_{max,\gamma}(\phi, \mu)$ fit to the contour data	45
Figure 19 Force only PRBM comparisons.....	46
Figure 20 Contours of Θ vs θ , demonstrating the near linear relationship between them.....	47
Figure 21 Surface of $c\theta$ showing the persistent discontinuity.....	48
Figure 22 Surface of the $c\theta(\phi, \mu)$ fit to values stored in the data structure	49
Figure 23 Comparisons between PRBMs for $c\theta$ when $\mu = 1$	50
Figure 24 FEA data and the associated polynomial fit used to calculate the work done on a beam by a force applied at 90°	55
Figure 25 The K_θ values as predicted by the non-dimensional strain energy and the K_θ values as predicted by an independent FEA investigation.....	56
Figure 26 The force only case showing close agreement between the K_θ calculated from the strain energy and the K_θ calculated by the transverse load index	57
Figure 27 The distributed load only case showing the discrepancy between the K_θ calculated from the strain energy and the K_θ calculated by the transverse load index.....	58
Figure 28 K_θ surfaces which are now sharing a common root due to the introduction of a constant factor of 0.4 to the distributed transverse load index	59
Figure 29 $K_{\theta,init}(\phi, \mu)$ surface used to find correction factors.....	62
Figure 30 Correction factor of force contribution for $\phi \leq 100^\circ$	64
Figure 31 Correction factor of distributed contribution for $\phi \leq 100^\circ$	65
Figure 32 Correction factor of force contribution for $\phi > 100^\circ$	66
Figure 33 Correction factor of distributed contribution for $\phi > 100^\circ$	67
Figure 34 Effect of correction factors on a single contour for $\mu = 1$ and $\phi = 90^\circ$	68
Figure 35 The departure from verticality for $\mu = 0.6$	69
Figure 36 The final surface fit for K_θ	70

Figure 37 The PRBM for cantilever beams undergoing combined parallel loading	72
Figure 38 Contours of % relative error in $\tilde{\delta}$ for $\mu = 0$	78
Figure 39 Contours of % relative error in $\tilde{\delta}$ for $\mu = 1$	79
Figure 40 Contours of % relative error in $\tilde{\delta}$ associated with the PRBM constructed by Howell et al	79
Figure 41 Output vectors from the PRBM functions of μ	82
Figure 42 The phase plane view of the data used in the dimensional error analysis	84
Figure 43 The % relative error in μ when F is the unknown	85
Figure 44 The % relative error in predicted F values	85
Figure 45 The % relative error in μ when w is the unknown	86
Figure 46 The % relative error in predicted w values	86

ABSTRACT

This dissertation describes the development and effectiveness of a mathematical model used to predict the behavior of cantilever beams whose loading conditions include parallel combinations of evenly distributed loads and endpoint forces. The large deflection of cantilever beams has been widely studied. A number of models and mathematical techniques have been utilized in predicting the endpoint path coordinates and load-deflection relationships of such beams. The Pseudo-Rigid-Body Model (PRBM) is one such method which replaces the elastic beam with rigid links of a parameterized pivot location and torsional spring stiffness. In this paper, the PRBM method is extended to include cases of a constant distributed load combined with a parallel endpoint force. The phase space of the governing differential equations is used to store information relevant to the characterization of the PRBM parameters. Correction factors are also given to decrease the error in the load-deflection relationship and extend the angular range of the model, thereby further aiding compliant mechanism design. The calculations suggest a simple way of representing the effective torque caused by a distributed load in a PRBM as a function of easily calculated model parameters.

CHAPTER 1: INTRODUCTION

1.1 Objective

The objective of this dissertation is to present the first planar Pseudo-Rigid-Body Model (PRBM) which describes the large deflection behavior of cantilever beams being acted upon by a combination of evenly distributed loads and endpoint forces acting in parallel. This model provides kinematic predictions of deflected endpoint locations and orientations. It also provides load-deflection relationships, whereby cantilever beam characteristics, such as material type and second moment of area, are tied to the loading conditions and the resulting displacements.

1.2 Motivation

The motivation of this work is to develop a mathematical construct whereby designers of compliant mechanisms can now incorporate the mass of the beam itself into their design considerations. Prior PRBM's have provided designers with a simplified interpretation of large deflection behavior. This has allowed designers to forego the use of complicated mathematics, such as elliptic integrals, to describe the highly nonlinear behavior inherent to large deflection analysis. Such simplifications have proven successful at producing new compliant mechanism designs and concepts.

As will be discussed, the loading conditions present in this analysis give rise to differential equations which, as yet, do not have closed-form solutions. Prior planar PRBM's have ignored the mass of the cantilever beam in their formulations and have, as such, benefitted from the existence of an exact closed-form solution. While the form of such a solution to the governing differential

equations, which will now incorporate an evenly distributed load, remains elusive, the designer can still be equipped with a set of relationships that describe the behavior of compliant components, relative to the beam's characteristics. So equipped, an engineer may still initialize a compliant mechanism design by neglecting the mass effects of the cantilever beam, but may now do so more by choice, rather than necessity. It will also now be possible to initialize the design of compliant mechanisms that interact with external fields or otherwise inertial reference frames.

1.3 Scope

The scope of this work is to first describe the development of a non-dimensional form of the governing differential equations associated with the parallel combination of evenly distributed loads and endpoint forces. The resulting equations will then be used to relate the load and beam characteristics to various PRBM parameters. The method by which the information extracted from the governing equations is thus related, herein named the *contour method*, will be described in detail, as well. The resulting model, with its adjoining equations, will then undergo an error analysis.

1.4 Overview

Chapter 2 serves to introduce the reader to the general bodies of knowledge associated with this work. These include the concept of planar compliant mechanisms, the family of curves, known as the *elastica*, and the concept of the Pseudo-Rigid-Body Model as a means of describing cantilever states undergoing large deflections. Lastly, the system of differential equations, used to describe the combined loading case under investigation, are shown and described in detail.

Chapter 3 describes the methods by which the differential equations, listed in Chapter 2, are non-dimensionalized. A new parameter is introduced which later serves in the development of the PRBM for this set of combined loading conditions.

Chapter 4 introduces the concept of phase space and demonstrates the utility of viewing the governing system of differential equations within that context. Also, initial Finite Element Analysis (FEA) results are used to validate the choice of the non-dimensional factor employed.

Chapter 5 discusses the contour method by first explicitly defining the data structure, including how the information is initially stored and then how it evolves through the process of numerical integration. Chapter 5 also involves the parameterization of all relevant PRBM variables and relationships. It also introduces and justifies the final version of the PRBM which incorporates distributed loading contributions acting in parallel to endpoint forces.

Chapter 6 investigates the effectiveness of the PRBM constructed in Chapter 5 by conducting error analyses. This is done both in a non-dimensional sense and in a dimensional sense. The forms of the equations describing the force-deflection relationships are introduced, as they vary depending upon the variables sought after. An example problem is also given, thereby demonstrating, again, the effectiveness and use of the model.

Chapter 7 concludes this work by summarizing its contributions. Recommendations to those who may wish to use the model are given. Ideas for future work are also given at the end.

CHAPTER 2: BACKGROUND

2.1 Compliant Mechanisms

Generally speaking, mechanisms are constructions which transform or transfer motion, force, or energy [1, 2]. Mechanisms which are the most numerous and familiar are those primarily composed of rigid links which are connected with *kinematic pairs*, such as pin joints or sliders. Such an assembly allows the transformation of force into motion, be it angular for a pin joint or linear for slider.

A *compliant mechanism*, however, is one which gains most, or all, of its motion by the deflection of flexible members, rather than from rigid-body joints, exclusively [3]. Flexible members are distinguishable from rigid members in that they are generally thinner, and are thus more susceptible to deformation under a given loading condition. Simple examples of compliant mechanisms include the bow and shampoo caps with a pop-top. Compliant mechanisms may still incorporate kinematic pairs, such as pin joints or sliders, and are thus labeled *partially compliant mechanisms*. By contrast, *fully compliant mechanisms* have no such kinematic pairs and seem more akin to structures. They gain all of their motion by the deformation of the flexible members. Were the cross-sectional characteristics of the flexible members changed in such a way as to increase their resistance to bending, their deflections could be made nearly indiscernible, as is the case with stable structures, such as buildings.

Compliant mechanisms offer several advantages over their rigid-body counterparts. These include a reduced part-count, thereby reducing assembly time, less wear and maintenance, reduced

weight and a simplified manufacturing process [4]. By not incorporating pin joints, the absence of rubbing between adjoining members reduces vibration, thereby increasing precision. Consequently, compliant mechanisms have been incorporated into high-precision instrumentation [5, 6], including lens focusing [7, 8]. Other examples of macroscopic compliant mechanisms include centrifugal clutches [9-11], pantographs [12], and plastic based products [13, 14].

By contrast, compliant mechanisms have also found extensive use in Micro-Electro-Mechanical Systems (MEMS). Figure (1) shows the use of compliant flexures at the MEMS scale. As they are relatively easy to miniaturize [15-24], when compared with their rigid body

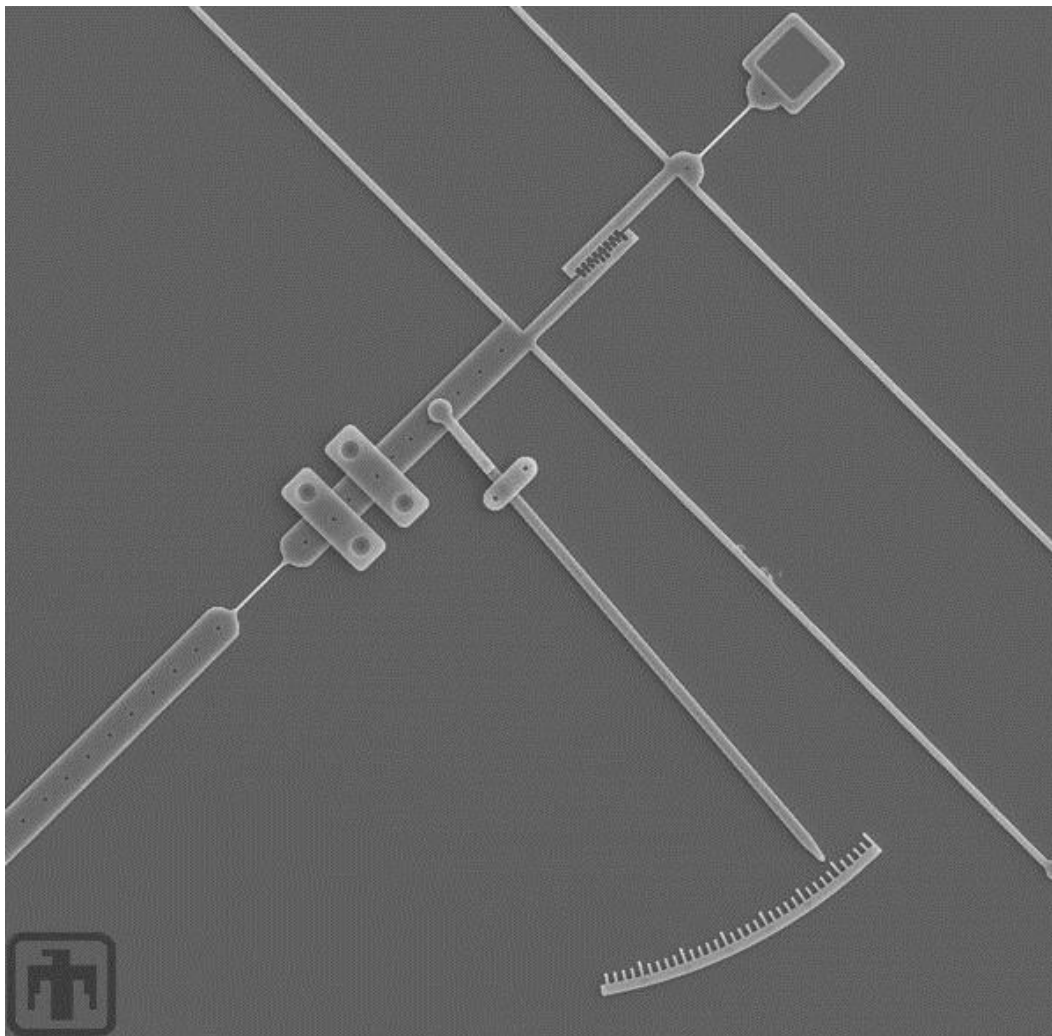


Figure 1 The use of a compliant segment is visible on the left side of this MEMS device. (Courtesy of Sandia National Laboratories, www.sandia.gov). This image is in the public domain.

counterparts, whose pin joint behavioral characteristics change significantly upon scale reduction, compliant mechanisms offer several advantages when used in MEMS devices. First, MEMS fabrication techniques are highly planar by nature [25], due to the growth of various layers via deposition onto planar substrates. Another advantage is found with respect to the micro-machined pin joints, the kinematics of which flexible members are able to imitate. These pin joints have a large surface roughness, as compared to their diameter. Continued rotation thus causes increasing dust creation around the joint and a subsequent enlargement of the difference between the diameters of the outside of the pin and the inside of the joint. This increase in the spacing between the pin and the joint decreases performance and precision, and ultimately leads to failure. Figure (2) shows an example of this type of failure in a MEMS pin joint. Yet another advantage of compliant MEMS is that there is no assembly required. The number of steps required to create

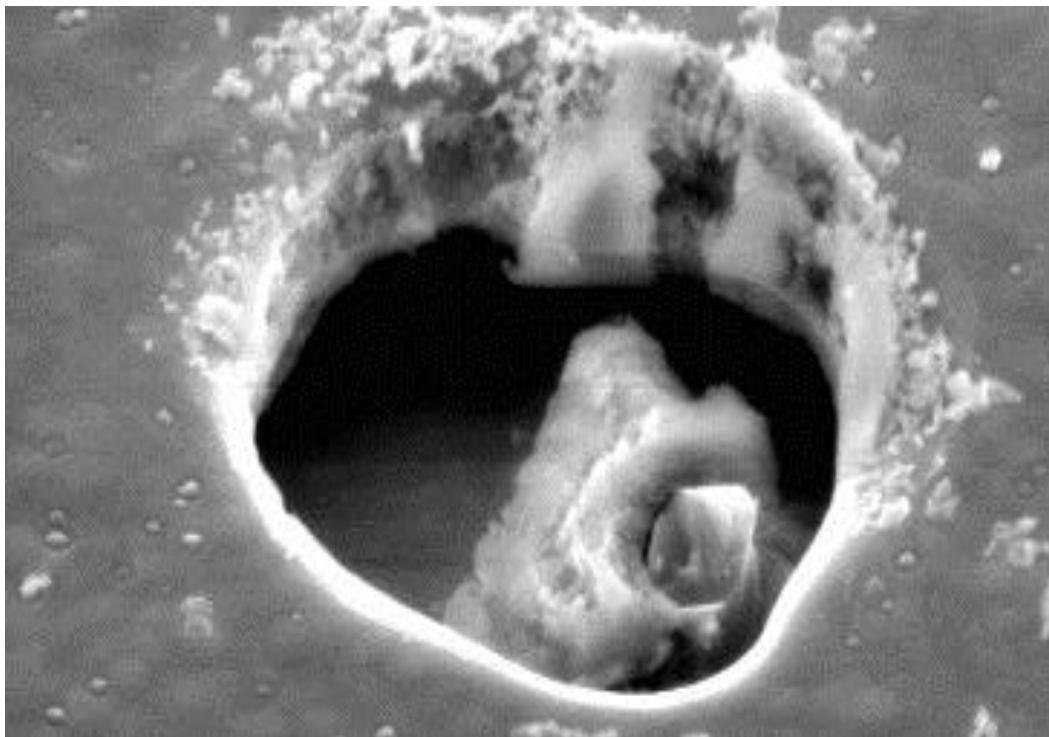


Figure 2 An example of the wear associated with pin joints created on the MEMS scale (Courtesy of Sandia National Laboratories, www.sandia.gov). This image is in the public domain.

free-standing structures on the MEMS scale is thereby significantly reduced when incorporating flexible segments.

However, compliant mechanisms do have their detractors. By nature, the deformation of flexible segments is accompanied by an increase in their stored energy [26]. This energy takes the form of strain energy and may not always be useful. There are many types of designs which use this stored strain energy to maintain a state of deformation indefinitely, even amidst small disturbances [27]. This phenomenon is termed *bi-stability* and has been used in several applications, including bi-stable switches and the Bi-stable Young mechanism [17].

While this is a productive use of the strain energy storage found within deformed flexible links, several problems persist. One problem is that of mechanical advantage. As the stored strain energy results from work being performed on the flexible segment via an input force, or torque, the resulting output force, or torque, will often be less than that of a rigid-link counterpart. This is the result of the input force, or torque, causing both deformation and motion in the flexible segments. Thus, there may be a significant energy loss between the energy supplied at the input and the energy resulting at the output. Another disadvantage is that of *creep* [3]. If a compliant mechanism is allowed to remain in its deformed state for long periods of time, and/or at elevated temperatures, stress relaxation often occurs. This arises from molecular rearrangement of the deformed material, resulting in permanent deformation [28]. In other words, as the loading conditions, which have caused the deformation, are removed, the flexible links will not return to their original positions. This can lead to degraded performance and precision. Lastly, a major disadvantage to using compliant mechanisms is that accurately predicting their behavior is challenging. This is due to the highly nonlinear behavior associated with large deflections.

2.2 The Elastica

The problem of beam deflection under applied loads is an old one. Merely posing the problem in a precise mathematical formulation required some of the brightest intellects in history. Galileo helped to set the stage by introducing the concept of moments relating to applied forces, thereby framing the problem of beam deflection in firm mathematical terminology [29]. Hooke introduced the idea of interconnected springs to attempt an explanation of the change in the *curvature* of a beam as the length changes, but was later shown to be an inadequate framing of the problem for anything but small deflections [30]. What was necessary was a precise mathematical understanding of curvature. This was not available until Newton published, through the use of his calculus, the formulation of the radius of curvature in Cartesian coordinates:

$$\rho = \frac{(1 + y'^2)^{\frac{3}{2}}}{y''} \quad (1)$$

Here, y' is the slope of a curve traversing the xy -plane, y'' is the change of that slope with respect to the horizontal coordinate x , and ρ is defined as the radius of curvature. The radius of curvature is related to the curvature via:

$$\kappa = \frac{1}{\rho} \quad (2)$$

where κ is defined as the curvature.

Equipped with this new precise relationship, James Bernoulli was able to more precisely formulate the relationship between the *arc length*, which is used to describe the beam length when

under a state of deformation, and the angle of deflection. The equations used to describe the problem in this fashion were later recognized by Huygens to fall short of being a general solution [31]. Revisiting the matter, James Bernoulli published a more general solution. Using this, Daniel Bernoulli and Euler, by applying variational techniques, were able to make the insight that the arc length followed a curve which minimized the strain energy of the beam [32]. This formulation led to the classic Bernoulli-Euler beam equation:

$$M = EI\kappa \quad (3)$$

where M is the moment (force x length), E is the Young's modulus (force / length²) and I is the second moment of area (length⁴). Euler was then able to characterize all of the elastica formed by various forces and moments, also termed the *inflectional* and *non-inflectional states*, respectively. Figure (3) shows one such elastica with its associated variation in curvature, and subsequent variation in the internal moment, per equation (3). Euler was then equipped to develop elliptic functions and their associated elliptic integrals [33]. These integrals would eventually lead to exact closed-form solutions to the elastica equations.

The various forms of elliptic integrals associated with the elastica have been extensively studied using a variety of techniques. The initial closed-form solutions of the elastica used Jacobi elliptic functions and were found by Saalschutz [34]. The approach of studying the deflection path of beams with forces and moments only applied to the ends has been used by Love [35] and Southwell [36], among others. Timoshenko and Gere [37] used the exact form of the Bernoulli-Euler beam equation, equation (3), to analyze the deflection path. Bisshopp and Drucker [38] and Frisch-Fay [39] also provided exact closed-form solutions to the elliptic integrals which arose

through application of the Bernoulli-Euler beam equation. However, while exact, the use of elliptic integrals proved cumbersome when used in conjunction with compliant mechanism design.

In response, Howell and Midha [40] developed a Pseudo-Rigid-Body Model to incorporate the solutions of elliptic integrals into a conceptual reformulation of the elastic cantilever beam undergoing large deflections, whereby two rigid links, connected at a single joint, could be used to predict the deflection path of the beam end. This followed work by Burns and Crossley [41], who utilized an approximation of the beam's deflection path by assuming that it swept a circular arc, whose radius was equal to $5/6$ the length of the beam. Incorporating work done by Norton [42], a load-deflection relationship was established for large ranges of angular deflection [43]. Currently, however, there is no PRBM which simplifies the analysis of beams experiencing a uniformly distributed load, also termed *heavy beams*.

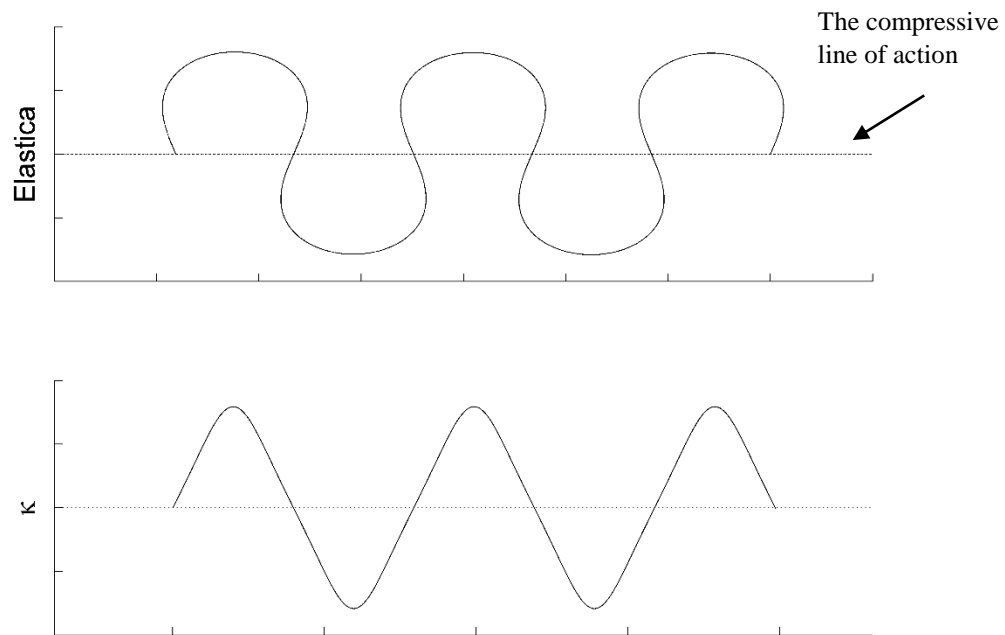


Figure 3 The ribbon-like curve of the elastica and its corresponding curvature. One can notice that the curvature goes to zero at the inflection points of the elastica.

While the structure of the Bernoulli-Euler equation which incorporates a uniformly distributed load, is, algebraically, simple to formulate, its solution is not able to be arrived at by the use of elliptic integrals. There have been several approaches to partial solutions. Falk [44] employed the use of series inversion and power series expansion of the trigonometric terms. This, however, provided only a partial solution as the coefficients of the resulting series, themselves, were never entirely complete, being dependent on each term within the adjoining power series. Solution methods, involving numerical approaches, have been employed by Holden [45]. A solution for beams with a uniformly distributed load normal to the elastica was found by Mitchell [46], and the cases of horizontal and vertical beams under a uniformly distributed load was carried out by Frisch-Fay [47], but, as yet, there remain no closed-form solutions to the general case of the heavy cantilever.

2.3 The Pseudo-Rigid-Body Model

In this section, the prior work done for the PRBM, which describes a cantilever beam whose free end is acted upon by a point force, is presented. The associated parameterized kinematics and load-deflection relationships are then discussed.

The PRBM is a methodology employed to simplify the nonlinear characteristics of beams undergoing large deflections by reducing such beams to connected rigid links which have similar force-deflection characteristics and kinematics [3]. Doing so allows compliant mechanism designs, incorporating nonlinear behavior, to be initialized by first analyzing the kinematics and force-deflection performance of a system of rigid links connected by pin joints. While developed primarily for compliant mechanism design, the PRBM is applicable to any elastic system undergoing large deflections [40].

2.3.1 Parameterized Kinematics

Figure (4) shows the PRBM of a large deflection beam. Here, the pin joint is located a distance γl from the beam tip, where l is the length of the beam and γ represents the fraction of the total length which will undergo circular motion. The product γl is called the *characteristic radius* and γ is called the *characteristic radius factor*. The pin joint itself is called the *characteristic pivot* and the angle through which the characteristic radius moves, Θ , is called the *pseudo-rigid-body angle*. Located at the characteristic pivot is a torsional spring, the stiffness of which governs the

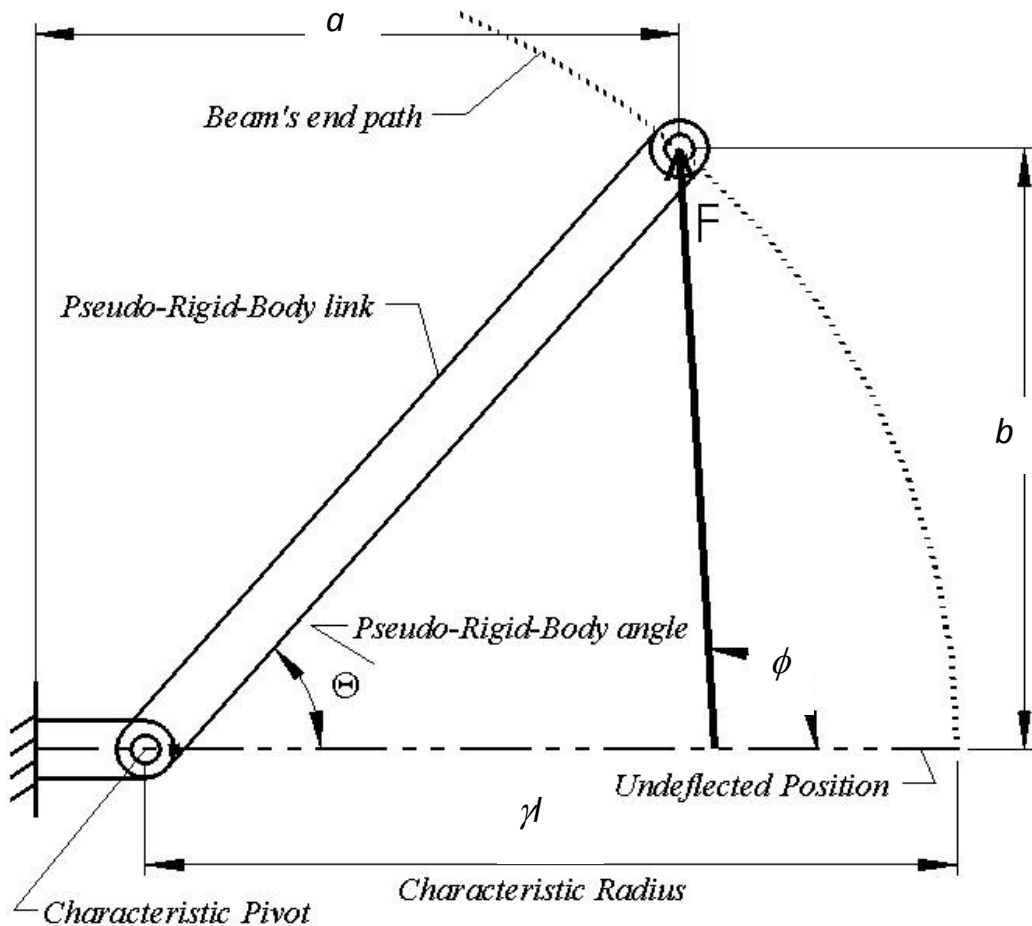


Figure 4 The Pseudo-Rigid-Body Model adapted from Howell [3].

force-deflection characteristics. Incorporating contributions from Su [48], the horizontal coordinate, a , and vertical coordinate, b , of the beam tip are related to γ and Θ by:

$$\frac{a}{l} = 1 - \gamma(1 - \cos\Theta) \quad (4)$$

$$\frac{b}{l} = \gamma \sin\Theta \quad (5)$$

$$\gamma = \frac{\left(\frac{b}{l}\right)^2 + \left(1 - \frac{a}{l}\right)^2}{2\left(1 - \frac{a}{l}\right)} \quad (6)$$

$$\Theta = \tan^{-1}\left(\frac{b}{a - l(1 - \gamma)}\right) \quad (7)$$

Using these relationships, an optimization routine was developed to express γ solely as a function of the load angle ϕ [40], which is expressed relative to the frame of reference found at the wall. The parameter γ actually changes with both ϕ and Θ , however, so an angular deflection limit, or Θ_{max} , was sought for each degree of the load angle. A relative error criteria of 0.5% of the deflection path, δ_e , as calculated from elliptic integrals, was established. The Golden Section elimination method was employed to find the value of γ for which Θ_{max} was maximized, but the relative error remained less than 0.5% for all values of Θ below Θ_{max} . The equation for the relative error is:

$$\frac{\text{error}}{\delta_e} = \frac{\sqrt{\left(\frac{a}{l} - (1 - \gamma(1 - \cos\Theta))\right)^2 + \left(\frac{b}{l} - \gamma \sin\Theta\right)^2}}{\sqrt{\left(1 - \frac{a}{l}\right)^2 + \left(\frac{b}{l}\right)^2}} \quad (8)$$

Using this routine, constant values of γ could be produced for each value of ϕ , thereby excluding the dependence of γ on Θ , and rendering γ a mere function of ϕ . Through polynomial fits, a series of three equations described $\gamma(\phi)$ in a piecewise fashion for three connected regions of ϕ , with a range of $0^\circ < \phi < 180^\circ$. Once the Θ_{max} values were known for each value of ϕ , a parametric approximation of the beam end angular deflection could be calculated following:

$$\theta_0 = c_\theta \Theta \quad (9)$$

Here, c_θ is termed the *parametric angle coefficient* and describes the near-linear relationship between the beam tip angular deflection, θ_0 , and Θ . The value of c_θ , for a given load angle ϕ , was determined by fitting the near-linear curve of Θ vs θ_0 with a line. This was done by using a least-squares regression, extending the fit to the Θ_{max} for that value of ϕ , and setting the slope of that line equal to c_θ . The PRBM, up to this point, allows one to take the PRBM parameters γ , Θ and c_θ , and transform these values into the actual beam variables a , b , and θ_0 . Modifications to both $\gamma(\phi)$ and $c_\theta(\phi)$ were later made for highly tensile load angles [49], but also reflected the use of piecewise solutions in ϕ .

2.3.2 Parameterized Load-Deflection Relationships

The relationship between the applied load and the subsequent deflection of the characteristic radius has been able to be described through the use of the non-dimensional transverse load index $(\alpha^2)_t$ [42] where:

$$(\alpha^2)_t = \frac{F_t l^2}{EI} \quad (10)$$

and

$$F_t = F \sin(\phi - \Theta) \quad (11)$$

Here F is defined as the force, E is Young's Modulus, and I is the second moment of area. It was noted that a slowly varying relationship, one which could be approximated as linear, existed between the pseudo-rigid-body angle, Θ , and the transverse load index, $(\alpha^2)_t$, which can be described by:

$$(\alpha^2)_t = K_\Theta \Theta \quad (12)$$

Meaning:

$$K_\Theta \Theta = \left(\frac{F l^2}{EI} \right) \sin(\phi - \Theta) \quad (13)$$

Here, K_Θ is termed the *stiffness coefficient* and is a non-dimensional quantity relating the transverse force component to the angular deflection of the PRBM. The stiffness coefficient itself relates to

the dimensional spring stiffness of the torsional spring, located at the characteristic pivot, by conducting a torque analysis via:

$$T = K\Theta = F_t \gamma l \quad (14)$$

Here, T is defined as the torque with units of (force x length) and K is the torsional spring constant with units of (force x length)/radian. Solving for F_t , and substituting the resulting expression into equation (13) leads to:

$$K = \frac{\gamma K_\Theta EI}{l} \quad (15)$$

This equation relates the stiffness coefficient to the torsional spring stiffness. The stiffness coefficient was also parameterized with respect to ϕ . At each degree of load angle ϕ , a constant value of K_Θ was recommended and another piecewise application of polynomial equations in ϕ was produced. This method thus divided the angular range of ϕ , again being $0^\circ < \phi < 180^\circ$, into three sections, akin to the parameterization of $\gamma(\phi)$ [43]. Thus, if one knows the load angle, one knows $K_\Theta(\phi)$, and can thereby define the relationship between the angular deflection and the magnitude of the load. However, a new Θ_{max} was recorded at different values of ϕ , each of which were lower than the values of Θ_{max} recorded for γ , often times significantly. For example at $\phi = 33.7^\circ$, $\Theta_{max}(\gamma) = 28.7^\circ$ and $\Theta_{max}(K_\Theta) = 23.6^\circ$, with a difference of only 5.1° . At $\phi = 135^\circ$, however, $\Theta_{max}(\gamma) = 94.8^\circ$ and $\Theta_{max}(K_\Theta) = 67.5^\circ$, with a difference of 27.3° . In other words, the angular deflection range of the parameterized kinematic model exceeds that of the parameterized force-

deflection relationship, with a discrepancy that increases as the load angles become increasingly compressive. Modifications to $K_{\theta}(\phi)$ were later made, again for highly tensile loads, which, again, reflected piecewise solutions in ϕ [49].

2.4 Cantilever Beam Differential Equations

The planar cantilever beam chosen for this analysis follows Bernoulli-Euler beam assumptions and is thus isotropic, inextensible and elastic, with a constant Young's Modulus E and a constant second moment of area I . Here, the uniformly distributed load w , which has units of (force/length), and the end force F are always pointed in the same direction, with a dot product always equal to unity. The cantilever beam seen in Figure (5) shows three angles, θ , ϕ and ϕ_0 along with two reference frames B and T , designating the base and tip frames, respectively. While the values of the angles are invariant under coordinate transformations, the x and y coordinates of the beam tip are not. This will later play a role in the formulation of the PRBM. The angular deflection of the beam tip, with respect to the x -axis in the B -frame, which is located an arc length distance s from the tip itself, is designated θ . The angle that the loads make, with respect to the x -axis in the T -frame, is designated ϕ_0 and the angle that the loads make, with respect to the x -axis in the B -frame, is designated ϕ . These angles follow a simple relation:

$$\theta = \phi - \phi_0 \quad (16)$$

As the numerical integration will always be carried out from the tip to some arbitrary arc length s , the angle that the loads make, with respect to the beam tip, will remain constant. Loads which have a constant angle with respect to the beam tip, regardless of beam deformation, are termed *follower loads* and result in the following differential relationships:

$$\frac{d\phi_0}{ds} = 0 \quad (17)$$

And therefore:

$$\frac{d\theta}{ds} = \frac{d\phi}{ds} \quad (18)$$

Under these definitions, the following system of differential equations can be used to describe the bending of a Bernoulli-Euler beam along its neutral axis, as a function of arc length:

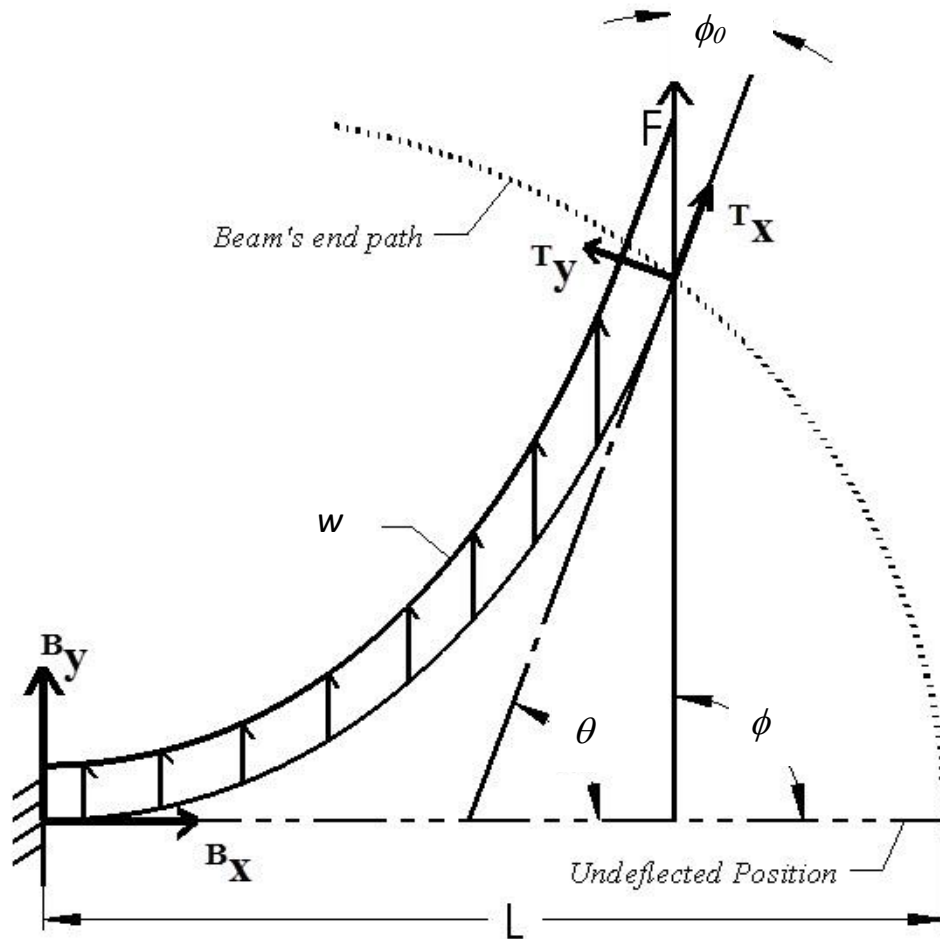


Figure 5 Cantilever beam with uniformly distributed load w and end force F pointing in the same direction.

$$\frac{d\theta}{ds} = \frac{d\phi}{ds} = \kappa \quad (19)$$

$$\frac{dM}{ds} = EI \frac{d\kappa}{ds} = EI \frac{d^2\phi}{ds^2} = (F + ws)\sin(\phi) \quad (20)$$

$$\frac{d^T x}{ds} = -\cos(\theta) = -\cos(\phi - \phi_0) \quad (21)$$

$$\frac{d^T y}{ds} = \sin(\theta) = \sin(\phi - \phi_0) \quad (22)$$

where κ is the curvature of the beam at an arc length distance s from the beam tip, M is the moment, $^T x$ is the horizontal coordinate of the neutral axis an arc length distance s from the beam tip (measured in the tip frame T), while $^T y$ is the vertical coordinate of the neutral axis an arc length distance s from the beam tip (again, measured in the tip frame T). Here, there is a negative sign present in equation (21), but absent from equation (22). This results from $^T x$ becoming increasingly negative as s increases away from the beam tip, while $^T y$ becomes increasingly positive.

CHAPTER 3: NON-DIMENSIONAL FORM OF DIFFERENTIAL EQUATIONS

In the force only case, the non-dimensional form of the differential equations can be arrived at by use of a characteristic length scale [50], expressed as:

$$\lambda_1 = \sqrt{\frac{EI}{F}} \quad (23)$$

Here, λ_1 has units of length and can, therefore, be used to cancel the units of both s and κ , where κ has units of (1/length). In the distributed load only case, the characteristic length scale can be expressed as:

$$\lambda_2 = \sqrt[3]{\frac{EI}{w}} \quad (24)$$

which, again, has units of length. However, in the combined loading case, a characteristic length scale must be chosen which both has units of length and reduces to λ_1 or λ_2 when the distributed load goes to zero or the force goes to zero, respectively. These requirements lead to the following expression:

$$\lambda_3 = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\sqrt{\frac{EI}{F}} \sqrt[3]{\frac{EI}{w}}}{\sqrt{\frac{EI}{F}} + \sqrt[3]{\frac{EI}{w}}} \quad (25)$$

As can be seen, λ_3 has units of length. However, as either F or w approach zero, both the numerator and denominator approach infinity. The limit can be evaluated if both the numerator and denominator are multiplied by the inverse of the term approaching infinity. Hence:

$$\lim_{F \rightarrow 0} \lambda_3 = \lim_{F \rightarrow 0} \frac{\sqrt[3]{\frac{EI}{w}}}{1 + \sqrt[3]{\frac{EI}{w}} \sqrt{\frac{F}{EI}}} = \lambda_2 \quad (26)$$

Similarly:

$$\lim_{w \rightarrow 0} \lambda_3 = \lim_{w \rightarrow 0} \frac{\sqrt{\frac{EI}{F}}}{\sqrt[3]{\frac{w}{EI}} \sqrt{\frac{EI}{F}} + 1} = \lambda_1 \quad (27)$$

Thus, λ_3 possesses the desired properties and can now be termed the *reduced characteristic length scale*. Applying λ_3 to equation (20) and dividing by EI results in:

$$\lambda_3^2 \frac{d\kappa}{ds} = \frac{d\tilde{\kappa}}{d\tilde{s}} = \left(\frac{\lambda_3^2}{\lambda_1^2} + \frac{\lambda_3^3}{\lambda_2^3} \frac{s}{\lambda_3} \right) \sin(\phi) \quad (28)$$

The use of the tilde in equation (28) indicates that the marked variable is now dimensionless. Reformulating equation (28) can prove useful by applying the following non-dimensional parameter:

$$\mu = \frac{\lambda_2}{\lambda_1 + \lambda_2} \quad (29)$$

Here, μ represents the fraction of the sum of the length scales, associated with both F and w , which is contributed by w alone, and may, therefore, be termed the *distributed contribution factor* (DCF). The DCF obtains useful values in the limit that either the distributed load or the end force approach zero, in that:

$$\lim_{F \rightarrow 0} \mu = \frac{\lambda_2}{\infty} = 0 \quad (30)$$

and

$$\lim_{w \rightarrow 0} \mu = \lim_{w \rightarrow 0} \frac{\sqrt[3]{\frac{EI}{w}}}{\sqrt{\frac{F}{EI}} + \sqrt[3]{\frac{EI}{w}}} = \lim_{w \rightarrow 0} \frac{1}{\sqrt[3]{\frac{w}{EI}} \sqrt{\frac{F}{EI}} + 1} = 1 \quad (31)$$

In other words, when $w = 0$, $\mu = 1$, and when $F = 0$, $\mu = 0$. Any intermediate value of μ , between zero and one, represents a combination of the magnitudes of F and w for a given EI . Figure (6) shows the surface of μ versus F and w . Applying the DCF to equation (28) and applying λ_3 to equations (19), (21) and (22) results in the final system of equations which are employed in constructing the phase space and evaluating the beam path:

$$\frac{d\phi}{d\tilde{s}} = \tilde{\kappa} \quad (32)$$

$$\frac{d\tilde{\kappa}}{d\tilde{s}} = \frac{d^2\phi}{d\tilde{s}^2} = (\mu^2 + (1 - \mu)^3 \tilde{s}) \sin(\phi) \quad (33)$$

$$\frac{d\tilde{s}}{d\tilde{s}} = 1 \quad (34)$$

$$\frac{d\mu}{d\tilde{s}} = 0 \quad (35)$$

$$\frac{d\phi_0}{d\tilde{s}} = 0 \quad (36)$$

$$\frac{d^T \tilde{x}}{d\tilde{s}} = -\cos(\phi - \phi_0) \quad (37)$$

$$\frac{d^T \tilde{y}}{d\tilde{s}} = \sin(\phi - \phi_0) \quad (38)$$

Here, equations (32)-(34) are used to define the phase space, while equations (37) and (38) are used to define the beam path.

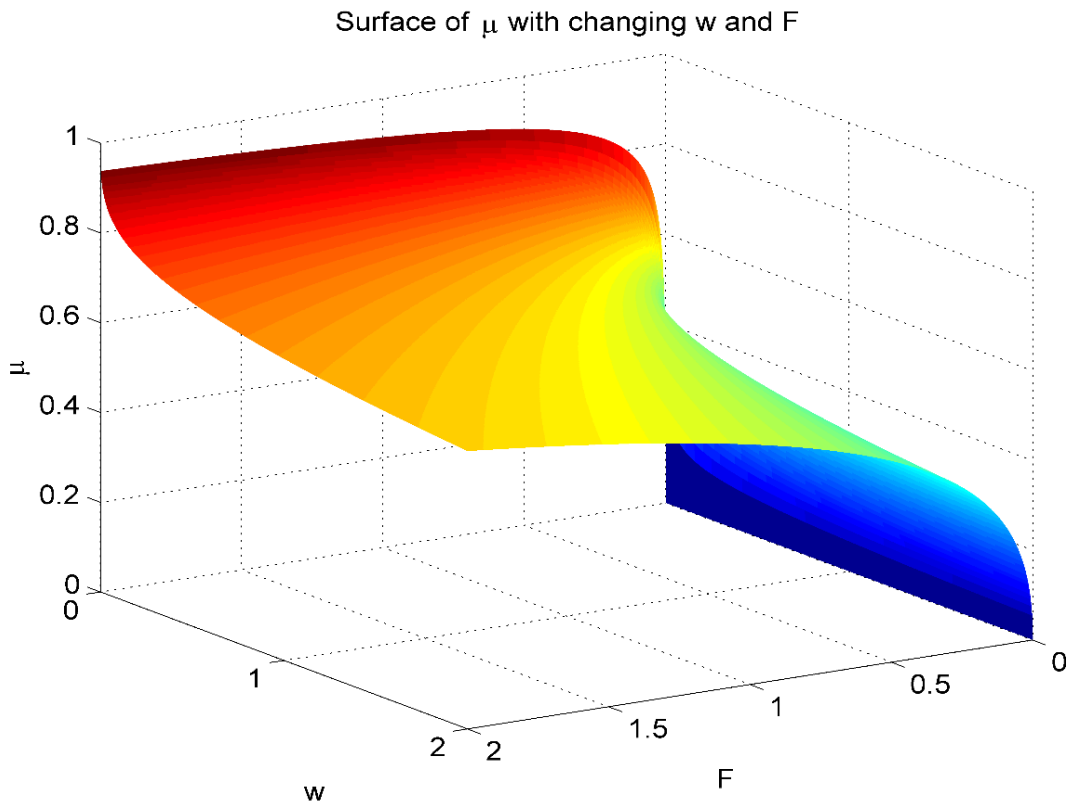


Figure 6 The distributed contribution factor versus w and F .

CHAPTER 4: PHASE SPACE AND INITIAL VALIDATIONS

The phase space of a system of equations, often dynamical, can be defined as the space in which all possible states of the system are depicted [51]. Every possible state of the system corresponds to unique phase space coordinates, the variety of which is represented by the dimension of the phase space. The phase space for Bernoulli-Euler beams undergoing a single end force has been used before to describe the system's mechanics. It was determined to be analogous to the phase space for that of a pendulum [52].

With regards to this system, the term *phase portrait* is a more precise description, as the system is absent a dependence on \tilde{s} , being thus labeled *autonomous*, and can, therefore, be represented on a plane. Figure (7) shows the phase portrait for the force only case. The curves shown are termed *trajectories* and represent the evolution of a set of initial conditions as \tilde{s} increases. Figure (7) and all subsequent figures and analyses are the results of numerically integrating the system of equations defined by equations (32)-(38) using the 'ode45' command in the commercial programming software Matlab. This command implements a Runge-Kutta (4,5) iterative solving routine for ordinary differential equations. The use of all four quadrants in Figure (7a) displays the periodicity of the trajectories of this system and physically represents the inflectional states, hence containing states of inflection where the moment goes to zero, of an infinite elastica undergoing buckling [35]. Non-inflectional states are beyond the scope of this work. This periodic behavior, however, can also be the result of symmetry. If a cantilever were used instead of an infinite elastica, the entire inflectional phase portrait could be reproduced by

merely inverting the coordinate systems by which the moments and/or force directions are defined as positive. Because of this symmetry, it is, therefore, allowable to use only a single quadrant to accurately analyze the behavior of the system. Figure (3b) shows the use of the second quadrant, as this represents the variable definitions and coordinate system orientations used by the PRBM [3] and those shown in Figure (5). This quadrant will, therefore, be used henceforth and represents

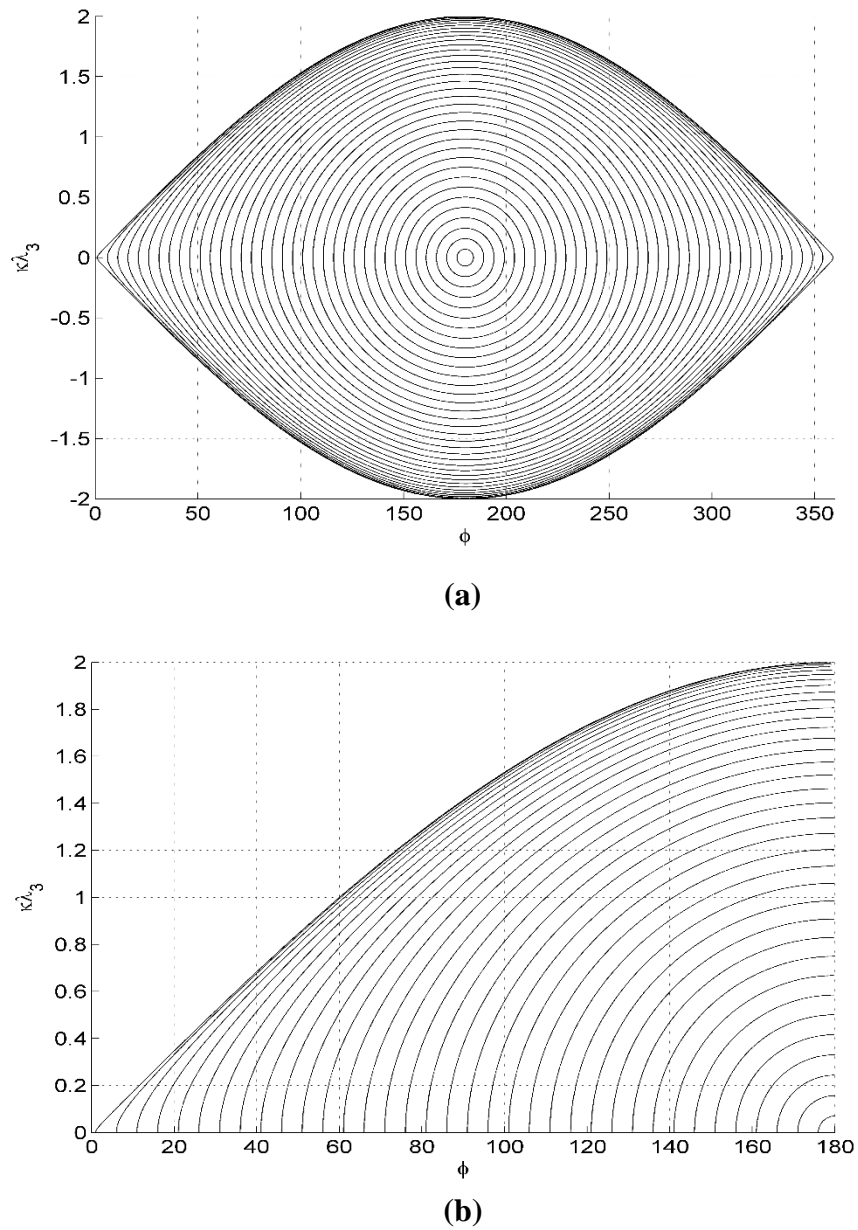


Figure 7 Phase portraits for the force only case. Part (a) shows the entire inflectional phase plane for the force only case. Part (b) shows the second quadrant in greater detail.

positive moments and positive vertical components of the force directions, relative to both the T -frame and the B -frame. Here, trajectories begin at $\tilde{s} = 0$, $\tilde{\kappa} = 0$, and $\phi = \phi_0$, which represents the beam tip. Every other point of the phase portrait, above the horizontal axis, now describes a potential base of the beam, or wall, with some value of curvature, corresponding to the maximum internal moment, some non-zero value of arc length, and some $\phi > \phi_0$.

The phase space for the distributed load only case is, however, of a very different nature from that of the force only case. This is due to the presence of \tilde{s} on the right-hand side of equation (33), which can now be described as a second order, non-autonomous, nonlinear differential equation. Equations (32)-(34) define a three dimensional phase space for a given value of μ . The phase space coordinates of this system are \tilde{s} , $\tilde{\kappa}$, and ϕ . Figure (8) shows a single trajectory for the distributed load only case against a backdrop of the force only phase portrait, for comparative purposes. This trajectory has the initial conditions of $\tilde{s} = 0$, $\tilde{\kappa} = 0$, and $\phi_0 = 45^\circ$. The trajectory then extends to $\tilde{s} = 50$. The lack of periodicity is noticeable. This may be interpreted, in a dimensional sense, as the magnitude of the force on the beam, caused by the product of ws , increasing as s increases. This is not behavior witnessed in the force only case. However, the second quadrant can still be employed, as the cantilever states under consideration always have their maximum internal moments, and, therefore, curvatures, located at the wall. Protrusions of trajectories from the second quadrant into the first would violate this principle and would thus correspond to buckling. Such behavior is beyond the scope of this paper.

To summarize, within the second quadrant, all states corresponding to load angles between 0° and 180° are represented. Additionally, all cantilever beams whose maximum curvature is located at the base, or wall, are represented in the second quadrant. Lastly, the definitions of

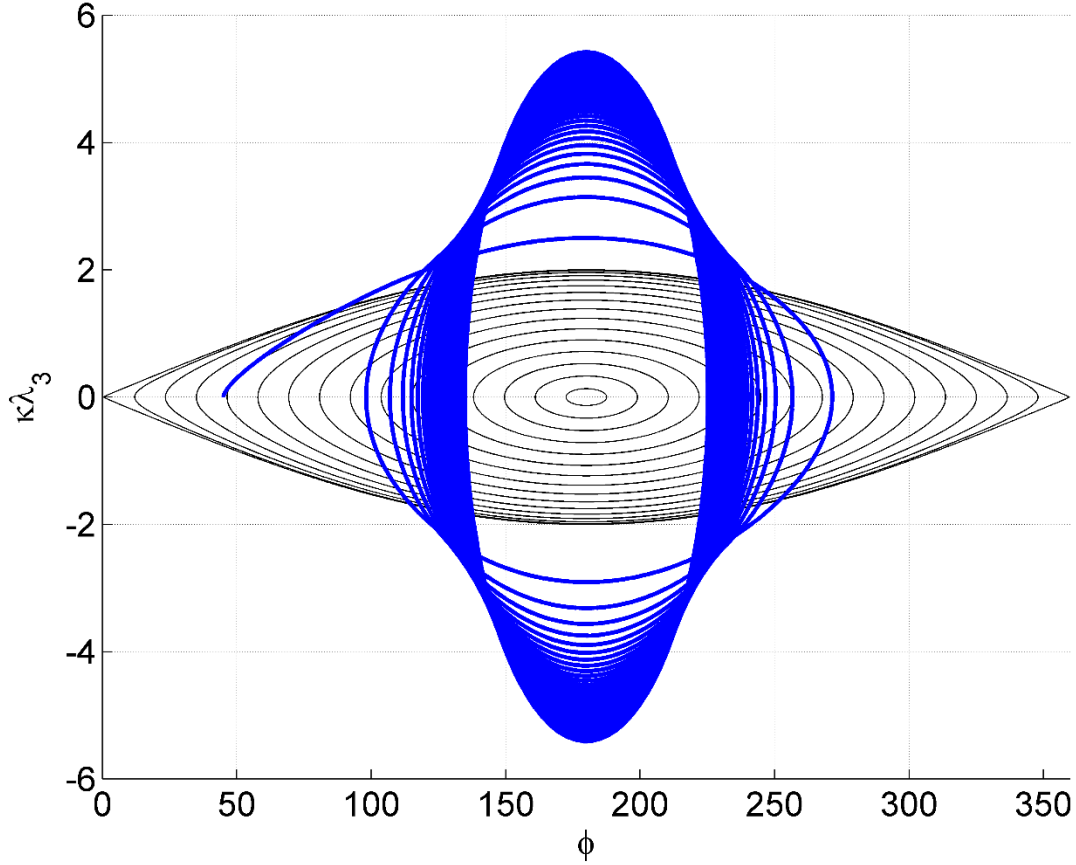


Figure 8 Depiction of a single trajectory originating at $\pi/4$ for the distributed load only case.

positive forces and moments, in conjunction with a right-handed coordinate system, where $\tilde{\kappa}$ increases upward and ϕ increases to the right, are represented in the second quadrant.

An initial validation of the governing differential equations can be provided by the commercial software ANSYS. This program uses finite element analysis (FEA) and can determine the deflections of cantilever beams under a variety of loads. A line body composed of 70 beam elements was chosen for the analysis, with a length of $L = 0.44174$ m, a Young's Modulus of $E = 200$ GPa and a second moment of area of $I = 5.4752 \times 10^{-13}$ m⁴. Figure (9) shows the deformation of this line body with $\phi = 105^\circ$. Load angles were chosen at 15° increments, ranging from 15° to 165° , with respect to the B frame, and are designated ϕ_B . At each beam node, the vertical and

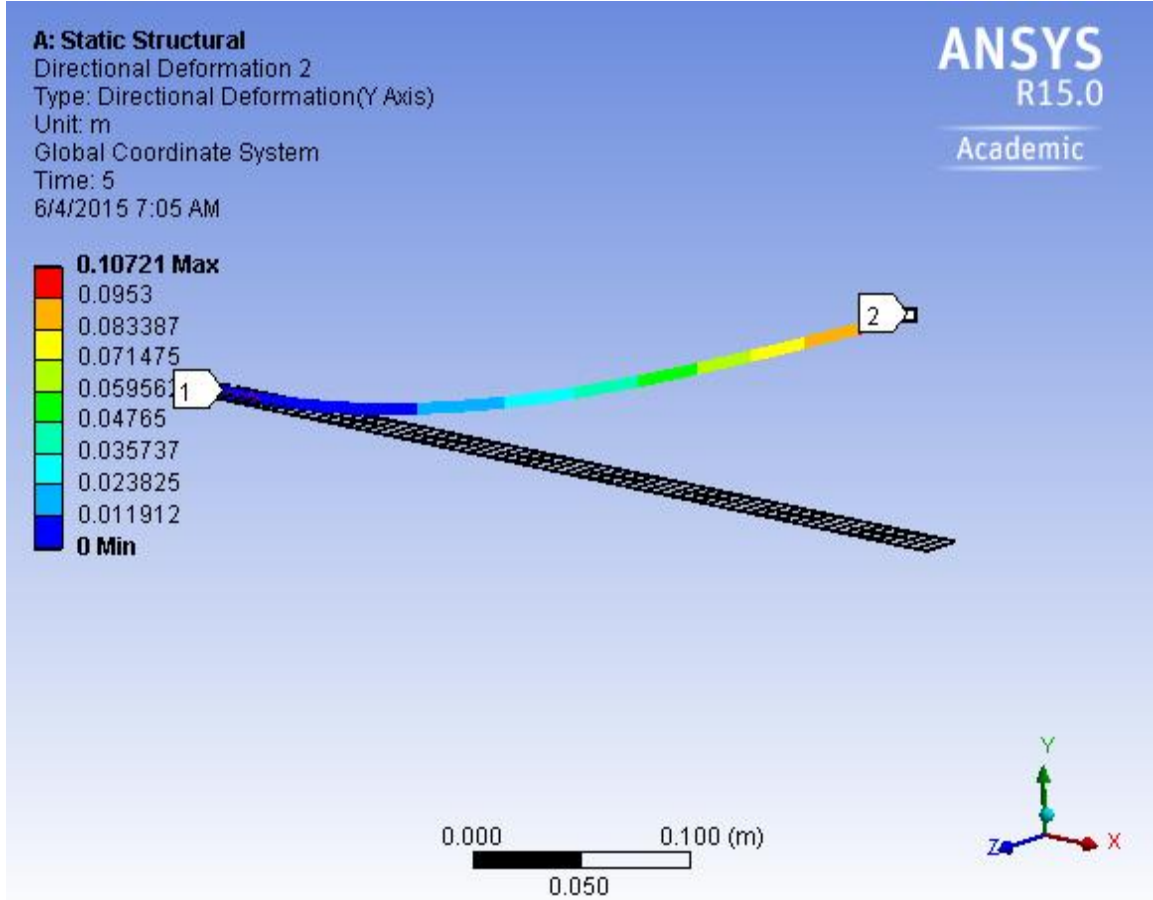


Figure 9 The results from a single trial in ANSYS with $\phi = 105^\circ$.

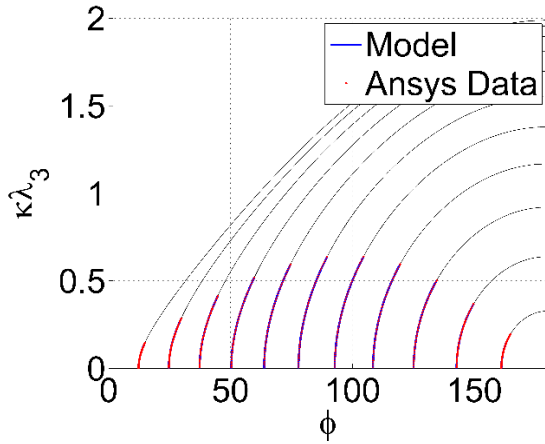
horizontal deflections, Δy_i and Δx_i , respectively, are measured, as is the moment M_i . That information can be transformed into the phase space coordinates for each element i , where $i = 1$ is located at the beam tip, by the following equations:

$$\tilde{\kappa}_i = \frac{\lambda_3 M_i}{EI} \quad (39)$$

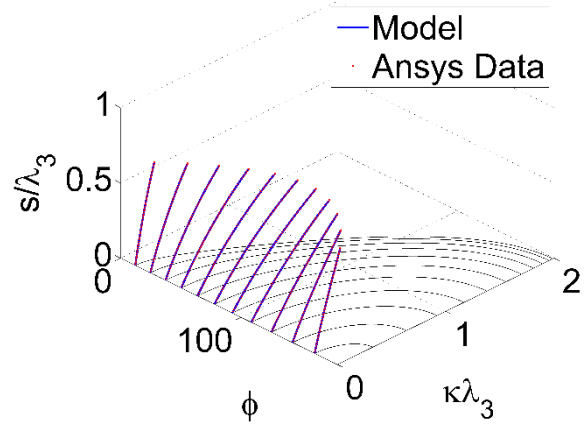
$$\tilde{s}_i = \frac{s_i}{\lambda_3} \quad (40)$$

$$\phi_i = \phi_B - \cos^{-1} \left(\frac{[(s_{71-i} + \Delta x_i) - (s_{71-(i+2)} + \Delta x_{i+2})]}{\sqrt{[(s_{71-i} + \Delta x_i) - (s_{71-(i+2)} + \Delta x_{i+2})]^2 + [\Delta y_i - \Delta y_{i+2}]^2}} \right) \quad (41)$$

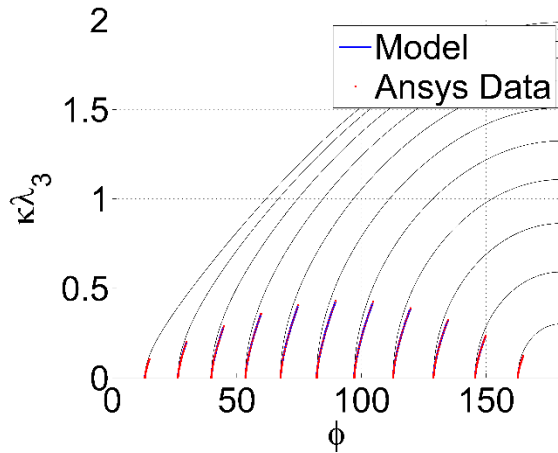
where $s_{71-i} = s_{70} = L$, when $i = 1$ at the beam tip. Here, equation (41) utilizes a central divided difference scheme to calculate ϕ , and, therefore, cannot do so at the first and last points of the beam. This equation provides only an approximation of ϕ , as an instantaneous slope is not provided from the FEA at each node. The approximation error increases as the curvature increases and can, therefore, be expected to diverge from the predicted values as $\tilde{\kappa}$ becomes larger. Increasing the number of nodes will increase the precision of equation (41), but any finite number of nodes will inevitably be an approximation. However, the approximation is close enough to qualitatively evaluate the agreement between equations (32)-(34) and the non-dimensional phase space coordinates calculated from equations (39)-(41), thereby justifying further investigation. The inputs into the differential equation solver, ode45, were the initial conditions of the beam tip. The solver was then tasked with performing the numerical integration for values of \tilde{s} ranging from 0 to \tilde{s}_{L-1} , in increments of 0.01 and with a maximum step size of 0.01. Figures (10a) and (10b) show the force only case, with $F = 0.24$ N and with $w = 10^{-100}$ N/m, as written in Matlab, and with $w = 0$ N/m, as written in ANSYS. The Matlab value of w was chosen to avoid singularities associated with calculating λ_3 term by term and results in $\mu = 1$. The force only phase portrait is provided as a backdrop for all plots in Figures (10) and (11), again for comparative purposes. While, for the force only case, evaluation in the dimension of \tilde{s} is unnecessary, it does prove useful in confirming the three dimensional agreement between ANSYS and the differential equations, as seen in Figure (10b). The distributed load only case is shown in Figures (10c) and (10d). Here, $w = 1.062355767$ N/m and $F = 10^{-100}$ N, as written in Matlab, while $F = 0$ N, as written in ANSYS. This results in



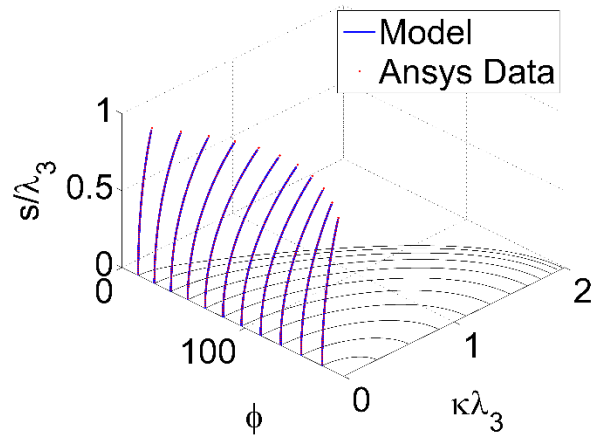
(a)



(b)



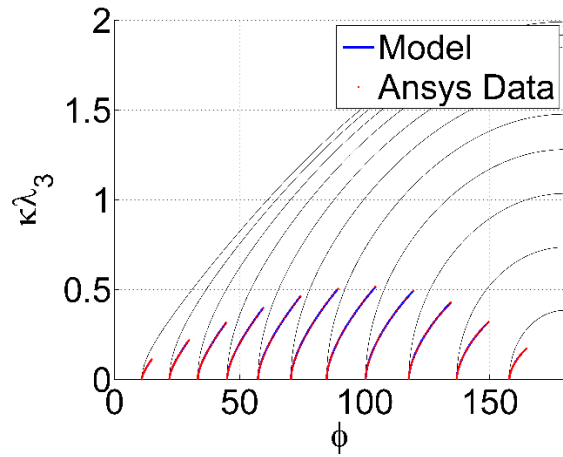
(c)



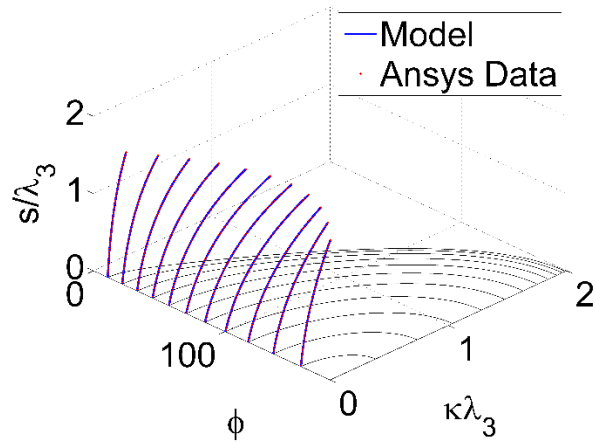
(d)

Figure 10 Phase space trajectory comparison with FEA for $\mu = 1$ and $\mu = 0$. Parts (a) and (b) depict trajectories for the force only case with $\mu = 1$. Parts (c) and (d) depict the distributed load only case with $\mu = 0$.

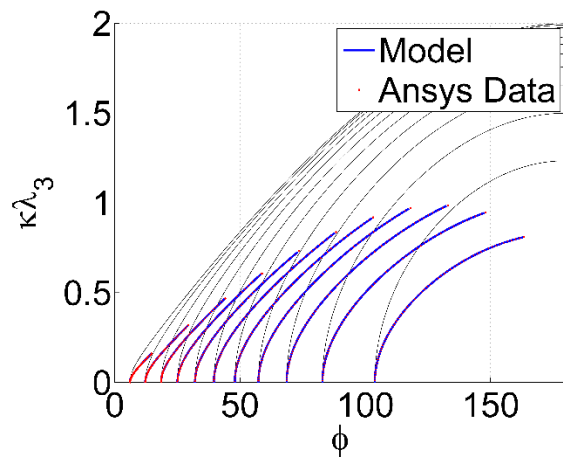
$\mu = 1.4169 \times 10^{-50}$. Figures (11a) and (11b) show a combined loading case, with $F = 0.24$ N and $w = 1.062355767$ N/m, which results in $\mu = 0.40973$. Figures (11c) and (11d) show another combined loading case, with $F = 1.2$ N and $w = 1.062355766$ N/m, which results in $\mu = 0.60817$. For each of these cases, an out of plane view is provided which shows the evolution of the trajectories as \tilde{s}



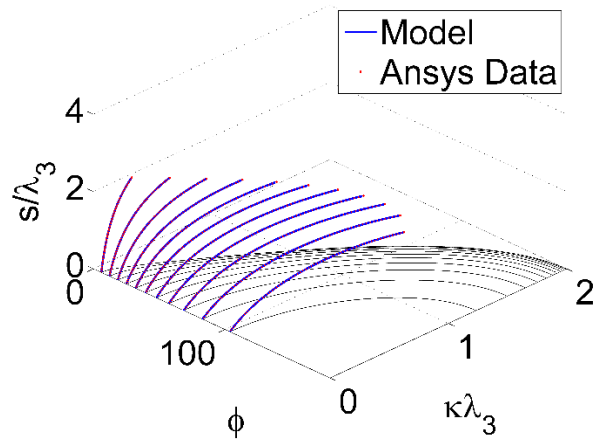
(a)



(b)



(c)



(d)

Figure 11 Phase space trajectory comparison with FEA for $\mu = 0.40973$ and $\mu = 0.60817$. Parts (a) and (b) depict trajectories for $\mu = 0.40973$. Parts (b) and (c) depict trajectories for $\mu = 0.60817$.

increases and demonstrates the close adherence of the ANSYS results to the predictions of the differential equations, even for approximate values of ϕ .

CHAPTER 5: THE CONTOUR METHOD

The method for extending the PRBM into situations of combined loading will now be described. The aim of this method was to develop expressions for the aforementioned PRBM parameters, but now in terms of ϕ and μ . While the PRBMs previously discussed employed the use of closed-form solutions to perform error analysis, i.e. elliptic integrals, this method was not available for the combined loading situations investigated. Numerical integration of the governing equations was, therefore, employed so that three dimensional surfaces of the relevant information could be constructed and their level-curves, or contours, could be analyzed. Based on this analysis, additional spatial surfaces were constructed and fit to equations for $\gamma(\phi, \mu)$, $c_d(\phi, \mu)$, and $K_e(\phi, \mu)$. Where appropriate, results will be compared with prior models. Correction factors will also be discussed, with their relevance and form described.

5.1 The Data Structure

Numerically solving equations (32)-(36) allows for the population of the phase space with an arbitrary number of trajectories, which evolve to an arbitrary extent. However, the primary benefit of such an approach is not to merely view the geometry of the phase space, but to extract information from it, to manipulate that information to suit one's needs, and then to use the data structure containing the values of the phase space coordinates themselves, \tilde{s} , \tilde{k} , and ϕ , to store the results of those manipulations. Doing so allows a direct correlation, within the data structure, between the locations of the results of manipulations and the locations of the phase space

coordinates. This can be demonstrated as follows; let the input vector into the differential equation solver resemble the initial conditions of these variables:

$$Inputs = [\phi \quad \tilde{\kappa} \quad \tilde{s} \quad \mu \quad \phi_0 \quad {}^T\tilde{x} \quad {}^T\tilde{y}] \quad (42)$$

The outputs of the solver will evolve according to equations (32)-(38), where ϕ , $\tilde{\kappa}$, \tilde{s} and ${}^T\tilde{y}$ will become increasingly positive, μ and ϕ_0 will remain the same, and ${}^T\tilde{x}$ will become increasingly negative. The solver will output values for the vector of input variables based on their initial conditions and according to the prescribed values of \tilde{s} , which range from 0 to 7 in increments of 0.01. As the solver outputs new values of each of the input variables, it stores them as column vectors for each variable. For \tilde{s} , this resembles:

$$\tilde{s}_{\phi_0, \mu} = \begin{bmatrix} 0 \\ 0.01 \\ \vdots \\ 7 \end{bmatrix} \quad (43)$$

Hence, for a given μ and ϕ_0 , a column vector of \tilde{s} values will be generated. The same will occur for all of the other input variables. If ϕ_0 is allowed to change from 1° to 179° , in 0.5° increments, then a matrix of \tilde{s} values will be generated. This matrix will have a size of 701 rows by 357 columns and will correspond to a single value of μ as:

$$\tilde{s}_\mu = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0.01 & 0.01 & \dots & 0.01 \\ \vdots & \vdots & \ddots & \vdots \\ 7 & 7 & 7 & 7 \end{bmatrix} \quad (44)$$

By stacking these matrices for changing values of μ , from 0 to 1 in increments of 0.02, the overall data structure of a single variable, \tilde{s} , has been created. This structure has a total size of 701 rows, 357 columns and 51 stacks. As long as every other variable is stored in exactly the same fashion, there will be a one-to-one positional correspondence between the elements of different variables within the data structure. Hence, for example:

$$\tilde{s}_{row,col,stack} \rightarrow \tilde{\kappa}_{row,col,stack} \quad (45)$$

This means that the non-dimensional curvature located at some (row, col, stack) coordinate corresponds to the non-dimensional arc length located at the same (row, col, stack) coordinate. These, taken together with the value of ϕ located at the same (row, col, stack) coordinate, will correspond to the phase space coordinates of one point along a single trajectory. This trajectory began at some data coordinates of (1, col, stack), where col is relatable to the value of ϕ_0 and stack is relatable to the value μ , and progresses to (row, col, stack) via the incremental progression of \tilde{s} .

Using this method, the entire second quadrant of the phase plane can be populated with information regarding a beam's curvature, the horizontal and vertical coordinates, the length, the B -frame load angle, etc. All of that information can be stored, accessed and manipulated via

consistent reference to the coordinates of the data contained within each variable. For instance, the deflection angle can be stored at every data location within the phase space via:

$$\theta_{ijk} = \phi_{ijk} - \phi_{0,ijk} \quad (46)$$

where the subscripts i, j , and k will henceforth denote row, col, and stack for the sake of simplicity. At this point, it is worth noting that every column of each variable was truncated, from one row above to every row below the row number of ϕ in which ϕ became larger than 180° . This ensured that only the second quadrant of the phase plane was used. It is also worth noting that the first three rows of all variables were removed so that numerical instabilities, associated with using PRBM equations containing variables of then near zero values, could be avoided. Hence, while the differential equation solver was initialized at $\tilde{s} = 0$, the data storage initializes at $\tilde{s} = 0.03$. The value of 0.02 is two orders of magnitude below the observed maximum values of \tilde{s} and corresponds to regions of very small deflections. A loss of generality was, therefore, deemed negligible.

5.2 Finding $\gamma(\phi, \mu)$

The precise value of γ can be found at each location within the phase space by using equation (6). Notice that equation (6) is absent any dependence on the pseudo-rigid-body angle, θ , but does depend on a and b , which are the x -axis and y -axis coordinates of the beam tip, as measured in the B frame. At this point, however, the x -axis and y -axis coordinates available are those of the location of the base, as measured from the tip frame. To convert ${}^T\tilde{x}$ and ${}^T\tilde{y}$ into the non-dimensional a and b coordinates, a clock-wise rotation of the reference frame T must be made by an amount equal to the beam's deflection angle, θ . The negative of both of the resulting x -axis

and y -axis coordinates will be equal to the non-dimensional a and b . This was done for every element within the data structures of the ${}^T \tilde{x}$ and ${}^T \tilde{y}$ variables as follows:

$$\begin{bmatrix} \tilde{a}_{ijk} \\ \tilde{b}_{ijk} \end{bmatrix} = (-1) \begin{bmatrix} \cos(\phi_{ijk} - \phi_{0,ijk}) & \sin(\phi_{ijk} - \phi_{0,ijk}) \\ -\sin(\phi_{ijk} - \phi_{0,ijk}) & \cos(\phi_{ijk} - \phi_{0,ijk}) \end{bmatrix} \begin{bmatrix} {}^T \tilde{x}_{ijk} \\ {}^T \tilde{y}_{ijk} \end{bmatrix} \quad (47)$$

The precise value of γ was then calculated for every element within the data structure by first recognizing:

$$\frac{a_{ijk}}{l_{ijk}} = \frac{a_{ijk}}{s_{ijk}} = \frac{\tilde{a}_{ijk}}{\tilde{s}_{ijk}} \quad (48)$$

$$\frac{b_{ijk}}{l_{ijk}} = \frac{b_{ijk}}{s_{ijk}} = \frac{\tilde{b}_{ijk}}{\tilde{s}_{ijk}} \quad (49)$$

therefore:

$$\gamma_{ijk} = \frac{\left(\frac{\tilde{b}_{ijk}}{\tilde{s}_{ijk}}\right)^2 + \left(1 - \frac{\tilde{a}_{ijk}}{\tilde{s}_{ijk}}\right)^2}{2\left(1 - \frac{\tilde{a}_{ijk}}{\tilde{s}_{ijk}}\right)} \quad (50)$$

Figure (12) shows a qualitative example of the utility of this method. Here, the phase space coordinate \tilde{s} is replaced by γ . The variation of γ with changing ϕ and $\tilde{\kappa}$, for the distributed load

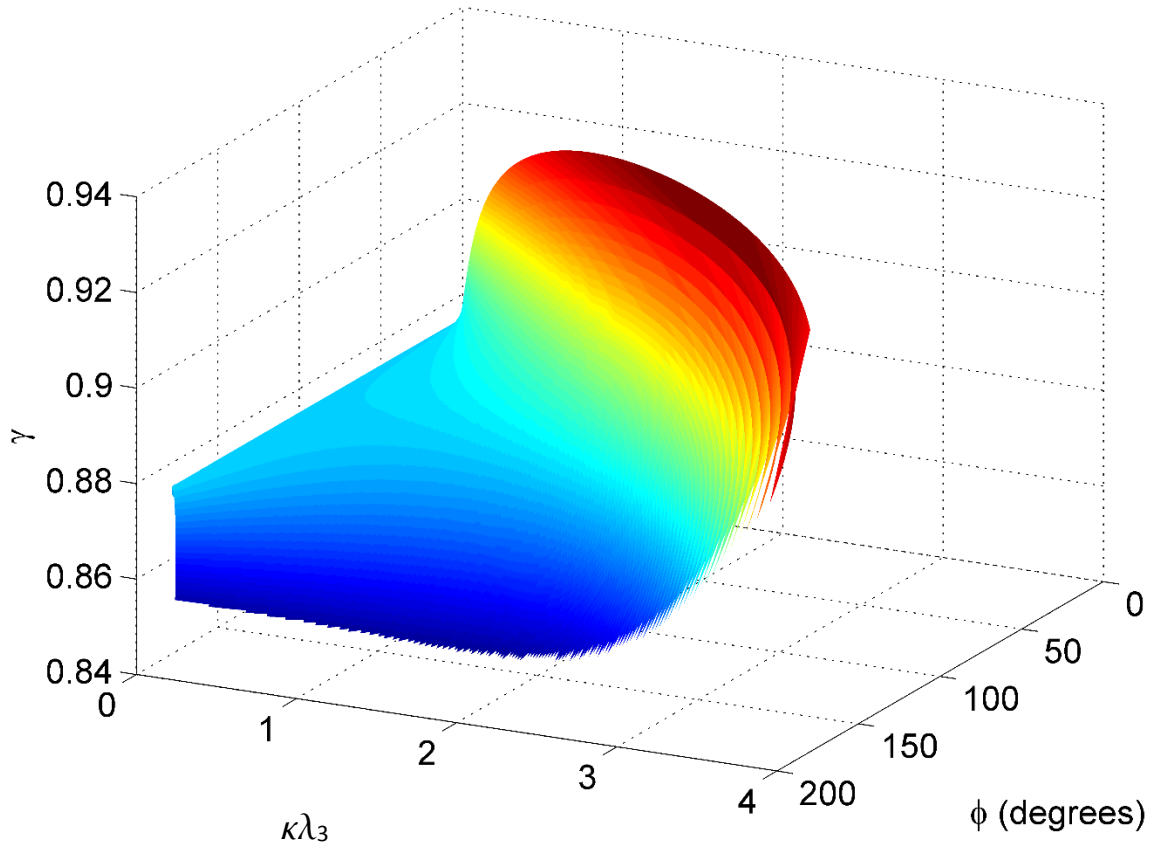


Figure 12 Here, γ is populating the phase space, thereby replacing \tilde{s} .

only case, with $\mu = 0$, can now be viewed and analyzed. However, it is the variation of γ , as the pseudo-rigid-body angle, Θ , changes, which must be analyzed quantitatively, and must be done so for individual load angles and individual values of the distributed contribution factor, μ . This necessitates the calculation of Θ for each location within the data structure as follows:

$$\Theta_{ijk} = \tan^{-1} \left(\frac{\tilde{b}_{ijk}}{\tilde{a}_{ijk} - \tilde{s}_{ijk} (1 - \gamma_{ijk})} \right) \quad (51)$$

This value of Θ represents the precise value, as it is calculated using the precise value of γ at each (i, j, k) data coordinate. From this, a three dimensional surface can be constructed for each value of μ analyzed. Figure (13) shows one such surface for the force only case, with $\mu = 1$. From this surface, contours were taken with respect to ϕ , ranging from $\phi = 20^\circ$ to $\phi = 170^\circ$, in 0.5° increments. This was done using the “contour” command within Matlab, with $\gamma_{all,all,k}$ as the x -argument input matrix, $\Theta_{all,all,k}$ as the y -argument input matrix, and $\phi_{all,all,k}$ as the z -argument matrix. Figure (14) shows a single contour which represents the variation of γ as Θ varies, for the level of $\phi = 90^\circ$ and the surface of $\mu = 1$.

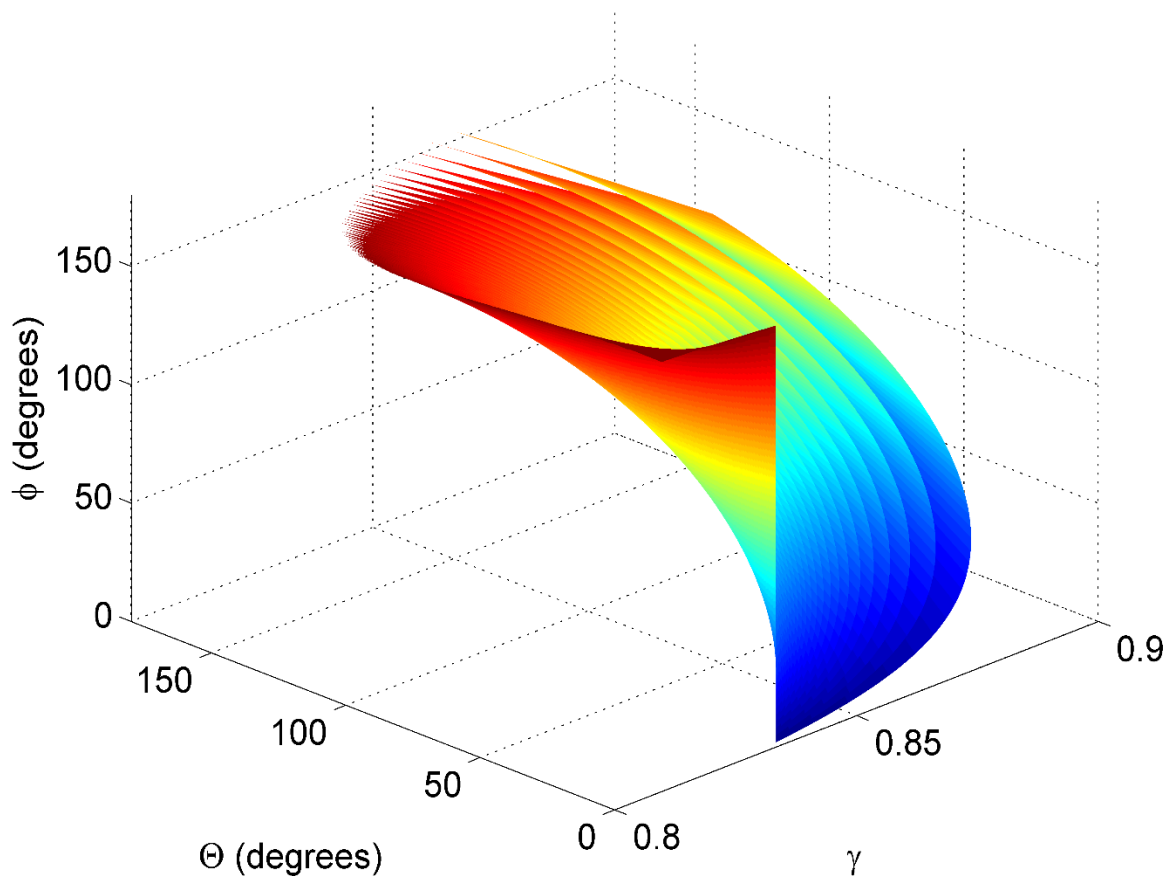


Figure 13 This represents the surface from which the contours, or levels curves, of γ were extracted, with $\mu = 1$.

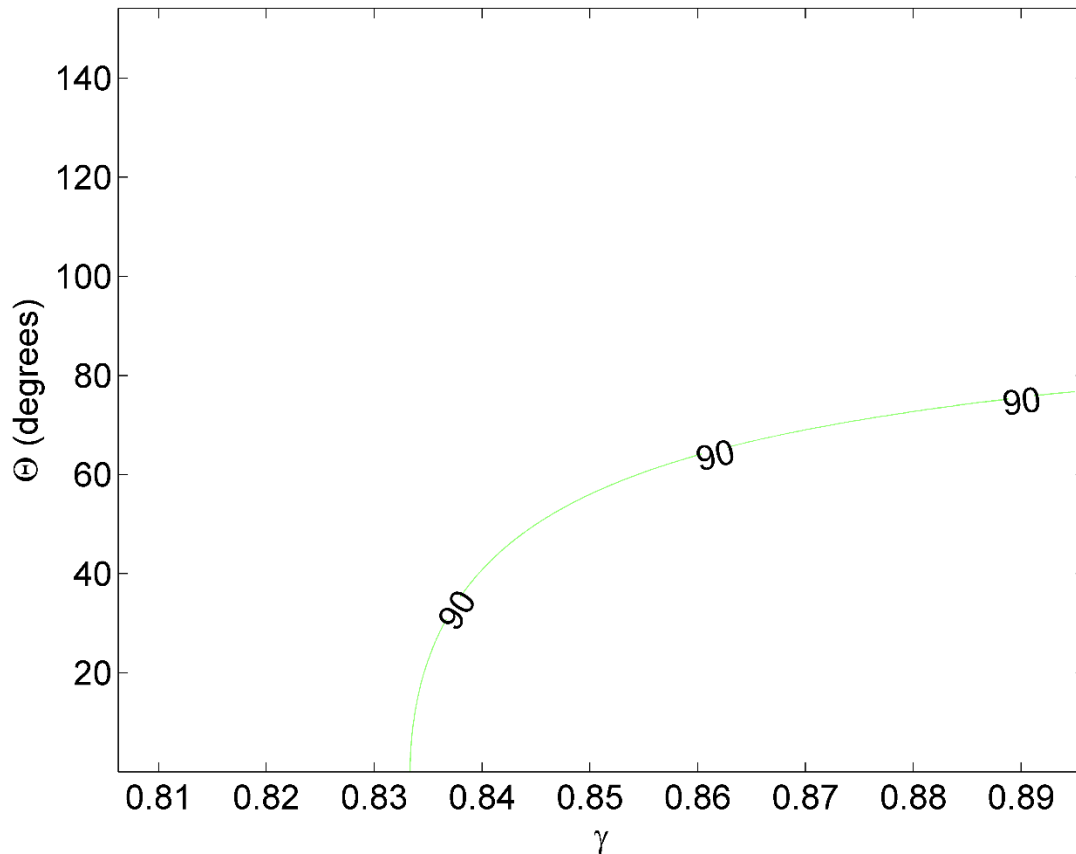


Figure 14 The contour of γ , at $\phi = 90^\circ$, as taken from the surface shown in Figure (13).

At this point, the information associated with this contour was stored into two rows, one for the γ values and one for the Θ values. If the contour matrix can be called C , then $C_{1,all}$ represents all of the γ coordinates of the contour, while $C_{2,all}$ represents all of the Θ coordinates of the contour. As this contour represents the exact values of γ , for each value of Θ , when $\phi = 90^\circ$ and $\mu = 1$, $C_{1,all}$ and $C_{2,all}$ can be used to construct the exact $(a/l, b/l)$ coordinates of the beam for every (γ, Θ) along the contour. This is possible by using equations (4) and (5), and introducing a dummy variable “ q ” to signify which column of C is being analyzed, and thus which $(\gamma, \Theta)_q$ is being analyzed. The exact $(a/l, b/l)$ coordinates were calculated as follows:

$$\left(\frac{a}{l}\right)_q = 1 - C_{1,q} (1 - \cos(C_{2,q})) \quad (52)$$

and

$$\left(\frac{b}{l}\right)_q = C_{1,q} \sin(C_{2,q}) \quad (53)$$

By creating a fine-grained vector of γ values, called γ_{vec} , ranging from $\gamma=0.7$ to $\gamma=1$, in increments of 0.001, a corresponding vector of Θ values, called Θ_{vec} , was created at each $(\gamma, \Theta)_q$ by the following:

$$\Theta_{q,vec} = \tan^{-1} \left(\frac{\left(\frac{b}{l}\right)_q}{\left(\frac{a}{l}\right)_q - (1 - \gamma_{vec})} \right) \quad (54)$$

Hence, at each contour data point $(\gamma, \Theta)_q$, also represented as $(C_{1,q}, C_{2,q})$, a vector of γ values was used to create a corresponding vector of calculated Θ values. This was done for the sole purpose of generating a relative error surface around the contour. Thus, for each contour data point $(\gamma, \Theta)_q$, a vector of relative error values was generated via:

$$relE_{q,vec} = \frac{\sqrt{\left(\left(\frac{a}{l}\right)_q - (1 - \gamma_{vec} (1 - \cos\Theta_{q,vec}))\right)^2 + \left(\left(\frac{b}{l}\right)_q - \gamma_{vec} \sin\Theta_{q,vec}\right)^2}}{\sqrt{\left(1 - \left(\frac{a}{l}\right)_q\right)^2 + \left(\frac{b}{l}\right)_q^2}} \quad (55)$$

where $relE_{q,vec} \times 100\%$ is the percent relative error vector generated by the contour data point $(\gamma, \Theta)_q$. Figure (15) shows the percent relative error surface surrounding the contour associated with the variation of γ , as Θ varies, for $\phi = 90^\circ$ and $\mu = 1$.

It was at this point that another contour could be taken; that of the percent relative error surface itself. Doing so allowed the error criteria of 0.5% to be evaluated at each (ϕ, μ) coordinate for γ . Shown in Figure (16), the contour of the percent relative error was able to be split into two matrices of similar dimension to the C matrix defined earlier. These can be called $leftE$ and $rightE$. By taking the minimum γ value of $rightE$, the largest approximation of γ , corresponding to the

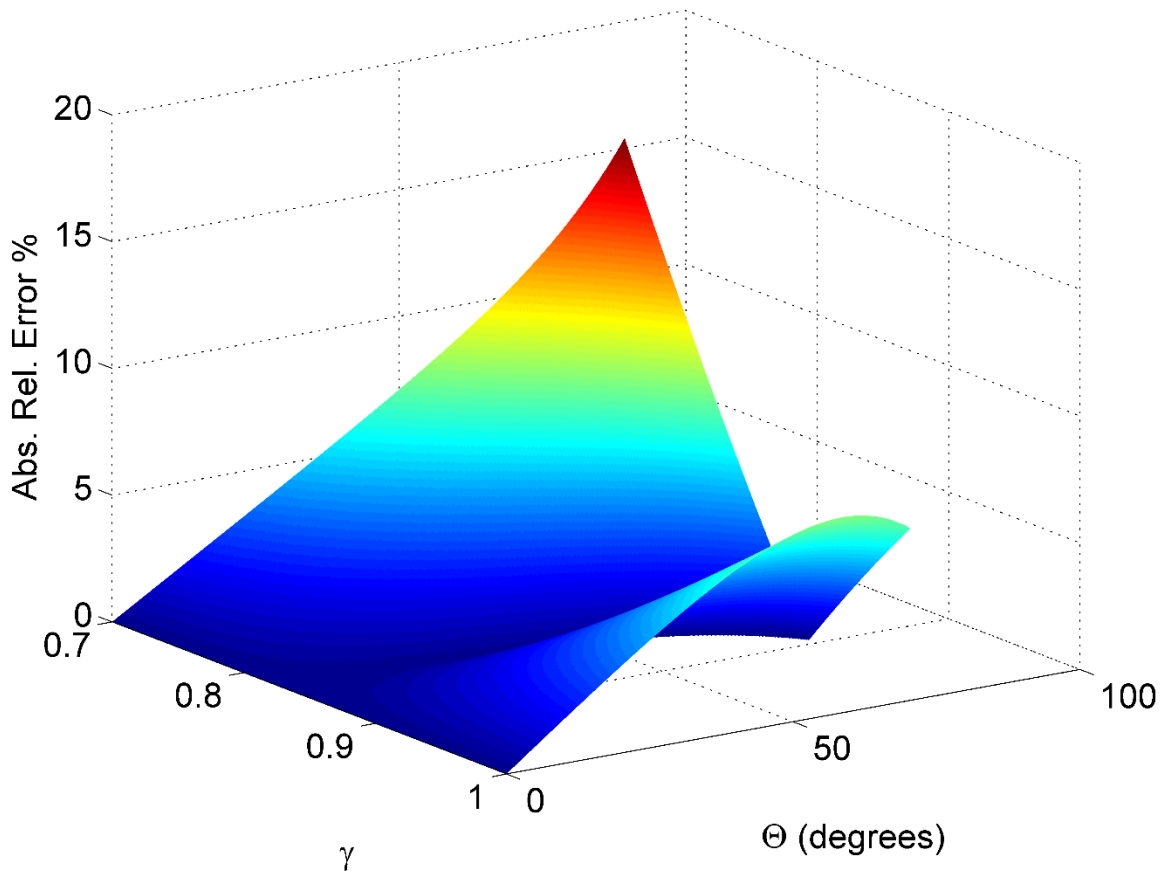


Figure 15 The relative error surface surrounding the contour of γ , when $\mu = 1$ and $\phi = 90^\circ$.

value of γ with the largest angular range in Θ , and still satisfying the error criteria, was found. Also shown in Figure (16), is the vertical line generated using this value. Following this vertical line upwards, through Θ , until it intersects the left error contour, or *leftE*, allowed linear interpolation to fix the value of $\Theta_{max,\gamma}$. Here, the variable $\Theta_{max,\gamma}$ is introduced to specify that this is the maximum Θ associated with γ . To summarize, using the contour method, with $\phi = 90^\circ$ and $\mu = 1$, it was found that $\gamma(\phi,\mu) = 0.8519$ and $\Theta_{max,\gamma}(\phi,\mu) = 64.4^\circ$. By comparison, for $\phi = 90^\circ$ and $\mu = 1$, the values of

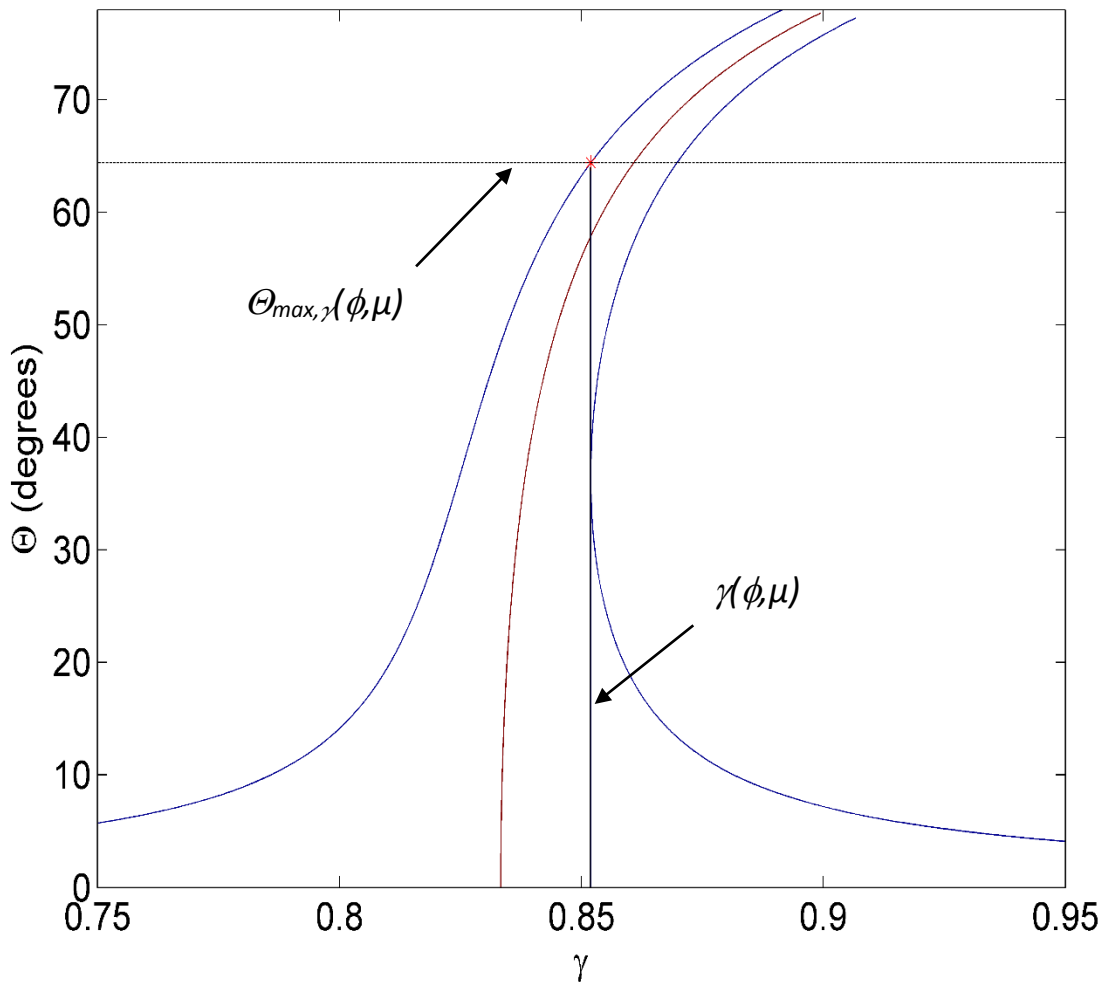


Figure 16 Observed values of γ and $\Theta_{max,\gamma}$ found by using the contour method, with $\phi = 90^\circ$ and $\mu = 1$.

$\gamma(\phi, \mu)$ and $\Theta_{max, \gamma}(\phi, \mu)$ found by Howell et al. are 0.8517 and 64.3°, respectively. This represents an absolute relative approximate error of 0.0235% and 0.1555%, respectively.

This process of contour analysis, then error surface generation, and then further contour analysis, was repeated for values of ϕ ranging from 20° to 170°, with 0.5° increments, and values of μ ranging from 0 to 1, with increments of 0.02. In other words, equations (52)-(55), and their accompanying processes, were repeated a total of $301 \times 51 = 15,351$ times. This corresponds to 15,351 values of $\gamma(\phi, \mu)$ and $\Theta_{max, \gamma}(\phi, \mu)$, which populate the spaces shown in Figures (17) and (18), respectively.

Accompanying the data points are the resulting fits as well. The sigmoidal increase in $\gamma(\phi, \mu)$, as μ decreases from 1 to 0, is visible. This results in curve resembling the shape of the letter “s” after being stretched, in a tensile fashion, by the endpoints. In other words, as the primary contribution to the applied load becomes increasingly of a distributed nature, the characteristic radius factor increases in value and the corresponding location of the characteristic pivot moves closer to the base. This results in a larger radius through which the pseudo-rigid link rotates. Having generated surfaces for $\gamma(\phi, \mu)$ and $\Theta_{max, \gamma}(\phi, \mu)$, with respect to varying values of ϕ and μ , surface fits were then produced. This was done using the Matlab “fit” command by guessing the form of the equation, inputting the coefficients as symbolic characters and allowing the default algorithm associated with the “fit” command to search for their values. This resulted in the following equations:

$$\gamma(\phi, \mu) = \quad (56)$$

$$\left(0.7987 + \frac{0.1758}{\ln\left(5.647 \tan^3\left(\frac{\phi}{2}\right) - 5.774 \tan^2\left(\frac{\phi}{2}\right) + 26.31 \tan\left(\frac{\phi}{2}\right) + 1.1\right)} \right) + \left(0.0003257 + \frac{1.272}{(1 + \exp(9.889 \mu^2 - 0.4737 \mu - 1.296))} \right) \left(0.04462 + \frac{-0.01106}{\ln\left(8.296 \tan^3\left(\frac{\phi}{2}\right) - 11.33 \tan^2\left(\frac{\phi}{2}\right) + 26.11 \tan\left(\frac{\phi}{2}\right) + 1.336\right)} \right)$$

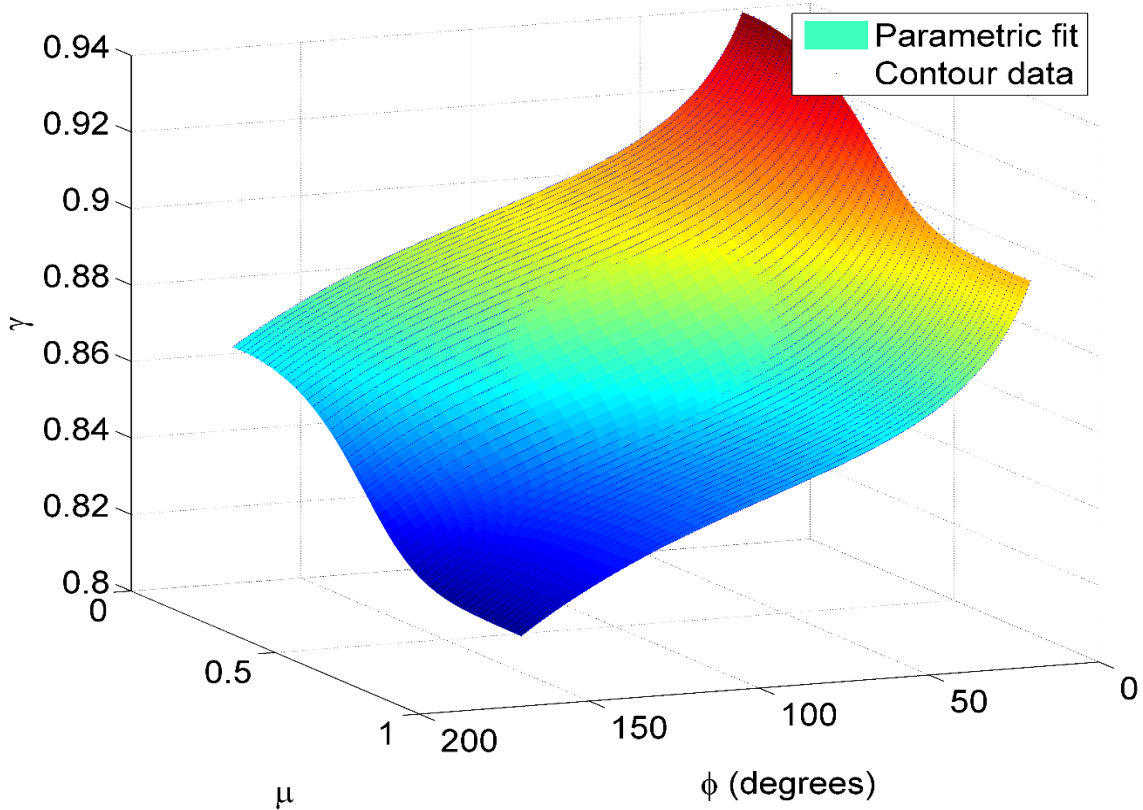


Figure 17 Surface of the $\gamma(\phi, \mu)$ fit to the contour data.

$$\Theta_{\max,\gamma}(\phi, \mu) = \quad (57)$$

$$\left(0.3267 + \left(\frac{0.0336}{1 + \exp(14.78\mu^3 - 1.754\mu^2 + 0.4065\mu - 0.2566)} \right) \right)$$

$$+ \left(-0.7882 \left(\frac{\phi}{\pi} \right)^4 + 2.258 \left(\frac{\phi}{\pi} \right)^3 - 1.944 \left(\frac{\phi}{\pi} \right)^2 + 1.723 \left(\frac{\phi}{\pi} \right) - 0.1701 \right)$$

$$\left(\frac{5.162\mu^9 - 164\mu^8 + 605.6\mu^7 - 934.3\mu^6 + 731.4\mu^5 - 299.6\mu^4 + 62.17\mu^3 - 6.786\mu^2 + 0.24\mu + 1.929}{} \right)$$

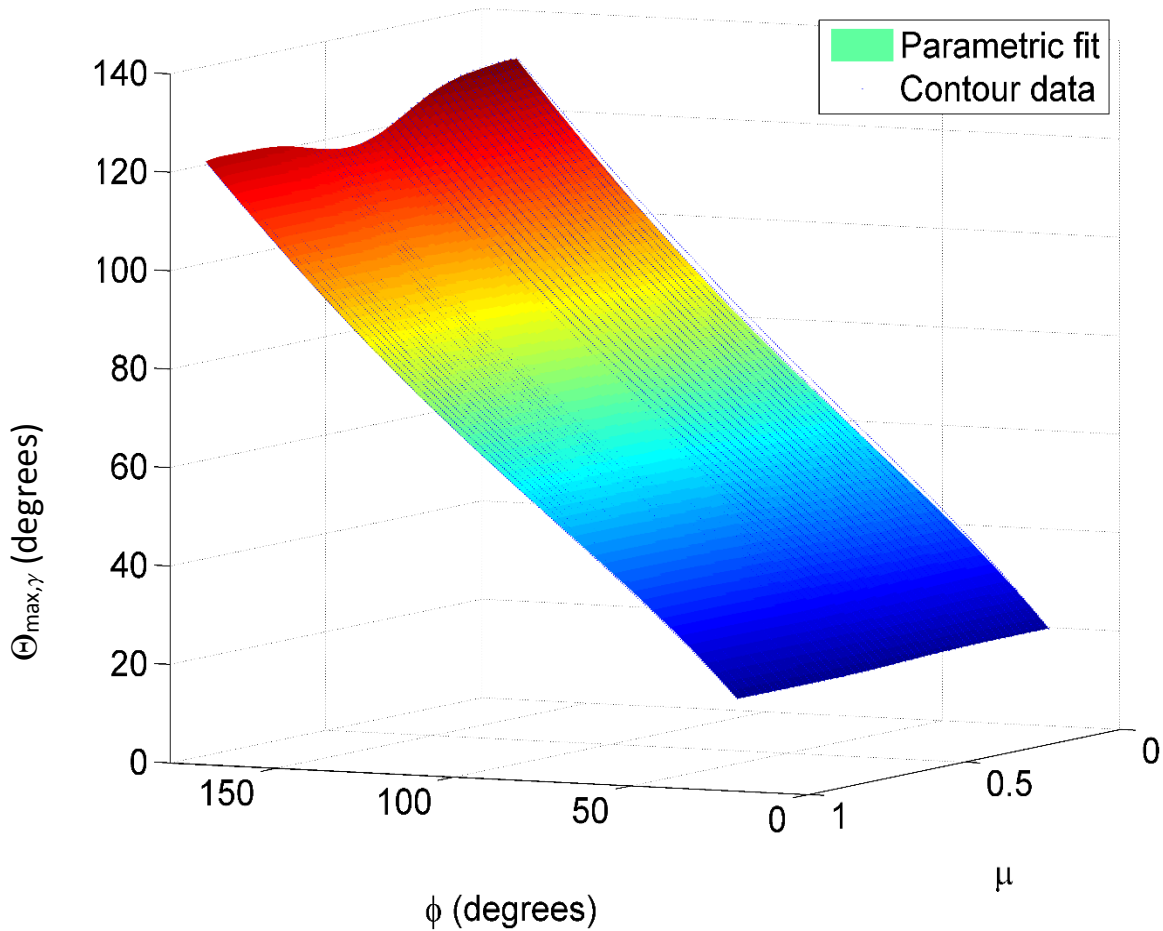


Figure 18 Surface of the $\Theta_{\max,\gamma}(\phi, \mu)$ fit to the contour data.

Here, $\Theta_{max,\gamma}(\phi,\mu)$ is defined as the function of the Θ_{max} values associated with γ and has units of radians. Figure (19) shows a comparison between the values of γ as predicted by equation (56), for $\mu = 1$ and $20^\circ < \phi < 170^\circ$, and the values of γ as predicted by Howell et al. [40] and the adjusted equations put forth by Pauly et al. [49]. Also, equations (56) and (57) should be viewed in the following context:

$$parameter = (term_1) + (term_2)(term_3) \quad (58)$$

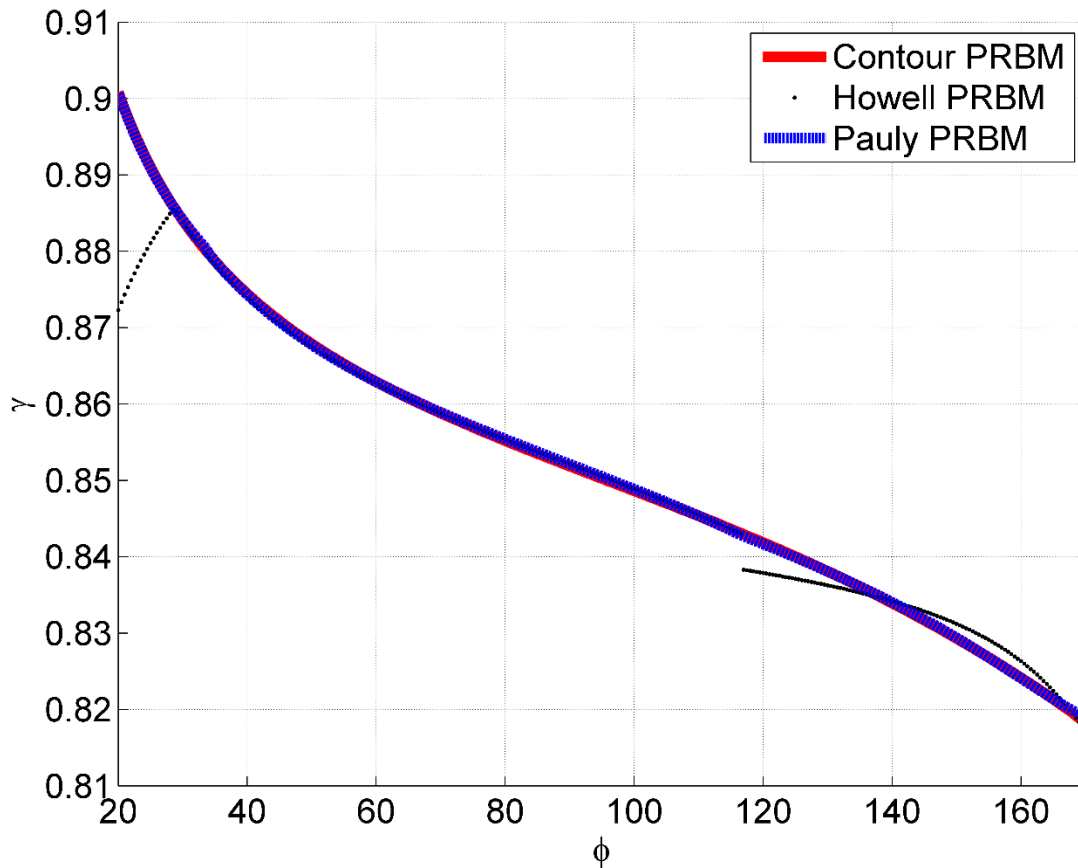


Figure 19 Force only PRBM comparisons.

where $term_1$ is defined as the equation for one edge of the surface being fitted, $term_3$ is defined as the equation describing the difference between the edge defined by $term_1$ and the edge opposite, and $term_2$ is defined as the function which distributes that difference across the intervening space. Here, $term_1$ in equation (56) is, by design, all that is required to fit the equation for γ in the force only case and, in that context, has a correlation coefficient, or r^2 , value of 1. It is also worth noting that this term is not a piecewise solution in ϕ . Equation (56), in its totality across the specified ϕ and μ space, has $r^2 = 1$, as well. Equation (57) has $r^2 = 0.9990$.

5.3 Finding $c_{\theta}(\phi, \mu)$

The process for finding $c_{\theta}(\phi, \mu)$ was simpler than that which had found $\gamma(\phi, \mu)$. While the contour method was still employed, the process of generating error spaces around each of the 15,351 individual contours, and then taking contours of those surfaces and processing the results, was not necessary. By employing the stored values of ϕ_{ijk} , along with the values stored in equations

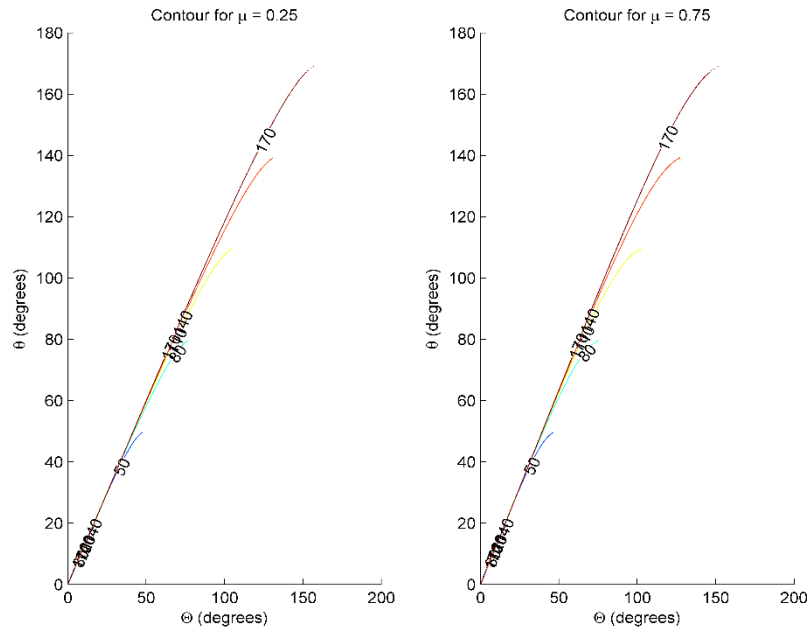


Figure 20 Contours of Θ vs θ , demonstrating the near linear relationship between them.

(46) and (51), three dimensional surfaces were constructed for each value of μ analyzed. By taking contours with respect to ϕ , level curves can again be produced and analyzed. Figure (20) shows several such level curves demonstrating the near-linear relationship between the pseudo-rigid-body angle, Θ , and the beam end angle, θ_0 , for $\mu = 0.25$ and $\mu = 0.75$. By fitting each contour, obtained from values of ϕ ranging from 20° to 170° , in 0.5° increments, and values of μ ranging from 0 to 1, with increments of 0.02, with a line rooted at the origin, and up to the limit of Θ as defined by $\Theta_{max,\gamma}(\phi,\mu)$ in equation (57), the surface seen in Figure (21) was produced. Notice the discontinuity for tensile loads near $\phi = 30^\circ$. A similar behavior was also observed by Pauly et al. [49]. For this surface, however, the location of the discontinuity in ϕ varies as μ varies. Therefore,

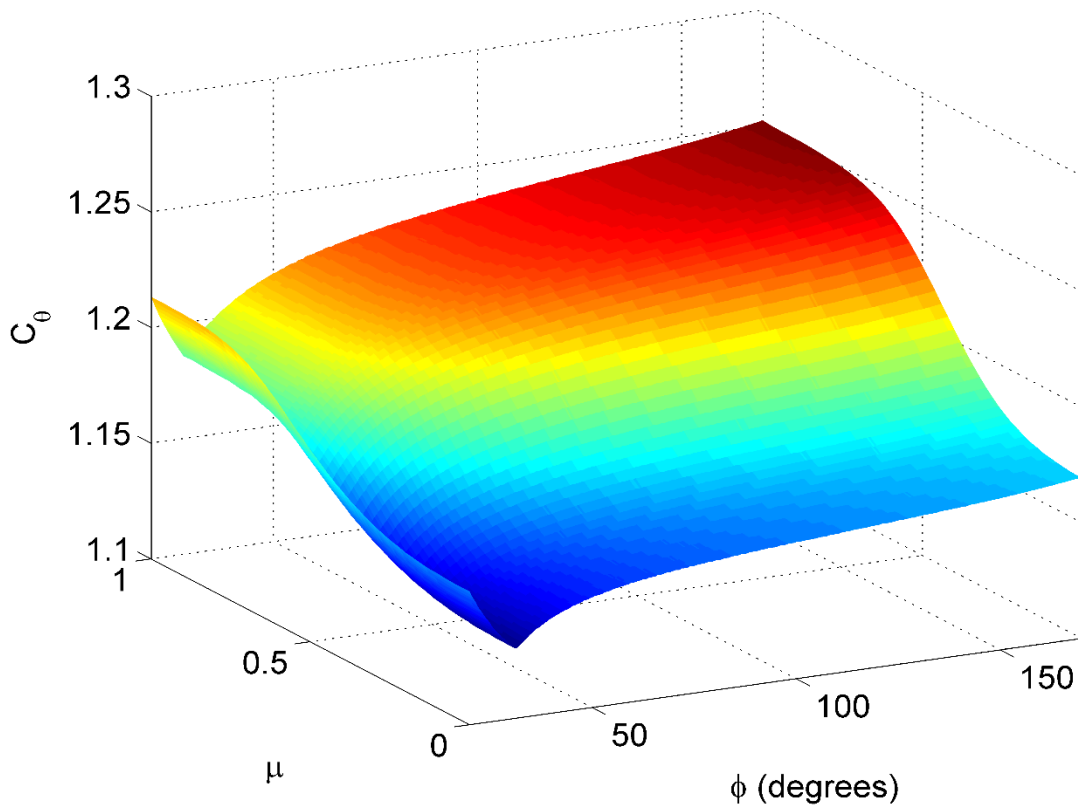


Figure 21 Surface of c_θ showing the persistent discontinuity.

a piecewise approach to the surface fits was deemed ill-suited to the task, as at least 102 different $c\theta(\phi)$ equations would be required to describe the data set. At this point, the data set was truncated for all values of ϕ below 31.5° so that at least the majority (92.36%) of the data set could be accurately described by a single equation. The surface it defines, as well as the non-truncated portion of the data set, can be seen in Figure (22). This single equation is given by:

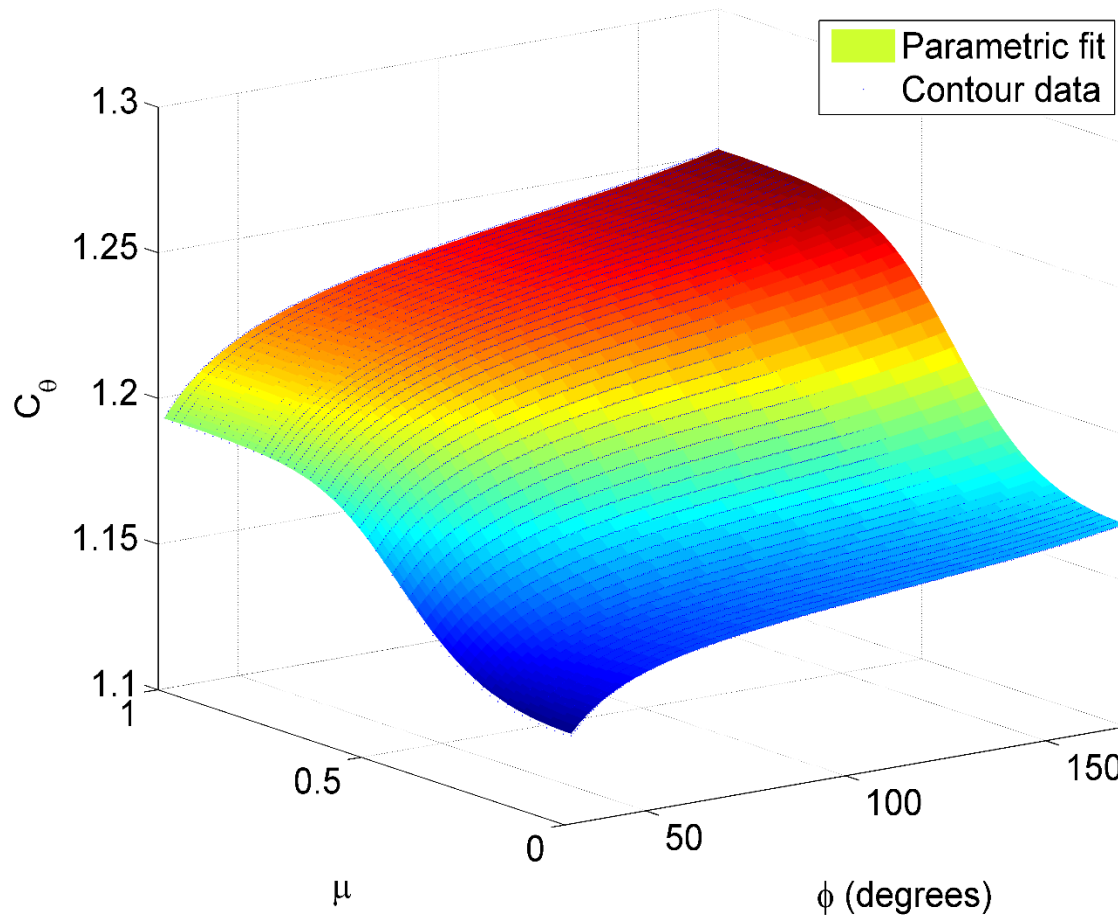


Figure 22 Surface of the $c\theta(\phi, \mu)$ fit to values stored in the data structure. Visible is the incorporation of the truncation.

$$c_{\theta}(\phi, \mu) = \quad (59)$$

$$\left(1.262 + \frac{-0.1221}{\ln \left(1.803 \tan^5 \left(\frac{\phi}{2} \right) + 2.399 \tan^4 \left(\frac{\phi}{2} \right) + 14.26 \tan^3 \left(\frac{\phi}{2} \right) + 50.4 \tan^2 \left(\frac{\phi}{2} \right) + 1.1 \tan \left(\frac{\phi}{2} \right) + 1.262 \right)} \right) + \left(\frac{-1.333}{1 + \exp(-15.05 \mu^4 + 16.2 \mu^3 + 4.905 \mu^2 + 0.04808 \mu - 1.113)} \right) \left(0.08554 + \frac{-0.03772}{\ln \left(1.555 \tan^5 \left(\frac{\phi}{2} \right) + 10.03 \tan^4 \left(\frac{\phi}{2} \right) + 13.39 \tan^3 \left(\frac{\phi}{2} \right) + 16.63 \tan^2 \left(\frac{\phi}{2} \right) + 1.1 \tan \left(\frac{\phi}{2} \right) + 3.043 \right)} \right)$$

This equation has $r^2 = 0.9999$. For comparative purposes, the values of c_{θ} described by Pauly et al. have been plotted alongside those predicted by equation (59), for $\mu = 1$, and can be seen in Figure (23).

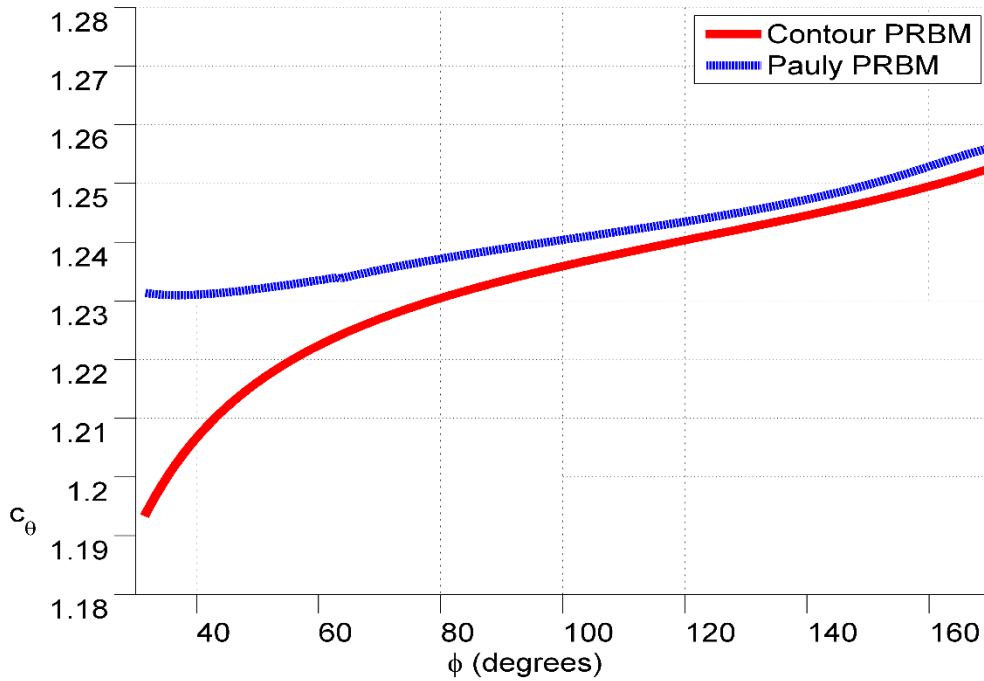


Figure 23 Comparisons between PRBMs for c_{θ} when $\mu = 1$.

5.4 Finding $K_{\theta}(\phi, \mu)$

Relating the applied loads to the resulting deflections was done in both a dimensional sense and a non-dimensional sense. The transverse load index in equation (10) is, itself, non-dimensional, but had to first be reformulated so that the information contained within the data structure could define it. This was done via:

$$(\alpha^2)_t = \left(\frac{s}{\lambda_1} \right)^2 \sin(\phi - \Theta) = \mu^2 \tilde{s}^2 \sin(\phi - \Theta) \quad (60)$$

In terms of referencing elements within the data structure, equation (60) can be written as:

$$(\alpha^2)_{t,ijk} = \mu_k^2 \tilde{s}_{ijk}^2 \sin(\phi_{ijk} - \Theta_{ijk}) \quad (61)$$

This was used to describe the force contribution within K_{θ} . The transverse distributed load index can be described in a similar fashion as:

$$(\alpha^3)_t = \frac{w_t l^3}{EI} = \left(\frac{s}{\lambda_2} \right)^3 \sin(\phi - \Theta) = (1 - \mu)^3 \tilde{s}^3 \sin(\phi - \Theta) \quad (62)$$

Again, in terms of referencing elements within the data structure:

$$(\alpha^3)_{t,ijk} = (1 - \mu_k)^3 \tilde{s}_{ijk}^3 \sin(\phi_{ijk} - \Theta_{ijk}) \quad (63)$$

While equation (63) was considered to be a plausible extension of K_{θ} , it did not produce a viable PRBM, as the total magnitude of the distributed load, given by w_s , would have been applied at the beam tip. This can be seen from a torque analysis similar to equation (14). If, for the force only case:

$$T = (\alpha^2)_t \left(\frac{EI}{l^2} \right) (\mathcal{M}) = F_t (\mathcal{M}) \quad (64)$$

then the (\mathcal{M}) term signifies the moment arm of the transverse loading. A similar torque analysis of the distributed load only case resulted in:

$$T = (\alpha^3)_t \left(\frac{EI}{l^2} \right) (\mathcal{M}) = lw_t (\mathcal{M}) \quad (65)$$

Here, the moment arm of the concentration of the distributed load was found to be located at the beam end, which appeared to defy credulity.

5.5 Non-dimensional Strain Energy

At this point, the strain energy, associated with the beam, was used as a guide to determine where the moment arm, associated with the concentration of the distributed load, should be placed.

The strain energy of an elastic beam undergoing deformation is given by:

$$V = \int_0^L \frac{M^2}{2EI} ds \quad (66)$$

where V is the strain energy, in units of (force x length), and M is the moment, in units of (force x length), as well. This equation was reformulated into a non-dimensional expression by equating it to the work done in moving thru the resistance posed by the torsional spring of the PRBM. Using the relationship between the spring stiffness and the stiffness coefficient, given by equation (15), the work was expressed as:

$$W = \frac{1}{2} K\Theta^2 = \frac{K_{\Theta}\gamma EI\Theta^2}{2l} \quad (67)$$

By setting the work equal to the strain energy and recalling that $M = EI\kappa$, the following equation was found:

$$\frac{K_{\Theta}\gamma EI\Theta^2}{2l} = \int_0^L \frac{\kappa^2 EI}{2} ds \quad (68)$$

or

$$K_{\Theta} = \frac{l \int_0^L \kappa^2 ds}{\gamma\Theta^2} \quad (69)$$

This equation was rendered non-dimensional by first recognizing that each non-zero \tilde{s} within the phase space is, itself, a wall. Hence $(l/\lambda_3) = (s/\lambda_3)$ and, therefore:

$$K_{\Theta} = \left(\frac{\lambda_3^2}{\lambda_3^2} \right) \frac{l \int_0^L \kappa^2 ds}{\gamma\Theta^2} = \frac{\tilde{s} \int_0^L \tilde{\kappa}^2 d\tilde{s}}{\gamma\Theta^2} = \frac{\tilde{s} \tilde{V}}{\gamma\Theta^2} \quad (70)$$

Here, \tilde{V} is termed the non-dimensional strain energy and is solved for numerically by incorporating it into the differential equation solver, astride equations (32)-(38), the expression:

$$\frac{d\tilde{V}}{d\tilde{s}} = \tilde{\kappa}^2 \quad (71)$$

Equation (70) represents a path to calculating K_{θ} which is independent of the transverse load indices. This equation was validated by using FEA to analyze the work done by the same beam created for the initial validations in Section 4. The force only case was used, with a load angle of 90° . This simplified the analysis, as only the vertical deflection needed to be considered. Figure (24) shows the force-deflection data taken from the FEA for the case described. By using a 7th order polynomial to fit the force-deflection curve, and then integrating it at each data point, the work done by the force could be calculated. K_{θ} could then be found at each data point via:

$$K_{\theta,i} = \frac{2lW_i}{EI\gamma_i\Theta_i^2} \quad (72)$$

Here, the subscript i is being used as the same dummy variable as that used in equations (39)-(41), and should not be confused with the subscripts ijk used in subsequent equations. W_i is the work calculated at each data point. Also, γ_i and Θ_i were found using equations (6) and (7), respectively, and at each data point.

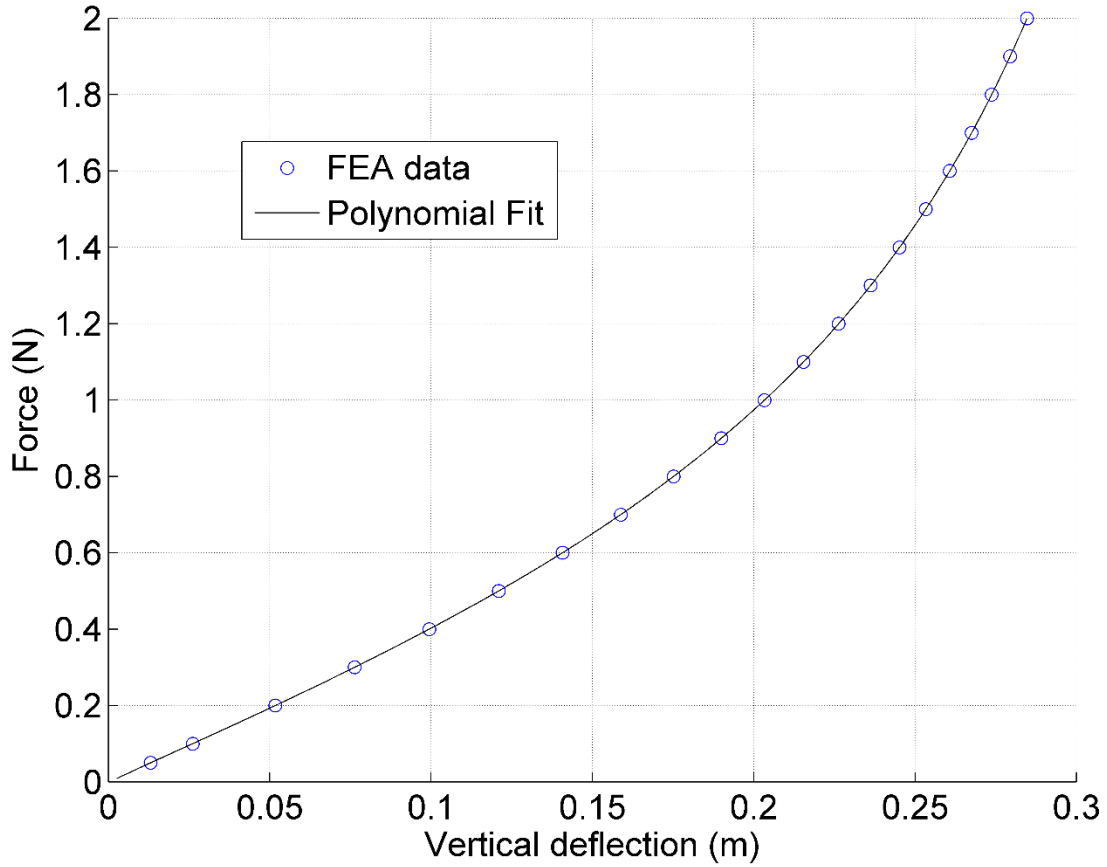


Figure 24 FEA data and the associated polynomial fit used to calculate the work done on a beam by a force applied at 90°.

At this point, the contour method was again employed so that contours of K_{θ} could be found. The K_{θ} associated with the non-dimensional strain energy was entered into the data structure via:

$${}^{SE} K_{\Theta,ijk} = \frac{\tilde{s}_{ijk} \tilde{V}_{ijk}}{\gamma_{ijk} \Theta_{ijk}^2} \quad (73)$$

Here, $^{SE} K_{\Theta}$ is used to differentiate between the K_{Θ} calculated via the transverse load indices and the K_{Θ} calculated via the strain energy. The K_{Θ} associated with the transverse load indices was also entered into the data structure as:

$$K_{\Theta,ijk} = \left(\mu_k^2 \tilde{s}_{ijk}^2 + (1 - \mu_k)^3 \tilde{s}_{ijk}^3 \right) \sin(\phi_{ijk} - \Theta_{ijk}) \quad (74)$$

With the strain energy $^{SE} K_{\Theta,ijk}$ inside the data structure, surfaces corresponding to $^{SE} K_{\Theta,ijk}$, Θ_{ijk} , and ϕ_{ijk} were generated for each value of μ analyzed. By taking the contour of the surface associated with $\mu = 1$ and at the level of $\phi = 90^\circ$, the FEA prediction of K_{Θ} was tested. The results are shown in Figure (25). There is a strong level of agreement in the region of large deflection.

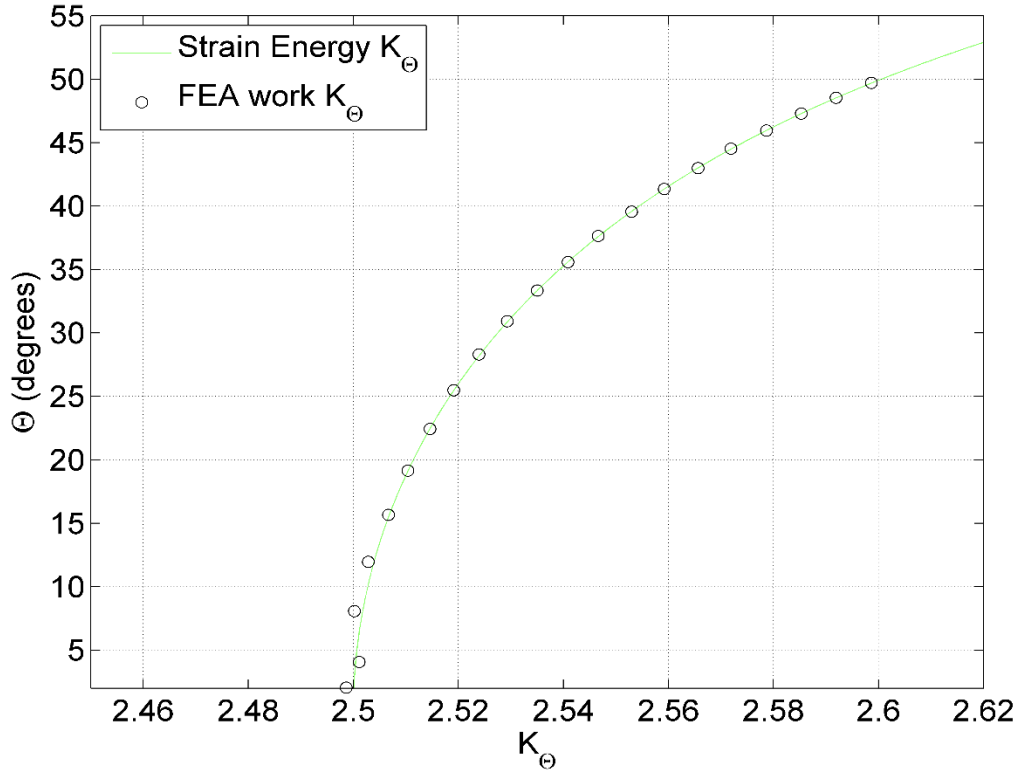


Figure 25 The K_{Θ} values as predicted by the non-dimensional strain energy and the K_{Θ} values as predicted by an independent FEA investigation.

With a K_{θ} generated independently from the transverse load indices in hand, comparisons could now be made across the entire phase plane for different loading cases. Figure (26) shows the K_{θ} predicted by both the force only transverse load index and that of the accompanying non-dimensional strain energy for $\mu = 1$. Figure (27) shows the $\mu = 0$ case, with a distributed load only. It was immediately noticed that, while the different surfaces of K_{θ} were qualitatively similar for the force only case, they were very different for the distributed load only case. However, in the force only case, the roots of K_{θ} , where both $\Theta = 0$ and $\tilde{\kappa} = 0$, were aligned. *This* was the guidance required from an independent calculation of K_{θ} . In order to align the roots of K_{θ} , for the distributed

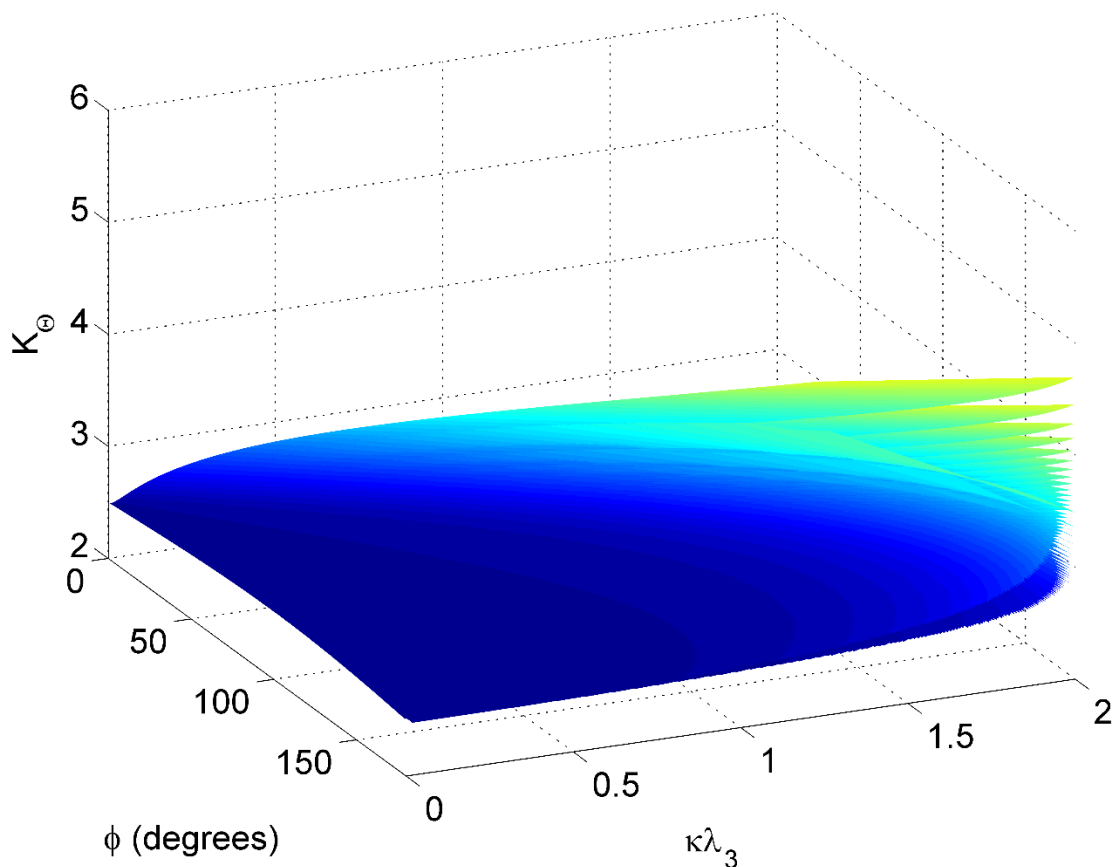


Figure 26 The force only case showing close agreement between the K_{θ} calculated from the strain energy and the K_{θ} calculated by the transverse load index.

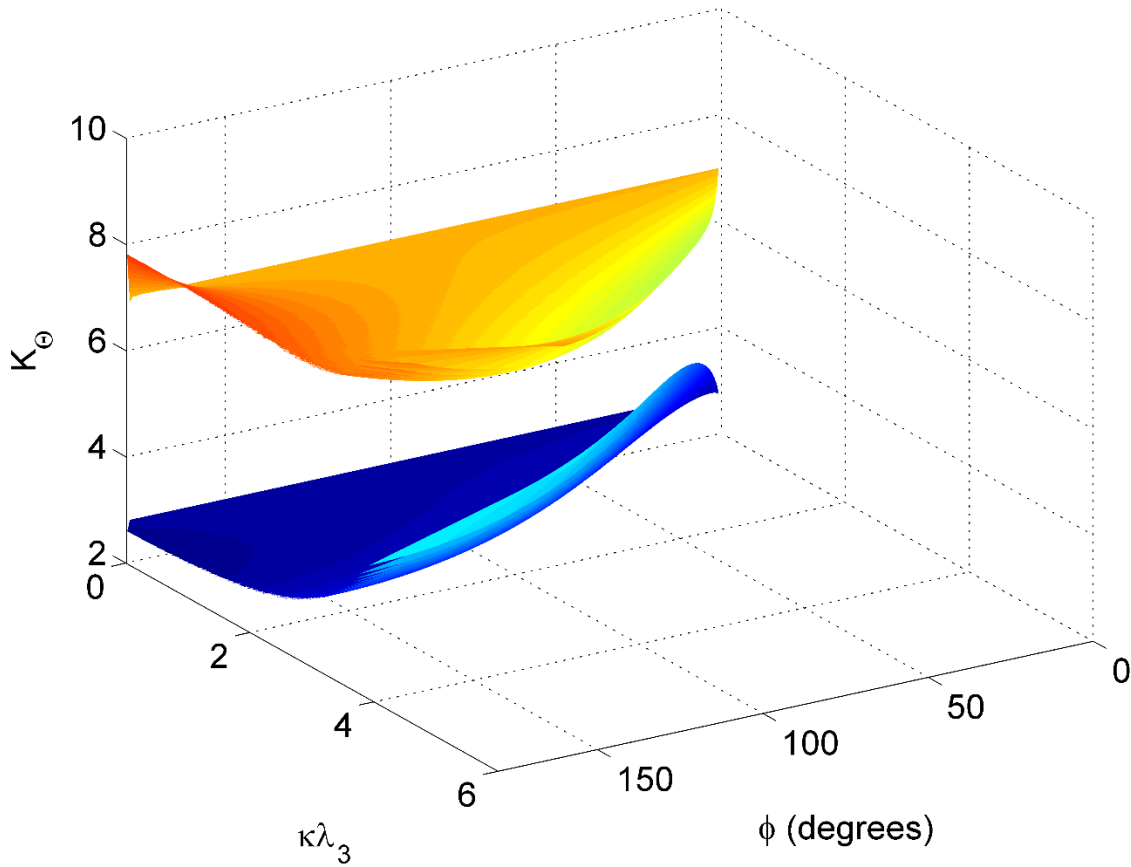


Figure 27 The distributed load only case showing the discrepancy between the K_{θ} calculated by the strain energy and the K_{θ} calculated by the transverse load index.

load only case, a constant value of 0.4 needed to be multiplied to the transverse load index. From this, a PRBM for distributed loads could be constructed. This means that while the entire beam length is incorporated into the calculation of the total force imparted by the distributed load, the moment arm from the characteristic pivot to the concentration point of that total force is equal to 0.4λ . An intermediate case, one incorporating this value of 0.4 into the distributed transverse load index, can now be investigated. The K_{θ} surfaces for $\mu = 0.5$ are shown in Figure (28).

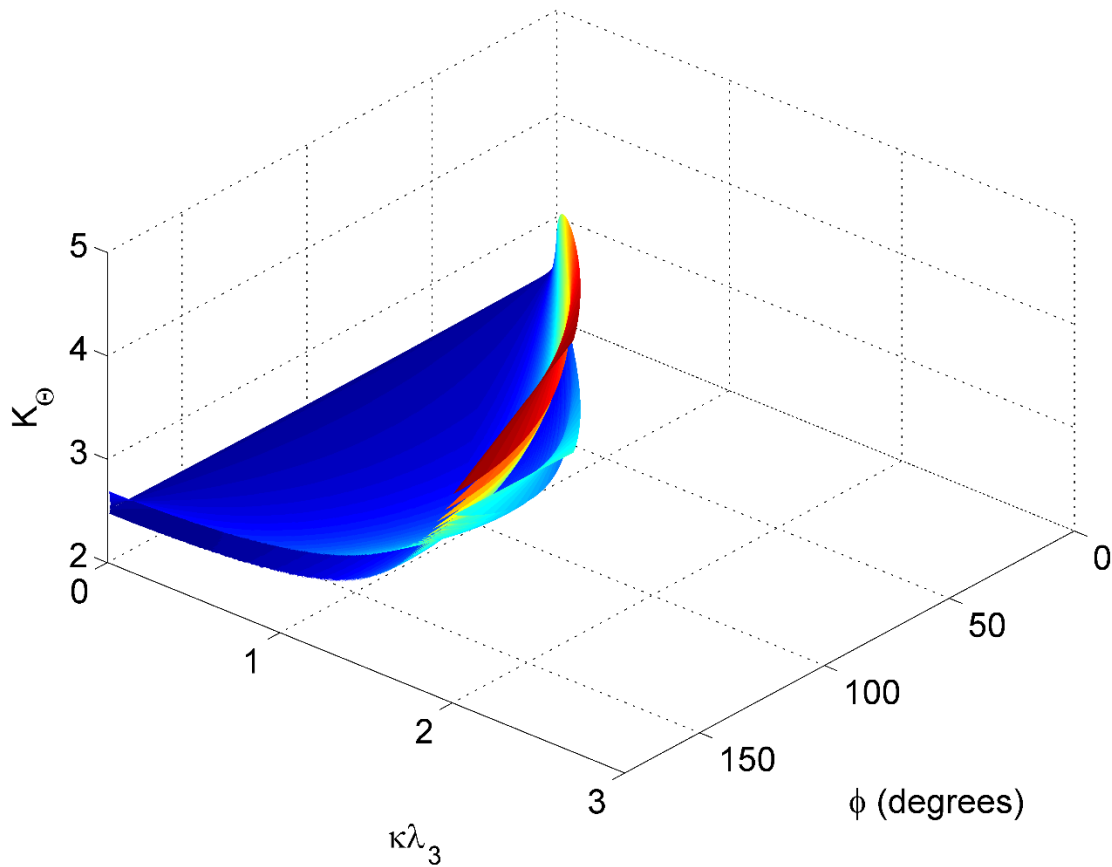


Figure 28 K_{θ} surfaces which are now sharing a common root due to the introduction of a constant factor of 0.4 to the distributed transverse load index. Here, $\mu = 0.5$.

5.6 Correction Factors

Even so, however, the qualitative dissimilarities between the K_{θ} values calculated by the transverse load index of the distributed load and that by the strain energy had to be addressed. Also, values of $K_{\theta}(\phi, \mu)$ needed to be recommended in such a way as to mitigate error in force or distributed load prediction. Correction factors were then decided to be employed. It was realized that if the contours of K_{θ} , as θ varied for a given ϕ , could be manipulated at the extremes of μ , i.e. at values of 0 and 1, then those manipulations would be persistent for intermediate values of μ . It was also realized that if the K_{θ} values could be manipulated as they traversed θ , for a given ϕ , and

if they were to ultimately be approximated by a straight vertical line, then the correction factors should be designed to make each K_{Θ} contour a straight vertical line. Doing so would allow $K_{\Theta}(\phi, \mu)$ to be defined by the roots of the K_{Θ} contours themselves, and would ultimately mitigate the errors associated with using the parameterized $K_{\Theta}(\phi, \mu)$ equation to calculate F, w, E, I or l .

At this point, it was decided to distinguish how Θ was to be calculated. The equation decided upon was:

$$\Theta_{ijk} = \sin^{-1} \left(\frac{\tilde{b}_{ijk}}{\tilde{s}_{ijk} \gamma(\phi, \mu)} \right) \quad (75)$$

There are two advantages to this equation. The first is that any imperfections associated with the fit for $\gamma(\phi, \mu)$, equation (56), will be incorporated into the correction factors, astride the undesired behavior already being corrected for. The second advantage has to do with how this model is intended to be used. If a designer were to employ a PRBM that calculated Θ from the arctangent function of equation (7), they would have to know the values of both a and b . By using an arcsin function, however, they would only need to know the vertical deflection, b . Unfortunately, this does place an upper limit to the amount of angular deflection through which the force-deflection relationship will remain accurate; that being 90° . This is because the arcsin function cannot distinguish between angles symmetric about the vertical axis, like 80° and 100° .

K_{Θ} can now be defined as a function of ϕ alone for the force only case by setting $\mu = 1$ and using the equation:

$$K_{\Theta,ijk} \Theta_{ijk} = \left(\mu_k^2 \tilde{s}_{ijk}^2 + 0.4(1 - \mu_k)^3 \tilde{s}_{ijk}^3 \right) \sin(\phi_{ijk} - \Theta_{ijk}) \quad (76)$$

It was realized that different correction factors were needed for the different extremes of μ , hence one for the distributed contribution and one for the force contribution. It was also realized that the correction factors would have to be piecewise solutions in ϕ . This was a result of the 90° upper limit placed on Θ . The form of the correction factors to be multiplied to the respective transverse load indices were thus to take the form of:

$$\chi_F\left(\frac{\Theta}{\phi}, \frac{\phi}{\pi}\right) \rightarrow \phi \leq 100^\circ \quad (77)$$

$$\chi_D\left(\frac{\Theta}{\phi}, \frac{\phi}{\pi}\right) \rightarrow \phi \leq 100^\circ \quad (78)$$

$$\chi_F\left(\frac{\Theta}{\pi}, \frac{\phi}{\pi}\right) \rightarrow \phi > 100^\circ \quad (79)$$

$$\chi_D\left(\frac{\Theta}{\pi}, \frac{\phi}{\pi}\right) \rightarrow \phi > 100^\circ \quad (80)$$

Here, the arguments of the correction factors were designed to produce normalized surfaces. Also, the subscripts D and F stand for distributed and force, respectively. An equation representing $K_\Theta(\phi, \mu)$ was then constructed by obtaining the root values of the K_Θ contours, each at $\Theta = 0$, for load angles ranging from 20° to 170° , in increments of 0.5° , and for μ values ranging from 0 to 1, in increments of 0.02. Figure (29) shows the resulting surface which can be fit by the following equation:

$$K_{\Theta,init}(\phi, \mu) = \quad (81)$$

$$\left(2.398 + \frac{0.5074}{\ln\left(5.062 \tan^3\left(\frac{\phi}{2}\right) - 5.485 \tan^2\left(\frac{\phi}{2}\right) + 24.11 \tan\left(\frac{\phi}{2}\right) + 1.18\right)} \right) + \frac{\left(312.5\mu^{10} - 52.56\mu^9 - 176.6\mu^8 - 126.1\mu^7 - 64.67\mu^6 - 20.29\mu^5 + 42.68\mu^4 + 107.8\mu^3 - 18.19\mu^2 + 1.299\mu + 0.3864\right)}{\left(1 + \exp(11\mu^3 - 14.76\mu^2 + 18.03\mu - 1.305)\right)}$$

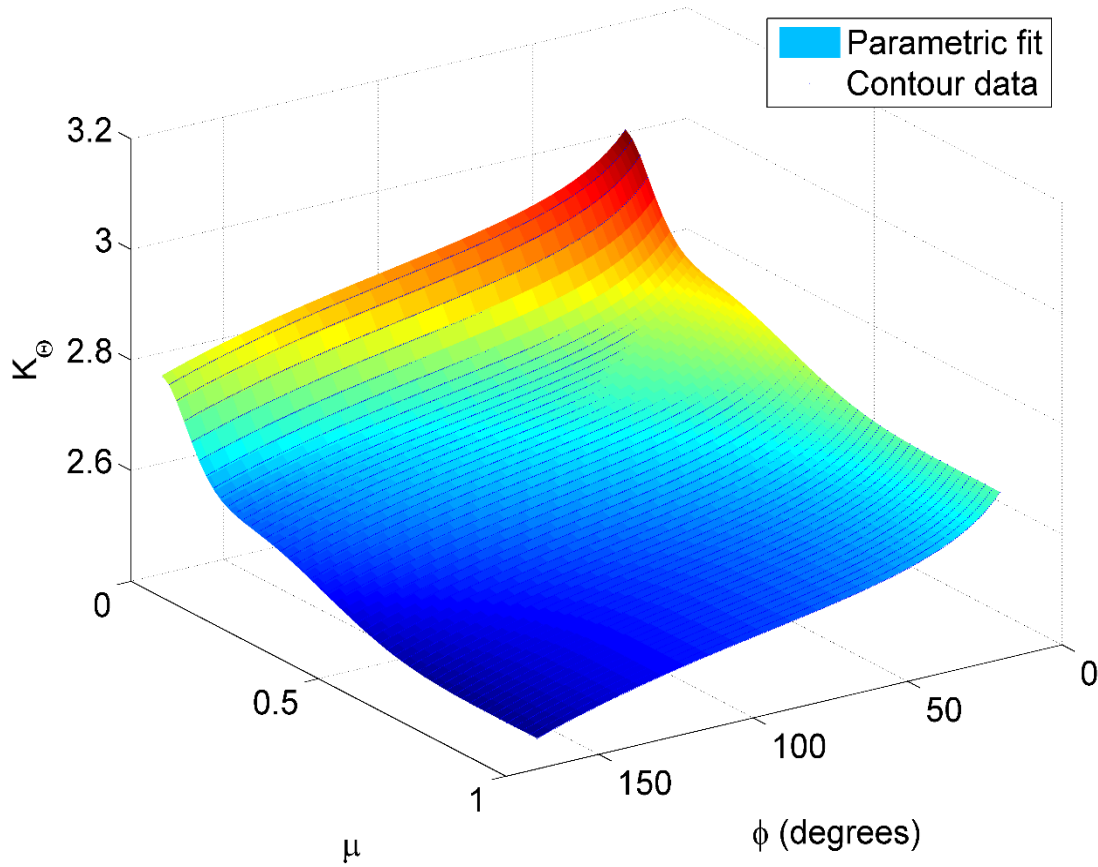


Figure 29 $K_{\Theta,init}(\phi, \mu)$ surface used to find correction factors.

Here, $K_{\Theta,init}(\phi, \mu)$ is defined as the initial fit for $K_{\Theta}(\phi, \mu)$, and has $r^2 = 0.9996$, but more importantly, the first term, which describes the force only case, has $r^2 = 1$. Notice the lack of a $term_3$ from equation (81). It was discovered that the difference between the force only and distributed only cases was so close to a constant value, with variations from that constant value being on the order of 10^{-3} , that attempting to fit the difference actually increased the error. Recall that the aim at this point was to very closely define $K_{\Theta,init}(\phi, \mu)$ for specific μ values, namely $\mu = 1$ and $\mu = 0$. Having done so, the correction surfaces can now be directly solved for at each element location within the data structure via:

$$\chi_{D,ijk} = \frac{K_{\Theta,init}(\phi, \mu = 0)_{ijk} \Theta_{ijk}}{0.4 \tilde{s}_{ijk}^3 \sin(\phi_{ijk} - \Theta_{ijk})} \quad (82)$$

and

$$\chi_{F,ijk} = \frac{K_{\Theta,init}(\phi, \mu = 1)_{ijk} \Theta_{ijk}}{\tilde{s}_{ijk}^2 \sin(\phi_{ijk} - \Theta_{ijk})} \quad (83)$$

With this information now stored throughout the data structure, surfaces pertaining to the force only and distributed only cases could be analyzed. At this point, the correction factors could now incorporate, and adjust for, the imperfections resulting from the fits of both $\chi(\phi, \mu)$, as now manifested in a non-linear expression of Θ_{ijk} , and $K_{\Theta,init}(\phi, \mu)$. By using $\chi_{F,ijk}$, Θ_{ijk} , and ϕ_{ijk} , level contours of ϕ could be taken at 0.5° increments. The same was done for $\chi_{D,ijk}$. If the ϕ contours being taken were less than or equal to 100° , the contour containing each level curve of ϕ was truncated after exceeding $0.8(\Theta / \phi)$. If, however, the ϕ contours being taken were greater than 100° , the contour containing the level curve of ϕ was truncated upon exceeding 80° . The information contained therein was stored in four sets of three matrices corresponding to the χ , Θ

and ϕ coordinates of the data points along each contour. Figures (30)-(33) show the resulting surfaces and their corresponding fits. The resulting equations for the correction factors corresponding to load angles at or beneath 100° are:

$$\begin{aligned} \chi_F \left(\frac{\Theta}{\phi}, \frac{\phi}{\pi} \right) = & 0.9572 + 0.5625 \left(\frac{\phi}{\pi} \right) + 0.4362 \left(\frac{\Theta}{\phi} \right) - 3.358 \left(\frac{\phi}{\pi} \right)^2 - 1.796 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\phi} \right) \\ & - 3.918 \left(\frac{\Theta}{\phi} \right)^2 + 10.6 \left(\frac{\phi}{\pi} \right)^3 + 0.6124 \left(\frac{\phi}{\pi} \right)^2 \left(\frac{\Theta}{\phi} \right) + 13.34 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\phi} \right)^2 + 9.698 \left(\frac{\Theta}{\phi} \right)^3 \\ & - 16.79 \left(\frac{\phi}{\pi} \right)^4 + 6.454 \left(\frac{\phi}{\pi} \right)^3 \left(\frac{\Theta}{\phi} \right) - 18.3 \left(\frac{\phi}{\pi} \right)^2 \left(\frac{\Theta}{\phi} \right)^2 - 17.73 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\phi} \right)^3 - 11.79 \left(\frac{\Theta}{\phi} \right)^4 \\ & + 10.41 \left(\frac{\phi}{\pi} \right)^5 - 7.822 \left(\frac{\phi}{\pi} \right)^4 \left(\frac{\Theta}{\phi} \right) + 10.28 \left(\frac{\phi}{\pi} \right)^3 \left(\frac{\Theta}{\phi} \right)^2 + 6.785 \left(\frac{\phi}{\pi} \right)^2 \left(\frac{\Theta}{\phi} \right)^3 \\ & + 10.8 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\phi} \right)^4 + 5.143 \left(\frac{\Theta}{\phi} \right)^5 \end{aligned} \quad (84)$$

$$r^2 = 0.9952$$

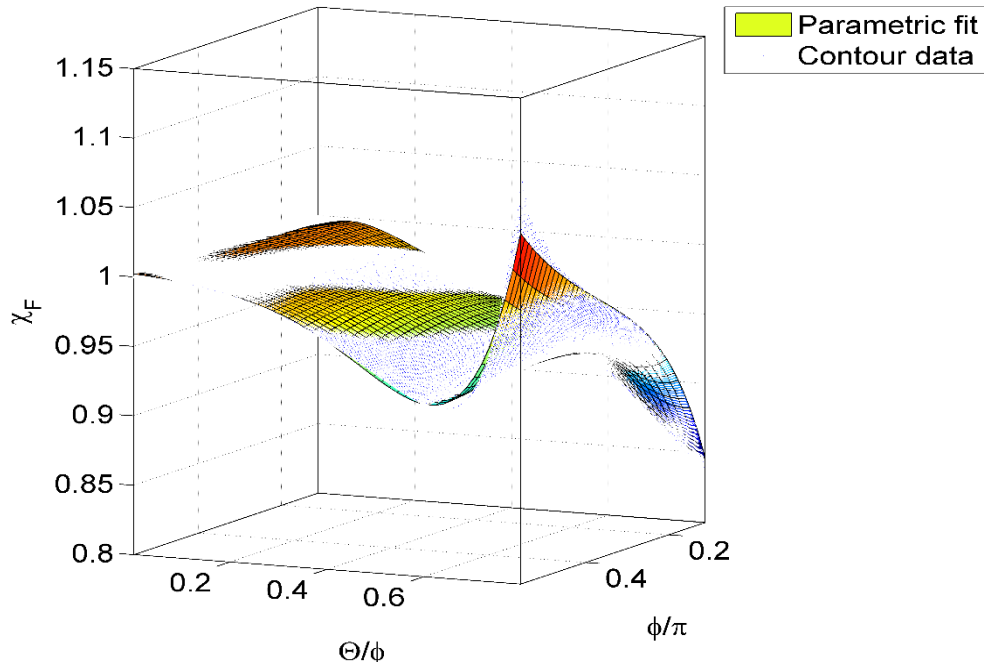


Figure 30 Correction factor of force contribution for $\phi \leq 100^\circ$.

$$\begin{aligned}
\chi_D\left(\frac{\Theta}{\phi}, \frac{\phi}{\pi}\right) = & 0.977 + 0.3589\left(\frac{\phi}{\pi}\right) + 0.3\left(\frac{\Theta}{\phi}\right) - 2.355\left(\frac{\phi}{\pi}\right)^2 - 0.4842\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\phi}\right) \\
& - 2.522\left(\frac{\Theta}{\phi}\right)^2 + 7.796\left(\frac{\phi}{\pi}\right)^3 - 1.915\left(\frac{\phi}{\pi}\right)^2\left(\frac{\Theta}{\phi}\right) + 7.867\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\phi}\right)^2 + 7.207\left(\frac{\Theta}{\phi}\right)^3 \\
& - 12.6\left(\frac{\phi}{\pi}\right)^4 + 7.218\left(\frac{\phi}{\pi}\right)^3\left(\frac{\Theta}{\phi}\right) - 10.76\left(\frac{\phi}{\pi}\right)^2\left(\frac{\Theta}{\phi}\right)^2 - 10.62\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\phi}\right)^3 - 9.493\left(\frac{\Theta}{\phi}\right)^4 \\
& + 7.87\left(\frac{\phi}{\pi}\right)^5 - 7.536\left(\frac{\phi}{\pi}\right)^4\left(\frac{\Theta}{\phi}\right) + 8.156\left(\frac{\phi}{\pi}\right)^3\left(\frac{\Theta}{\phi}\right)^2 + 1.761\left(\frac{\phi}{\pi}\right)^2\left(\frac{\Theta}{\phi}\right)^3 \\
& + 7.767\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\phi}\right)^4 + 4.517\left(\frac{\Theta}{\phi}\right)^5
\end{aligned} \tag{85}$$

$$r^2 = 0.9986$$

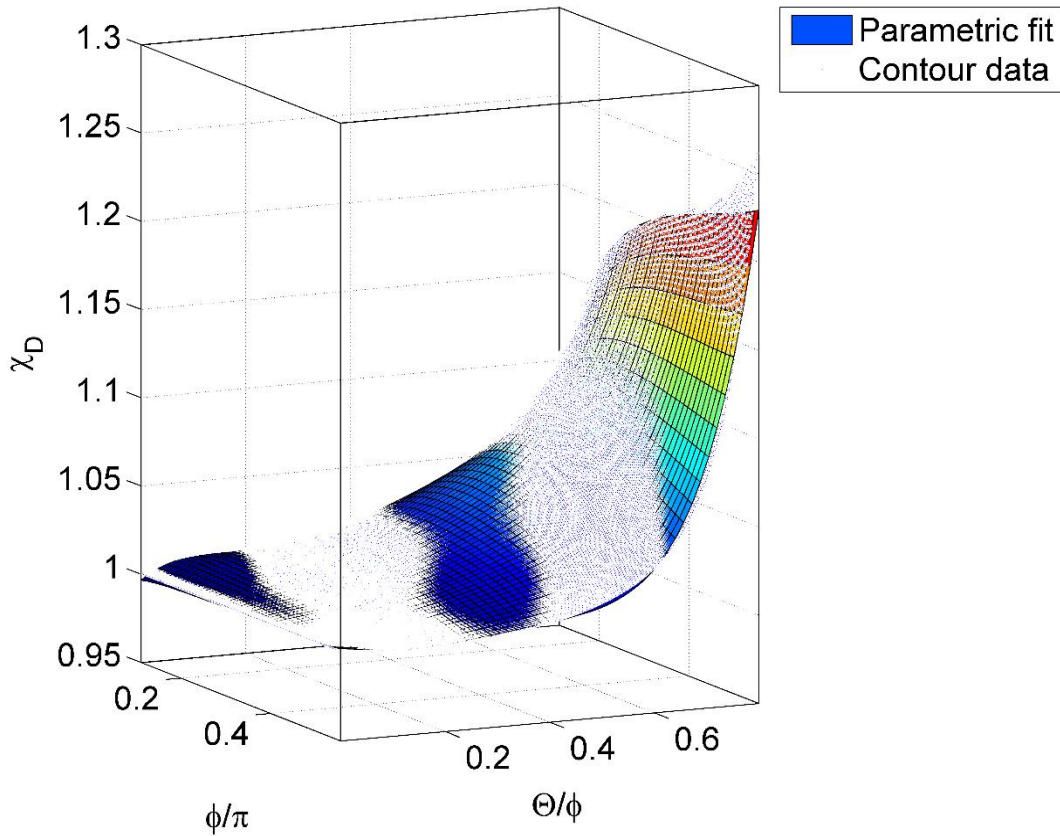


Figure 31 Correction factor of distributed contribution for $\phi \leq 100^\circ$.

The resulting equations for the correction factors corresponding to load angles above 100° are:

$$\begin{aligned} \chi_F \left(\frac{\Theta}{\pi}, \frac{\phi}{\pi} \right) = & 1.608 - 4.706 \left(\frac{\phi}{\pi} \right) + 2.225 \left(\frac{\Theta}{\pi} \right) + 14.46 \left(\frac{\phi}{\pi} \right)^2 - 15.06 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\pi} \right) \\ & + 2.722 \left(\frac{\Theta}{\pi} \right)^2 - 22.11 \left(\frac{\phi}{\pi} \right)^3 + 38.49 \left(\frac{\phi}{\pi} \right)^2 \left(\frac{\Theta}{\pi} \right) - 25.48 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\pi} \right)^2 + 11.91 \left(\frac{\Theta}{\pi} \right)^3 \\ & + 16.86 \left(\frac{\phi}{\pi} \right)^4 - 44.84 \left(\frac{\phi}{\pi} \right)^3 \left(\frac{\Theta}{\pi} \right) + 68.46 \left(\frac{\phi}{\pi} \right)^2 \left(\frac{\Theta}{\pi} \right)^2 - 94.92 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\pi} \right)^3 + 62.37 \left(\frac{\Theta}{\pi} \right)^4 \\ & - 5.134 \left(\frac{\phi}{\pi} \right)^5 + 19.9 \left(\frac{\phi}{\pi} \right)^4 \left(\frac{\Theta}{\pi} \right) - 54.01 \left(\frac{\phi}{\pi} \right)^3 \left(\frac{\Theta}{\pi} \right)^2 + 116.2 \left(\frac{\phi}{\pi} \right)^2 \left(\frac{\Theta}{\pi} \right)^3 \\ & - 119.8 \left(\frac{\phi}{\pi} \right) \left(\frac{\Theta}{\pi} \right)^4 + 32.37 \left(\frac{\Theta}{\pi} \right)^5 \end{aligned} \quad (86)$$

$$r^2 = 0.9954$$

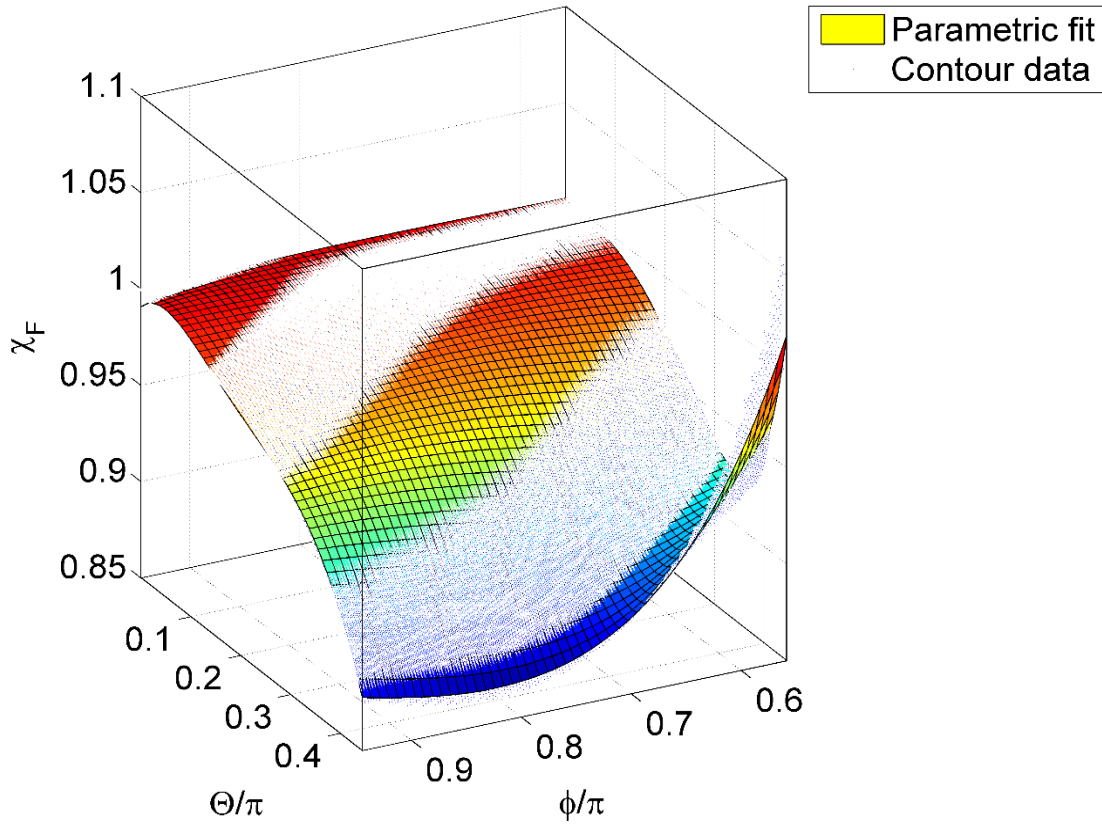


Figure 32 Correction factor of force contribution for $\phi > 100^\circ$.

$$\begin{aligned}
\chi_D\left(\frac{\Theta}{\pi}, \frac{\phi}{\pi}\right) = & 2.658 - 12.66\left(\frac{\phi}{\pi}\right) + 5.379\left(\frac{\Theta}{\pi}\right) + 38.49\left(\frac{\phi}{\pi}\right)^2 - 32.82\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\pi}\right) \\
& + 9.967\left(\frac{\Theta}{\pi}\right)^2 - 58.19\left(\frac{\phi}{\pi}\right)^3 + 76.2\left(\frac{\phi}{\pi}\right)^2\left(\frac{\Theta}{\pi}\right) - 49.82\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\pi}\right)^2 + 16.61\left(\frac{\Theta}{\pi}\right)^3 \\
& + 43.74\left(\frac{\phi}{\pi}\right)^4 - 79.15\left(\frac{\phi}{\pi}\right)^3\left(\frac{\Theta}{\pi}\right) + 87.3\left(\frac{\phi}{\pi}\right)^2\left(\frac{\Theta}{\pi}\right)^2 - 79.33\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\pi}\right)^3 + 42.31\left(\frac{\Theta}{\pi}\right)^4 \\
& - 13.07\left(\frac{\phi}{\pi}\right)^5 + 29.61\left(\frac{\phi}{\pi}\right)^4\left(\frac{\Theta}{\pi}\right) - 46.27\left(\frac{\phi}{\pi}\right)^3\left(\frac{\Theta}{\pi}\right)^2 + 67.32\left(\frac{\phi}{\pi}\right)^2\left(\frac{\Theta}{\pi}\right)^3 \\
& - 54.47\left(\frac{\phi}{\pi}\right)\left(\frac{\Theta}{\pi}\right)^4 + 4.942\left(\frac{\Theta}{\pi}\right)^5
\end{aligned} \tag{87}$$

$$r^2 = 0.9937$$

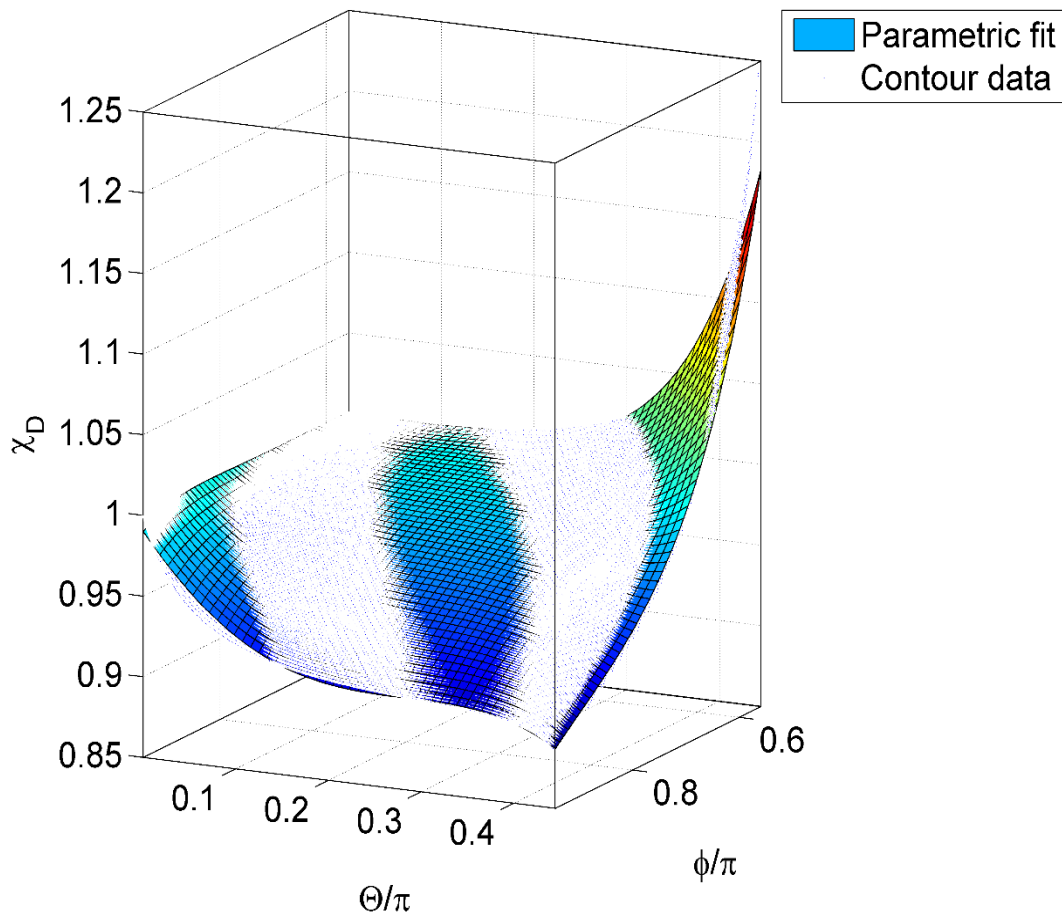


Figure 33 Correction factor of distributed contribution for $\phi > 100^\circ$.

The effect of these correction factors is shown in Figure (34). As can be seen, the contour, which once readily diverged from the K_{θ} located at the root, now conforms to the vertical line recommended by $K_{\theta,init}(\phi, \mu)$. However, it was realized that contours associated with intermediate values of μ , between 0 and 1, were displaced from the vertical line centered at the root. While the behavior imparted upon them by the correction factors persisted, this newfound verticality was shifted to the right or left. The cause of this behavior remains unknown at this juncture, but it can still be corrected for. Upon realizing this, new values of $K_{\theta}(\phi, \mu)$ were chosen by, again, using the contour method described earlier. This time, however, the analysis was conducted from $\mu = 0.02$

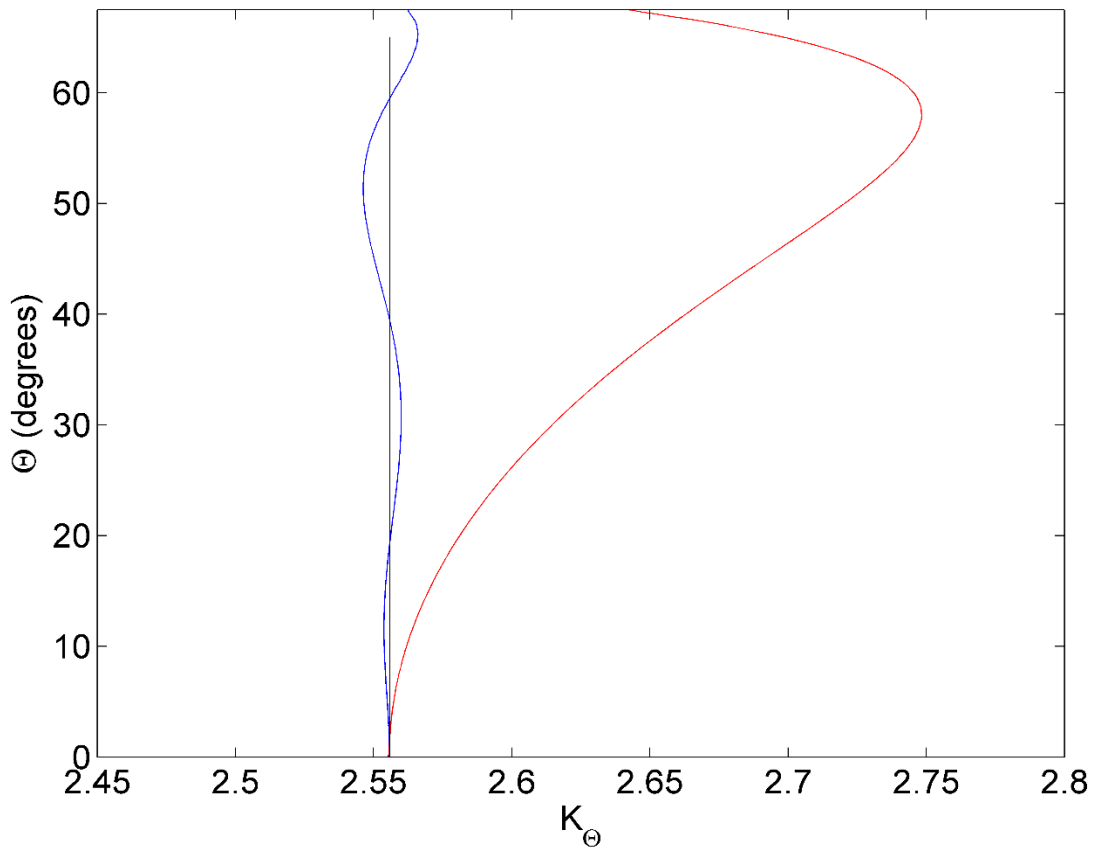


Figure 34 Effect of correction factors on a single contour for $\mu = 1$ and $\phi = 90^\circ$. The red line is the result of the transverse load index alone. The black line is the recommended value of K_{θ} and the blue line is the corrected contour.

to $\mu = 0.98$. This was done to avoid destroying the interrelationship between the correction factors and $K_{\Theta,init}(\phi, \mu)$, for values of $\mu = 0$ and $\mu = 1$, thereby holding constant the vertical nature of the contours within the intermediate values of μ . At the same load angle intervals and range as was used to construct equation (81), the analysis now used the *average* value of the contour as the recommended K_{Θ} at each (ϕ, μ) coordinate, instead of the root value. Figure (35) shows the utility of applying this method to a single contour. The resulting surface was then used to append the

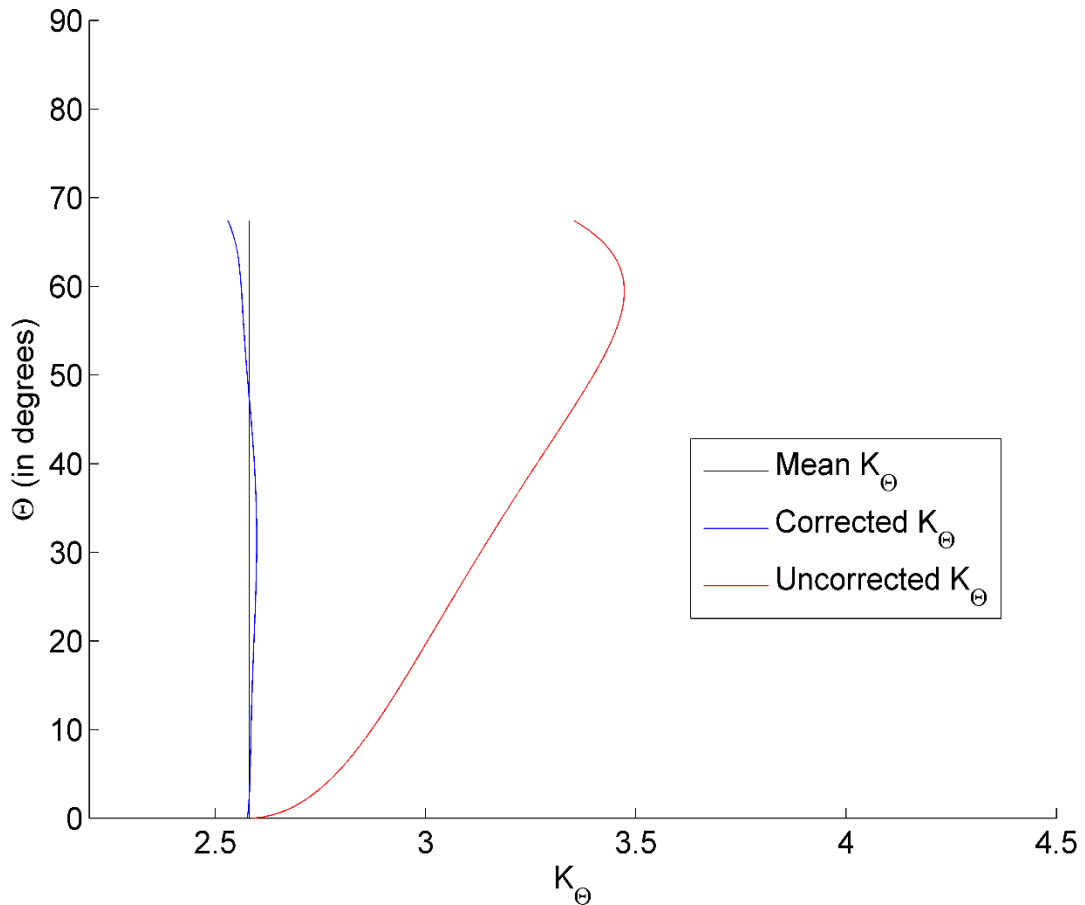


Figure 35 The departure from verticality for $\mu = 0.6$. The red line is the contour with only the uncorrected transverse load indices while the blue line

Model and data comparison for K_{θ}

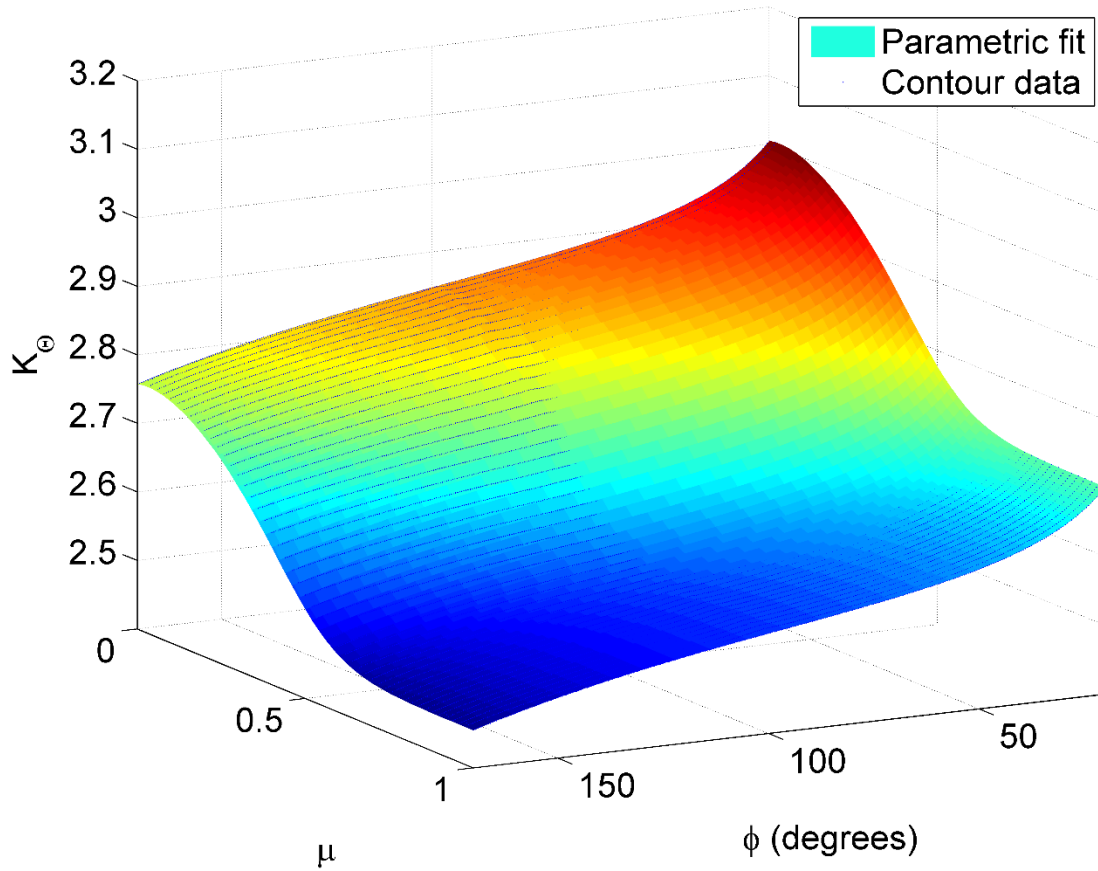


Figure 36 The final surface fit for K_{θ} .

original $K_{\theta,init}(\phi, \mu)$ surface described by equation (81). In other words, the $\mu = 0$ and $\mu = 1$ contour data was kept, while all else was replaced by the new average values of K_{θ} . Doing so resulted in the surface shown in Figure (36). The equation which was fit to this surface represents the final $K_{\theta}(\phi, \mu)$ equation used in the PRBM and is described below as:

$$K_{\Theta}(\phi, \mu) = \quad (88)$$

$$\left(2.398 + \frac{0.5074}{\ln\left(5.062 \tan^3\left(\frac{\phi}{2}\right) - 5.485 \tan^2\left(\frac{\phi}{2}\right) + 24.11 \tan\left(\frac{\phi}{2}\right) + 1.18\right)} \right) + \left(\frac{-0.248\mu^2 - 0.2399\mu + 0.3882}{(1 + \exp(-11.21\mu^3 + 22.52\mu^2 - 4.814\mu - 1.258))} \right)$$

$$r^2 = 0.9996$$

The resulting equations describing the values of $\Theta_{max}(K_{\Theta})$ for varying values of ϕ are given as:

$$\phi \leq 100^{\circ} \rightarrow \Theta_{max,K} = 0.75\phi \quad (89)$$

$$\phi > 100^{\circ} \rightarrow \Theta_{max,K} = 75^{\circ} \quad (90)$$

Here, $\Theta_{max,K}$ describes the maximum Θ values associated with K_{Θ} . Also, the angular range of $\Theta_{max,K}$ was shortened due to imperfections in the fits of the correction factors, mainly clustered around the higher values of Θ . The final dimensional form of K_{Θ} can now be described as:

$$K_{\Theta}(\phi, \mu)^{\Theta} = \left(\frac{Fl^2 \chi_F}{EI} + \frac{0.4wl^3 \chi_D}{EI} \right) \sin(\phi - \Theta) \quad (91)$$

The final non-dimensional form of K_{Θ} is given by:

$$K_{\Theta}(\phi, \mu)\Theta = (\mu^2 \tilde{s}^2 \chi_F + 0.4(1 - \mu)^3 \tilde{s}^3 \chi_D) \sin(\phi - \Theta) \quad (92)$$

Lastly, Figure (37) shows the PRBM associated with cantilever beams undergoing combinations of endpoint forces and uniformly distributed loads applied in parallel.

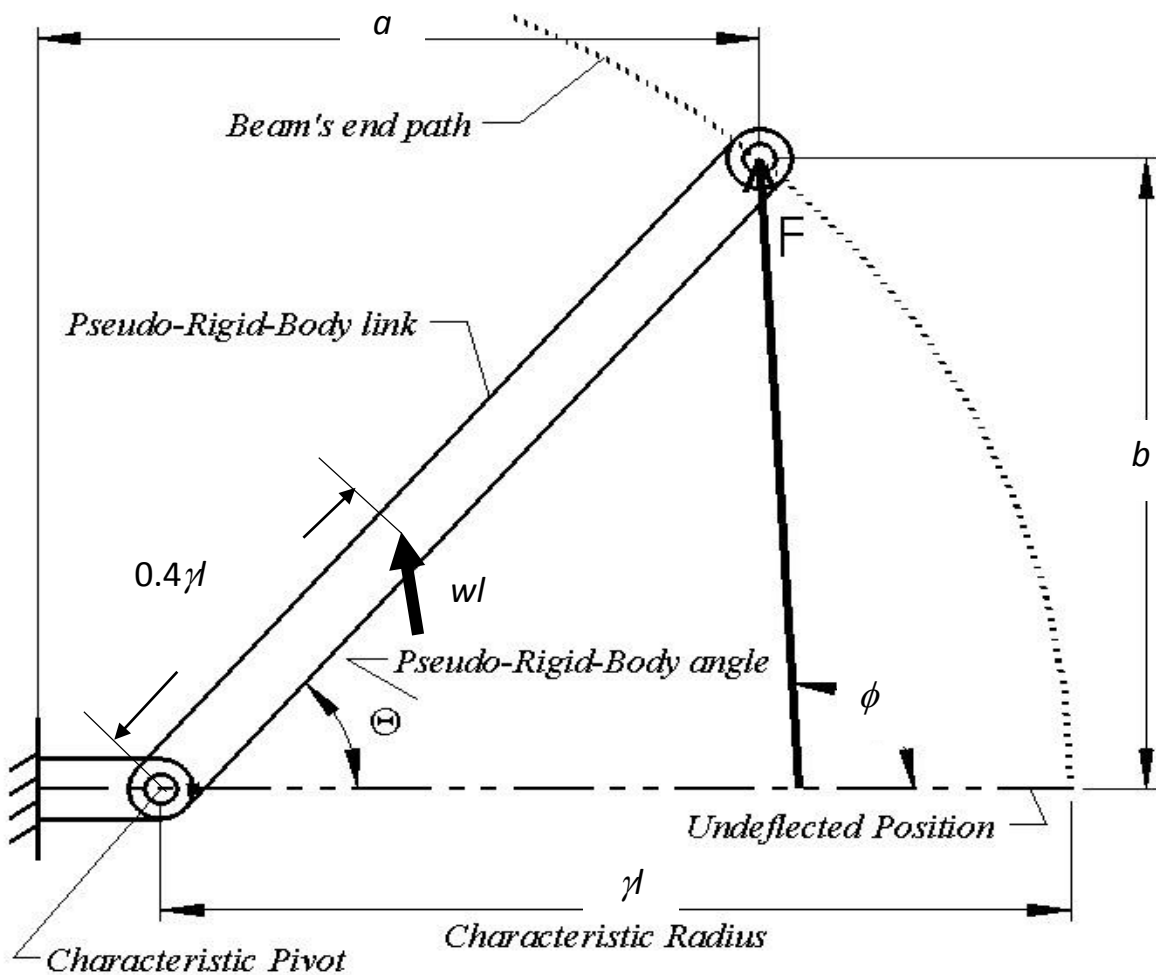


Figure 37 The PRBM for cantilever beams undergoing combined parallel loading. This shows the concentration of the distributed load, wl , acting at a distance of 0.4γ away from the characteristic pivot.

CHAPTER 6: RESULTS

Now that the PRBM parametric equations have been constructed, an error analysis can be performed. However, if one acknowledges the sheer quantity of comparisons which can be made, this presents a dilemma. As a relative approximate error, now defined as ε , can represent one dimension, this error must, in some fashion, traverse the space generated by varying values of ϕ , μ , and Θ . This represents a four dimensional object of error analysis. As there are several variables within which this error can be manifested, i.e. F , l , E , I and w , there are seemingly a multitude of four dimensional objects required for a proper investigative analysis of the error. Doing so in tabular form may be considered less than illuminating, as this would require a veritable pamphlet of tables.

However, it will be shown that the analysis can be drastically simplified for $\mu = 1$ and $\mu = 0$ if the errors associated with each of the variables can be related to the errors associated with one variable. Contour plots of that single error can then be given, thereby summarizing the errors associated with all variables of interest. It is here that the non-dimensional arc length is, again, employed. Then, an example problem will demonstrate the determination of μ for an intermediate case, where $0 < \mu < 1$. Lastly, the methods employed during the example problem will be extended to a greater number of beam cases, so that a more thorough investigation of the error in μ , for an intermediate μ value, can be conducted. Justifications for the use of FEA to evaluate the error will also be given.

6.1 Error Evaluation for $\mu = 1$ and $\mu = 0$

The relative approximate error associated with \tilde{s} can be given as:

$$\varepsilon_{\tilde{s}} = \left(\frac{\tilde{s}_a - \tilde{s}}{\tilde{s}} \right) \times 100\% = \left(\frac{\tilde{s}_a}{\tilde{s}} - 1 \right) \times 100\% \quad (93)$$

where \tilde{s}_a is defined as the approximate value of \tilde{s} . First, it should be demonstrated that the errors associated with s , which is equivalent to l , are themselves equivalent to the errors associated with \tilde{s} . This can be shown via:

$$s = \tilde{s} \lambda_3 \therefore \frac{s_a}{s} = \frac{\tilde{s}_a \lambda_3}{\tilde{s} \lambda_3} = \frac{\tilde{s}_a}{\tilde{s}} \quad (94)$$

therefore

$$\varepsilon_s = \varepsilon_{\tilde{s}} \quad (95)$$

Now, if, for $\mu = 1$, the force is the only unknown or if, for $\mu = 0$, the distributed load is the only unknown, they can be described in terms of \tilde{s} by recognizing the following relations:

$$\begin{aligned} \tilde{s} &= \frac{s}{\lambda_1 \mu} = \frac{s}{\lambda_2 (1 - \mu)} \\ \therefore \tilde{s} &= \frac{s}{\mu} \sqrt{\frac{F}{EI}} = \frac{s}{(1 - \mu)} \sqrt[3]{\frac{w}{EI}} \end{aligned} \quad (96)$$

Therefore, the error in F , as defined by:

$$\varepsilon_F = \left(\frac{F_a}{F} - 1 \right) \times 100\% \quad (97)$$

can be rewritten in terms of \tilde{s} via:

$$\varepsilon_F = \left(\frac{\left(\frac{\tilde{s}_a \mu}{s} \right)^2 EI}{\left(\frac{\tilde{s} \mu}{s} \right)^2 EI} - 1 \right) \times 100\% = \left(\left(\frac{\tilde{s}_a}{\tilde{s}} \right)^2 - 1 \right) \times 100\% \quad (98)$$

Upon appropriate substitution, the error in F , for $\mu = 1$, can now be defined in terms of the error in \tilde{s} as:

$$\varepsilon_F = \left(\left(\frac{\varepsilon_{\tilde{s}}}{100} + 1 \right)^2 - 1 \right) \times 100\% \quad (99)$$

Employing a similar methodology, it can be shown that the error in w , for $\mu = 0$, can now be defined in terms of the error in \tilde{s} as:

$$\varepsilon_w = \left(\left(\frac{\varepsilon_{\tilde{s}}}{100} + 1 \right)^3 - 1 \right) \times 100\% \quad (100)$$

Lastly, if E or I is the only unknown, then either F or w is known, for $\mu = 1$ or $\mu = 0$, respectively. Hence, either λ_1 or λ_2 may be used to relate the errors. Therefore, again employing a similar methodology, the error in E or I can now be represented in terms of $\tilde{\delta}$, for $\mu = 1$ and $\mu = 0$, via:

$$\varepsilon_{E,I} = \left(\left(\frac{1}{\left(\frac{\varepsilon_{\tilde{\delta}}}{100} + 1 \right)} \right)^3 - 1 \right) \times 100\% \quad (101)$$

for the distributed load only case, and:

$$\varepsilon_{E,I} = \left(\left(\frac{1}{\left(\frac{\varepsilon_{\tilde{\delta}}}{100} + 1 \right)} \right)^2 - 1 \right) \times 100\% \quad (102)$$

for the force only case. Equations (95) and (99)-(102) will now allow the transformation of the errors between the approximate values of $\tilde{\delta}$, as predicted by the PRBM, and the precise values of $\tilde{\delta}$, as calculated by numerical integration, into errors associated with whichever variable is of interest, for $\mu = 1$ and $\mu = 0$.

At this point, contour plots of $\varepsilon_{\tilde{\delta}}$ must be created. This is done by populating the data structure with PRBM predictions of the value of $\tilde{\delta}$ and then comparing these values with those already input into the differential equation solver, and, with regards to the data structure,

subsequently stored therein. This was done by solving for the roots of the following expression at each location within the data structure corresponding to $\mu = 1$ and $\mu = 0$:

$$\frac{K_{\Theta,ijk}(\phi_{ijk}, \mu_k) \Theta_{ijk}(\phi_{ijk}, \mu_k)}{\sin(\phi_{ijk} - \Theta_{ijk}(\phi_{ijk}, \mu_k))} - \mu_k^2 \tilde{s}_{ijk}^2 \chi_{F,ijk}(\phi_{ijk}, \mu_k) - 0.4(1 - \mu_k)^3 \tilde{s}_{ijk}^3 \chi_{D,ijk}(\phi_{ijk}, \mu_k) = 0 \quad (103)$$

Here, all variables associated with the PRBM, including Θ , are represented in terms of functions of ϕ and μ . For the case of Θ , and terms which are functions of Θ , this dependence arises from the dependence of γ on ϕ and μ . At this point, an interesting problem arises. If one were to employ the PRBM to attempt to find l , which is represented by s , equation (103) would not provide a direct method of obtaining the roots. This is because of the definition of Θ , where s appears in the denominator. Equation (103) would then become a higher order trigonometric polynomial in s , because of the 5th order polynomial of Θ , located within χ_D , resulting in a 5th order polynomial of arctangents inversely related to s , which then interacts with the sin function within the denominator. There is, of course, χ_F to consider, as well. In lieu of this observation, it is highly recommended not to attempt to directly solve equation (103) for s , if that is the only unknown. If a value of s were so desired, a recommended course of action would be to express equation (103) symbolically within a programming software and then to populate that expression with a densely populated vector of s values and use linear interpolation to solve for the root in a region where realistic s values are presumed to exist. This is the method employed in finding μ for the example problem later given. It should be noted that, while equation (95) is logically sound, it will not be

able to relate the errors in s to the errors in \tilde{s} by solving for the roots of equation (103), and will thus be exempt from the proposed error relationships.

If, however, the beam length is known beforehand, then the argument of the arctangent is a known function of ϕ and μ , in recognition of equation (49). Therefore, all of the error is manifested inside of \tilde{s} and equations (99)-(102) can then be employed. Figures (38) and (39) display the results of the relative approximate error of the realistic root, often the only non-negative root, to equation (103). Also shown, in Figure (40), is a contour plot of the error in \tilde{s} associated with the PRBM developed by Howell et al [43].

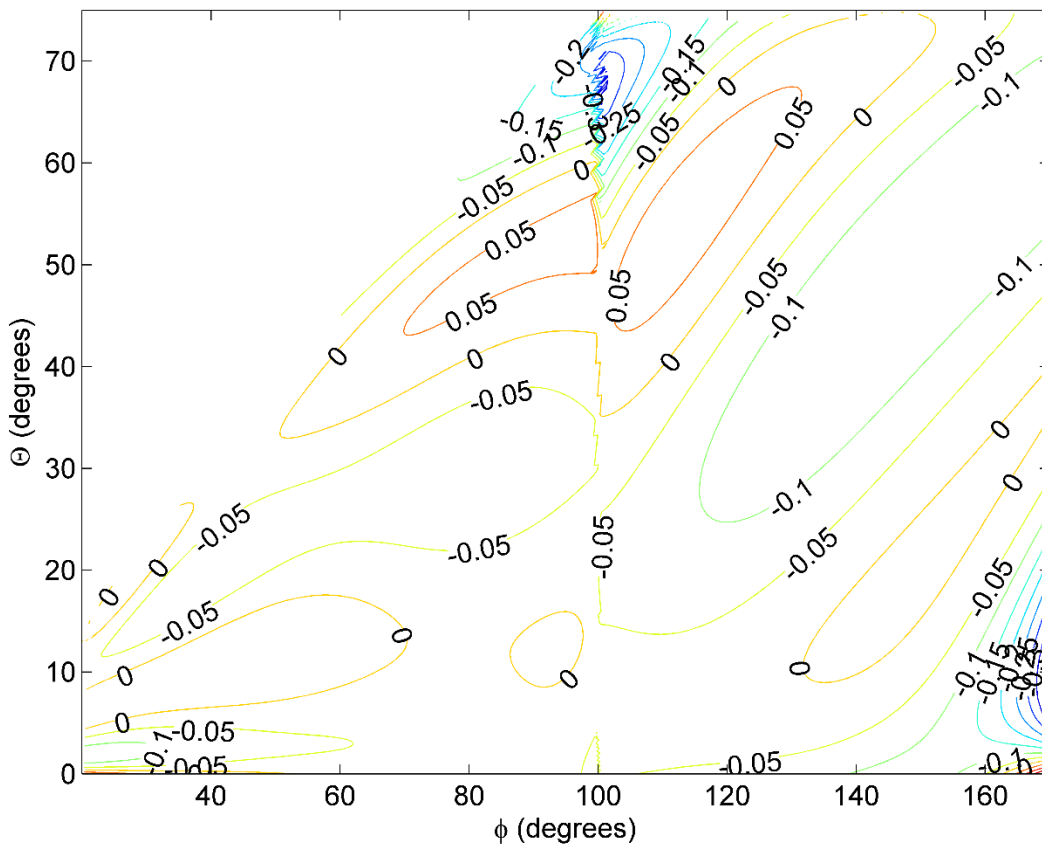


Figure 38 Contours of % relative error in \tilde{s} for $\mu = 0$.

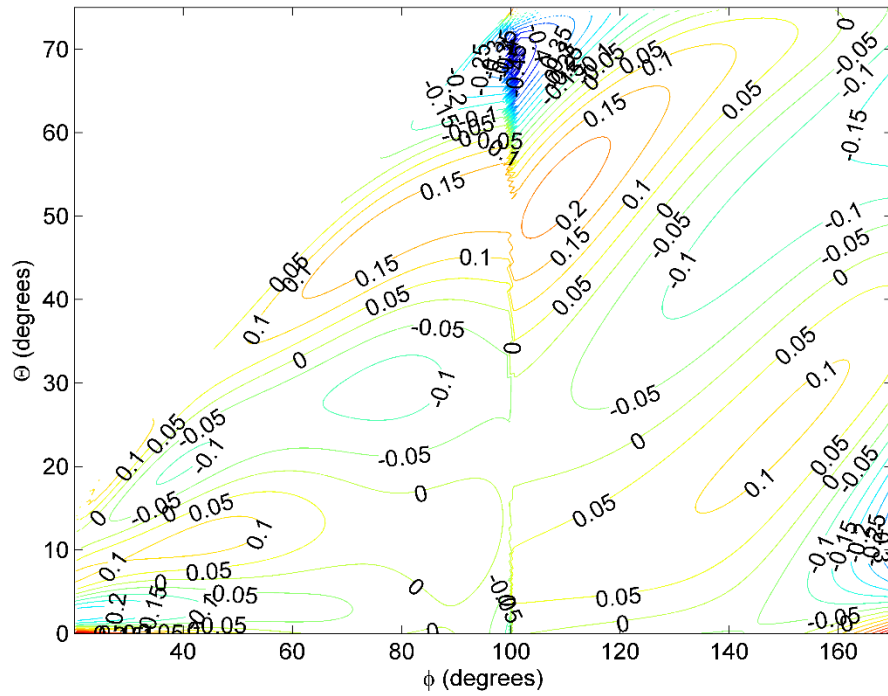


Figure 39 Contours of % relative error in \tilde{s} for $\mu = 1$.

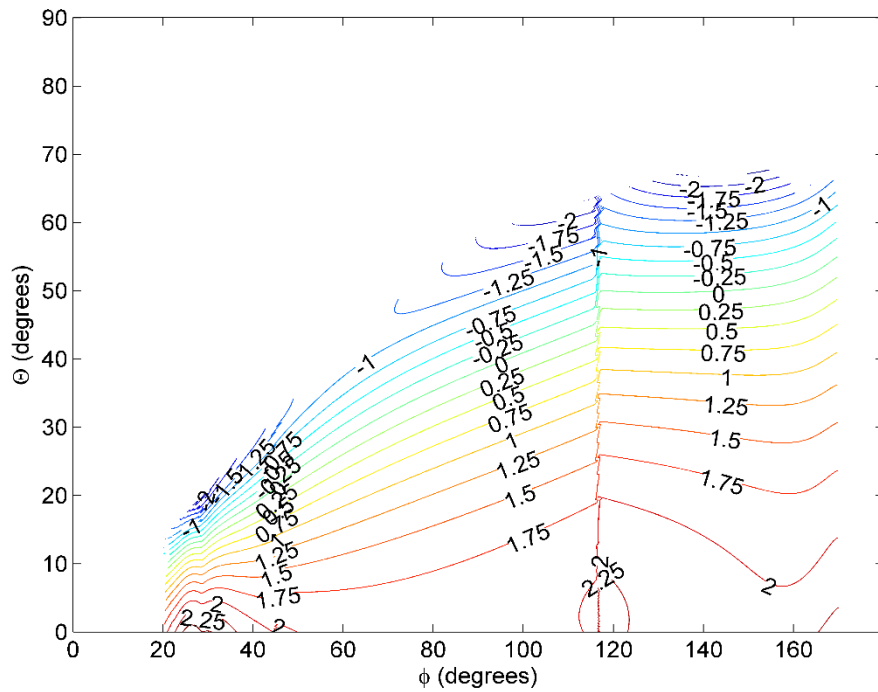


Figure 40 Contours of % relative error in \tilde{s} associated with the PRBM constructed by Howell et al [3].

6.2 Example Problem

The same beam used to compare the FEA non-dimensional beam characteristics with the predictions from the differential equations, as shown in Figures (10) and (11), will now be used to verify the results of the PRBM in an actual combined loading problem. Assume the following: the modulus of elasticity, $E = 200 \times 10^9$ Pa, the second moment of area, $I = 5.4752 \times 10^{-13}$ m⁴, the beam length, $l = 0.44174$ m and the vertical deflection, $b = 0.10363$ m. The load angle is given by $\phi = 90^\circ$. The load magnitudes required to create the deflection, b , given the initial conditions, I , E , L and ϕ , were those applied during the FEA, and have values of force, $F = 0.24$ N, and distributed load, $w = 1.0623557$ N/m. These magnitudes, along with the values of E and I , result in a precise value of $\mu = 0.4097$.

Now, if the value of the distributed load is unknown, but all other information is present, the PRBM designed herein results in a one-equation, one-unknown system via:

$$\frac{K_{\Theta}(\mu)\Theta(\mu)}{\sin(\phi - \Theta(\mu))} - \chi_F(\mu)\left(\frac{L^2 F}{EI}\right) - 0.4L^3 \chi_D(\mu)\left(\frac{F}{EI}\right)^{\frac{3}{2}}\left(\frac{1-\mu}{\mu}\right)^3 = 0 \quad (104)$$

Notice how all PRBM parameters, including the correction factors, are functions of μ . It is, therefore possible to evaluate the above expression for μ values within the range of 0 to 1. Directly solving the above expression for μ is, however, highly discouraged. Not only would such an attempt prove computationally costly and time consuming, but would also produce values of μ which lie outside of the realistic boundaries of μ , namely 0 and 1. The realistic root to the above expression was, however, found by inputting a vector of μ values ranging from 0.001 to 0.999, in increments of 0.001, and then using linear interpolation to find the root. This resulted in a predicted

μ value of 0.4103 and a corresponding predicted w value of 1.0555 N/m. This represents a relative approximate error of 0.1283% in μ and -0.6496% in w .

If, however, the value of the force is unknown, but all other information is present, the PRBM function of μ to evaluate is given by:

$$\frac{K_{\Theta}(\mu)\Theta(\mu)}{\sin(\phi - \Theta(\mu))} - L^2 \chi_F(\mu) \left(\frac{w}{EI}\right)^{\frac{2}{3}} \left(\frac{\mu}{1-\mu}\right)^2 - 0.4 \chi_D(\mu) \left(\frac{L^3 w}{EI}\right) = 0 \quad (105)$$

The above expression, containing a solution which was found by using the previously described method, resulted in a predicted value for μ of 0.4091 and a corresponding predicted value for the endpoint force of 0.2388 N. This represents a relative approximate error of -0.1429% in μ and -0.4832% in F . Figure (41) depicts a graphical representation of the output vectors of equations (104) and (105), as well as the precise value of μ .

Here, solving the dimensional form of K_{Θ} , equation (91), entails reformulating the unknown quantities into functions of μ . To this end, the following relations may prove useful:

$$w = \frac{F^{\frac{3}{2}}}{\sqrt{EI}} \left(\frac{1-\mu}{\mu}\right)^3 \quad (106)$$

$$F = \sqrt[3]{EI} w^{2/3} \left(\frac{\mu}{1-\mu}\right)^2 \quad (107)$$

$$EI = \frac{F^3}{w^2} \left(\frac{1-\mu}{\mu}\right)^6 \quad (108)$$

Using equations (106)-(108), one can also derive the PRBM function of μ for when either E or I is the only unknown, as both give the same form of the equation, via:

$$\frac{K_{\Theta}(\mu)\Theta(\mu)}{\sin(\phi - \Theta(\mu))} - \left(\frac{1-\mu}{\mu}\right)^6 \left[\chi_F(\mu)\left(\frac{LW}{F}\right)^2 + \chi_D(\mu)\left(\frac{LW}{F}\right)^3 \right] = 0 \quad (109)$$

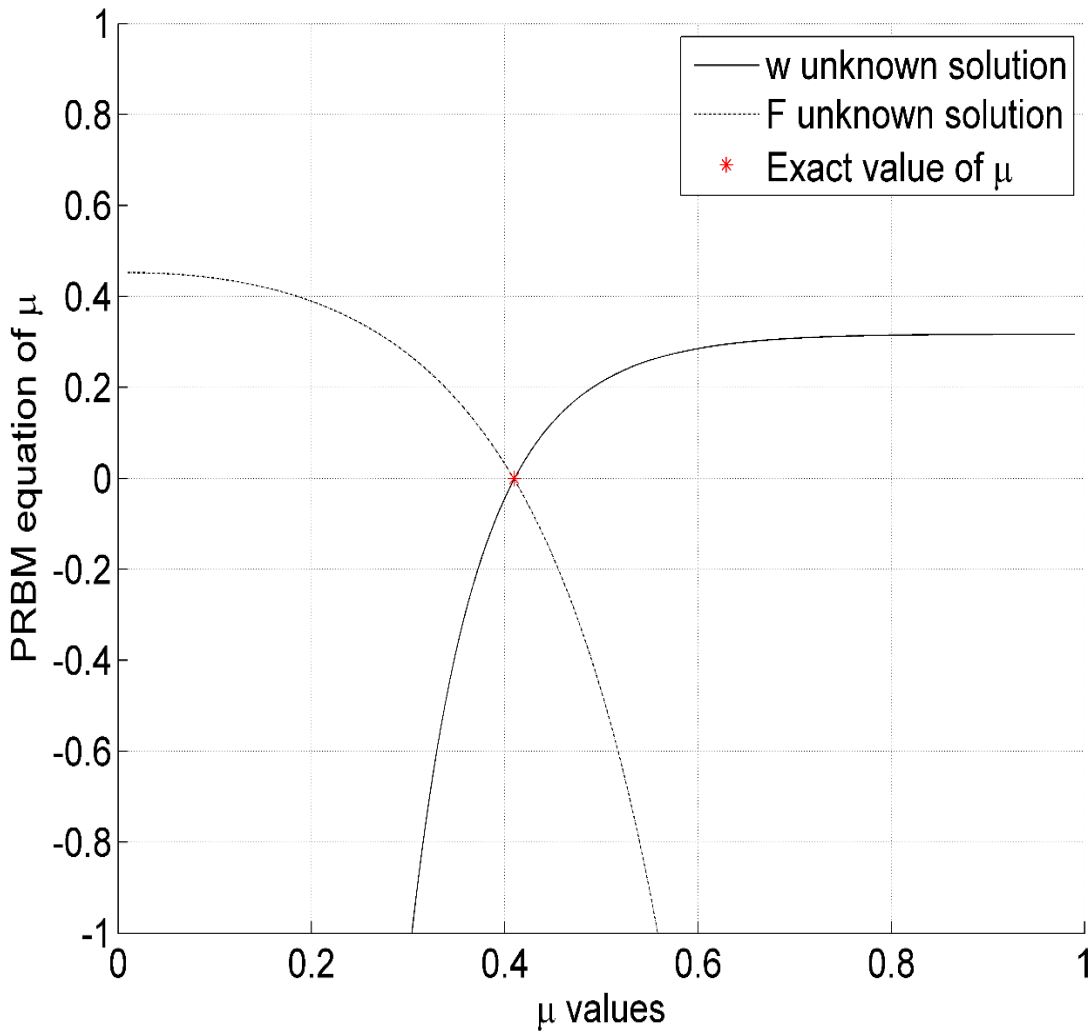


Figure 41 Output vectors from the PRBM functions of μ (equations (104) and (105)).

6.3 Error Analysis of Intermediate μ Values

At this point, error contours similar to Figures (38) and (39), for intermediate values of μ , may be deemed appropriate. Such contours would represent the % relative error in μ , as opposed to $\tilde{\kappa}$. However, in the process of constructing these contours, by using the non-dimensional equation (103) to provide a prediction of μ at each location within the data structure, it was discovered that equation (103) led to simultaneous solutions within the expected range of μ values, namely between 0 and 1. While root values obtained by restricting the investigation to the local neighborhood of the exact μ value, namely between $\mu_{exact} - 0.025$ and $\mu_{exact} + 0.025$, did produce promising results, with a % relative error in μ less than 2% for almost all regions, this error was found to be unrelated to the errors obtained by solving the dimensional cases, namely equations (104) and (105). This is because the forms of the equations are different, being different functions of μ . In other words, depending upon which value is being sought, be it w , F , E , or I , the PRBM developed herein will predict different values of μ and by different curves.

This can be illustrated by, again, using FEA to investigate the dimensional forms of the $K\theta$ equation, namely equations (104) and (105). This was done for the entirety of the beam cases studied in Figures (11a) and (11b) for $\mu = 0.40973$. The methods employed in Section 6.2 were used to predict the value of μ when both F was unknown and when w was unknown. Those values of μ were then used to calculate the values of F and w . Figure (42) shows the data from the perspective of the phase plane. The % relative error in μ , with F being unknown, and the % relative error in F are shown in Figures (43) and (44), respectively. The % relative error in μ , with w being unknown, and the % relative error in w are shown in Figures (45) and (46), respectively. While there are apparent fluctuations in the errors for small $\tilde{\kappa}$ values, these correspond to regions of small deflections. In lieu of this data, it is recommended that the PRBM developed herein should only

be used for large deflection beam cases, if a combined loading situation is, indeed, present. Closer analysis of Figures (43)-(46) reveals that the errors become less than 2% after angular deflections of $\theta \approx 5^\circ$ and the accuracy of the model increases as the deflection of the beam becomes larger. It should also be noted that the calculation of ϕ for each data point was done using equation (41) and represents a close approximation of ϕ , but an approximation nonetheless. Also, the difference between predicted μ values, when either F is unknown or w is unknown, is visible. Throughout these calculations, the range of μ values input into equations (104) and (105) was between 0.001 and 0.999. Hence, the data shown in Figures (43)-(46) represent unique values within the range $0 < \mu < 1$, and thereby avoid the simultaneous solutions found in a non-dimensional error analysis.

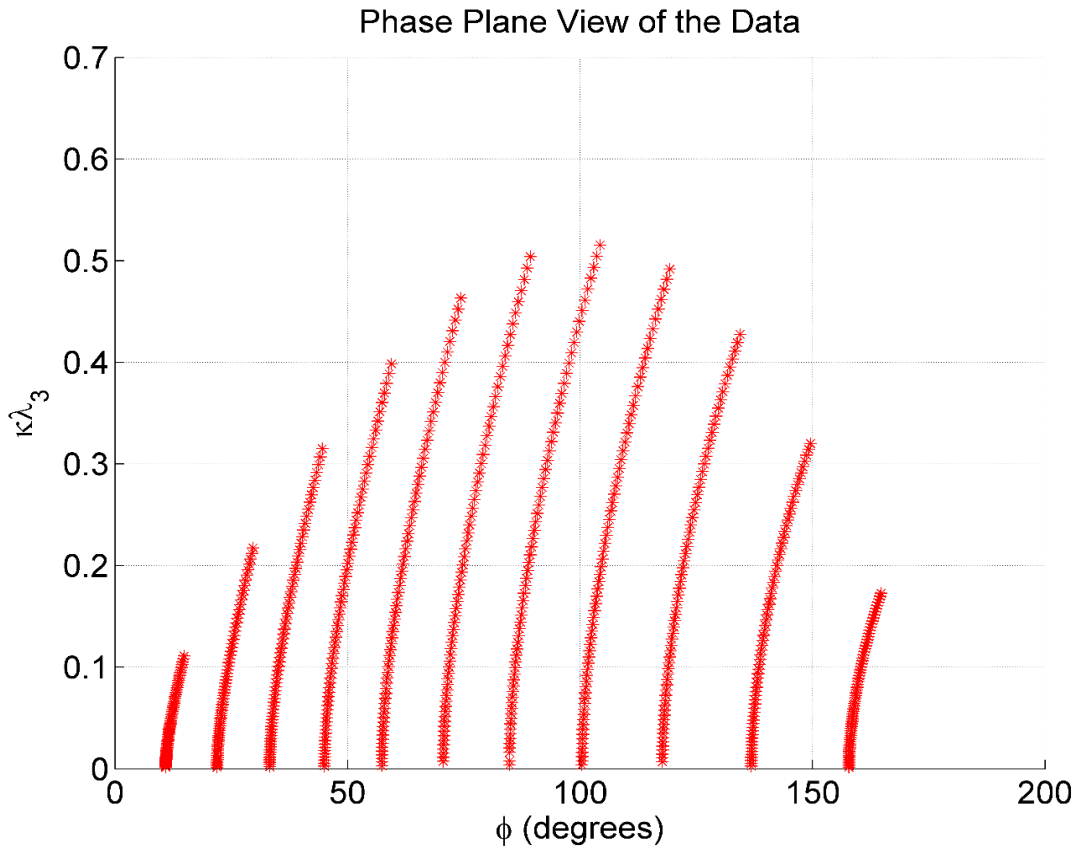


Figure 42 The phase plane view of the data used in the dimensional error analysis.

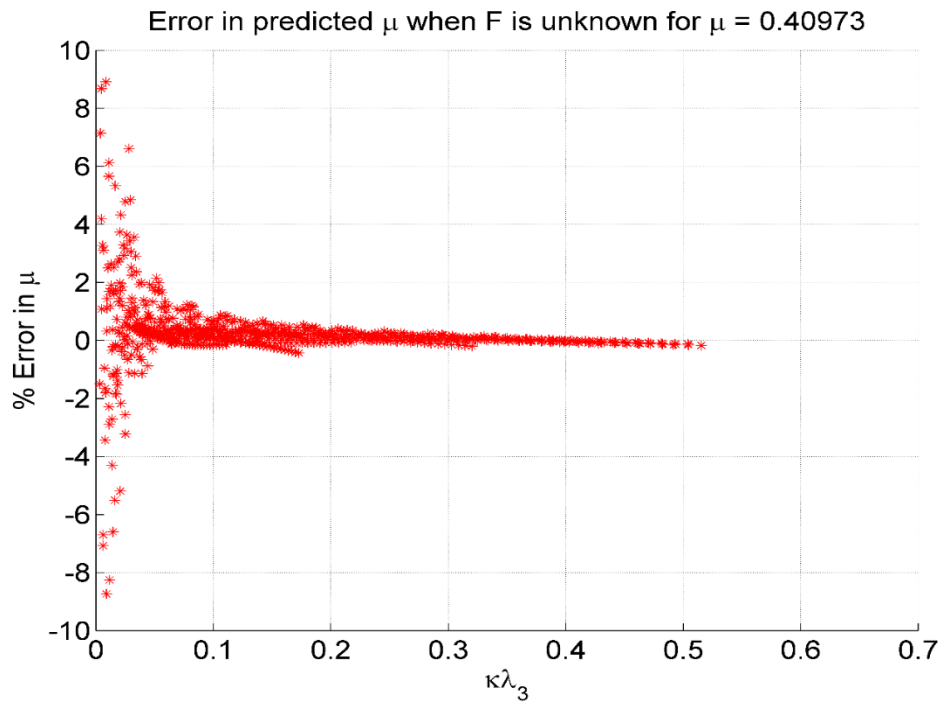


Figure 43 The % relative error in μ when F is the unknown.

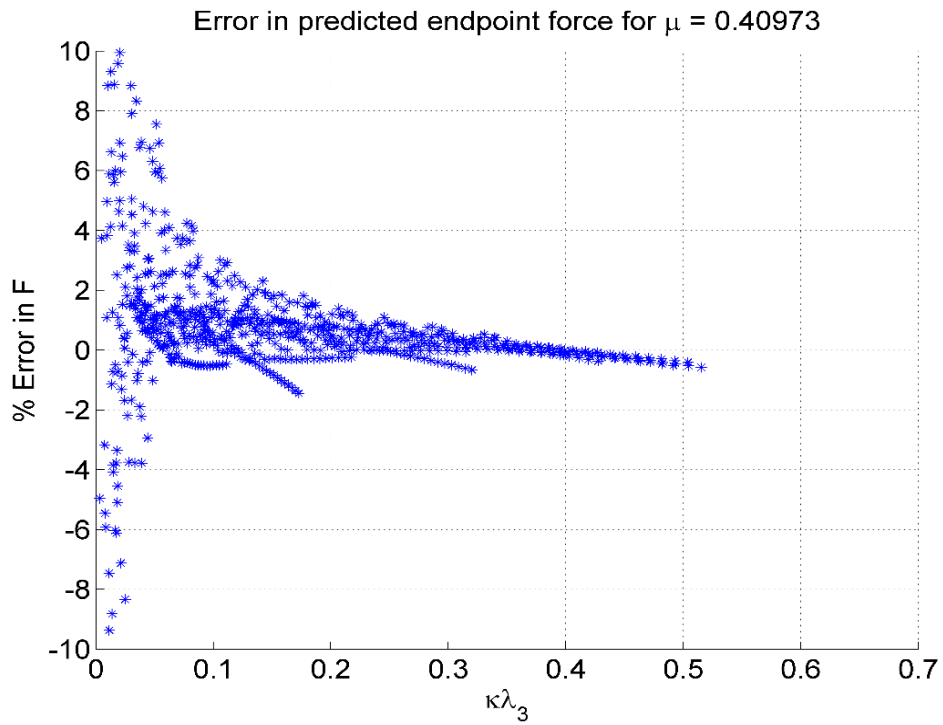


Figure 44 The % relative error in predicted F values.

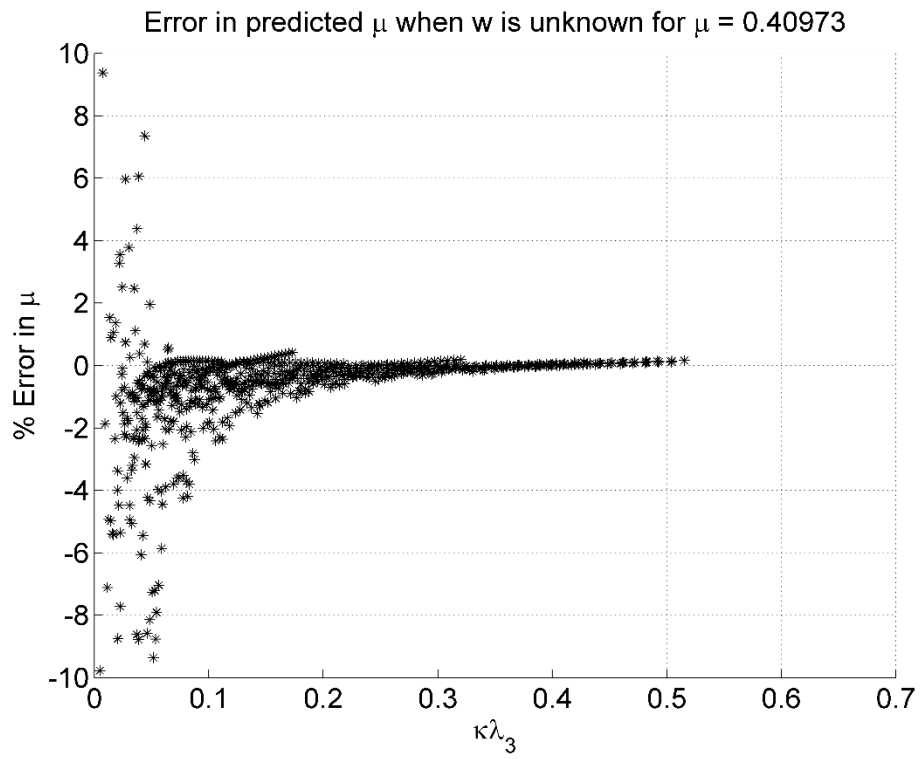


Figure 45 The % relative error in μ when w is the unknown.

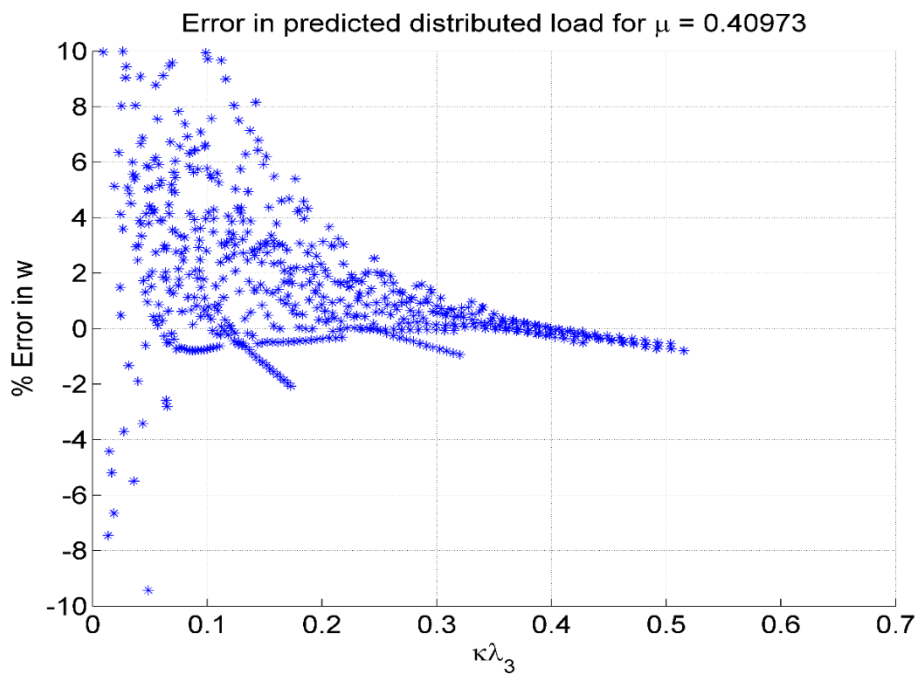


Figure 46 The % relative error in predicted w values.

CHAPTER 7: CONCLUSION

The Pseudo-Rigid-Body Model has been previously used to simplify the analysis of geometrically nonlinear beam characteristics by replacing flexible members with rigid links joined at a characteristic pivot. This method employed the closed form solution provided by elliptic integrals to develop the recommended PRBM parameters. This paper extends the prior work by developing a PRBM to describe combined loading situations containing unidirectional end force and uniformly distributed load contributions. This is done without the aid of an exact solution, but rather employs numerical integration and the analysis of relevant contours of information contained within data structures. These data structures were created by adhering to a one-to-one positional correspondence between defined quantities and the phase space coordinates of the trajectories used to populate the phase space. The resulting load-deflection relationships and parameterized kinematics provide geometric insight into the effects of unidirectionally applying distributed loads onto beams already experiencing an end force. This analysis can now be employed to aid in the design of compliant mechanisms expected to interact with external fields, or otherwise experience inertial reference frames.

7.1 Recommendations for Use

It is recognized by the author that there are a multitude of equations in this work. Most of them are not usable in a dimensional sense, but rather were used to relate the governing differential equations to justifiably recommendable and approximate relationships. Consequently, comparatively few are intended for use by designers. Table (1) lists the equations that can be used

in the context of compliant mechanism design, or, indeed, in the design of any flexible member expected to undergo large deflections.

Table 1 Equations recommended for use by designers

Intended Use	Recommended Equations	Determination of θ	Check value of θ_{max}
Finding γ and $c\theta$	Equations (56) & (59), respectively	Use arctan function	Equation (57)
Finding $K\theta$ (subsequently F , w , E , or I)	Equations (84)-(88) to determine $K\theta$ and the correction factors (making sure to check the value of ϕ relative to 100°) Equations (91) & (106)-(108) to determine the form of the equation used to determine μ and subsequently solve for F , w , E , or I (hence: Equation (105) for finding F , Equation (104) for finding w , and Equation (109) for finding E or I)	Use arcsin function	Equations (89) & (90)

It is worth noting that there is a column devoted to how one goes about finding the value of θ . Even if both the horizontal and vertical coordinates of the beam tip are known, it is still recommended to use only the vertical coordinate and an arcsin function to find, or express, θ when determining $K\theta$, as this was how the PRBM equations were formulated. As the recommended values of γ are not the precise values of γ , the resulting use of them in determining or expressing θ , will produce different results depending on what functions of γ are being used, the arcsin or arctan. Hence, θ will deviate from its precise value in different ways, depending on what function of the imprecise γ is being used. It is worth noting that this trait was present in the original

formulation of the force only PRBM developed by Howell et al [6] and has thus been something which requires attention during its use.

7.2 Future Work

The PRBM developed herein may serve as a conceptual springboard by which future engineers may investigate more complex loading situations. For instance, if the force and the distributed load act in different directions, the phase space will change, but the distributed contribution factor may still play an integral part in reformulating the governing differential equations, as, at least in the context of this dissertation, it is merely a magnitude argument and not a directional one. Admittedly, this was the easiest case to solve, as the difference between the load angles remained fixed at zero.

The next easiest case to solve may be a loading condition where the force angle and the distributed load angle are different, but still planar and of a very characteristic nature. It may be possible to continue to use the contour method in a single quadrant of the phase plane by restricting the difference between the load angles to the following criteria:

$$\cos(|angle_F - angle_w|) > 0 \quad (110)$$

This would allow for the absence of buckling, assuming, of course, that both action vectors had either positive or negative vertical components.

If the criteria found in equation (110) is not found to be true, then a location of, at least a local, maximum moment, and therefore curvature, would occur at a location away from the wall. This would correspond to the intrusion of trajectories from one quadrant of the phase portrait into another. As there would be two inflection points, where the curvature goes to zero, located at the

beam tip and between the points of maximum curvature, it appears that the use of three quadrants of the phase portrait would be required. The contour method may again be employed, but it appears likely that a complex combination of separate PRBM's would be required to explain the behavior.

Another direction this research could take is the extension of the loading conditions examined herein to a non-planar, or spatial, context. A 3D PRBM, incorporating the effects of distributed loading, may be possible to construct.

REFERENCES

- [1] Erdman, A.G., and Sandor, G. N., 1997, Mechanism Design: Analysis and Synthesis, Vol. 1, 3rd Ed., Prentice Hall, Upper Saddle River, NJ.
- [2] Shigley, J. E., and Uicker, J. J., 1995, Theory of Machines and Mechanisms, 2nd Ed., McGraw-Hill, New York.
- [3] Howell, L. (2001). Compliant mechanisms. New York, NY: John Wiley & Sons, INC.
- [4] Salamon, B. A., 1989, "Mechanical Advantage Aspects in Compliant Mechanisms Design," M.S. Thesis, Purdue University.
- [5] Motesinger, R. N., 1964, "Flexural Devices in Measurement Systems," Chapter 11 in Measurement Engineering, by P. K. Stein, Stein Engineering Services, Phoenix, AZ.
- [6] Tuttle, S. B., 1967, "Semifixed Flexural Mechanisms," Chapter 8 in Mechanisms for Engineering Design, Wiley, New York.
- [7] Her, I., 1986, "Methodology for Compliant Mechanisms Design," Ph.D. dissertation, Purdue University, West Lafayette, IN.
- [8] Eijk, J. V., 1985, "On the Design of Plate-Spring Mechanisms," Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands.
- [9] Crane, N.B., Howell L.L., and Weight, B. L., 2000, "Design and Testing of a Compliant Floating-Opposing Arm (FOA) Centrifugal Clutch," Proceedings of 8th International Power Transmission and Gearing Conference, 2000 ASME Design Engineering Technical Conferences DETC2000/PTG-14451.
- [10] Roach, G.M., and Howell, L. L., 2000, "Compliant Overrunning clutch with centrifugal Throw out," U.S. patent 6,148,979, Nov 21.
- [11] Roach, G.M., Lyon, S. M., and Howell, L. L., 1998, "A Compliant, Over-running Ratchet and Pawl Clutch with Centrifugal Throw-out," Proceedings of the 1998 ASME Design Engineering Technical Conferences, DETC98/MECH-5819.
- [12] Nielson, A.J. and Howell, L.L., 1998, "Compliant Pantographs via the Pseudo-Rigid-Body Model," Proceedings of the 1998 ASME Design Engineering Technical Conferences, DETC98/MECH-5930.

- [13] Paros, J. M., and Weisbord, L., 1965, "How to Design Flexural Hinges," *Machine Design*, Nov. 25, pp. 151-156.
- [14] Chow, W. W., 1981, "Hinges and Straps," Chapter 12 in *Plastics Products Design Handbook*, Part B, E. Miller, ed., Marcel Dekker, New York.
- [15] Derderian, J. M., 1996, "The Pseudo-Rigid-Body Model Concept and Its Application to Micro Compliant Mechanisms," M.S. thesis, Brigham Young University, Provo, UT.
- [16] Jensen, B. D., Howell, L. L., Gunyan, D. B., and Salmon, L. G., 1997, "The Design and Analysis of Compliant MEMS Using the Pseudo-Rigid-Body Model," *Microelectromechanical Systems (MEMS)*, at the 1997 ASME International Mechanical Engineering Congress and Exposition, DSC-Vol. 62, pp. 119-126.
- [17] Jensen, B. D., Howell, L. L., and Salmon, L. G., 1999, "Design of Two-Link, In-Plane, Bistable Compliant Micro-Mechanisms," *Journal of Mechanical Design*, Trans. ASME, Vol. 121, No. 3, pp. 416-423.
- [18] Ananthasuresh, G. K., and Kota, S., 1995, "Designing Compliant Mechanisms," *Mechanical Engineering*, Vol. 117, No. 11, pp. 93-96.
- [19] Ananthasuresh, G. K., Kota, S., and Gianchandani, Y., 1994, "A Methodical Approach to the Design of Compliant Mechanisms," *Solid-State Sensor and Actuator Workshop*, Hilton Head Island, SC, pp. 189-192.
- [20] Larsen, V. D., Sigmund, O., and Bouwstra, S., 1996, "Design and Fabrication of Compliant Micromechanisms and Structures with Negative Poisson's Ratio," *Proceedings of the 9th Annual International Workshop on Micro Electro Mechanical Systems*, San Diego, CA, Feb. 11-15, pp. 365-371.
- [21] Baker, M. S., Lyon, S. M., and Howell, L. L., 2000, "A Linear Displacement Bistable Micromechanism," *Proceedings of the 26th Biennial Mechanisms and Robotics Conference*, 2000 ASME Design Engineering Technical Conferences, DETC2000/MECH-141117.
- [22] Clements, D., 2000, "Implementing Compliant Mechanisms in Micro-Electro-Mechanical Systems (MEMS)," M.S. thesis, Brigham Young University, Provo, UT.
- [23] Parkinson, M. B., Jensen, B. D., and Roach, G. M., 2000, "Optimization Based Design of a Fully-Compliant Bistable Micromechanism," *Proceedings of the 26th Biennial Mechanisms and Robotics Conference*, DETC2000/MECH-14119.
- [24] Salmon, L. G., Gunyan, D. B., Derderian, J. M., Opdahl, P. G., and Howell, L. L., 1996, "Use of the Pseudo-Rigid-Body Model to Simplify the Description of Compliant Micro-Mechanisms," *1996 IEEE Solid-State and Actuator Workshop*, Hilton Head Island, SC, pp. 136-139.

- [25] Hsu, T. R., 2008, “Microsystems Fabrication Processes”, Chapter 8 in MEMS and Microsystems: Design, Manufacture, and Nanoscale Engineering, 2nd Ed., 2008, John Wiley & Sons, Inc., Hoboken, New Jersey.
- [26] Hoetmer, Karin, Herder, Just L and Kim, Charles. “A Building Block Approach for the Design of Statically Balanced Compliant Mechanisms”. International Design Engineering Technical Conference San Diego, California, USA, 2009. Vols. DETC2009 87451.
- [27] Opdahl, P. G., 1996, “Modelling and Analysis of Compliant Bi-stable Mechanisms Using the Pseudo-Rigid-Body Model,” M.S. thesis, Brigham Young University, Provo, UT.
- [28] Askeland, D. R., Fulay, P. P., and Wright, W. J., 2011, “Mechanical Properties: Part Two”, Chapter 7 in The Science and Engineering of Materials, 6th Ed., 2011, Cengage Learning, Stamford, CT.
- [29] Isaac Todhunter. A History of the Theory of Elasticity and the Strength of Materials, vol. 1. Cambridge University Press, 1886.
- [30] C. Truesdell. The Rational Mechanics of Flexible or Elastic Bodies: 1638-1788. Leonard Euler, Opera Omnia. Birkhauser, 1960.
- [31] I. J. Schoenberg. Selected Papers, vol. 1. Birkhauser, Boston, 1988. Edited by Carl de Boor.
- [32] Daniel Bernoulli. The 26th letter to Euler. In Correspondence Mathematique et Physique, vol. 2. P. H. Fuss, October 1742.
- [33] R. Sridharan. Physics to mathematics: from lintearia to lemniscate – I. Resonnace, pages 21-29, April 2004.
- [34] Louis Saaschutz. Der belastete Stab unter Einwirkung einer seitlichen Kraft. B. G. Teubner, Leipzig, 1880.
- [35] Love, A. E. H. (1927). A treatise on the mathematical theory of elasticity (4th ed.) Cambridge, Cambridge Univ. Press.
- [36] Southwell, R.V. (1941). An introduction to the theory of elasticity for engineers and physicists (2nd ed.) Oxford Univ. Press.
- [37] Timoshenko, S. P. and Gere, J. M. (1961). Theory of elastic stability (2nd ed.) New York, McGraw-Hill.
- [38] Bisshopp, K. E. and Drucker, D. C. (1945). Large Deflection of Cantilever Beams. Quarterly of Applied Mathematics, Vol. 3, No. 3, pp. 272-275.
- [39] Frisch-Fay, R., 1962, Flexible Bars, Butterworth, Washington, D.C.

- [40] Howell, L. L., and Midha, A., 1995, "Parametric Deflection Approximations for End-Loaded, Large-Deflection Beams in Compliant Mechanisms", *Journal of Mechanical Design*, Trans. ASME, Vol. 117, No. 1, pp. 156-165.
- [41] Burns, R. H., and Crossley, F. R. E., 1968, "Kinetostatic Synthesis of Flexible Link Mechanisms," ASME Paper No. 68-Mech-36.
- [42] Norton, T. W., 1991, "On the Nomenclature and Classification, and Mobility of Compliant Mechanisms," M.S. Thesis, Purdue University.
- [43] Howell, L. L., Midha, A., and Norton, T. W., 1996, "Evaluation of Equivalent Spring Stiffness for Use in a Pseudo-Rigid-Body Model of Large-Deflection Compliant Mechanisms," *Journal of Mechanical Design*, Trans. ASME, Vol. 118, No.1, pp. 126-131.
- [44] Falk, S. (1974). "Buckling of Straight Bars Subject to Their Own Weight," *Ing.-Arch.*, Vol. 43, pp. 110-7.
- [45] Holden, J. T. (1972). "On the Finite Deflections of Thin Beams," *Jour. Solid Struct.*, Vol. 8, pp. 1051-5.
- [46] Mitchel, T. P. (1959). "The Nonlinear Bending of Thin Rods," *Trans. ASME, Jour. Appl. Mech.*, pp. 40-3
- [47] Frisch-Fay, R. (1961a). "The Analysis of a Vertical and Horizontal Cantilever Under Uniformly Distributed Load," *Jour. Franklin Inst.*, Vol. 271, pp. 192-9.
- [48] Su, H., 2009. "A Pseudo-rigid-body 3R model for determining large-deflection of cantilever beams subject to tip-loads". *J. Mechanisms Robotics* 1 (2): 021008.
- [49] Pauly, J. and Midha, A., "Improved Pseudo-Rigid-Body Model Parameter Values for End-Force-Loaded Compliant Beams," ASME DETC2004-57580, pp. 1513-17.
- [50] Lusk, C. P., (2011). "Quantifying Uncertainty for Planar Pseudo-Rigid-Body Models", ASME DETC2011-47456, pp. 67-73.
- [51] Strogatz, S. H., *Nonlinear Dynamics and Chaos: With Applications To Physics, Biology, Chemistry and Engineering (Studies in Nonlinearity)*, Westview Press, 1994.
- [52] Kirchhoff G (1859) "Über das Gleichgewicht und die Bewegung eines unendlich dunnen elastischen Stabes," *Journal für die reine und angewandte Mathematik* 56:285-316.

APPENDIX A: NOMENCLATURE

ρ	Radius of curvature
x	Horizontal coordinate of any curve
y	Vertical coordinate of any curve
κ	Curvature
M	Moment
E	Young's Modulus
I	Second moment of area
a	Horizontal coordinate of beam tip
b	Vertical coordinate of beam tip
l	Length of the beam
L	Length of the beam
γ	Characteristic radius factor
Θ	PRBM angle
ϕ	Load angle from the frame of reference of the beam base neutral axis
ϕ_0	Load angle from the frame of reference of the beam tip neutral axis
δ_e	Length of the deflection vector of the beam tip
c_θ	Parametric angle coefficient
θ_0	Beam tip angular displacement with respect to the base frame of reference
θ	Beam tip angular displacement with respect to an intermediate beam location

$(\alpha^2)_t$	Transverse load index for the force contribution
F_t	Transverse force located at the beam tip
F	Force located at the beam tip
w	Evenly distributed load
K_θ	Stiffness coefficient
K	Torsional spring stiffness
T	Torque
$\Theta_{max}(\gamma)$	Angular limit of the PRBM when using γ
$\Theta_{max}(K_\theta)$	Angular limit of the PRBM when using K_θ
s	Arc length of the beam (synonymous with l)
T_x	Tip frame horizontal axis
T_y	Tip frame vertical axis
B_x	Base frame horizontal axis
B_y	Base frame vertical axis
λ_1	Characteristic length scale associated with force
λ_2	Characteristic length scale associated with distributed loads
λ_3	Characteristic length scale associated with combined loading
μ	Distributed contribution factor
\tilde{s}	Non-dimensional arc length
$\tilde{\kappa}$	Non-dimensional curvature
$\chi(\phi, \mu)$	PRBM fitted equation for γ
$c\theta(\phi, \mu)$	PRBM fitted equation for $c\theta$
$K\theta(\phi, \mu)$	PRBM fitted equation for $K\theta$

$K_{\Theta,init}(\phi, \mu)$	Fitted equation for strain energy K_{Θ} root values (not to be used by designers)
${}^T \tilde{y}$	Non-dimensional tip frame vertical coordinate
${}^T \tilde{x}$	Non-dimensional tip frame horizontal coordinate
ijk	Subscripts denoting row, column, and stack coordinates of information within the data structure
\tilde{a}	Non-dimensional horizontal coordinate of the beam tip
\tilde{b}	Non-dimensional vertical coordinate of the beam tip
C	γ contour matrix containing γ and Θ values obtained at level curves of ϕ
q	Subscript denoting the columns of C
γ_{vec}	Vector of γ values used in error surface generation about the contour C
Θ_{vec}	Vector of Θ values used in error surface generation about the contour C
$relE_{q,vec}$	The relative error vector of values generated at each point within the contour C
$leftE$	Left error contour matrix of values
$rightE$	Right error contour matrix of values
$\Theta_{max,\gamma}(\phi, \mu)$	Equation of the angular limit of the PRBM associated with γ
$(\alpha^3)_t$	Transverse load index associated with distributed loads
w_t	Transverse distributed load
V	Strain energy
W	Work
\tilde{V}	Non-dimensional strain energy
${}^{SE} K_{\Theta}$	The K_{Θ} values predicted by the non-dimensional strain energy
$\Theta_{max,K}$	The angular limit of the PRBM associated with K_{Θ}
χ_F	The correction factor for the force transverse load index

χ_D	The correction factor for the distributed transverse load index
$\varepsilon_{\tilde{s}}$	The relative error in \tilde{s}
ε_s	The relative error in s
ε_F	The relative error in F
ε_w	The relative error in w
$\varepsilon_{E,I}$	The relative error in E or I (they are the same error value)

APPENDIX B: MATLAB CODE

B.1 Function File

```
function dg = beam_all_cases_wSE(s,phi)
dg=zeros(8,1);
dg(1) = phi(2);
dg(2) = ((phi(4))^2 + (1-phi(4))^3 * phi(3))*sin(phi(1));
dg(3) = 1;
dg(4)=0;
dg(5)=0;
dg(6)=-cos(phi(1)-phi(5));
dg(7)=-sin(phi(1)-phi(5));
dg(8)=phi(2)^2;
```

B.2 Elastica Example

```
clc
clear all
close all
clf

%% Inputs and initializations

h=0;
g=0;

mustart=1;
mustep=1;
mustop=1;

phistart=75;
phistep=1;
phistop=75;

sstart=0;
sstep=0.01;
sstop=19.85;

options=odeset('MaxStep',sstep);
```

```

%% Code

for mu=mustart:mustep:mustop
    h=h+1;
    g=0;

    for phi=phistart:phistep:phistop
        test=phi*pi/180;
        [sgrav,ggrav]=ode45('beam_all_cases_wSE',(sstart:sstep:ssstop),...
            [test 0 0 mu test 0 0 0],options);
        rota=abs(atan(ggrav(end,6)/ggrav(end,7)))+3*pi/2;

        for k=1:length(sgrav)
            tempval=[cos(rota) sin(rota);-sin(rota) cos(rota)]*...
                [ggrav(k,6);ggrav(k,7)];
            ggrav(k,6)=-tempval(1);
            ggrav(k,7)=-tempval(2);
        end
    end
end

subplot(2,1,1)
hold on
plot(ggrav(:,6),ggrav(:,7),'k')
plot(-7:1:1,zeros(size(-7:1:1)), 'k-.' )
hold off
set(gca,'XTickLabel','')
set(gca,'YTickLabel','')
xlim([-7 1])
ylabel('Elastica')

subplot(2,1,2)
hold on
plot(1:length(sgrav),ggrav(:,2),'k')
plot(-400:100:2400,zeros(length(-400:100:2400)), 'k:')
hold off
xlim([-390 2305])
set(gca,'XTickLabel','')
set(gca,'YTickLabel','')
ylabel('\kappa')

B.3 Surface of  $\mu$ 

clc
clf

```

```

clear all
close all

EI=0.109504;
w0=3;
F0=3;

minlim=0;
incr=0.0025;
maxlim=2;

w=minlim:incr:maxlim;
F=minlim:incr:maxlim;

C_w_0=((EI/w).^(2/3))./((EI/F0)+(2*(EI/F0)^(1/2)*(EI/w).^(1/3))+...
(EI/w).^(2/3));
D_w_0=((EI/F0)^(3/2))./((EI/F0)^(3/2)+(3*(EI/F0)*(EI/w).^(1/3))+...
(3*(EI/F0)^(1/2)*(EI/w).^(2/3))+(EI/w));
C_F_0=((EI/w0)^(2/3))./((EI/F)+(2*(EI/F).^(1/2)*(EI/w0)^(1/3))+...
(EI/w0)^(2/3));
D_F_0=((EI/F).^(3/2))./((EI/F).^(3/2)+(3*(EI/F)*(EI/w0)^(1/3))+...
(3*(EI/F).^(1/2)*(EI/w0)^(2/3))+(EI/w0));
Factor_w_0=((EI/w).^(1/3)*(EI/F0)^(1/2))./((EI/w).^(1/3)+(EI/F0)^(1/2));
Factor_F_0=((EI/w0)^(1/3)*(EI/F).^(1/2))./((EI/w0)^(1/3)+(EI/F).^(1/2));

figure(1)
hold on
plot(w,C_w_0,'k-')
plot(w,D_w_0,'k-')
plot(F,C_F_0,'r-')
plot(F,D_F_0,'r-')
grid on
xlabel('Changing w or F')
ylabel('Values of Nondimensionalization Factors')
title('Variation of Nondimensionalization Factors with changing w or F')
legend('C as w goes to zero','D as w goes to zero','C as F goes to zero'...
,'D as F goes to zero')
hold off

figure(2)
hold on
plot(w,Factor_w_0,'k-')
plot(F,Factor_F_0,'b-')
grid on
xlabel('Changing w or F')
ylabel('Values of Primary Nondimensionalization Factor')

```



```

title('Variation of Primary Nondimensionalization Factor with...
changing w or F')
legend('\lambda_3 as w goes to zero', '\lambda_3 as F goes to zero')
hold off

```

```

g=0;
Factor=zeros(length(w),length(w));
w_mat=zeros(length(w),length(w));
f_mat=zeros(length(w),length(w));
mu=zeros(length(w),length(w));

```

```

for f=minlim:incr:maxlim
    g=g+1;
    Factor(:,g)=((EI/(w+0.001)).^(1/3)*(EI/(f+0.001))^(1/2))./...
        ((EI/(w+0.001)).^(1/3)+(EI/(f+0.001))^(1/2));
    mu(:,g)=((EI/w).^(1/3))./((EI/w).^(1/3)+(EI/f)^(1/2));
    w_mat(:,g)=w;
    f_mat(1:length(w),g)=f;
    lam1_mat=(EI/f_mat).^(1/2);
    lam2_mat=(EI/w_mat).^(1/3);
end

```

```

figure(3)
hold on
surf(f_mat,w_mat,Factor)
shading flat
grid on
xlabel('f')
ylabel('w')
zlabel('\lambda_3')
title('Surface of Primary Nondimensionalization Factor (\lambda_3) with...
changing w and F')
hold off

```

```

figure(4)
hold on
surf(f_mat,w_mat,mu)
shading flat
grid on
xlabel('f')
ylabel('w')
zlabel('\mu')
title('Surface of \mu with changing w and F')
hold off

```

```

figure(5)

```

```

hold on
surf(lam1_mat,lam2_mat,Factor)
shading flat
grid on
xlabel('\lambda_1')
ylabel('\lambda_2')
zlabel('\lambda_3')
title('Surface of \lambda_3 with changing \lambda_1 and \lambda_2')
hold off

```

```

figure(6)
hold on
surf(lam1_mat,lam2_mat,mu)
shading flat
grid on
xlabel('\lambda_1')
ylabel('\lambda_2')
zlabel('\mu')
title('Surface of \mu with changing \lambda_1 and \lambda_2')
hold off

```

```

figure(7)
hold on
surf(lam1_mat,lam2_mat,mu)
surf(lam1_mat,lam2_mat,Factor)
shading flat
grid on
xlabel('\lambda_1')
ylabel('\lambda_2')
zlabel('\mu and \lambda_3')
title('Surface of \mu and \lambda_3 with changing \lambda_1 and \lambda_2')
hold off

```

B.4 Distributed Phase Trajectory

```

clc
clf
clear all
close all

test=pi/4;
options=odeset('MaxStep',0.01);
figure(1)

for i=0.01:0.2:pi-0.01
    [s,g]=ode45('beam_nograv',(0.01:0.01:30),[i 0 0],options);

```

```

hold on
plot3(g(:,1)*180/pi,g(:,2),g(:,3),'k-')
hold off
end

[sgrav,ggrav]=ode45('beam_all_cases_wSE',(0.01:0.01:50.2),...
[test 0 0 0 test 0 0 0],options);
hold on
plot3(ggrav(:,1)*180/pi,ggrav(:,2),ggrav(:,3),'b-','Linewidth',2)
hold off
grid on
xlabel('\phi')
ylabel('\kappa\lambda_3')
zlabel('s/\lambda_3')

```

B.5 ANSYS Comparisons

```

clc
clear all
close all
clf

%% Constants (combined loading)

E=200*10^9;
I=5.4752*10^-13;
F=0.24;
w=1.062355767;
lambda1=(E*I/F)^(1/2);
lambda2=(E*I/w)^(1/3);
lambda3=(lambda1*lambda2)/(lambda1+lambda2);
magratio=lambda2/(lambda1+lambda2);

%% Ansys Data (combined loading)
phi_deg=(165:-15:15)*(pi/180);
dataloop_max=length(phi_deg);

%data entry
% for hide=1:1
%end data entry

%% Data Manipulation

options=odeset('MaxStep',0.01);

for a=1:dataloop_max

```

```

eval(['phi_meas' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
eval(['K_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
eval(['S_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
eval(['xvec_length(a)=length(xdisp' num2str(a) ')'-1;'])
eval(['Mvec_length(a)=length(Mvec' num2str(a) ');'])

```

```

for j=2:xvec_length(a)
    eval(['tempx=(svec(j+1) + xdisp' num2str(a) ...
        '(j+1)) - (svec(j-1) + xdisp' num2str(a) '(j-1));'])
    eval(['tempy=ydisp' num2str(a) '(j+1)-ydisp' num2str(a) '(j-1);'])
    tempmag=sqrt(tempx^2+tempy^2);
    tempalpha=abs(acos(tempx/tempmag));
    eval(['phi_meas' num2str(a) '(j)=phi_deg(a)-tempalpha;'])
    eval(['S_tilde' num2str(a) '(j)=svec(length(svec)+1-j)/lambda3;'])
end

```

```

for n=1:2:Mvec_length(a)
    eval(['K_tilde' num2str(a) '((n+1)/2)=(lambda3*-'Mvec' num2str(a)...
        '(n)) / (E*I);'])
end

```

```

eval(['test(a)=phi_meas' num2str(a) '(j);'])
eval(['PM=phi_meas' num2str(a) ';'])
eval(['KT=K_tilde' num2str(a) ';'])
eval(['ST=S_tilde' num2str(a) ';'])
PM(1)=[];
KT(1)=[];
ST(1)=[];
[sgrav,ggrav]=ode45('beam_all_cases',...
    [0:0.01:(svec(70)-svec(2))/lambda3],...
    [test(a) 0 0 magratio test(a) 0 0],options);
figure(1)
hold on
plot3(ggrav(:,1)*180/pi,ggrav(:,2),ggrav(:,3),'b-','Linewidth',2)
plot3(PM*180/pi,KT,ST,'r.')
hold off
end

```

```

for i=1:dataloop_max
    [s,g]=ode45('beam_nograv',(0:0.01:20),[test(i) 0 0],options);

    for mm=1:length(s)
        if g(mm,1)>pi || g(mm,2)<0
            g(mm:length(s),:)=[];
            s(mm:length(s),:)=[];
            break
        end
    end
end

```

```

    end
end

figure(1)
hold on
plot3(g(:,1)*180/pi,g(:,2),g(:,3),'k-')
hold off
end

figure(1)
grid on
xlabel('\phi')
ylabel('\kappa\lambda_3')
zlabel('s\lambda_3')
legend('Model','Ansys Data')

%% Constants (distributed load only)

E=200*10^9;
I=5.4752*10^-13;
F=10^-100;
w=1.062355767;
lambda1=(E*I/F)^(1/2);
lambda2=(E*I/w)^(1/3);
lambda3=(lambda1*lambda2)/(lambda1+lambda2);
magratio=lambda2/(lambda1+lambda2);

%% Ansys Data (distributed load only)

phi_deg=(165:-15:15)*(pi/180);
dataloop_max=length(phi_deg);

%data entry
% for hide=1:1
%end data entry

%% Data Manipulation

options=odeset('MaxStep',0.01);

for a=1:dataloop_max
    eval(['phi_meas' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['K_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['S_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['xvec_length(a)=length(xdisp' num2str(a) ')'-1;'])
    eval(['Mvec_length(a)=length(Mvec' num2str(a) ')';'])

```

```

for j=2:xvec_length(a)
    eval(['tempx=(svec(j+1) + xdisp' num2str(a)...
        '(j+1) - (svec(j-1) + xdisp' num2str(a) '(j-1));'])
    eval(['tempy=ydisp' num2str(a) '(j+1)-ydisp' num2str(a) '(j-1);'])
    tempmag=sqrt(tempx^2+tempy^2);
    tempalpha=abs(acos(tempx/tempmag));
    eval(['phi_meas' num2str(a) '(j)=phi_deg(a)-tempalpha;'])
    eval(['S_tilde' num2str(a) '(j)=svec(length(svec)+1-j)/lambda3;'])
end

for n=1:2:Mvec_length(a)
    eval(['K_tilde' num2str(a) '((n+1)/2)=(lambda3*-'Mvec' num2str(a)...
        '(n) / (E*I);'])
end

eval(['test(a)=phi_meas' num2str(a) '(j);'])
eval(['PM=phi_meas' num2str(a) ';'])
eval(['KT=K_tilde' num2str(a) ';'])
eval(['ST=S_tilde' num2str(a) ';'])
PM(1)=[];
KT(1)=[];
ST(1)=[];
[sgrav,ggrav]=ode45('beam_all_cases',...
    [0:0.01:(svec(70)-svec(2))/lambda3],...
    [test(a) 0 0 magratio test(a) 0 0],options);
figure(2)
hold on
plot3(ggrav(:,1)*180/pi,ggrav(:,2),ggrav(:,3),'b-','Linewidth',2)
plot3(PM*180/pi,KT,ST,'r.')
hold off
end

for i=1:dataloop_max
    [s,g]=ode45('beam_nograv',(0:0.01:20),[test(i) 0 0],options);

    for mm=1:length(s)
        if g(mm,1)>pi || g(mm,2)<0
            g(mm:length(s),:)=[];
            s(mm:length(s),:)=[];
            break
        end
    end
end

figure(2)
hold on

```

```

    plot3(g(:,1)*180/pi,g(:,2),g(:,3),'k-')
    hold off
end

figure(2)
grid on
xlabel('\phi')
ylabel('\kappa\lambda_3')
zlabel('s/\lambda_3')
legend('Model','Ansys Data')

%% Constants (force load only)

E=200*10^9;
I=5.4752*10^-13;
F=0.24;
w=1e-100;
lambda1=(E*I/F)^(1/2);
lambda2=(E*I/w)^(1/3);
lambda3=(lambda1*lambda2)/(lambda1+lambda2);
magratio=lambda2/(lambda1+lambda2);

%% Ansys Data (force load only)

phi_deg=(165:-15:15)*(pi/180);
dataloop_max=length(phi_deg);

%data entry
% for hide=1:1
%end data entry

%% Data Manipulation

options=odeset('MaxStep',0.01);

for a=1:dataloop_max
    eval(['phi_meas' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['K_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['S_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['xvec_length(a)=length(xdisp' num2str(a) ')'-1;'])
    eval(['Mvec_length(a)=length(Mvec' num2str(a) ');'])

    for j=2:xvec_length(a)
        eval(['tempx=(svec(j+1) + xdisp' num2str(a)...
            '(j+1)) - (svec(j-1) + xdisp' num2str(a) '(j-1));'])
        eval(['tempy=ydisp' num2str(a) '(j+1)-ydisp' num2str(a) '(j-1);'])
    end
end

```

```

    tempmag=sqrt(tempx^2+tempy^2);
    tempalpha=abs(acos(tempx/tempmag));
    eval(['phi_meas' num2str(a) '(j)=phi_deg(a)-tempalpha;'])
    eval(['S_tilde' num2str(a) '(j)=svec(length(svec)+1-j)/lambda3;'])
end

for n=1:2:Mvec_length(a)
    eval(['K_tilde' num2str(a) '((n+1)/2)=(lambda3*-Mvec' num2str(a)...
        '(n)) / (E*I);'])
end

eval(['test(a)=phi_meas' num2str(a) '(j);'])
eval(['PM=phi_meas' num2str(a) ';'])
eval(['KT=K_tilde' num2str(a) ';'])
eval(['ST=S_tilde' num2str(a) ';'])
PM(1)=[];
KT(1)=[];
ST(1)=[];
[sgrav,ggrav]=ode45('beam_all_cases',...
    (0:0.01:(svec(70)-svec(2))/lambda3),...
    [test(a) 0 0 magratio test(a) 0 0],options);
figure(3)
hold on
plot3(ggrav(:,1)*180/pi,ggrav(:,2),ggrav(:,3),'b-','Linewidth',2)
plot3(PM*180/pi,KT,ST,'r.')
hold off
end

for i=1:dataloop_max
    [s,g]=ode45('beam_nograv',(0:0.01:20),[test(i) 0 0],options);

    for mm=1:length(s)
        if g(mm,1)>pi || g(mm,2)<0
            g(mm:length(s),:)=[];
            s(mm:length(s),:)=[];
            break
        end
    end
end

figure(3)
hold on
plot3(g(:,1)*180/pi,g(:,2),g(:,3),'k-')
hold off
end

figure(3)

```



```

grid on
xlabel('\phi')
ylabel('\kappa\lambda_3')
zlabel('s/\lambda_3')
legend('Model','Ansys Data')

%% Constants (combined loading)

E=200*10^9;
I=5.4752*10^-13;
F=1.2;
w=1.062355767;
lambda1=(E*I/F)^(1/2);
lambda2=(E*I/w)^(1/3);
lambda3=(lambda1*lambda2)/(lambda1+lambda2);
magratio=lambda2/(lambda1+lambda2);

%% Ansys Data (combined loading)
phi_deg=(165:-15:15)*(pi/180);
dataloop_max=length(phi_deg);

%data entry
% for hide=1:1
%end data entry

%% Data Manipulation

options=odeset('MaxStep',0.01);

for a=1:dataloop_max
    eval(['phi_meas' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['K_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['S_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
    eval(['xvec_length(a)=length(xdisp' num2str(a) ')'-1;'])
    eval(['Mvec_length(a)=length(Mvec' num2str(a) ');'])

    for j=2:xvec_length(a)
        eval(['tempx=(svec(j+1) + xdisp' num2str(a) ...
            '(j+1)) - (svec(j-1) + xdisp' num2str(a) '(j-1));'])
        eval(['tempy=ydisp' num2str(a) '(j+1)-ydisp' num2str(a) '(j-1);'])
        tempmag=sqrt(tempx^2+tempy^2);
        tempalpha=abs(acos(tempx/tempmag));
        eval(['phi_meas' num2str(a) '(j)=phi_deg(a)-tempalpha;'])
        eval(['S_tilde' num2str(a) '(j)=svec(length(svec)+1-j)/lambda3;'])
    end
end

```

```

for n=1:2:Mvec_length(a)
    eval(['K_tilde' num2str(a) '((n+1)/2)=(lambda3*-Mvec' num2str(a)...
        '(n)) / (E*I);'])
end

eval(['test(a)=phi_meas' num2str(a) '(j);'])
eval(['PM=phi_meas' num2str(a) ';'])
eval(['KT=K_tilde' num2str(a) ';'])
eval(['ST=S_tilde' num2str(a) ';'])
PM(1)=[];
KT(1)=[];
ST(1)=[];
[sgrav,ggrav]=ode45('beam_all_cases',...
    (0:0.01:(svec(70)-svec(2))/lambda3),...
    [test(a) 0 0 magratio test(a) 0 0],options);
figure(4)
hold on
plot3(ggrav(:,1)*180/pi,ggrav(:,2),ggrav(:,3),'b-','Linewidth',2)
plot3(PM*180/pi,KT,ST,'r.')
hold off
end

for i=1:dataloop_max
    [s,g]=ode45('beam_nograv',(0:0.01:20),[test(i) 0 0],options);

    for mm=1:length(s)
        if g(mm,1)>pi || g(mm,2)<0
            g(mm:length(s),:)=[];
            s(mm:length(s),:)=[];
            break
        end
    end
end

figure(4)
hold on
plot3(g(:,1)*180/pi,g(:,2),g(:,3),'k-')
hold off
end

figure(4)
grid on
xlabel('\phi')
ylabel('\kappa\lambda_3')
zlabel('s\lambda_3')
legend('Model','Ansys Data')

```

B.6 Phase Space of γ

```
clc
clear all
close all
clf

%% Inputs and initializations

h=0;
g=0;
HH=0;

mustart=0;
mustep=1;
mustop=1;

phistart=2;
phistep=1;
phistop=178;

sstart=0;
sstep=0.01;
sstop=7;

mat_phi=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_K=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_s=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_g=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_C_theta=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_THETA=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_K_theta=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
muvec=zeros(1,abs((mustop-mustart)/mustep)+1);

options=odeset('MaxStep',sstep);

%% Code

for mu=mustart:mustep:mustop
```

```

h=h+1;
g=0;

for phi=phistart:phistep:phistop
    test=phi*pi/180;
    [sgrav,ggrav]=ode45('beam_all_cases',(sstart:sstep:sstop),...
        [test 0 0 mu test 0 0],options);

    for mm=1:length(sgrav)
        if ggrav(mm,1)>pi || ggrav(mm,2)<0
            ggrav(mm:length(sgrav),:)=[];
            sgrav(mm:length(sgrav),:)=[];
            break
        end
    end

    for nn=1:length(sgrav)
        if ggrav(nn,3)>0.02
            ggrav(1:nn,:)=[];
            sgrav(1:nn,:)=[];
            break
        end
    end

    for k=1:length(sgrav)
        tempval=[cos((ggrav(k,1)-ggrav(k,5))) ...
            sin((ggrav(k,1)-ggrav(k,5)));...
            -sin((ggrav(k,1)-ggrav(k,5))) ...
            cos((ggrav(k,1)-ggrav(k,5)))]*[ggrav(k,6);ggrav(k,7)];
        ggrav(k,6)=-tempval(1);
        ggrav(k,7)=-tempval(2);
    end

    gamma=((ggrav(:,7)./sgrav).^2+(1-ggrav(:,6)./sgrav).^2)./...
        (2*(1-ggrav(:,6)./sgrav));
    THETA=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma)))));
    C_theta=abs((ggrav(:,1)-ggrav(:,5))./THETA);
    K_theta=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3/2)).*...
        sin(ggrav(:,1)-THETA)./THETA;
    g=g+1;
    mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
    mat_K(1:length(sgrav),g,h)=ggrav(:,2);
    mat_s(1:length(sgrav),g,h)=ggrav(:,3);
    mat_g(1:length(sgrav),g,h)=gamma;
    mat_C_theta(1:length(sgrav),g,h)=C_theta;
    mat_THETA(1:length(sgrav),g,h)=THETA;

```

```

        mat_K_theta(1:length(sgrav),g,h)=K_theta;
        muvec(h)=mu;
    end
end

[rownum colnum stacknum]=size(mat_phi);

for x=1:rownum
    for y=1:colnum
        for z=1:stacknum
            if mat_phi(x,y,z)==0
                mat_phi(x,y,z)=NaN;
            end
            if mat_s(x,y,z)==0
                mat_s(x,y,z)=NaN;
            end
            if mat_g(x,y,z)==0
                mat_g(x,y,z)=NaN;
            end
            if mat_C_theta(x,y,z)==0
                mat_C_theta(x,y,z)=NaN;
            end
            if mat_THETA(x,y,z)==0
                mat_THETA(x,y,z)=NaN;
            end
            if mat_K_theta(x,y,z)==0
                mat_K_theta(x,y,z)=NaN;
            end
            if mat_K(x,y,z)==0
                mat_K(x,y,z)=NaN;
            end
        end
    end
end

figure(1)

for pp=1:1
    hold on
    surf((mat_phi(:,:,1))*180/pi,(mat_K(:,:,1)),(mat_g(:,:,1)))
end

shading flat
hold off
xlabel('\phi (degrees)')
ylabel('\kappa\lambda_3')

```

```

xlabel('\gamma')

figure(2)

for pp=h:h
    hold on
    surf((mat_g(:,:,pp)),(mat_THETA(:,:,pp))*180/pi,(mat_phi(:,:,pp))*180/pi)
end

shading flat
hold off
xlabel('\gamma')
ylabel('\Theta (degrees)')
xlabel('\phi (degrees)')

philvls=10:20:170;

figure(3)

for tt=1:h
    if h>2
        subplot(2,ceil(h/2),tt)
        hold on
        [C,v]=contour3((mat_g(:,:,tt)),...
            (mat_THETA(:,:,tt))*180/pi,(mat_phi(:,:,tt))*180/pi,philvls);
        clabel(C,v)
        title(['Contour for \mu = ' num2str(muvec(tt))])
        xlabel('\gamma')
        ylabel('\Theta (degrees)')
        xlabel('\phi (degrees)')
    elseif h==1
        hold on
        [C,v]=contour3((mat_g(:,:,tt)),...
            (mat_THETA(:,:,tt))*180/pi,(mat_phi(:,:,tt))*180/pi,philvls);
        clabel(C,v)
        title(['Contour for \mu = ' num2str(muvec(tt))])
        xlabel('\gamma')
        ylabel('\Theta (degrees)')
        xlabel('\phi (degrees)')
    elseif h==2
        subplot(1,2,tt)
        hold on
        [C,v]=contour3((mat_g(:,:,tt)),...
            (mat_THETA(:,:,tt))*180/pi,(mat_phi(:,:,tt))*180/pi,philvls);
        clabel(C,v)
        title(['Contour for \mu = ' num2str(muvec(tt))])
    end
end

```

```

        xlabel('\gamma')
        ylabel('\Theta (degrees)')
        zlabel('\phi (degrees)')
    end
    xlim([0.8 0.92])
    ylim([0 180])
end

hold off

figure(4)
[C,v]=contour((mat_g(:,h)),...
    (mat_THETA(:,h))*180/pi,(mat_phi(:,h))*180/pi,[90 90]);
clabel(C,v)
i=1;

while i<length(C)
    if C(1,i)>1
        C(:,i)=[];
        continue
    end
    i=i+1;
end

title(['Contour for \mu = ' num2str(muvec(tt)) ' and \phi=90'])
xlabel('\gamma')
ylabel('\Theta (degrees)')
zlabel('\phi (degrees)')

C(:,1)=[];
[rowC,colC]=size(C);
gamvec=0.7:0.001:1;
a_C=zeros(1,colC);
b_C=zeros(1,colC);
rel_err=zeros(colC,length(gamvec));
mat_THETAvec=zeros(colC,length(gamvec));
for qq=1:colC
    a_C(qq)=1-C(1,qq)+C(1,qq)*cosd(C(2,qq));
    b_C(qq)=C(1,qq)*sind(C(2,qq));
    mat_THETAvec(qq,:)=atan2(b_C(qq),a_C(qq)-(1-gamvec));
    rel_err(qq,:)=sqrt((a_C(qq)-...
        (1-gamvec.*(1-cos(mat_THETAvec(qq,:))))).^2+(b_C(qq)-...
        gamvec.*sin(mat_THETAvec(qq,:))).^2)/sqrt((1-a_C(qq))^2+(b_C(qq))^2);
end
mat_gamvec=ones(1,colC)*gamvec;

```

```

figure(5)
surf(mat_gamvec,mat_THETAvec*180/pi,rel_err*100)
shading flat
title('Error surface for \phi=90')
xlabel('\gamma')
ylabel('\Theta')
zlabel('Absolute relative error (in %)')
xlim([min(gamvec) max(gamvec)])

errlvl=0:0.5:5;
Thetavals=0:80;
gammaval=0.8517*ones(length(Thetavals));

figure(6)
[D,v]=contour(mat_gamvec,mat_THETAvec*180/pi,rel_err*100,errlvl);
clabel(D,v)
title('Contour of error levels for \mu=1 and \phi=90')
xlabel('\gamma')
ylabel('\Theta (degrees)')
zlabel('Absolute relative error (in %)')
xlim([min(gamvec) max(gamvec)])
ylim([0 max(max(mat_THETAvec*180/pi))])
hold on
plot(gammaval,Thetavals,'k--')
plot(0.8517,64.3,'r*')
hold off

```

B.7 γ Fit

```

clc
clear all
close all
clf

```

%% Inputs and initializations

```

phi_mat=dlmread('Phi_matrix.txt');
mu_mat=dlmread('Mu_matrix.txt');
gamma_found=dlmread('Gamma_matrix.txt');
THETA_found=dlmread('THETA_matrix.txt');
K_found=dlmread('Ktheta_matrix.txt');
a=1;
b=1;
c=1;
d=1;
f=1;

```



```

g=1;
h=1;

%% Code

phi_mat=180-phi_mat;
figure(1)
surf(phi_mat,mu_mat,gamma_found)
shading flat
title('\gamma surface for changing \mu and \phi')
xlabel('\phi')
ylabel('\mu')
zlabel('\gamma')
axis([0 180 0 1 0.7 1])

figure(2)
surf(phi_mat,mu_mat,THETA_found)
shading flat
title('\Theta surface for changing \mu and \phi')
xlabel('\phi')
ylabel('\mu')
zlabel('\Theta (degrees)')

x(:,1)=(170:-0.5:20)./180;
y=0:0.02:1;

for i=1:length(y)
    xvec(a:a+length(x)-1,1)=x;
    a=a+length(x);
end

for j=1:length(y)
    yvec(b:b+length(x)-1,1)=y(j);
    b=b+length(x);
end

gamma_found=gamma_found';
[rowG,colG]=size(gamma_found);

for k=1:colG
    gamvec(c:c+rowG-1,1)=gamma_found(1:rowG,k);
    c=c+rowG;
end

THETA_found=(THETA_found')*pi/180;
[rowT,colT]=size(THETA_found);

```

```

for m=1:colT
    THETAvec(d:d+rowT-1,1)=THETA_found(1:rowT,m);
    d=d+rowT;
end

K_found=K_found';
[rowK,colK]=size(K_found);

for n=1:colK
    Kvec(g:g+rowK-1,1)=K_found(1:rowK,n);
    g=g+rowK;
end

eqn20_T=(0.3267+0.0336./(1+exp(14.78*y.^3-1.754*y.^2+0.4065*y-0.2566)));
diffT=THETA_found(1,1:colT)-eqn20_T;
diffT_eq='5.162*mu^9-164*mu^8+605.6*mu^7-934.3*mu^6+731.4*mu^5-299.6*...
mu^4+62.17*mu^3-6.786*mu^2+0.24*mu+1.929';

% ftypeT=fitype('A+B/(1+exp(C*mu^3+D*mu^2+E*mu+F))','indep',...
% {'mu'},'depen','y')
% opts=fityptions(ftypeT);
% opts.Lower=[-100 -100 -100 -100 -100 -100];
% opts.Upper=[100 100 100 100 100 100];
% opts.StartPoint=[1 1 1 1 1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fT,gofT_alt]=fit([y'],THETA_found(rowT,1:colT)',ftypeT,opts)
% figure(4)
% plot(fT,[y'],THETA_found(rowT,1:colT)')
% title('Model and data comparison for \Theta')
% xlabel('\mu')
% ylabel('\Theta')
% legend('Parametric fit','Contour data')

% % ftypeT=fitype('A*mu^5+B*mu^4+C*mu^3+D*mu^2+E*mu+F','indep',{'mu'},...
% 'depen','y')
% % opts=fityptions(ftypeT);
% % opts.Lower=[-100 -100 -100 -100 -100 -100];
% % opts.Upper=[100 100 100 100 100 100];
% % opts.StartPoint=[1 1 1 1 1 1];
% % opts.Robust='LAR';
% % opts.MaxIter=100000000;
%
% % [fT,gofT_alt]=fit([y'],diffT,ftypeT,opts)

```

```

% [fT,gofT_alt]=fit([y'],diffT,'poly9')
% figure(5)
% plot(fT,[y'],diffT)
% title('Model and data comparison for \Theta')
% xlabel('\mu')
% ylabel('\Theta')
% legend('Parametric fit','Contour data')

% ftypeT=fitype('(0.3267+0.0336/(1+exp(14.78*\mu^3-1.754*\mu^2+...
%0.4065*\mu-0.2566)))+(A*\phi^4+B*\phi^3+C*\phi^2+D*\phi+E)*(5.162*\mu^9-...
%164*\mu^8+605.6*\mu^7-934.3*\mu^6+731.4*\mu^5-299.6*\mu^4+62.17*\mu^3-...
%6.786*\mu^2+0.24*\mu+1.929)','indep',{ 'phi','mu'},'depen','z')
% opts=fitoptions(ftypeT);
% opts.Lower=[-100 -100 -100 -100 -100];
% opts.Upper=[100 100 100 100 100];
% opts.StartPoint=[1 1 1 1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fT,gofT]=fit([xvec,yvec],THETAvec,ftypeT,opts)
% figure(6)
% plot(fT,[xvec,yvec],THETAvec)
% title('Model and data comparison for \Theta_{max}')
% xlabel('\phi/pi')
% ylabel('\mu')
% zlabel('\Theta_{max}')
% legend('Parametric fit','Contour data')

THETAmax_eq=(0.3267+0.0336./(1+exp(14.78*\mu_mat.^3-1.754*...
mu_mat.^2+0.4065*\mu_mat-0.2566)))+(-0.7882*(phi_mat/180).^4+...
2.258*(phi_mat/180).^3-1.944*(phi_mat/180).^2+1.723*(phi_mat/180)...
-0.1701).*(5.162*\mu_mat.^9-164*\mu_mat.^8+605.6*\mu_mat.^7-934.3*...
mu_mat.^6+731.4*\mu_mat.^5-299.6*\mu_mat.^4+62.17*\mu_mat.^3-6.786*...
mu_mat.^2+0.24*\mu_mat+1.929);
figure(7)
hold on
surf(phi_mat,mu_mat,THETAmax_eq*180/pi)
plot3(phi_mat,mu_mat,THETA_found*180/pi,'b.','markersize',1)
hold off
shading flat
grid on
xlabel('\phi (degrees)')
ylabel('\mu')
zlabel('\Theta_max (degrees)')
legend('Parametric fit','Contour data')
x_alt(:,1)=tand((170:-0.5:20)./2);

```

```

for n=1:length(y)
    xvec_alt(f:f+length(x_alt)-1,1)=x_alt;
    f=f+length(x_alt);
end

% gstartFO=(0.7987+(0.1758/log(5.647*tphio2^3-5.774*tphio2^2+...
% 26.31*tphio2+1.1)));
% gstartDL=(0.8434+(0.1639/log(5.346*tphio2^3-5.336*tphio2^2+...
% 25.96*tphio2+1.1)));
% gdif=(0.04462+(-0.01106/log(8.296*tphio2^3-11.33*tphio2^2+...
% 26.11*tphio2+1.336)));
%
% ftypeG=fittype('(0.7987+(0.1758/log(5.647*tphio2^3-5.774*tphio2^2+...
% 26.31*tphio2+1.1)))+(A/(1+exp(B*mu^2+C*mu+D)))+E)*(0.04462+...
% (-0.01106/log(8.296*tphio2^3-11.33*tphio2^2+26.11*tphio2+1.336))),...
% 'indep',{ 'tphio2','mu'},'depen','z')
% opts=fitoptions(ftypeG);
% opts.Lower=[-100 -100 -100 -100 -100];
% opts.Upper=[100 100 100 -1 100];
% opts.StartPoint=[1 1 1 -1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fG_alt,gofG_alt]=fit([xvec_alt,yvec],gamvec,ftypeG,opts)
% figure(7)
% plot(fG_alt,[xvec_alt,yvec],gamvec)
% title('Model and data comparison for \gamma')
% xlabel('tan(\phi/2)')
% ylabel('\mu')
% zlabel('\gamma')
% legend('Parametric fit','Contour data')

% ftypeG=fittype('(A+(B/log(C*tphio2^3+D*tphio2^2+E*tphio2+F)))',...
% 'indep',{ 'tphio2'},'depen','y')
% opts=fitoptions(ftypeG);
% opts.Lower=[-1000 -1000 1.1 -1000 1.1 1.1];
% opts.Upper=[1000 1000 1000 1000 1000 1000];
% opts.StartPoint=[1 1 1.1 1 1.1 1.1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fG_alt,gofG_alt]=fit([xvec_alt(1:301)],gamvec(1:301),ftypeG,opts)
% figure(7)
% plot(fG_alt,[xvec_alt(1:301)],gamvec(1:301))
% title('Model and data comparison for \gamma')

```

```

% xlabel('tan(\phi/2)')
% ylabel('\mu')
% zlabel('\gamma')
% legend('Parametric fit','Contour data')

gamma_eq=(0.7987+(0.1758./log(5.647*tand(phi_mat/2).^3-5.774*...
    tand(phi_mat/2).^2+26.31*tand(phi_mat/2)+1.1)))+((1.272./...
    (1+exp(9.889*mu_mat.^2-0.4737*mu_mat-1.296)))+0.0003257).*...
    (0.04462+(-0.01106./log(8.296*tand(phi_mat/2).^3-11.33*...
    tand(phi_mat/2).^2+26.11*tand(phi_mat/2)+1.336)));
figure(8)
hold on
surf(phi_mat,mu_mat,gamma_eq)
plot3(phi_mat,mu_mat,gamma_found','b.','markersize',1)
hold off
shading flat
grid on
xlabel('\phi (degrees)')
ylabel('\mu')
zlabel('\gamma')
legend('Parametric fit','Contour data')

```

B.8 Finding $c\theta$

```

clc
clear all
close all
clf

%% Inputs and initializations

h=0;
g=0;

mustart=0;
mustep=0.02;
mustop=1;

phistart=0.5;
phistep=0.5;
phistop=179.5;

sstart=0;
sstep=0.01;
sstop=9;

```

```

mat_phi=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_K=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_s=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_g=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_C_theta=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_THETA=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_K_theta=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_theta=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs((mustop-mustart)/mustep)+1);
mat_THETA_approx=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_gamma_eq=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_C_theta_approx=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
muvec=zeros(1,abs((mustop-mustart)/mustep)+1);

```

```
options=odeset('MaxStep',sstep);
```

```
%% Code
```

```

for mu=mustart:mustep:mustop
    h=h+1;
    g=0;

    for phi=phistart:phistep:phistop
        test=phi*pi/180;
        [sgrav,ggrav]=ode45('beam_all_cases_wSE',...
            (sstart:sstep:sstop),[test 0 0 mu test 0 0 0],options);

        for mm=1:length(sgrav)
            if ggrav(mm,1)>pi || ggrav(mm,2)<0
                ggrav(mm:length(sgrav),:)=[];
                sgrav(mm:length(sgrav),:)=[];
                break
            end
        end
    end

    for nn=1:length(sgrav)

```

```

if ggrav(nn,3)>0.02
    ggrav(1:nn,:)=[];
    sgrav(1:nn,:)=[];
    break
end
end

for k=1:length(sgrav)
    tempval=[cos((ggrav(k,1)-ggrav(k,5))) sin((ggrav(k,1)-ggrav(k,5)))-sin((ggrav(k,1)-
ggrav(k,5))) cos((ggrav(k,1)-ggrav(k,5)))]*[ggrav(k,6);ggrav(k,7)];
    ggrav(k,6)=-tempval(1);
    ggrav(k,7)=-tempval(2);
end

gamma=((ggrav(:,7)./sgrav).^2+(1-ggrav(:,6)./sgrav).^2)/(2*(1-ggrav(:,6)./sgrav));
THETA=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma)))));
C_theta=abs((ggrav(:,1)-ggrav(:,5))./THETA);
gamma_eq=(0.7987+(0.1758./log(5.647*tan(ggrav(:,1)/2).^3-
5.774*tan(ggrav(:,1)/2).^2+26.31*tan(ggrav(:,1)/2)+1.1)))+((1.272./(1+exp(9.889*mu^2-
0.4737*mu-1.296)))+0.0003257).*(0.04462+(-0.01106./log(8.296*tan(ggrav(:,1)/2).^3-
11.33*tan(ggrav(:,1)/2).^2+26.11*tan(ggrav(:,1)/2)+1.336)));
THETA_approx=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma_eq)))));
C_theta_approx=abs((ggrav(:,1)-ggrav(:,5))./THETA_approx);
K_theta=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3/2)).*sin(ggrav(:,1)-THETA)./THETA;
theta=ggrav(:,1)-ggrav(:,5);
g=g+1;
mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
mat_K(1:length(sgrav),g,h)=ggrav(:,2);
mat_s(1:length(sgrav),g,h)=ggrav(:,3);
mat_g(1:length(sgrav),g,h)=gamma;
mat_C_theta(1:length(sgrav),g,h)=C_theta;
mat_THETA(1:length(sgrav),g,h)=THETA;
mat_K_theta(1:length(sgrav),g,h)=K_theta;
mat_theta(1:length(sgrav),g,h)=theta;
mat_THETA_approx(1:length(sgrav),g,h)=THETA_approx;
mat_gamma_eq(1:length(sgrav),g,h)=gamma_eq;
mat_C_theta_approx(1:length(sgrav),g,h)=C_theta_approx;
muvec(h)=mu;
end
end

[rownum colnum stacknum]=size(mat_phi);

for x=1:rownum
    for y=1:colnum
        for z=1:stacknum

```

```

if mat_phi(x,y,z)==0
    mat_phi(x,y,z)=NaN;
end
if mat_s(x,y,z)==0
    mat_s(x,y,z)=NaN;
end
if mat_g(x,y,z)==0
    mat_g(x,y,z)=NaN;
end
if mat_C_theta(x,y,z)==0
    mat_C_theta(x,y,z)=NaN;
end
if mat_THETA(x,y,z)==0
    mat_THETA(x,y,z)=NaN;
end
if mat_K_theta(x,y,z)==0
    mat_K_theta(x,y,z)=NaN;
end
if mat_K(x,y,z)==0
    mat_K(x,y,z)=NaN;
end
if mat_theta(x,y,z)==0
    mat_theta(x,y,z)=NaN;
end
if isreal(mat_THETA_approx(x,y,z))==0
    mat_THETA_approx(x,y,z)=NaN;
    mat_K(x,y,z)=NaN;
    mat_K_theta(x,y,z)=NaN;
    mat_THETA(x,y,z)=NaN;
    mat_C_theta(x,y,z)=NaN;
    mat_g(x,y,z)=NaN;
    mat_s(x,y,z)=NaN;
    mat_phi(x,y,z)=NaN;
    mat_C_theta_approx(x,y,z)=NaN;
    mat_gamma_eq(x,y,z)=NaN;
end
end
end
end

```

```

philvls=20:30:170;
phi_vec=20:0.5:170;
phi_mat=ones(length(muvec),1)*phi_vec;
mu_mat=muvec'*ones(1,length(phi_vec));
Ctheta_f=zeros(length(muvec),length(phi_vec));
mat_THETA_max=zeros(length(muvec),length(phi_vec));

```



```

for M=1:h
    for P=1:length(phi_vec)
        [D]=contours((mat_THETA_approx(:, :, M))*180/pi,...
            (mat_theta(:, :, M))*180/pi,(mat_phi(:, :, M))*180/pi,...
            [phi_vec(P) phi_vec(P)]);
        D(:,1)=[];
        D(:,length(D))=[];
        D=(sortrows(D',1))';
        mat_THETA_max(M,P)=(0.3267+0.0336/(1+exp(14.78*muvec(M)^3-...
            1.754*muvec(M)^2+0.4065*muvec(M)-0.2566)))+(-0.7882*...
            (phi_vec(P)/180).^4+2.258*(phi_vec(P)/180).^3-1.944*...
            (phi_vec(P)/180).^2+1.723*(phi_vec(P)/180)-0.1701)*...
            (5.162*muvec(M)^9-164*muvec(M)^8+605.6*muvec(M)^7-...
            934.3*muvec(M)^6+731.4*muvec(M)^5-299.6*muvec(M)^4+...
            62.17*muvec(M)^3-6.786*muvec(M)^2+0.24*muvec(M)+1.929);
        t=1;
        while t<=length(D)
            if D(2,t)>180 || D(2,t)<0 || D(1,t)>mat_THETA_max(M,P)...
                *180/pi || D(1,t)<0 || isnan(D(1,t))==1
                D(:,t)=[];
                continue
            end
            t=t+1;
        end
        tempfit=polyfit(D(1,1:t-1),D(2,1:t-1),1);
        Ctheta_f(M,P)=tempfit(1);
    end
end

```

```

figure(4)
surf(phi_mat,mu_mat,Ctheta_f)
shading flat
title('C_\theta surface for changing \mu and \phi')
xlabel('\phi')
ylabel('\mu')
zlabel('C_\theta')

```

```

figure(5)
surf(phi_mat,mu_mat,mat_THETA_max*180/pi)
shading flat
title('\Theta_max surface for changing \mu and \phi')
xlabel('\phi')
ylabel('\mu')
zlabel('\Theta_max')

```

```
dlmwrite('Ctheta_matrix_approx.txt',Ctheta_f,'delimiter','');
```

B.9 Fitting $c\theta$

```
clc
```

```
clear all
```

```
close all
```

```
clf
```

```
%% Inputs and initializations
```

```
mu_mat=dlmread('Mu_matrix_Kth_approx.txt');
```

```
C_mat=dlmread('Ctheta_matrix_approx.txt');
```

```
phi_mat=zeros(51,301);
```

```
for i=1:51
```

```
    for j=1:301
```

```
        phi_mat(i,j)=20+(j-1)*0.5;
```

```
    end
```

```
end
```

```
%% Code
```

```
figure(1)
```

```
surf(phi_mat,mu_mat,C_mat)
```

```
shading flat
```

```
title('C_\theta surface for changing \mu and \phi')
```

```
xlabel('\phi (degrees)')
```

```
ylabel('\mu')
```

```
zlabel('C_\theta')
```

```
C_mat(:,1:23)=[];
```

```
mu_mat(:,1:23)=[];
```

```
phi_mat(:,1:23)=[];
```

```
figure(2)
```

```
surf(phi_mat,mu_mat,C_mat)
```

```
shading flat
```

```
title('Modified C_\theta surface for changing \mu and \phi')
```

```
xlabel('\phi (degrees)')
```

```
ylabel('\mu')
```

```
zlabel('C_\theta')
```

```
phi_mat=(phi_mat)/180;
```

```
[rowP,colP]=size(phi_mat);
```

```

t=1;
for i=1:colP
    phivec(t:t+rowP-1,1)=phi_mat(1:rowP,i);
    t=t+rowP;
end

mu_mat=mu_mat';
[rowM,colM]=size(mu_mat);

t=1;
for i=1:colM
    muvec(t:t+rowM-1,1)=mu_mat(1:rowM,i);
    t=t+rowM;
end

C_mat=C_mat';
[rowC,colC]=size(C_mat);

t=1;
for i=1:colC
    Cvec(t:t+rowC-1,1)=C_mat(1:rowC,i);
    t=t+rowC;
end

tphio2=tan((phivec*pi)/2);
tphid=tand((phi_mat(1:rowP,1)*180)/(2));
dC=C_mat(1:rowC,colC)-C_mat(1:rowC:1);
Cstart='(-0.1221/log(1.803*tphio2^5+2.399*tphio2^4+14.26*tphio2^3+...
50.4*tphio2^2+1.1*tphio2+1.254)+1.262)';
Cdiff='(-0.03772/log(1.555*tphio2^5+10.03*tphio2^4+13.39*tphio2^3+...
16.63*tphio2^2+1.1*tphio2+3.043)+0.08554)';

% ftypeC=fitype('A/log(B*tphio2^5+C*tphio2^4+D*tphio2^3+E*tphio2^2+...
% F*tphio2+G)+H','indep',{ 'tphio2' },'depen','y')
% opts=fitoptions(ftypeC);
% opts.Lower=[-1000 1.1 -1001.1 1.1 -10061.1 1.1 1.1 -1000];
% opts.Upper=[1000 1000 1000 1000 1000 1000 1000 1000];
% opts.StartPoint=[1 1.1 1.1 1.1 1.1 1.1 1.1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fC_alt,gofC_alt]=fit([tphid],C_mat(1:rowC,colC),ftypeC,opts)
% figure(3)
% plot(fC_alt,[tphid],C_mat(1:rowC,colC))
% title('Model and data comparison for C_\theta and Force Only')
% xlabel('tan(\phi/2)')

```

```

% ylabel('C_\theta')
% legend('Parametric fit','Contour data')

%r^2=0.9999

% ftypeC=fitype('A/log(B*tphio2^5+C*tphio2^4+D*tphio2^3+E*tphio2^2+...
%F*tphio2+G)+H','indep',{ 'tphio2'},'depen','y')
% opts=fitoptions(ftypeC);
% opts.Lower=[-1000 1.1 1.1 1.1 1.1 1.1 1.1 -1000];
% opts.Upper=[1000 1000 1000 1000 1000 1000 1000 1000];
% opts.StartPoint=[1 1.1 1.1 1.1 1.1 1.1 1.1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fC_alt,gofC_alt]=fit([tphid],C_mat(1:rowC,colC)-C_mat(1:rowC,1),ftypeC,opts)
% figure(3)
% plot(fC_alt,[tphid],C_mat(1:rowC,colC)-C_mat(1:rowC,1))
% title('Model and data comparison for C_\theta difference')
% xlabel('tan(\phi/2)')
% ylabel('C_\theta')
% legend('Parametric fit','Contour data')

% ftypeC=fitype('((A*mu^5+B*mu^4+C*mu^3+D*mu^2+E*mu+F)/(1+exp(G*mu^4+...
%H*mu^3+I*mu^2+J*mu+K)))','indep',{'mu}','depen','y')
% opts=fitoptions(ftypeC);
% opts.Lower=[-1000 -1000 -1000 -1000 -1000 -1000 -1000 -1000 -1000 -1000 -1000];
% opts.Upper=[1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000];
% opts.StartPoint=[1 1 1 1 1 1 1 1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fC_alt,gofC_alt]=fit([mu_mat(rowC,1:colM)],C_mat(rowC,1:colC)',ftypeC,opts)
% % [fC_alt,gofC_alt]=fit([mu_mat(rowC,1:colM)],C_mat(rowC,1:colC)', 'poly9')
% figure(4)
% plot(fC_alt,[mu_mat(rowC,1:colM)],C_mat(rowC,1:colC)')
% title('Model and data comparison for C_\theta difference')
% xlabel('\mu')
% ylabel('C_\theta')
% legend('Parametric fit','Contour data')

% ftypeC=fitype('(-0.1221/log(1.803*tphio2^5+2.399*tphio2^4+14.26*...
%tphio2^3+50.4*tphio2^2+1.1*tphio2+1.254)+1.262)+(A/(1+exp(B*mu^4+...
%C*mu^3+D*mu^2+E*mu+F)))*(-0.03772/log(1.555*tphio2^5+10.03*tphio2^4+...
%13.39*tphio2^3+16.63*tphio2^2+1.1*tphio2+3.043)+0.08554)',...
% 'indep',{ 'tphio2','mu'},'depen','z')
% opts=fitoptions(ftypeC);

```

```

% opts.Lower=[-1000 -1000 -1000 -1000 -1000 -1000];
% opts.Upper=[1000 1000 1000 1000 1000 1000];
% opts.StartPoint=[1 1 1 1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fC_alt,gofC_alt]=fit([tphio2,muvec],Cvec,ftypeC,opts)
% figure(5)
% plot(fC_alt,[tphio2,muvec],Cvec)
% title('Model and data comparison for C_\theta')
% xlabel('tan(\phi/2)')
% ylabel('\mu')
% zlabel('C_\theta')
% legend('Parametric fit','Contour data')
%r^2=0.9999

C_eq=(1.262-0.1221./log(1.803*tan(phi_mat*pi/2).^5+2.399*...
tan(phi_mat*pi/2).^4+14.26*tan(phi_mat*pi/2).^3+50.4*...
tan(phi_mat*pi/2).^2+1.1*tan(phi_mat*pi/2)+1.262))+(-1.333./(1+...
exp(-15.05*mu_mat.^4+16.2*mu_mat.^3+4.905*mu_mat.^2+0.04808*mu_mat-...
1.113))).*(-0.03772./log(1.555*tan(phi_mat*pi/2).^5+10.03*...
tan(phi_mat*pi/2).^4+13.39*tan(phi_mat*pi/2).^3+16.63*...
tan(phi_mat*pi/2).^2+1.1*tan(phi_mat*pi/2)+3.043)+0.08554);

figure(6)
hold on
surf(phi_mat*180,mu_mat,C_eq)
plot3(phi_mat*180,mu_mat,C_mat,'b.','MarkerSize',1)
hold off
shading flat
grid on
title('Model and data comparison for C_\theta')
xlabel('\phi (degrees)')
ylabel('\mu')
zlabel('C_\theta')
legend('Parametric fit','Contour data')

```

B.10 Strain Energy K_{θ} Comparisons

```

clc
clear all
close all
clf

```

```

%% Inputs and initializations

```

```

h=0;
g=0;

mustart=0;
mustep=0.5;
mustop=1;

phistart=1;
phistep=1;
phistop=179;

sstart=0;
sstep=0.01;
sstop=9;

mat_phi=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_s=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_g=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_C_theta=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_FO=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_DL=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_V=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_gamma_eq=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta4=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta5=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta6=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta7=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta8=zeros((abs((sstop-sstart)/sstep))+1,...

```

```

(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta9=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_geo=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_try=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
muvec=zeros(1,abs(round((mustop-mustart)/mustep))+1);
options=odeset('MaxStep',sstep);

```

%% Code

```

for mu=mustart:mustep:mustop
    h=h+1;
    g=0;

    for phi=phistart:phistep:phistop
        test=phi*pi/180;
        [sgrav,ggrav]=ode45('beam_all_cases_wSE',...
            (sstart:sstep:sstop),[test 0 0 mu test 0 0 0],options);

        for mm=1:length(sgrav)
            if ggrav(mm,1)>pi || ggrav(mm,2)<0
                ggrav(mm:length(sgrav),:)=[];
                sgrav(mm:length(sgrav),:)=[];
                break
            end
        end

        for nn=1:length(sgrav)
            if ggrav(nn,3)>0.02
                ggrav(1:nn,:)=[];
                sgrav(1:nn,:)=[];
                break
            end
        end

        for k=1:length(sgrav)
            tempval=[cos((ggrav(k,1)-ggrav(k,5))) sin((ggrav(k,1)-ggrav(k,5)))-sin((ggrav(k,1)-
            ggrav(k,5))) cos((ggrav(k,1)-ggrav(k,5)))]*[ggrav(k,6);ggrav(k,7)];
            ggrav(k,6)=-tempval(1);
            ggrav(k,7)=-tempval(2);
        end

        gamma=((ggrav(:,7)./sgrav).^2+(1-ggrav(:,6)./sgrav).^2)/...
            (2*(1-ggrav(:,6)./sgrav));
    end
end

```

```

gamma_eq=(0.7987+(0.1758./log(5.647*tan(ggrav(:,1)/2).^3-5.774*...
    tan(ggrav(:,1)/2).^2+26.31*tan(ggrav(:,1)/2)+1.1)))+...
    ((1.272./(1+exp(9.889*mu^2-0.4737*mu-1.296)))+0.0003257).*...
    (0.04462+(-0.01106./log(8.296*tan(ggrav(:,1)/2).^3-11.33*...
    tan(ggrav(:,1)/2).^2+26.11*tan(ggrav(:,1)/2)+1.336)));
THETA=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma)))));
C_theta=abs((ggrav(:,1)-ggrav(:,5))./THETA);
K_theta=((mu^2*sgrav.^2)+(0.4*(1-mu)^3*sgrav.^3)).*...
    sin(ggrav(:,1)-THETA)./THETA;
K_theta4=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3.*(gamma.^4)).*...
    sin(ggrav(:,1)-THETA)./THETA;
K_theta5=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3.*(gamma.^5)).*...
    sin(ggrav(:,1)-THETA)./THETA;
K_theta6=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3.*(gamma.^6)).*...
    sin(ggrav(:,1)-THETA)./THETA;
K_theta7=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3.*(gamma.^7)).*...
    sin(ggrav(:,1)-THETA)./THETA;
K_theta8=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3.*(gamma.^8)).*...
    sin(ggrav(:,1)-THETA)./THETA;
K_theta9=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3.*(gamma.^9)).*...
    sin(ggrav(:,1)-THETA)./THETA;
K_theta_geo=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3.*(0.05./...
    (1-gamma))).*sin(ggrav(:,1)-THETA)./THETA;
K_theta_try=((mu^2*sgrav.^2).*sin(ggrav(:,1)-THETA)./THETA)+...
    ((0.4*(1-mu)^3*sgrav.^3).*sin(ggrav(:,1)-THETA)./THETA);
K_theta_FO=(mu^2*sgrav.^2).*sin(ggrav(:,1)-THETA)./THETA;
K_theta_DL=((1-mu)^3*sgrav.^3).*sin(ggrav(:,1)-THETA)./THETA;
g=g+1;
mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
mat_K(1:length(sgrav),g,h)=ggrav(:,2);
mat_s(1:length(sgrav),g,h)=ggrav(:,3);
mat_g(1:length(sgrav),g,h)=gamma;
mat_gamma_eq(1:length(sgrav),g,h)=gamma_eq;
mat_C_theta(1:length(sgrav),g,h)=C_theta;
mat_THETA(1:length(sgrav),g,h)=THETA;
mat_K_theta(1:length(sgrav),g,h)=K_theta;
mat_K_theta4(1:length(sgrav),g,h)=K_theta4;
mat_K_theta5(1:length(sgrav),g,h)=K_theta5;
mat_K_theta6(1:length(sgrav),g,h)=K_theta6;
mat_K_theta7(1:length(sgrav),g,h)=K_theta7;
mat_K_theta8(1:length(sgrav),g,h)=K_theta8;
mat_K_theta9(1:length(sgrav),g,h)=K_theta9;
mat_K_theta_geo(1:length(sgrav),g,h)=K_theta_geo;
mat_K_theta_try(1:length(sgrav),g,h)=K_theta_try;
mat_K_theta_FO(1:length(sgrav),g,h)=K_theta_FO;
mat_K_theta_DL(1:length(sgrav),g,h)=K_theta_DL;

```



```

        mat_V(1:length(sgrav),g,h)=ggrav(:,8);
        muvec(h)=mu;
    end
end

[rownum colnum stacknum]=size(mat_phi);
mat_TLI=zeros(rownum,colnum,stacknum);

for z=1:stacknum
    for x=1:rownum
        for y=1:colnum
            if mat_phi(x,y,z)==0
                mat_phi(x,y,z)=NaN;
            end
            if mat_s(x,y,z)==0
                mat_s(x,y,z)=NaN;
            end
            if mat_g(x,y,z)==0
                mat_g(x,y,z)=NaN;
            end
            if mat_C_theta(x,y,z)==0
                mat_C_theta(x,y,z)=NaN;
            end
            if mat_THETA(x,y,z)==0
                mat_THETA(x,y,z)=NaN;
            end
            if mat_K_theta(x,y,z)==0
                mat_K_theta(x,y,z)=NaN;
            end
            if mat_K(x,y,z)==0
                mat_K(x,y,z)=NaN;
            end
        end
    end
end

SE_K_theta=(mat_s.*mat_V)./(mat_g.*mat_THETA.^2);
mat_V_via_K=(mat_K_theta_try.*mat_g.*mat_THETA.^2)./mat_s;

% figure(1)
% for pp=h:h
% hold on
% surf((mat_phi(:,:,pp))*180/pi,(mat_K(:,:,pp)),(SE_K_theta(:,:,pp)))
% surf((mat_phi(:,:,pp))*180/pi,(mat_K(:,:,pp)),(mat_K_theta(:,:,pp)))
% end
% shading flat

```

```

% hold off
% title('Surface Plot of K_\Theta via TLI and SE (k^2 integrated)')
% xlabel('\phi (degrees)')
% ylabel('\kappa (nondimensional)')
% zlabel('K_\Theta')
% grid on

figure(2)
hold on
surf((mat_phi(:,:,2))*180/pi,(mat_K(:,:,2)),(SE_K_theta(:,:,2)))
surf((mat_phi(:,:,2))*180/pi,(mat_K(:,:,2)),(mat_K_theta(:,:,2)))
shading flat
hold off
%title('Distributed Only Surface Plot of K_\Theta via TLI and SE ...
%(k^2 integrated)')
xlabel('\phi (degrees)')
ylabel('\kappa\lambda_3')
zlabel('K_\Theta')
grid on

```

B.11 Finding K_θ Roots

```

clc
clear all
close all
clf

%% Inputs and initializations

h=0;
g=0;

mustart=0;
mustep=0.02;
mustop=1;

phistart=0.25;
phistep=0.25;
phistop=179;

sstart=0;
sstep=0.01;
sstop=9;

mat_phi=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
phistep))+1,abs(round((mustop-mustart)/mustep))+1);

```

```

mat_s=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_V=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_gamma_eq=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_denom_DL=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_denom_FO=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA_approx=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
muvec=zeros(1,abs(round((mustop-mustart)/mustep))+1);

options=odeset('MaxStep',sstep);

```

%% Code

```

for mu=mustart:mustep:mustop
    h=h+1;
    g=0;
    for phi=phistart:phistep:phistop
        test=phi*pi/180;
        [sgrav,ggrav]=ode45('beam_all_cases_wSE',(sstart:sstep:sstop),...
            [test 0 0 mu test 0 0 0],options);

        for mm=1:length(sgrav)
            if ggrav(mm,1)>pi || ggrav(mm,2)<0
                ggrav(mm:length(sgrav),:)=[];
                sgrav(mm:length(sgrav),:)=[];
                break
            end
        end

        for nn=1:length(sgrav)
            if ggrav(nn,3)>0.02
                ggrav(1:nn,:)=[];
                sgrav(1:nn,:)=[];
                break
            end
        end

        for k=1:length(sgrav)
            tempval=[cos((ggrav(k,1)-ggrav(k,5))) ...
                sin((ggrav(k,1)-ggrav(k,5)));...
                -sin((ggrav(k,1)-ggrav(k,5))) ...

```

```

        cos((ggrav(k,1)-ggrav(k,5)))*[ggrav(k,6);ggrav(k,7)];
    ggrav(k,6)=-tempval(1);
    ggrav(k,7)=-tempval(2);
end

gamma_eq=(0.7987+(0.1758./log(5.647*tan(ggrav(:,1)/2).^3-5.774*...
    tan(ggrav(:,1)/2).^2+26.31*tan(ggrav(:,1)/2)+1.1))+...
    ((1.272./(1+exp(9.889*mu^2-0.4737*mu-1.296)))+0.0003257).*...
    (0.04462+(-0.01106./log(8.296*tan(ggrav(:,1)/2).^3-11.33*...
    tan(ggrav(:,1)/2).^2+26.11*tan(ggrav(:,1)/2)+1.336)));
THETA_approx=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma_eq)))));
factorDE=ggrav(:,1);
arg=(THETA_approx./factorDE);
denom_DL=0.4*(1-mu)^3*factorDE.*sgrav.^3.*sin(ggrav(:,1)-THETA_approx);
denom_FO=mu^2*factorDE.*sgrav.^2.*sin(ggrav(:,1)-THETA_approx);
g=g+1;
mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
mat_s(1:length(sgrav),g,h)=ggrav(:,3);
mat_THETA_approx(1:length(sgrav),g,h)=THETA_approx;
mat_gamma_eq(1:length(sgrav),g,h)=gamma_eq;
mat_V(1:length(sgrav),g,h)=ggrav(:,8);
mat_denom_DL(1:length(sgrav),g,h)=denom_DL;
mat_denom_FO(1:length(sgrav),g,h)=denom_FO;
muvec(h)=mu;
end
end

```

```

SE_K_theta_approx=(mat_s.*mat_V)./(mat_gamma_eq.*mat_THETA_approx.^2);
[rownum colnum stacknum]=size(mat_phi);

```

```

for z=1:stacknum
    for x=1:rownum
        for y=1:colnum
            if mat_phi(x,y,z)==0
                mat_phi(x,y,z)=NaN;
            end
            if mat_s(x,y,z)==0
                mat_s(x,y,z)=NaN;
            end
            if isreal(mat_THETA_approx(x,y,z))==0
                mat_THETA_approx(x,y,z)=NaN;
                mat_s(x,y,z)=NaN;
                mat_phi(x,y,z)=NaN;
                mat_V(x,y,z)=NaN;
                mat_gamma_eq(x,y,z)=NaN;
                mat_denom_DL(x,y,z)=NaN;
            end
        end
    end
end

```

```

        mat_denom_FO(x,y,z)=NaN;
    end
end
end
end

phi_vec=20:0.5:170;
phi_mat=ones(length(muvec),1)*phi_vec;
mu_mat=muvec'*ones(1,length(phi_vec));
Kth_found=zeros(length(muvec),length(phi_vec));
flag=1;
test_phival=90;
test_muval1=0.5;
test_muval2=0.25;
test_muval3=0.75;
M0=1;
M1=1;
f_of_THETA_DL=zeros(800,length(phi_vec));
f_of_THETA_FO=zeros(800,length(phi_vec));
THETA_DL=zeros(800,length(phi_vec));
THETA_FO=zeros(800,length(phi_vec));
PHI_DL=zeros(800,length(phi_vec));
PHI_FO=zeros(800,length(phi_vec));

for M=1:h
    for P=1:length(phi_vec)

        [K]=contours((SE_K_theta_approx(:, :, M)), ...
            (mat_THETA_approx(:, :, M))*180/pi, ...
            (mat_phi(:, :, M))*180/pi, [phi_vec(P) phi_vec(P)]);
        K=(sortrows(K',2))';
        K(:,1)=[];
        K(:,length(K))=[];
        t=1;
        while t<=length(K)
            if K(1,t)>4 || K(1,t)<1 || isnan(K(1,t))==1
                K(:,t)=[];
                continue
            end
            t=t+1;
        end

        K2start=K(1,1);
        Klim=K2start+0.4*K2start;

        t=1;
    end
end
end
end

```

```

if phi_vec(P)<110
    while t<=length(K)
        if K(1,t)>Klim || K(2,t)>0.8*phi_vec(P)
            K(:,t)=[];
            continue
        end
        t=t+1;
    end
else
    while t<=length(K)
        if K(1,t)>Klim || K(2,t)>0.8*phi_vec(P)
            K(:,t)=[];
            continue
        end
        t=t+1;
    end
end

factorPRBM=phi_vec(P)*pi/180;
FOlim=3;
DLlim=3;
elim=0.5;
Kth_found(M,P)=K(1,1);

if phi_vec(P)==test_phival && muvec(M)==0
    [DconDS,dconD]=contours(mat_denom_DL(:, :,M),...
        mat_THETA_approx(:, :,M),...
        mat_phi(:, :,M)*180/pi,[phi_vec(P) phi_vec(P)]);
    DconDS=(sortrows(DconDS',2))';
    DconDS(:,1)=[];
    DconDS(:,length(DconDS))=[];
    t=1;
    while t<=length(DconDS)
        if DconDS(2,t)>=phi_vec(P)*pi/180
            DconDS(:,t:length(DconDS))=[];
            break
        end
        t=t+1;
    end

    f_of_THETA_DL_sample=log(Kth_found(M,P)*...
        ones(1,1:length(DconDS)).*DconDS(2,:).^2./...
        DconDS(1,:))./log(DconDS(2,:)/(factorPRBM));
    t=1;
    while t<=length(f_of_THETA_DL_sample)
        if f_of_THETA_DL_sample(t)>DLlim || isnan(f_of_THETA_DL_sample(t))==1

```

```

        f_of_THETA_DL_sample(t)=[];
        DconDS(:,t)=[];
        continue
    end
    t=t+1;
end

figure(1)
plot(DconDS(2,:)*180/pi,f_of_THETA_DL_sample,'b')
title(['Distributed Only K_\Theta correction factor with \phi = '...
    num2str(test_phival)])
xlabel('\Theta (degrees)')
ylabel('f(\Theta)')
eqn_DL=polyfit(DconDS(2,:),f_of_THETA_DL_sample,4);
testing_eqn_DL=polyval(eqn_DL,DconDS(2,:));
hold on
plot(DconDS(2,:)*180/pi,testing_eqn_DL,'r')
hold off
legend('Precise f(\Theta)', 'Polynomial fit')
end

if muvec(M)==0
[DconD,dconD]=contours(mat_denom_DL(:,M),...
    mat_THETA_approx(:,M),mat_phi(:,M)*180/pi,...
    [phi_vec(P) phi_vec(P)]);
DconD=(sortrows(DconD',2))';
DconD(:,1)=[];
DconD(:,length(DconD))=[];
t=1;

while t<=length(DconD)
    if DconD(2,t)>=phi_vec(P)*pi/180
        DconD(:,t:length(DconD))=[];
        break
    end
    t=t+1;
end
f_of_THETA_DL(1:length(DconD),M0)=log(Kth_found(M,P)*...
    ones(1,1:length(DconD)).*DconD(2,:).^2./DconD(1,:))./...
    log(DconD(2,:)/(factorPRBM));
t=1;

while t<=length(f_of_THETA_DL(:,M0))
    if f_of_THETA_DL(t,M0)>DLLim
        f_of_THETA_DL(t,M0)=NaN;
        DconD(:,t)=NaN;
    end
end

```

```

        continue
    end
    t=t+1;
end
THETA_DL(1:length(DconD),M0)=DconD(2,:);
PHI_DL(1:length(DconD),M0)=phi_vec(P)*pi/180*ones(length(DconD(2,:)),1);
M0=M0+1;
end

if phi_vec(P)==test_phival && muvec(M)==1
    [DconFS,dconF]=contours(mat_denom_FO(:,M),...
        mat_THETA_approx(:,M),mat_phi(:,M)*180/pi,...
        [phi_vec(P) phi_vec(P)]);
    DconFS=(sortrows(DconFS',2))';
    DconFS(:,1)=[];
    DconFS(:,length(DconFS))=[];
    t=1;
    while t<=length(DconFS)
        if DconFS(2,t)>phi_vec(P)*pi/180
            DconFS(:,t:length(DconFS))=[];
            break
        end
        t=t+1;
    end
    f_of_THETA_FO_sample=log(Kth_found(M,P)*...
        ones(1,1:length(DconFS)).*DconFS(2,:).^2./DconFS(1,:))./...
        log(DconFS(2,:)/(factorPRBM));
    t=1;
    while t<=length(f_of_THETA_FO_sample)
        if f_of_THETA_FO_sample(t)>FOlim || isnan(f_of_THETA_FO_sample(t))==1
            f_of_THETA_FO_sample(t)=[];
            DconFS(:,t)=[];
            continue
        end
        t=t+1;
    end
    figure(2)
    plot(DconFS(2,:)*180/pi,f_of_THETA_FO_sample,'b')
    title('Force Only K_\Theta correction factor determination with \phi = 90')
    xlabel('\Theta (degrees)')
    ylabel('f(\Theta)')
    eqn_FO=polyfit(DconFS(2,:),f_of_THETA_FO_sample,4);
    testing_eqn_FO=polyval(eqn_FO,DconFS(2,:));
    hold on
    plot(DconFS(2,:)*180/pi,testing_eqn_FO,'r')
    hold off
end

```



```

legend('Precise f(\Theta)', 'Polynomial fit')
end
if muvec(M)==1
[DconF,dconF]=contours(mat_denom_FO(:, :, M),...
    mat_THETA_approx(:, :, M),mat_phi(:, :, M)*180/pi,...
    [phi_vec(P) phi_vec(P)]);
DconF=(sortrows(DconF',2))';
DconF(:,1)=[];
DconF(:,length(DconF))=[];
t=1;
while t<=length(DconF)
    if DconF(2,t)>phi_vec(P)*pi/180
        DconF(:,t:length(DconF))=[];
        break
    end
    t=t+1;
end
f_of_THETA_FO(1:length(DconF),M1)=log(Kth_found(M,P)*...
    ones(1,1:length(DconF)).*DconF(2,:).^2./DconF(1,:))./...
    log(DconF(2,:)/(factorPRBM));
t=1;
while t<=length(f_of_THETA_FO(:,M1))
    if f_of_THETA_FO(t,M1)>FOlim
        f_of_THETA_FO(t,M1)=NaN;
        DconF(:,t)=NaN;
        continue
    end
    t=t+1;
end
THETA_FO(1:length(DconF),M1)=DconF(2,:);
PHI_FO(1:length(DconF),M1)=phi_vec(P)*pi/180*ones(length(DconF(2,:)),1);
M1=M1+1;
end
end
end

[rowFD,colFD,stackFD]=size(f_of_THETA_DL);
for z=1:stackFD
    for x=1:rowFD
        for y=1:colFD
            if f_of_THETA_DL(x,y,z)==0 || THETA_DL(x,y,z)==0 ||
                PHI_DL(x,y,z)==0 || isreal(f_of_THETA_DL(x,y,z))==0 ||
                isreal(THETA_DL(x,y,z))==0 || isreal(PHI_DL(x,y,z))==0 ||
                THETA_DL(x,y,z)>0.8*PHI_DL(x,y,z)

                f_of_THETA_DL(x,y,z)=NaN;
            end
        end
    end
end

```

```

        THETA_DL(x,y,z)=NaN;
        PHI_DL(x,y,z)=NaN;
    end
end
end
end

[rowFF,colFF,stackFF]=size(f_of_THETA_FO);
for z=1:stackFF
    for x=1:rowFF
        for y=1:colFF
            if f_of_THETA_FO(x,y,z)==0 || THETA_FO(x,y,z)==0 ||
                PHI_FO(x,y,z)==0 || isreal(f_of_THETA_FO(x,y,z))==0 ||
                isreal(THETA_FO(x,y,z))==0 || isreal(PHI_FO(x,y,z))==0 ||
                THETA_FO(x,y,z)>0.8*PHI_FO(x,y,z)

                f_of_THETA_FO(x,y,z)=NaN;
                THETA_FO(x,y,z)=NaN;
                PHI_FO(x,y,z)=NaN;
            end
        end
    end
end
end

```

```

THETA_FO=THETA_FO./PHI_FO;
THETA_DL=THETA_DL./PHI_DL;
PHI_FO=PHI_FO/pi;
PHI_DL=PHI_DL/pi;

```

```

figure(3)
surf(phi_mat,mu_mat,Kth_found)
shading flat
title('K_\Theta surface for changing \mu and \phi')
xlabel('\phi')
ylabel('\mu')
zlabel('K_\Theta')

```

```

figure(4)
surf(PHI_DL,THETA_DL,f_of_THETA_DL)
shading flat
title('Distributed only f(\Theta) surface for changing \Theta and \phi')
xlabel('\phi')
ylabel('\Theta')
zlabel('f(\Theta)')

```

```

figure(5)

```

```

surf(PHI_FO,THETA_FO,f_of_THETA_FO)
shading flat
title('Force only f(\Theta) surface for changing \Theta and \phi')
xlabel('\phi')
ylabel('\Theta')
zlabel('f(\Theta)')

dlmwrite('Phi_matrix_Kth_constantK.txt',phi_mat,'delimiter',' ');
dlmwrite('Mu_matrix_Kth_constantK.txt',mu_mat,'delimiter',' ');
dlmwrite('Kth_matrix_constantK.txt',Kth_found,'delimiter',' ');
dlmwrite('f_of_THETA_matrix_FO_constantK.txt',f_of_THETA_FO,'delimiter',' ');
dlmwrite('f_of_THETA_matrix_DL_constantK.txt',f_of_THETA_DL,'delimiter',' ');
dlmwrite('THETA_over_PHI_matrix_FO_constantK.txt',THETA_FO,'delimiter',' ');
dlmwrite('THETA_over_PHI_matrix_DL_constantK.txt',THETA_DL,'delimiter',' ');
dlmwrite('PHI_over_pi_matrix_FO_constantK.txt',PHI_FO,'delimiter',' ');
dlmwrite('PHI_over_pi_matrix_DL_constantK.txt',PHI_DL,'delimiter',' ');

```

B.12 Fitting $K_{\theta,init}$

```

clc
clear all
close all
clf

%% Inputs and initializations

mu_mat=dlmread('Mu_matrix_Kth_approx.txt');
Kth_mat=dlmread('Kth_matrix_constantK.txt');
phi_mat=zeros(51,301);

for i=1:51
    for j=1:301
        phi_mat(i,j)=20+(j-1)*0.5;
    end
end

%% Code

figure(1)
surf(phi_mat,mu_mat,Kth_mat)
shading flat
title('K_\Theta surface for changing \mu and \phi')
xlabel('\phi (degrees)')
ylabel('\mu')
zlabel('K_\Theta')

```

```

phi_mat=(phi_mat)/180;
[rowP,colP]=size(phi_mat);

t=1;
for i=1:colP
    phivec(t:t+rowP-1,1)=phi_mat(1:rowP,i);
    t=t+rowP;
end

mu_mat=mu_mat';
[rowM,colM]=size(mu_mat);

t=1;
for i=1:colM
    muvec(t:t+rowM-1,1)=mu_mat(1:rowM,i);
    t=t+rowM;
end

Kth_mat=Kth_mat';
[rowK,colK]=size(Kth_mat);

t=1;
for i=1:colK
    Kvec(t:t+rowK-1,1)=Kth_mat(1:rowK,i);
    t=t+rowK;
end

tphio2=tan(phivec*pi/2);
tphid=tand(phi_mat(1:rowK,1)*180/2);
dK=Kth_mat(1:rowK,1)-Kth_mat(1:rowK,colK);
starteqK='(2.398+(0.5074/log(5.062*tphio2^3-5.485*tphio2^2+24.11*tphio2+1.18)))';
endeqK='(2.699+(0.5252/log(5.349*tphio2^3-5.309*tphio2^2+26.03*tphio2+1.099)))';
eq4diffK='0.09712+(1.119/log(27.77*tphio2^3-13.06*tphio2^2+85.38*tphio2+10.02))';

%
ftypeK=fittype('(A+B/log(C*tphio2^3+D*tphio2^2+E*tphio2+F))','indep',{'tphio2'},'depen','y')
% opts=fitoptions(ftypeK);
% opts.Lower=[-100 -100 0.1 -100 0.1 0.1];
% opts.Upper=[100 100 100 100 100 100];
% opts.StartPoint=[1 1 1.1 1 1.1 1.1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fK_alt,gofK_alt]=fit([tphid],Kth_mat(1:rowK,colK),ftypeK,opts)
% figure(2)
% plot(fK_alt,[tphid],Kth_mat(1:rowK,colK))

```

```

% title('Model and data comparison for K_\Theta')
% xlabel('tan(\phi/2)')
% ylabel('K_\Theta')
% legend('Parametric fit','Contour data')

%
ftypeK=fitype('(A+B/log(C*tphio2^3+D*tphio2^2+E*tphio2+F))','indep',{ 'tphio2'},'depen','y')
% opts=fitoptions(ftypeK);
% opts.Lower=[-100 -100 0.1 -100 0.1 0.1];
% opts.Upper=[100 100 100 100 100 100];
% opts.StartPoint=[1 1 1.1 1 1.1 1.1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fK_alt,gofK_alt]=fit(tphid,Kth_mat(1:rowK,1),ftypeK,opts)
% figure(3)
% plot(fK_alt,tphid,Kth_mat(1:rowK,1))
% title('Model and data comparison for K_\Theta difference')
% xlabel('tan(\phi/2)')
% ylabel('K_\Theta difference')
% legend('Parametric fit','Contour data')

% ftypeK=fitype('(2.398+(0.5074/log(5.062*tphio2^3-5.485*tphio2^2+24.11...
% *tphio2+1.18)))+(A*mu^10+B*mu^9+C*mu^8+D*mu^7+E*mu^6+F*mu^5+G*mu^4+...
% H*mu^3+I*mu^2+J*mu+K)/(1+exp(L*mu^3+M*mu^2+N*mu+O))',...
% 'indep',{ 'tphio2','mu'},'depen','z')
% opts=fitoptions(ftypeK);
% opts.Lower=[-1000 -1000 -1000 -1000 -1000 -1000 -1000 -1000 -1000 ...
%-1000 -1000 -1000 -1000 -1000 -1000];
% opts.Upper=[1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000...
% 1000 1000 1000];
% opts.StartPoint=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fK_alt,gofK_alt]=fit([tphio2,muvec],Kvec,ftypeK,opts)
% figure(5)
% plot(fK_alt,[tphio2,muvec],Kvec)
% title('Model and data comparison for K_\Theta')
% xlabel('tan(\phi/2)')
% ylabel('\mu')
% zlabel('K_\Theta')
% legend('Parametric fit','Contour data')

```

```

K_eq=(2.398+(0.5074./log(5.062*tan(phi_mat*pi/2).^3-5.485*...
tan(phi_mat*pi/2).^2+24.11*tan(phi_mat*pi/2)+1.18)))+...

```

```
(312.5*mu_mat.^10-52.56*mu_mat.^9-176.6*mu_mat.^8-126.1*...
mu_mat.^7-64.67*mu_mat.^6-20.29*mu_mat.^5+42.68*mu_mat.^4+...
107.8*mu_mat.^3-18.19*mu_mat.^2+1.299*mu_mat+0.3864)./...
(1+exp(11*mu_mat.^3-14.76*mu_mat.^2+18.03*mu_mat-1.305));
```

```
figure(6)
hold on
surf(phi_mat*180,mu_mat,K_eq)
plot3(phi_mat*180,mu_mat,Kth_mat,'b.','MarkerSize',1)
hold off
shading flat
grid on
title('Model and data comparison for K_\Theta')
xlabel('\phi (degrees)')
ylabel('\mu')
zlabel('K_\Theta')
legend('Parametric fit','Contour data')
```

B.13 Finding Correction Factors

```
clc
clear all
close all
clf

%% Inputs and initializations

h=0;
g=0;

mustart=0;
mustep=1;
mustop=1;

phistart=0.25;
phistep=0.25;
phistop=179;

sstart=0;
sstep=0.01;
sstop=9;

mat_phi=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...
/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_s=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...
/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
```

```

mat_V=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...
    /phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_gamma_eq=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...
    /phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_chi_D=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...
    /phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_chi_F=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...
    /phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA_approx=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
muvec=zeros(1,abs(round((mustop-mustart)/mustep))+1);

```

```
options=odeset('MaxStep',sstep);
```

```
%% Code
```

```

for mu=mustart:mustep:mustop
    h=h+1;
    g=0;
    for phi=phistart:phistep:phistop
        test=phi*pi/180;
        [sgrav,ggrav]=ode45('beam_all_cases_wSE',(sstart:sstep:sstop),...
            [test 0 0 mu test 0 0 0],options);

        for mm=1:length(sgrav)
            if ggrav(mm,1)>pi || ggrav(mm,2)<0
                ggrav(mm:length(sgrav),:)=[];
                sgrav(mm:length(sgrav),:)=[];
                break
            end
        end

        for nn=1:length(sgrav)
            if ggrav(nn,3)>0.02
                ggrav(1:nn,:)=[];
                sgrav(1:nn,:)=[];
                break
            end
        end

        for k=1:length(sgrav)
            tempval=[cos((ggrav(k,1)-ggrav(k,5))) sin((ggrav(k,1)-ggrav(k,5)))-sin((ggrav(k,1)-
            ggrav(k,5))) cos((ggrav(k,1)-ggrav(k,5)))]*[ggrav(k,6);ggrav(k,7)];
            ggrav(k,6)=-tempval(1);
            ggrav(k,7)=-tempval(2);
        end
    end
end

```

```

gamma_eq=(0.7987+(0.1758./log(5.647*tan(ggrav(:,1)/2).^3-5.774*...
tan(ggrav(:,1)/2).^2+26.31*tan(ggrav(:,1)/2+1.1)))+...
((1.272./(1+exp(9.889*mu^2-0.4737*mu-1.296)))+0.0003257).*...
(0.04462+(-0.01106./log(8.296*tan(ggrav(:,1)/2).^3-11.33*...
tan(ggrav(:,1)/2).^2+26.11*tan(ggrav(:,1)/2+1.336)));
THETA_approx=abs(asin(ggrav(:,7)./(sgrav.*gamma_eq)));

```

```

for i=1:length(THETA_approx)
    if THETA_approx(i)*180/pi>80
        THETA_approx(i:length(THETA_approx))=NaN;
        break
    end
end
end

```

```

factorDE=ggrav(:,1);
arg=(THETA_approx./factorDE);
K_theta_PRBM=(2.398+(0.5074./log(5.062*tan(ggrav(:,1)/2).^3-...
5.485*tan(ggrav(:,1)/2).^2+24.11*tan(ggrav(:,1)/2+1.18)))+...
(312.5*mu^10-52.56*mu^9-176.6*mu^8-126.1*mu^7-64.67*...
mu^6-20.29*mu^5+42.68*mu^4+107.8*mu^3-18.19*mu^2+1.299*...
mu+0.3864)/(1+exp(11*mu^3-14.76*mu^2+18.03*mu-1.305));
chi_D=(K_theta_PRBM.*THETA_approx)./(0.4*(1-mu)^3*sgrav.^3.*...
sin(ggrav(:,1)-THETA_approx));
chi_F=(K_theta_PRBM.*THETA_approx)./(mu^2*sgrav.^2.*...
sin(ggrav(:,1)-THETA_approx));
g=g+1;
mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
mat_s(1:length(sgrav),g,h)=ggrav(:,3);
mat_THETA_approx(1:length(sgrav),g,h)=THETA_approx;
mat_gamma_eq(1:length(sgrav),g,h)=gamma_eq;
mat_V(1:length(sgrav),g,h)=ggrav(:,8);
mat_chi_D(1:length(sgrav),g,h)=chi_D;
mat_chi_F(1:length(sgrav),g,h)=chi_F;
muvec(h)=mu;
end
end

```

```

SE_K_theta_approx=(mat_s.*mat_V)./(mat_gamma_eq.*mat_THETA_approx.^2);
[rownum colnum stacknum]=size(mat_phi);

```

```

for z=1:stacknum
    for x=1:rownum
        for y=1:colnum
            if mat_phi(x,y,z)==0
                mat_phi(x,y,z)=NaN;
            end
        end
    end
end

```



```

end
if mat_s(x,y,z)==0
    mat_s(x,y,z)=NaN;
end
if isreal(mat_THETA_approx(x,y,z))==0 ||
    isnan(mat_THETA_approx(x,y,z))==1 ||
    mat_THETA_approx(x,y,z)/mat_phi(x,y,z)>0.8

    mat_THETA_approx(x,y,z)=NaN;
    mat_s(x,y,z)=NaN;
    mat_phi(x,y,z)=NaN;
    mat_V(x,y,z)=NaN;
    mat_gamma_eq(x,y,z)=NaN;
    mat_chi_D(x,y,z)=NaN;
    mat_chi_F(x,y,z)=NaN;
end
end
end
end

```

```

figure(1)
surf(mat_phi(:,:,1)/pi,mat_THETA_approx(:,:,1),mat_chi_D(:,:,1))
shading flat
grid on
title('\chi_D Mapping')
xlabel('\phi/\pi')
ylabel('\Theta')
zlabel('\chi_D')

```

```

figure(2)
surf(mat_phi(:,:,h)/pi,mat_THETA_approx(:,:,h),mat_chi_F(:,:,h))
shading flat
grid on
title('\chi_F Mapping')
xlabel('\phi/\pi')
ylabel('\Theta')
zlabel('\chi_F')

```

```

phi_vec=20:0.5:170;
phi_mat=ones(length(muvec),1)*phi_vec;
mu_mat=muvec'*ones(1,length(phi_vec));
test_phival=90;
M01=1;
M02=1;
M11=1;
M12=1;

```

```

cD_20_100=zeros(800,length(phi_vec));
cD_100_170=zeros(800,length(phi_vec));
cF_20_100=zeros(800,length(phi_vec));
cF_100_170=zeros(800,length(phi_vec));
THETA_DL_20_100=zeros(800,length(phi_vec));
THETA_DL_100_170=zeros(800,length(phi_vec));
THETA_FO_20_100=zeros(800,length(phi_vec));
THETA_FO_100_170=zeros(800,length(phi_vec));
PHI_DL_20_100=zeros(800,length(phi_vec));
PHI_DL_100_170=zeros(800,length(phi_vec));
PHI_FO_20_100=zeros(800,length(phi_vec));
PHI_FO_100_170=zeros(800,length(phi_vec));

for M=1:h
    for P=1:length(phi_vec)
        FOlim=3;
        DLLim=3;

        if muvec(M)==0 && phi_vec(P)<=100
            [DconD,dconD]=contours(mat_chi_D(:, :, M),...
                mat_THETA_approx(:, :, M),mat_phi(:, :, M)*...
                180/pi,[phi_vec(P) phi_vec(P)]);
            DconD=(sortrows(DconD',2))';
            DconD(:,1)=[];
            DconD(:,length(DconD))=[];
            t=1;
            while t<=length(DconD)
                if DconD(2,t)>=phi_vec(P)*pi/180 || DconD(1,t)>DLLim
                    DconD(:,t:length(DconD))=[];
                    break
                end
                t=t+1;
            end
            cD_20_100(1:length(DconD),M01)=DconD(1,:);
            THETA_DL_20_100(1:length(DconD),M01)=DconD(2,:);
            PHI_DL_20_100(1:length(DconD),M01)=phi_vec(P)*...
                pi/180*ones(length(DconD(2,:)),1);
            M01=M01+1;
        end
        if muvec(M)==0 && phi_vec(P)>100
            [DconD,dconD]=contours(mat_chi_D(:, :, M),...
                mat_THETA_approx(:, :, M),mat_phi(:, :, M)*180/pi,...
                [phi_vec(P) phi_vec(P)]);
            DconD=(sortrows(DconD',2))';
            DconD(:,1)=[];
            DconD(:,length(DconD))=[];
        end
    end
end

```

```

t=1;
while t<=length(DconD)
    if DconD(2,t)>=phi_vec(P)*pi/180 || DconD(1,t)>DLlim
        DconD(:,t:length(DconD))=[];
        break
    end
    t=t+1;
end
cD_100_170(1:length(DconD),M02)=DconD(1,:);
THETA_DL_100_170(1:length(DconD),M02)=DconD(2,:);
PHI_DL_100_170(1:length(DconD),M02)=phi_vec(P)*...
    pi/180*ones(length(DconD(2,:)),1);
M02=M02+1;
end
if muvec(M)==1 && phi_vec(P)<=100
    [DconF,dconF]=contours(mat_chi_F(:,M),...
        mat_THETA_approx(:,M),mat_phi(:,M)*180/pi,...
        [phi_vec(P) phi_vec(P)]);
    DconF=(sortrows(DconF',2))';
    DconF(:,1)=[];
    DconF(:,length(DconF))=[];
    t=1;
    while t<=length(DconF)
        if DconF(2,t)>phi_vec(P)*pi/180 || DconF(1,t)>FOLim
            DconF(:,t:length(DconF))=[];
            break
        end
        t=t+1;
    end
    cF_20_100(1:length(DconF),M11)=DconF(1,:);
    THETA_FO_20_100(1:length(DconF),M11)=DconF(2,:);
    PHI_FO_20_100(1:length(DconF),M11)=phi_vec(P)*pi/180*...
        ones(length(DconF(2,:)),1);
    M11=M11+1;
end
if muvec(M)==1 && phi_vec(P)>100
    [DconF,dconF]=contours(mat_chi_F(:,M),...
        mat_THETA_approx(:,M),mat_phi(:,M)*180/pi,...
        [phi_vec(P) phi_vec(P)]);
    DconF=(sortrows(DconF',2))';
    DconF(:,1)=[];
    DconF(:,length(DconF))=[];
    t=1;
    while t<=length(DconF)
        if DconF(2,t)>phi_vec(P)*pi/180 || DconF(1,t)>FOLim
            DconF(:,t:length(DconF))=[];

```

```

        break
    end
    t=t+1;
end
cF_100_170(1:length(DconF),M12)=DconF(1,:);
THETA_FO_100_170(1:length(DconF),M12)=DconF(2,:);
PHI_FO_100_170(1:length(DconF),M12)=phi_vec(P)*...
    pi/180*ones(length(DconF(2,:)),1);
M12=M12+1;
end
end
end

[rowFD,colFD,stackFD]=size(cD_20_100);
for z=1:stackFD
    for x=1:rowFD
        for y=1:colFD
            if cD_20_100(x,y,z)==0 || THETA_DL_20_100(x,y,z)==0 ||
                PHI_DL_20_100(x,y,z)==0

                cD_20_100(x,y,z)=NaN;
                THETA_DL_20_100(x,y,z)=NaN;
                PHI_DL_20_100(x,y,z)=NaN;
            end
        end
    end
end

[rowFD,colFD,stackFD]=size(cD_100_170);
for z=1:stackFD
    for x=1:rowFD
        for y=1:colFD
            if cD_100_170(x,y,z)==0 || THETA_DL_100_170(x,y,z)==0 ||
                PHI_DL_100_170(x,y,z)==0

                cD_100_170(x,y,z)=NaN;
                THETA_DL_100_170(x,y,z)=NaN;
                PHI_DL_100_170(x,y,z)=NaN;
            end
        end
    end
end

[rowFF,colFF,stackFF]=size(cF_20_100);
for z=1:stackFF
    for x=1:rowFF
        for y=1:colFF

```

```

    if cF_20_100(x,y,z)==0 || THETA_FO_20_100(x,y,z)==0 ||
        PHI_FO_20_100(x,y,z)==0
        cF_20_100(x,y,z)=NaN;
        THETA_FO_20_100(x,y,z)=NaN;
        PHI_FO_20_100(x,y,z)=NaN;
    end
end
end
end

[rowFF,colFF,stackFF]=size(cF_100_170);
for z=1:stackFF
    for x=1:rowFF
        for y=1:colFF
            if cF_100_170(x,y,z)==0 || THETA_FO_100_170(x,y,z)==0 ||
                PHI_FO_100_170(x,y,z)==0
                cF_100_170(x,y,z)=NaN;
                THETA_FO_100_170(x,y,z)=NaN;
                PHI_FO_100_170(x,y,z)=NaN;
            end
        end
    end
end
end

```

```

THETA_FO_20_100=THETA_FO_20_100./PHI_FO_20_100;
THETA_DL_20_100=THETA_DL_20_100./PHI_DL_20_100;
THETA_FO_100_170=THETA_FO_100_170/pi;
THETA_DL_100_170=THETA_DL_100_170/pi;
PHI_FO_20_100=PHI_FO_20_100/pi;
PHI_DL_20_100=PHI_DL_20_100/pi;
PHI_FO_100_170=PHI_FO_100_170/pi;
PHI_DL_100_170=PHI_DL_100_170/pi;

```

```

figure(3)
surf(PHI_DL_20_100,THETA_DL_20_100,cD_20_100)
shading flat
title('Distributed only \chi_D surface for 20<\phi<100')
xlabel('\phi/\pi')
ylabel('\Theta/\phi')
zlabel('\chi_D')

```

```

figure(4)
surf(PHI_FO_20_100,THETA_FO_20_100,cF_20_100)
shading flat
title('Force only \chi_F surface for 20<\phi<100')
xlabel('\phi/\pi')

```

```
ylabel('\Theta/\pi')
xlabel('\chi_D')
```

```
figure(5)
surf(PHI_DL_100_170,THETA_DL_100_170,cD_100_170)
shading flat
title('Distributed only \chi_D surface for 100<\phi<170')
xlabel('\phi/\pi')
ylabel('\Theta/\pi')
xlabel('\chi_D')
```

```
figure(6)
surf(PHI_FO_100_170,THETA_FO_100_170,cF_100_170)
shading flat
title('Force only \chi_F surface for 100<\phi<170')
xlabel('\phi/\pi')
ylabel('\Theta/\pi')
xlabel('\chi_D')
```

```
dlmwrite('chi_F_20_100.txt',cF_20_100,'delimiter',' ');
dlmwrite('chi_F_100_170.txt',cF_100_170,'delimiter',' ');
dlmwrite('chi_D_20_100.txt',cD_20_100,'delimiter',' ');
dlmwrite('chi_D_100_170.txt',cD_100_170,'delimiter',' ');
dlmwrite('ToPhi_FO_20_100.txt',THETA_FO_20_100,'delimiter',' ');
dlmwrite('ToPhi_FO_100_170.txt',THETA_FO_100_170,'delimiter',' ');
dlmwrite('ToPhi_DL_20_100.txt',THETA_DL_20_100,'delimiter',' ');
dlmwrite('ToPhi_DL_100_170.txt',THETA_DL_100_170,'delimiter',' ');
dlmwrite('PhioPi_FO_20_100.txt',PHI_FO_20_100,'delimiter',' ');
dlmwrite('PhioPi_FO_100_170.txt',PHI_FO_100_170,'delimiter',' ');
dlmwrite('PhioPi_DL_20_100.txt',PHI_DL_20_100,'delimiter',' ');
dlmwrite('PhioPi_DL_100_170.txt',PHI_DL_100_170,'delimiter',' ');
```

B.14 Fitting Correction Factors

```
clc
clear all
close all
clf
```

```
%% Inputs and initializations
```

```
phi_mat_DL20=dlmread('PhioPi_DL_20_100.txt');
phi_mat_DL170=dlmread('PhioPi_DL_100_170.txt');
phi_mat_FO20=dlmread('PhioPi_FO_20_100.txt');
phi_mat_FO170=dlmread('PhioPi_FO_100_170.txt');
theta_mat_DL20=dlmread('ToPhi_DL_20_100.txt');
```

```

theta_mat_DL170=dlmread('ToPhi_DL_100_170.txt');
theta_mat_FO20=dlmread('ToPhi_FO_20_100.txt');
theta_mat_FO170=dlmread('ToPhi_FO_100_170.txt');
chi_D20=dlmread('chi_D_20_100.txt');
chi_D170=dlmread('chi_D_100_170.txt');
chi_F20=dlmread('chi_F_20_100.txt');
chi_F170=dlmread('chi_F_100_170.txt');

```

%% Code

```

figure(1)
surf(phi_mat_DL20,theta_mat_DL20,chi_D20)
shading flat
title('\chi_D surface for 20<\phi<100')
xlabel('\phi/\pi')
ylabel('\Theta/\pi')
zlabel('\chi_D')

figure(2)
surf(phi_mat_DL170,theta_mat_DL170,chi_D170)
shading flat
title('\chi_D surface for 100<\phi<170')
xlabel('\phi/\pi')
ylabel('\Theta/\pi')
zlabel('\chi_D')

figure(3)
surf(phi_mat_FO20,theta_mat_FO20,chi_F20)
shading flat
title('\chi_F surface for 20<\phi<100')
xlabel('\phi/\pi')
ylabel('\Theta/\pi')
zlabel('\chi_F')

figure(4)
surf(phi_mat_FO170,theta_mat_FO170,chi_F170)
shading flat
title('\chi_F surface for 100<\phi<170')
xlabel('\phi/\pi')
ylabel('\Theta/\pi')
zlabel('\chi_F')

[row,col]=size(phi_mat_DL20);

t=1;
r=1;

```

```

for i=1:col
    temp_phi_DL=phi_mat_DL20(:,i);
    temp_phi_FO=phi_mat_FO20(:,i);
    temp_theta_DL=theta_mat_DL20(:,i);
    temp_theta_FO=theta_mat_FO20(:,i);
    temp_fot_DL=chi_D20(:,i);
    temp_fot_FO=chi_F20(:,i);

    j=1;
    while j<=length(temp_phi_DL)
        if isnan(temp_phi_DL(j))==1 || isnan(temp_theta_DL(j))==1 || isnan(temp_fot_DL(j))==1
            temp_phi_DL(j)=[];
            temp_theta_DL(j)=[];
            temp_fot_DL(j)=[];
            continue
        end
        j=j+1;
    end

    k=1;
    while k<=length(temp_phi_FO)
        if isnan(temp_phi_FO(k))==1 || isnan(temp_theta_FO(k))==1 || isnan(temp_fot_FO(k))==1
            temp_phi_FO(k)=[];
            temp_theta_FO(k)=[];
            temp_fot_FO(k)=[];
            continue
        end
        k=k+1;
    end
    j=j-1;
    k=k-1;

    if i==1
        THETA20_DL20=temp_theta_DL;
        chi_D20_20=temp_fot_DL;
        THETA20_FO20=temp_theta_FO;
        chi_F20_20=temp_fot_FO;
    end
    if i==51
        THETA20_DL45=temp_theta_DL;
        chi_D20_45=temp_fot_DL;
    end
    if i==161
        THETA20_DL100=temp_theta_DL;
        chi_D20_100=temp_fot_DL;
        THETA20_FO100=temp_theta_FO;
    end
end

```



```

    chi_F20_100=temp_fot_FO;
end

phivec_DL20(t:t+j-1,1)=temp_phi_DL;
THETAvec_DL20(t:t+j-1,1)=temp_theta_DL;
chivec_D20(t:t+j-1,1)=temp_fot_DL;
phivec_FO20(r:r+k-1,1)=temp_phi_FO;
THETAvec_FO20(r:r+k-1,1)=temp_theta_FO;
chivec_F20(r:r+k-1,1)=temp_fot_FO;

t=t+j;
r=r+k;
end

[row,col]=size(phi_mat_DL170);

t=1;
r=1;
for i=1:col
    temp_phi_DL=phi_mat_DL170(:,i);
    temp_phi_FO=phi_mat_FO170(:,i);
    temp_theta_DL=theta_mat_DL170(:,i);
    temp_theta_FO=theta_mat_FO170(:,i);
    temp_fot_DL=chi_D170(:,i);
    temp_fot_FO=chi_F170(:,i);

    j=1;
    while j<=length(temp_phi_DL)
        if isnan(temp_phi_DL(j))==1 || isnan(temp_theta_DL(j))==1 || isnan(temp_fot_DL(j))==1
            temp_phi_DL(j)=[];
            temp_theta_DL(j)=[];
            temp_fot_DL(j)=[];
            continue
        end
        j=j+1;
    end

    k=1;
    while k<=length(temp_phi_FO)
        if isnan(temp_phi_FO(k))==1 || isnan(temp_theta_FO(k))==1 || isnan(temp_fot_FO(k))==1
            temp_phi_FO(k)=[];
            temp_theta_FO(k)=[];
            temp_fot_FO(k)=[];
            continue
        end
        k=k+1;
    end
end

```

```

end
j=j-1;
k=k-1;

if i==1
    THETA170_DL100=temp_theta_DL;
    chi_D170_100=temp_fot_DL;
    THETA170_FO100=temp_theta_FO;
    chi_F170_100=temp_fot_FO;
end
if i==140
    THETA170_DL170=temp_theta_DL;
    chi_D170_170=temp_fot_DL;
    THETA170_FO170=temp_theta_FO;
    chi_F170_170=temp_fot_FO;
end

phivec_DL170(t:t+j-1,1)=temp_phi_DL;
THETAvec_DL170(t:t+j-1,1)=temp_theta_DL;
chivec_D170(t:t+j-1,1)=temp_fot_DL;
phivec_FO170(r:r+k-1,1)=temp_phi_FO;
THETAvec_FO170(r:r+k-1,1)=temp_theta_FO;
chivec_F170(r:r+k-1,1)=temp_fot_FO;

t=t+j;
r=r+k;
end

[fDL,gofDL]=fit([phivec_DL20,THETAvec_DL20],chivec_D20,'poly55')
figure(5)
plot(fDL,[phivec_DL20,THETAvec_DL20],chivec_D20)
title('Model and data comparison for \chi_D')
xlabel('\phi/\pi')
ylabel('\Theta/\phi')
zlabel('\chi_D')
legend('Parametric fit','Contour data')
%r^2=.9986

[fF,gofF]=fit([phivec_FO20,THETAvec_FO20],chivec_F20,'poly55')
figure(6)
plot(fF,[phivec_FO20,THETAvec_FO20],chivec_F20)
title('Model and data comparison for \chi_F')
xlabel('\phi/\pi')
ylabel('\Theta/\phi')
zlabel('\chi_F')
legend('Parametric fit','Contour data')

```

```
%r^2=0.9952
```

```
[fD,gofD]=fit([phivec_DL170,THETAvec_DL170],chivec_D170,'poly55')  
figure(7)  
plot(fD,[phivec_DL170,THETAvec_DL170],chivec_D170)  
title('Model and data comparison for \chi_D')  
xlabel('\phi/\pi')  
ylabel('\Theta/\pi')  
zlabel('\chi_D')  
legend('Parametric fit','Contour data')  
%r^2=0.9937
```

```
[fF,gofF]=fit([phivec_FO170,THETAvec_FO170],chivec_F170,'poly55')  
figure(8)  
plot(fF,[phivec_FO170,THETAvec_FO170],chivec_F170)  
title('Model and data comparison for \chi_F')  
xlabel('\phi/\pi')  
ylabel('\Theta/\pi')  
zlabel('\chi_F')  
legend('Parametric fit','Contour data')  
%r^2=0.9954
```

B.15 Finding the Final K_{θ}

```
clc  
clear all  
close all  
clf
```

```
%% Inputs and initializations
```

```
h=0;  
g=0;  
  
mustart=0;  
mustep=0.02;  
mustop=1;  
  
phistart=0.5;  
phistep=0.5;  
phistop=179;  
  
sstart=0;  
sstep=0.01;  
sstop=9;
```

```

mat_phi=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_s=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_gamma_eq=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA_approx=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_eq_approx=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA_max=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
muvec=zeros(1,abs(round((mustop-mustart)/mustep))+1);

options=odeset('MaxStep',sstep);

```

%% Code

```

for mu=mustart:mustep:mustop
    h=h+1;
    g=0;
    for phi=phistart:phistep:phistop
        test=phi*pi/180;

        [sgrav,ggrav]=ode45('beam_all_cases_wSE',(sstart:sstep:sstop),...
            [test 0 0 mu test 0 0 0],options);

        for mm=1:length(sgrav)
            if ggrav(mm,1)>pi || ggrav(mm,2)<0
                ggrav(mm:length(sgrav),:)=[];
                sgrav(mm:length(sgrav),:)=[];
                break
            end
        end

        for nn=1:length(sgrav)
            if ggrav(nn,3)>0.02
                ggrav(1:nn,:)=[];
                sgrav(1:nn,:)=[];
                break
            end
        end

        for k=1:length(sgrav)
            tempval=[cos((ggrav(k,1)-ggrav(k,5))) sin((ggrav(k,1)-...
                ggrav(k,5)))-sin((ggrav(k,1)-ggrav(k,5))) cos((ggrav(k,1)-...

```

```

ggrav(k,5))]*[ggrav(k,6);ggrav(k,7)];
ggrav(k,6)=-tempval(1);
ggrav(k,7)=-tempval(2);
end

gamma_eq=(0.7987+(0.1758./log(5.647*tan(ggrav(:,1)/2).^3-5.774*...
tan(ggrav(:,1)/2).^2+26.31*tan(ggrav(:,1)/2)+1.1)))+...
((1.272./(1+exp(9.889*mu^2-0.4737*mu-1.296)))+0.0003257).*...
(0.04462+(-0.01106./log(8.296*tan(ggrav(:,1)/2).^3-11.33*...
tan(ggrav(:,1)/2).^2+26.11*tan(ggrav(:,1)/2)+1.336)));
THETA_approx=abs(asin(ggrav(:,7)/(sgrav.*gamma_eq)));
factorDE=ggrav(:,1);
arg=(THETA_approx./factorDE);
chi_D=zeros(length(sgrav),1);
chi_F=zeros(length(sgrav),1);
THETA_max=zeros(length(sgrav),1);
for i=1:length(sgrav)
    if isreal(THETA_approx(i))==0
        THETA_approx(i:length(sgrav))=NaN;
    end
    if ggrav(i,1)<=100*pi/180
        chi_D(i)=0.977+0.3589*(ggrav(i,1)/pi)+0.3*...
            (THETA_approx(i)/ggrav(i,1))-2.355*(ggrav(i,1)/pi)^2-...
            0.4842*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))-...
            2.522*(THETA_approx(i)/ggrav(i,1))^2+7.796*...
            (ggrav(i,1)/pi)^3-1.915*(ggrav(i,1)/pi)^2*...
            (THETA_approx(i)/ggrav(i,1))+7.867*(ggrav(i,1)/pi)*...
            (THETA_approx(i)/ggrav(i,1))^2+7.207*...
            (THETA_approx(i)/ggrav(i,1))^3-12.6*(ggrav(i,1)/pi)^4+...
            7.218*(ggrav(i,1)/pi)^3*(THETA_approx(i)/ggrav(i,1))-...
            10.76*(ggrav(i,1)/pi)^2*(THETA_approx(i)/ggrav(i,1))^2-...
            10.62*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))^3-...
            9.493*(THETA_approx(i)/ggrav(i,1))^4+7.87*...
            (ggrav(i,1)/pi)^5-7.536*(ggrav(i,1)/pi)^4*...
            (THETA_approx(i)/ggrav(i,1))+8.156*(ggrav(i,1)/pi)^3*...
            (THETA_approx(i)/ggrav(i,1))^2+1.761*(ggrav(i,1)/pi)^2*...
            (THETA_approx(i)/ggrav(i,1))^3+7.767*(ggrav(i,1)/pi)*...
            (THETA_approx(i)/ggrav(i,1))^4+4.517*...
            (THETA_approx(i)/ggrav(i,1))^5;
        chi_F(i)=0.9572+0.5625*(ggrav(i,1)/pi)+0.4362*...
            (THETA_approx(i)/ggrav(i,1))-3.358*(ggrav(i,1)/pi)^2-...
            1.796*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))-...
            3.918*(THETA_approx(i)/ggrav(i,1))^2+10.6*...
            (ggrav(i,1)/pi)^3+0.6124*(ggrav(i,1)/pi)^2*...
            (THETA_approx(i)/ggrav(i,1))+13.34*(ggrav(i,1)/pi)*...
            (THETA_approx(i)/ggrav(i,1))^2+9.698*...

```

```

(THETA_approx(i)/ggrav(i,1))^3-16.79*(ggrav(i,1)/pi)^4+...
6.454*(ggrav(i,1)/pi)^3*...
(THETA_approx(i)/ggrav(i,1))-18.3*(ggrav(i,1)/pi)^2*...
(THETA_approx(i)/ggrav(i,1))^2-17.73*(ggrav(i,1)/pi)*...
(THETA_approx(i)/ggrav(i,1))^3-11.79*...
(THETA_approx(i)/ggrav(i,1))^4+10.41*(ggrav(i,1)/pi)^5-...
7.822*(ggrav(i,1)/pi)^4*(THETA_approx(i)/ggrav(i,1))+...
10.28*(ggrav(i,1)/pi)^3*(THETA_approx(i)/ggrav(i,1))^2+...
6.785*(ggrav(i,1)/pi)^2*(THETA_approx(i)/ggrav(i,1))^3+...
10.8*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))^4+...
5.143*(THETA_approx(i)/ggrav(i,1))^5;
THETA_max(i)=0.75*ggrav(i,1);
elseif ggrav(i,1)>100*pi/180
chi_D(i)=2.658-12.66*(ggrav(i,1)/pi)+5.379*...
(THETA_approx(i)/pi)+38.49*(ggrav(i,1)/pi)^2-32.82*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)+9.967*...
(THETA_approx(i)/pi)^2-58.19*(ggrav(i,1)/pi)^3+76.2*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)-49.82*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^2+16.61*...
(THETA_approx(i)/pi)^3+43.74*(ggrav(i,1)/pi)^4-79.15*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)+87.3*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^2-79.33*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^3+42.31*...
(THETA_approx(i)/pi)^4-13.07*(ggrav(i,1)/pi)^5+...
29.61*(ggrav(i,1)/pi)^4*(THETA_approx(i)/pi)-46.27*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)^2+67.32*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^3-54.47*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^4+4.942*...
(THETA_approx(i)/pi)^5;
chi_F(i)=1.608-4.706*(ggrav(i,1)/pi)+2.225*...
(THETA_approx(i)/pi)+14.46*(ggrav(i,1)/pi)^2-15.06*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)+2.722*...
(THETA_approx(i)/pi)^2-22.11*(ggrav(i,1)/pi)^3+38.49*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)-25.48*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^2+11.91*...
(THETA_approx(i)/pi)^3+16.86*(ggrav(i,1)/pi)^4-44.84*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)+68.46*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^2-94.92*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^3+62.37*...
(THETA_approx(i)/pi)^4-5.134*(ggrav(i,1)/pi)^5+19.9*...
(ggrav(i,1)/pi)^4*(THETA_approx(i)/pi)-54.01*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)^2+116.2*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^3-119.8*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^4+32.37*...
(THETA_approx(i)/pi)^5;
THETA_max(i)=75*pi/180;

```

```

    end
end

K_theta_eq_approx=(mu^2*sgrav.^2.*chi_F.*sin(ggrav(:,1)-...
    THETA_approx)./THETA_approx) + (0.4*(1-mu)^3*sgrav.^3.*chi_D.*...
    sin(ggrav(:,1)-THETA_approx)./THETA_approx);
g=g+1;
mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
mat_s(1:length(sgrav),g,h)=ggrav(:,3);
mat_THETA_approx(1:length(sgrav),g,h)=THETA_approx;
mat_K_theta_eq_approx(1:length(sgrav),g,h)=K_theta_eq_approx;
mat_gamma_eq(1:length(sgrav),g,h)=gamma_eq;
mat_THETA_max(1:length(sgrav),g,h)=THETA_max;
muvec(h)=mu;
end
end

[rownum colnum stacknum]=size(mat_phi);

for z=1:stacknum
    for x=1:rownum
        for y=1:colnum
            if mat_phi(x,y,z)==0
                mat_phi(x,y,z)=NaN;
            end
            if mat_s(x,y,z)==0
                mat_s(x,y,z)=NaN;
            end
            if isreal(mat_THETA_approx(x,y,z))==0 ||
                mat_THETA_approx(x,y,z)>mat_THETA_max(x,y,z)

                mat_THETA_approx(x,y,z)=NaN;
                mat_s(x,y,z)=NaN;
                mat_phi(x,y,z)=NaN;
                mat_K_theta_eq_approx(x,y,z)=NaN;
                mat_gamma_eq(x,y,z)=NaN;
            end
        end
    end
end

phi_vec=20:0.5:170;
phi_mat=ones(length(muvec)-2,1)*phi_vec;
mu_mat=muvec(2:h-1)*ones(1,length(phi_vec));
Kth_found=zeros(length(muvec)-2,length(phi_vec));
flag=1;

```

```

test_phival=45;
test_muval1=0.25;
test_muval2=0.5;
test_muval3=0.75;

```

```

for M=2:h-1

```

```

    %for P=2*(test_phival-20)+1
    for P=1:length(phi_vec)

```

```

        [K]=contours((mat_K_theta_eq_approx(:,:,M),...
            (mat_THETA_approx(:,:,M))*180/pi,(mat_phi(:,:,M))*180/pi,...
            [phi_vec(P) phi_vec(P)]);

```

```

        K=(sortrows(K',2));

```

```

        K(:,1)=[];

```

```

        K(:,length(K))=[];

```

```

        t=1;

```

```

        while t<=length(K)

```

```

            if K(1,t)>4 || K(1,t)<1 || isnan(K(1,t))==1

```

```

                K(:,t)=[];

```

```

                continue

```

```

            end

```

```

            t=t+1;

```

```

        end

```

```

        Kth_found(M-1,P)=mean(K(1,:));

```

```

        Kvec=Kth_found(M-1,P)*ones(1,length(K));

```

```

        if phi_vec(P)==test_phival && muvec(M)==test_muval1

```

```

            figure(1)

```

```

            hold on

```

```

            plot(Kvec,K(2,:), 'k')

```

```

            plot(K(1,:),K(2,:), 'r')

```

```

            xlim([min(K(1,:))-1 max(K(1,:))+1])

```

```

            ylim([0 phi_vec(P)])

```

```

            title(['Mean K_\Theta and PRBM K_\Theta (\mu = ' num2str(muvec(M)) ')'])

```

```

            xlabel('K_\Theta')

```

```

            ylabel('\Theta (in degrees)')

```

```

            legend('Mean K_\Theta','PRBM K_\Theta')

```

```

            hold off

```

```

        end

```

```

        if phi_vec(P)==test_phival && muvec(M)==test_muval2

```

```

            figure(2)

```

```

            hold on

```

```

            plot(Kvec,K(2,:), 'k')

```

```

            plot(K(1,:),K(2,:), 'r')

```

```

            xlim([min(K(1,:))-1 max(K(1,:))+1])

```



```

ylim([0 phi_vec(P)])
title(['Mean K_\Theta and PRBM K_\Theta (\mu = ' num2str(muvec(M)) ')])
xlabel('K_\Theta')
ylabel('\Theta (in degrees)')
legend('Mean K_\Theta','PRBM K_\Theta')
hold off
end
if phi_vec(P)==test_phival && muvec(M)==test_muval3
figure(3)
hold on
plot(Kvec,K(2:,:), 'k')
plot(K(1,:),K(2:,:), 'r')
xlim([min(K(1,:))-1 max(K(1,:))+1])
ylim([0 phi_vec(P)])
title(['Mean K_\Theta and PRBM K_\Theta (\mu = ' num2str(muvec(M)) ')])
xlabel('K_\Theta')
ylabel('\Theta (in degrees)')
legend('Mean K_\Theta','PRBM K_\Theta')
hold off
end

end
end

figure(4)
surf(phi_mat,mu_mat,Kth_found)
shading flat
title('K_\Theta surface for changing \mu and \phi')
xlabel('\phi')
ylabel('\mu')
zlabel('K_\Theta')

dlmwrite('Phi_mat_mean_PRBM_Kth.txt',phi_mat,'delimiter',' ');
dlmwrite('Mu_mat_mean_PRBM_Kth.txt',mu_mat,'delimiter',' ');
dlmwrite('Kth_mat_mean_PRBM_Kth.txt',Kth_found,'delimiter',' ');

```

B.16 Fitting the Final K_{θ}

```

clc
clear all
close all
clf

%% Inputs and initializations

mu_mat=dlmread('Mu_matrix_Kth_approx.txt');

```

```

Kth_mat=dlmread('Kth_matrix_constantK.txt');
Kth_mat_middle=dlmread('Kth_mat_mean_PRBM_Kth.txt');
Kth_mat(2:50,:)=Kth_mat_middle;
phi_mat=zeros(51,301);

```

```

for i=1:51
    for j=1:301
        phi_mat(i,j)=20+(j-1)*0.5;
    end
end

```

%% Code

```

figure(1)
surf(phi_mat,mu_mat,Kth_mat)
shading flat
title('K_\Theta surface for changing \mu and \phi')
xlabel('\phi (degrees)')
ylabel('\mu')
zlabel('K_\Theta')

```

```

phi_mat=(phi_mat')/180;
[rowP,colP]=size(phi_mat);

```

```

t=1;
for i=1:colP
    phivec(t:t+rowP-1,1)=phi_mat(1:rowP,i);
    t=t+rowP;
end

```

```

mu_mat=mu_mat';
[rowM,colM]=size(mu_mat);

```

```

t=1;
for i=1:colM
    muvec(t:t+rowM-1,1)=mu_mat(1:rowM,i);
    t=t+rowM;
end

```

```

Kth_mat=Kth_mat';
[rowK,colK]=size(Kth_mat);

```

```

t=1;
for i=1:colK
    Kvec(t:t+rowK-1,1)=Kth_mat(1:rowK,i);
    t=t+rowK;
end

```

end

```
tphio2=tan(phivec*pi/2);
tphid=tand(phi_mat(1:rowK,1)*180/2);
dK=Kth_mat(1:rowK,1)-Kth_mat(1:rowK,colK);
starteqK='(2.398+(0.5074/log(5.062*tphio2^3-5.485*tphio2^2+24.11*tphio2+1.18)))';
endeqK='(2.699+(0.5252/log(5.349*tphio2^3-5.309*tphio2^2+26.03*tphio2+1.099)))';
eq4diffK='0.09712+(1.119/log(27.77*tphio2^3-13.06*tphio2^2+85.38*tphio2+10.02))';
original_K='(2.398+(0.5074./log(5.062*tan(ggrav(:,1)/2).^3-5.485*...
tan(ggrav(:,1)/2).^2+24.11*tan(ggrav(:,1)/2)+1.18)))+(312.5*mu^10-52.56...
*mu^9-176.6*mu^8-126.1*mu^7-64.67*mu^6-20.29*mu^5+42.68*mu^4+107.8*...
mu^3-18.19*mu^2+1.299*mu+0.3864)/(1+exp(11*mu^3-14.76*mu^2+18.03*mu-1.305))';

% ftypeK=fittype('(A+B/log(C*tphio2^3+D*tphio2^2+E*tphio2+F))',...
% 'indep',{ 'tphio2'},'depen','y')
% opts=fitoptions(ftypeK);
% opts.Lower=[-100 -100 0.1 -100 0.1 0.1];
% opts.Upper=[100 100 100 100 100 100];
% opts.StartPoint=[1 1 1.1 1 1.1 1.1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fK_alt,gofK_alt]=fit([tphid],Kth_mat(1:rowK,colK),ftypeK,opts)
% figure(2)
% plot(fK_alt,[tphid],Kth_mat(1:rowK,colK))
% title('Model and data comparison for K_\Theta')
% xlabel('tan(\phi/2)')
% ylabel('K_\Theta')
% legend('Parametric fit','Contour data')

% ftypeK=fittype('(A+B/log(C*tphio2^3+D*tphio2^2+E*tphio2+F))','indep',...
% {'tphio2'},'depen','y')
% opts=fitoptions(ftypeK);
% opts.Lower=[-100 -100 0.1 -100 0.1 0.1];
% opts.Upper=[100 100 100 100 100 100];
% opts.StartPoint=[1 1 1.1 1 1.1 1.1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fK_alt,gofK_alt]=fit(tphid,Kth_mat(1:rowK,1),ftypeK,opts)
% figure(3)
% plot(fK_alt,tphid,Kth_mat(1:rowK,1))
% title('Model and data comparison for K_\Theta difference')
% xlabel('tan(\phi/2)')
% ylabel('K_\Theta difference')
% legend('Parametric fit','Contour data')
```

```

% ftypeK=fittype('(2.398+(0.5074/log(5.062*tphio2^3-5.485*tphio2^2+24.11*...
% tphio2+1.18)))+(A*mu^2+B*mu+C)/(1+exp(D*mu^3+E*mu^2+F*mu+G))','indep',...
% {'tphio2','mu'},'depen','z')
% opts=fitoptions(ftypeK);
% opts.Lower=[-1000 -1000 -1000 -1000 -1000 -1000 -1000];
% opts.Upper=[1000 1000 1000 1000 1000 1000 1000];
% opts.StartPoint=[1 1 1 1 1 1 1];
% opts.Robust='LAR';
% opts.MaxIter=100000000;
%
% [fK_alt,gofK_alt]=fit([tphio2,muvec],Kvec,ftypeK,opts)
% figure(5)
% plot(fK_alt,[tphio2,muvec],Kvec)
% title('Model and data comparison for K_\Theta')
% xlabel('tan(\phi/2)')
% ylabel('\mu')
% zlabel('K_\Theta')
% legend('Parametric fit','Contour data')

%r^2=0.9996

```

```

K_eq=(2.398+(0.5074./log(5.062*tan(phi_mat*pi/2).^3-5.485*...
tan(phi_mat*pi/2).^2+24.11*tan(phi_mat*pi/2)+1.18)))+...
(-0.248*mu_mat.^2-0.2399*mu_mat+0.3882)/(1+exp(-11.21*...
mu_mat.^3+22.52*mu_mat.^2-4.814*mu_mat-1.258));

```

```

figure(6)
hold on
surf(phi_mat*180,mu_mat,K_eq)
plot3(phi_mat*180,mu_mat,Kth_mat,'b.','MarkerSize',1)
hold off
shading flat
grid on
title('Model and data comparison for K_\Theta')
xlabel('\phi (degrees)')
ylabel('\mu')
zlabel('K_\Theta')
legend('Parametric fit','Contour data')

```

B.17 Effects of Correction Factors

```

clc
clear all
close all
clf

```

```
%% Inputs and initializations
```

```
h=0;  
g=0;
```

```
mustart=0.6;  
mustep=0.4;  
mustop=1;
```

```
phistart=0.5;  
phistep=0.5;  
phistop=179;
```

```
sstart=0;  
sstep=0.01;  
sstop=9;
```

```
mat_phi=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...  
/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
mat_s=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...  
/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
mat_gamma_eq=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...  
/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
mat_THETA_approx=zeros((abs((sstop-sstart)/sstep))+1,...  
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
mat_K_theta_eq_approx=zeros((abs((sstop-sstart)/sstep))+1,...  
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
mat_THETA_max=zeros((abs((sstop-sstart)/sstep))+1,...  
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
mat_K_theta=zeros((abs((sstop-sstart)/sstep))+1,...  
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
mat_K_theta_PRBM=zeros((abs((sstop-sstart)/sstep))+1,...  
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);  
muvec=zeros(1,abs(round((mustop-mustart)/mustep))+1);  
  
options=odeset('MaxStep',sstep);
```

```
%% Code
```

```
for mu=mustart:mustep:mustop  
h=h+1;  
g=0;  
for phi=phistart:phistep:phistop  
test=phi*pi/180;  
[sgrav,ggrav]=ode45('beam_all_cases_wSE',...
```

```

(ssstart:ssstep:ssstop),[test 0 0 mu test 0 0 0],options);

for mm=1:length(sgrav)
    if ggrav(mm,1)>pi || ggrav(mm,2)<0
        ggrav(mm:length(sgrav),:)=[];
        sgrav(mm:length(sgrav),:)=[];
        break
    end
end

for nn=1:length(sgrav)
    if ggrav(nn,3)>0.02
        ggrav(1:nn,:)=[];
        sgrav(1:nn,:)=[];
        break
    end
end

for k=1:length(sgrav)
    tempval=[cos((ggrav(k,1)-ggrav(k,5))) ...
            sin((ggrav(k,1)-ggrav(k,5)));-sin((ggrav(k,1)-ggrav(k,5))) ...
            cos((ggrav(k,1)-ggrav(k,5)))]*[ggrav(k,6);ggrav(k,7)];
    ggrav(k,6)=-tempval(1);
    ggrav(k,7)=-tempval(2);
end

gamma_eq=(0.7987+(0.1758./log(5.647*tan(ggrav(:,1)/2).^3-5.774*...
    tan(ggrav(:,1)/2).^2+26.31*tan(ggrav(:,1)/2)+1.1)))+...
    ((1.272./(1+exp(9.889*mu^2-0.4737*mu-1.296)))+0.0003257).*...
    (0.04462+(-0.01106./log(8.296*tan(ggrav(:,1)/2).^3-11.33*...
    tan(ggrav(:,1)/2).^2+26.11*tan(ggrav(:,1)/2)+1.336)));
THETA_approx=abs(asin(ggrav(:,7))./(sgrav.*gamma_eq));
factorDE=ggrav(:,1);
arg=(THETA_approx./factorDE);
chi_D=zeros(length(sgrav),1);
chi_F=zeros(length(sgrav),1);
THETA_max=zeros(length(sgrav),1);

for i=1:length(sgrav)
    if isreal(THETA_approx(i))==0
        THETA_approx(i:length(sgrav))=NaN;
    end
    if ggrav(i,1)<=100*pi/180
        chi_D(i)=0.977+0.3589*(ggrav(i,1)/pi)+0.3*...
            (THETA_approx(i)/ggrav(i,1))-2.355*(ggrav(i,1)/pi)^2-...
            0.4842*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))-...

```

$2.522*(\text{THETA_approx}(i)/\text{ggrav}(i,1))^2+7.796*...$
 $(\text{ggrav}(i,1)/\pi)^3-1.915*(\text{ggrav}(i,1)/\pi)^2*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))+7.867*(\text{ggrav}(i,1)/\pi)*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^2+7.207*(\text{THETA_approx}(i)/...$
 $\text{ggrav}(i,1))^3-12.6*(\text{ggrav}(i,1)/\pi)^4+7.218*...$
 $(\text{ggrav}(i,1)/\pi)^3*(\text{THETA_approx}(i)/\text{ggrav}(i,1))-...$
 $10.76*(\text{ggrav}(i,1)/\pi)^2*(\text{THETA_approx}(i)/\text{ggrav}(i,1))^2-...$
 $10.62*(\text{ggrav}(i,1)/\pi)*(THETA_approx}(i)/\text{ggrav}(i,1))^3-...$
 $9.493*(\text{THETA_approx}(i)/\text{ggrav}(i,1))^4+7.87*...$
 $(\text{ggrav}(i,1)/\pi)^5-7.536*(\text{ggrav}(i,1)/\pi)^4*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))+8.156*(\text{ggrav}(i,1)/\pi)^3*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^2+1.761*(\text{ggrav}(i,1)/\pi)^2*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^3+7.767*(\text{ggrav}(i,1)/\pi)*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^4+4.517*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^5;$
 $\text{chi_F}(i)=0.9572+0.5625*(\text{ggrav}(i,1)/\pi)+0.4362*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))-3.358*(\text{ggrav}(i,1)/\pi)^2-...$
 $1.796*(\text{ggrav}(i,1)/\pi)*(THETA_approx}(i)/\text{ggrav}(i,1))-...$
 $3.918*(\text{THETA_approx}(i)/\text{ggrav}(i,1))^2+10.6*...$
 $(\text{ggrav}(i,1)/\pi)^3+0.6124*(\text{ggrav}(i,1)/\pi)^2*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))+13.34*(\text{ggrav}(i,1)/\pi)*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^2+9.698*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^3-16.79*(\text{ggrav}(i,1)/\pi)^4+...$
 $6.454*(\text{ggrav}(i,1)/\pi)^3*(THETA_approx}(i)/\text{ggrav}(i,1))-...$
 $18.3*(\text{ggrav}(i,1)/\pi)^2*(THETA_approx}(i)/\text{ggrav}(i,1))^2-...$
 $17.73*(\text{ggrav}(i,1)/\pi)*(THETA_approx}(i)/\text{ggrav}(i,1))^3-...$
 $11.79*(\text{THETA_approx}(i)/\text{ggrav}(i,1))^4+10.41*...$
 $(\text{ggrav}(i,1)/\pi)^5-7.822*(\text{ggrav}(i,1)/\pi)^4*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))+10.28*(\text{ggrav}(i,1)/\pi)^3*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^2+6.785*(\text{ggrav}(i,1)/\pi)^2*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^3+10.8*(\text{ggrav}(i,1)/\pi)*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^4+5.143*...$
 $(\text{THETA_approx}(i)/\text{ggrav}(i,1))^5;$
 $\text{THETA_max}(i)=0.75*\text{ggrav}(i,1);$
elseif $\text{ggrav}(i,1)>100*\pi/180$
 $\text{chi_D}(i)=2.658-12.66*(\text{ggrav}(i,1)/\pi)+5.379*...$
 $(\text{THETA_approx}(i)/\pi)+38.49*(\text{ggrav}(i,1)/\pi)^2-32.82*...$
 $(\text{ggrav}(i,1)/\pi)*(THETA_approx}(i)/\pi)+9.967*...$
 $(\text{THETA_approx}(i)/\pi)^2-58.19*(\text{ggrav}(i,1)/\pi)^3+76.2*...$
 $(\text{ggrav}(i,1)/\pi)^2*(THETA_approx}(i)/\pi)-49.82*...$
 $(\text{ggrav}(i,1)/\pi)*(THETA_approx}(i)/\pi)^2+16.61*...$
 $(\text{THETA_approx}(i)/\pi)^3+43.74*(\text{ggrav}(i,1)/\pi)^4-79.15*...$
 $(\text{ggrav}(i,1)/\pi)^3*(THETA_approx}(i)/\pi)+87.3*...$
 $(\text{ggrav}(i,1)/\pi)^2*(THETA_approx}(i)/\pi)^2-79.33*...$
 $(\text{ggrav}(i,1)/\pi)*(THETA_approx}(i)/\pi)^3+42.31*...$
 $(\text{THETA_approx}(i)/\pi)^4-13.07*(\text{ggrav}(i,1)/\pi)^5+29.61*...$

```

(ggrav(i,1)/pi)^4*(THETA_approx(i)/pi)-46.27*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)^2+67.32*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^3-54.47*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^4+4.942*...
(THETA_approx(i)/pi)^5;
chi_F(i)=1.608-4.706*(ggrav(i,1)/pi)+2.225*...
(THETA_approx(i)/pi)+14.46*(ggrav(i,1)/pi)^2-15.06*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)+2.722*...
(THETA_approx(i)/pi)^2-22.11*(ggrav(i,1)/pi)^3+38.49*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)-25.48*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^2+11.91*...
(THETA_approx(i)/pi)^3+16.86*(ggrav(i,1)/pi)^4-44.84*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)+68.46*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^2-94.92*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^3+62.37*...
(THETA_approx(i)/pi)^4-5.134*(ggrav(i,1)/pi)^5+19.9*...
(ggrav(i,1)/pi)^4*(THETA_approx(i)/pi)-54.01*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)^2+116.2*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^3-119.8*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^4+32.37*...
(THETA_approx(i)/pi)^5;
THETA_max(i)=75*pi/180;
end
end

K_theta_eq_approx=(mu^2*sgrav.^2.*chi_F.*sin(ggrav(:,1)-...
THETA_approx)./THETA_approx)...
+ (0.4*(1-mu)^3*sgrav.^3.*chi_D.*sin(ggrav(:,1)-...
THETA_approx)./THETA_approx);
K_theta_PRBM=(2.398+(0.5074./log(5.062*tan(ggrav(:,1)/2).^3-...
5.485*tan(ggrav(:,1)/2).^2+24.11*tan(ggrav(:,1)/2)+1.18))+...
(-0.248*mu^2-0.2399*mu+0.3882)/(1+exp(-11.21*mu^3+22.52*...
mu^2-4.814*mu-1.258));
K_theta=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3)).*sin(ggrav(:,1)-...
THETA_approx)./THETA_approx;

g=g+1;

mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
mat_s(1:length(sgrav),g,h)=ggrav(:,3);
mat_THETA_approx(1:length(sgrav),g,h)=THETA_approx;
mat_K_theta_eq_approx(1:length(sgrav),g,h)=K_theta_eq_approx;
mat_gamma_eq(1:length(sgrav),g,h)=gamma_eq;
mat_THETA_max(1:length(sgrav),g,h)=THETA_max;
mat_K_theta(1:length(sgrav),g,h)=K_theta;
mat_K_theta_PRBM(1:length(sgrav),g,h)=K_theta_PRBM;

```



```

        muvec(h)=mu;
    end
end

[rownum colnum stacknum]=size(mat_phi);

for z=1:stacknum
    for x=1:rownum
        for y=1:colnum
            if mat_phi(x,y,z)==0
                mat_phi(x,y,z)=NaN;
            end
            if mat_s(x,y,z)==0
                mat_s(x,y,z)=NaN;
            end
            if isreal(mat_THETA_approx(x,y,z))==0 ||
                mat_THETA_approx(x,y,z)>mat_THETA_max(x,y,z)

                mat_THETA_approx(x,y,z)=NaN;
                mat_s(x,y,z)=NaN;
                mat_phi(x,y,z)=NaN;
                mat_K_theta_eq_approx(x,y,z)=NaN;
                mat_gamma_eq(x,y,z)=NaN;
                mat_K_theta(x,y,z)=NaN;
                mat_K_theta_PRBM(x,y,z)=NaN;
            end
        end
    end
end
end

```

```

phi_vec=20:0.5:170;
phi_mat=ones(length(muvec)-2,1)*phi_vec;
mu_mat=muvec(2:h-1)*ones(1,length(phi_vec));
Kth_found=zeros(length(muvec)-2,length(phi_vec));
flag=1;
test_phival=90;
test_muval1=0.6;
test_muval2=1;
test_muval3=0.75;

```

```

for M=1:h
    %for P=2*(test_phival-20)+1
    for P=1:length(phi_vec)

```

```

        [K]=contours((mat_K_theta_eq_approx(:, :, M)), ...

```

```

    (mat_THETA_approx(:,:,M))*180/pi,(mat_phi(:,:,M))*180/pi,...
    [phi_vec(P) phi_vec(P)]);
K=(sortrows(K',2));
K(:,1)=[];
K(:,length(K))=[];
t=1;
while t<=length(K)
    if K(1,t)>4 || K(1,t)<1 || isnan(K(1,t))==1
        K(:,t)=[];
        continue
    end
    t=t+1;
end

Kth_found(M,P)=mean(K(1,:));
Kvec=Kth_found(M,P)*ones(1,length(K));

[Ke]=contours((mat_K_theta_PRBM(:,:,M)),...
    (mat_THETA_approx(:,:,M))*180/pi,(mat_phi(:,:,M))*180/pi,...
    [phi_vec(P) phi_vec(P)]);
Ke=(sortrows(Ke',2));
Ke(:,1)=[];
Ke(:,length(Ke))=[];
t=1;
while t<=length(Ke)
    if Ke(1,t)>4 || Ke(1,t)<1 || isnan(Ke(1,t))==1
        Ke(:,t)=[];
        continue
    end
    t=t+1;
end

[Kt]=contours((mat_K_theta(:,:,M)),...
    (mat_THETA_approx(:,:,M))*180/pi,(mat_phi(:,:,M))*180/pi,...
    [phi_vec(P) phi_vec(P)]);
Kt=(sortrows(Kt',2));
Kt(:,1)=[];
Kt(:,length(Kt))=[];
t=1;
while t<=length(Kt)
    if Kt(1,t)>4 || Kt(1,t)<1 || isnan(Kt(1,t))==1
        Kt(:,t)=[];
        continue
    end
    t=t+1;
end

```

```

if phi_vec(P)==test_phival && muvec(M)==test_muval1
    figure(1)
    hold on
    plot(Kvec,K(2:),'k')
    plot(K(1:),K(2:),'b')
    plot(Ke(1:),Ke(2:),'k:')
    plot(Kt(1:),Kt(2:),'r')
    xlim([min(K(1:))-1 max(K(1:))+1])
    ylim([0 phi_vec(P)])
    title(['K_\Theta Model Comparisons (\mu = ' num2str(muvec(M)) ')]')
    xlabel('K_\Theta')
    ylabel('\Theta (in degrees)')
    legend('Mean K_\Theta','Corrected K_\Theta',...
        'PRBM K_\Theta value','Uncorrected K_\Theta')
    hold off
end
if phi_vec(P)==test_phival && muvec(M)==test_muval2
    figure(2)
    hold on
    plot(Kvec,K(2:),'k')
    plot(K(1:),K(2:),'b')
    plot(Ke(1:),Ke(2:),'k:')
    plot(Kt(1:),Kt(2:),'r')
    xlim([min(K(1:))-1 max(K(1:))+1])
    ylim([0 phi_vec(P)])
    title(['K_\Theta Model Comparisons (\mu = ' num2str(muvec(M)) ')]')
    xlabel('K_\Theta')
    ylabel('\Theta (in degrees)')
    legend('Mean K_\Theta','Corrected K_\Theta',...
        'PRBM K_\Theta value','Uncorrected K_\Theta')
    hold off
end
if phi_vec(P)==test_phival && muvec(M)==test_muval3
    figure(3)
    hold on
    plot(Kvec,K(2:),'k')
    plot(K(1:),K(2:),'b')
    plot(Ke(1:),Ke(2:),'k:')
    plot(Kt(1:),Kt(2:),'r')
    xlim([min(K(1:))-1 max(K(1:))+1])
    ylim([0 phi_vec(P)])
    title(['K_\Theta Model Comparisons (\mu = ' num2str(muvec(M)) ')]')
    xlabel('K_\Theta')
    ylabel('\Theta (in degrees)')
    legend('Mean K_\Theta','Corrected K_\Theta',...

```

```

        'PRBM K_\Theta value','Uncorrected K_\Theta')
    hold off
end

```

```

end
end

```

B.18 Error Analysis for $\mu = 0$ and $\mu = 1$

```

clc
clear all
close all
clf

```

```

%% Inputs and initializations

```

```

h=0;
g=0;

```

```

mustart=0;
mustep=1;
mustop=1;

```

```

phistart=1;
phistep=0.5;
phistop=179;

```

```

sstart=0;
sstep=0.01;
sstop=9;

```

```

mat_phi=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_s=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_g=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)/...
    phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_gamma_eq=zeros((abs((sstop-sstart)/sstep))+1,(abs((phistop-phistart)...
    /phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA_approx=zeros((abs((sstop-sstart)/sstep))+1,...
    (abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_eq_approx=zeros((abs((sstop-sstart)/sstep))+1,...

```

```

(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_calc_s_approx=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_calc_s=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_error_s_approx=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_error_s=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_THETA_max=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_PRBM=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_K_theta_H=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_calc_s_H=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
mat_error_s_H=zeros((abs((sstop-sstart)/sstep))+1,...
(abs((phistop-phistart)/phistep))+1,abs(round((mustop-mustart)/mustep))+1);
muvec=zeros(1,abs(round((mustop-mustart)/mustep))+1);

options=odeset('MaxStep',sstep);

```

%% Code

```

for mu=mustart:mustep:mustop
    h=h+1;
    g=0;
    for phi=phistart:phistep:phistop
        test=phi*pi/180;
        [sgrav,ggrav]=ode45('beam_all_cases_wSE',...
            (sstart:sstep:sstop),[test 0 0 mu test 0 0 0],options);

        for mm=1:length(sgrav)
            if ggrav(mm,1)>pi || ggrav(mm,2)<0
                ggrav(mm:length(sgrav),:)=[];
                sgrav(mm:length(sgrav),:)=[];
                break
            end
        end

        for nn=1:length(sgrav)
            if ggrav(nn,3)>0.02
                ggrav(1:nn,:)=[];
                sgrav(1:nn,:)=[];
                break
            end
        end
    end
end

```

```
end
end
```

```
for k=1:length(sgrav)
    tempval=[cos((ggrav(k,1)-ggrav(k,5))) ...
            sin((ggrav(k,1)-ggrav(k,5)));-sin((ggrav(k,1)-ggrav(k,5))) ...
            cos((ggrav(k,1)-ggrav(k,5)))]*[ggrav(k,6);ggrav(k,7)];
    ggrav(k,6)=-tempval(1);
    ggrav(k,7)=-tempval(2);
end
```

```
gamma=((ggrav(:,7)./sgrav).^2+(1-ggrav(:,6)./sgrav).^2)/(2*(1-ggrav(:,6)./sgrav));
THETA=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma)))));
gamma_eq=(0.7987+(0.1758./log(5.647*tan(ggrav(:,1)/2).^3-5.774*...
    tan(ggrav(:,1)/2).^2+26.31*tan(ggrav(:,1)/2)+1.1)))+(1.272./...
    (1+exp(9.889*mu^2-0.4737*mu-1.296)))+0.0003257.*(0.04462+...
    (-0.01106./log(8.296*tan(ggrav(:,1)/2).^3-11.33*...
    tan(ggrav(:,1)/2).^2+26.11*tan(ggrav(:,1)/2)+1.336)));
%THETA_approx=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma_eq)))));
THETA_approx=abs(asin(ggrav(:,7)/(sgrav.*gamma_eq)));
K_theta_PRBM=(2.398+(0.5074./log(5.062*tan(ggrav(:,1)/2).^3-...
    5.485*tan(ggrav(:,1)/2).^2+24.11*tan(ggrav(:,1)/2)+1.18))+...
    (-0.248*mu^2-0.2399*mu+0.3882)/(1+exp(-11.21*mu^3+22.52*...
    mu^2-4.814*mu-1.258));
```

```
chi_D=zeros(length(sgrav),1);
chi_F=zeros(length(sgrav),1);
THETA_max=zeros(length(sgrav),1);
```

```
for i=1:length(sgrav)
    if ggrav(i,1)<=100*pi/180
        chi_D(i)=0.977+0.3589*(ggrav(i,1)/pi)+0.3*...
            (THETA_approx(i)/ggrav(i,1))-2.355*(ggrav(i,1)/pi)^2-...
            0.4842*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))-...
            2.522*(THETA_approx(i)/ggrav(i,1))^2+7.796*...
            (ggrav(i,1)/pi)^3-1.915*(ggrav(i,1)/pi)^2*...
            (THETA_approx(i)/ggrav(i,1))+7.867*(ggrav(i,1)/pi)*...
            (THETA_approx(i)/ggrav(i,1))^2+7.207*(THETA_approx(i)/...
            ggrav(i,1))^3-12.6*(ggrav(i,1)/pi)^4+7.218*...
            (ggrav(i,1)/pi)^3*(THETA_approx(i)/ggrav(i,1))-...
            10.76*(ggrav(i,1)/pi)^2*(THETA_approx(i)/ggrav(i,1))^2-...
            10.62*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))^3-...
            9.493*(THETA_approx(i)/ggrav(i,1))^4+7.87*...
            (ggrav(i,1)/pi)^5-7.536*(ggrav(i,1)/pi)^4*...
            (THETA_approx(i)/ggrav(i,1))+8.156*(ggrav(i,1)/pi)^3*...
            (THETA_approx(i)/ggrav(i,1))^2+1.761*(ggrav(i,1)/pi)^2*...
            (THETA_approx(i)/ggrav(i,1))^3+7.767*(ggrav(i,1)/pi)*...
```

```

    (THETA_approx(i)/ggrav(i,1))^4+4.517*...
    (THETA_approx(i)/ggrav(i,1))^5;
chi_F(i)=0.9572+0.5625*(ggrav(i,1)/pi)+0.4362*...
    (THETA_approx(i)/ggrav(i,1))-3.358*(ggrav(i,1)/pi)^2-...
    1.796*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))-...
    3.918*(THETA_approx(i)/ggrav(i,1))^2+10.6*...
    (ggrav(i,1)/pi)^3+0.6124*(ggrav(i,1)/pi)^2*...
    (THETA_approx(i)/ggrav(i,1))+13.34*(ggrav(i,1)/pi)*...
    (THETA_approx(i)/ggrav(i,1))^2+9.698*...
    (THETA_approx(i)/ggrav(i,1))^3-16.79*(ggrav(i,1)/pi)^4+...
    6.454*(ggrav(i,1)/pi)^3*(THETA_approx(i)/ggrav(i,1))-...
    18.3*(ggrav(i,1)/pi)^2*(THETA_approx(i)/ggrav(i,1))^2-...
    17.73*(ggrav(i,1)/pi)*(THETA_approx(i)/ggrav(i,1))^3-...
    11.79*(THETA_approx(i)/ggrav(i,1))^4+10.41*...
    (ggrav(i,1)/pi)^5-7.822*(ggrav(i,1)/pi)^4*...
    (THETA_approx(i)/ggrav(i,1))+10.28*(ggrav(i,1)/pi)^3*...
    (THETA_approx(i)/ggrav(i,1))^2+6.785*(ggrav(i,1)/pi)^2*...
    (THETA_approx(i)/ggrav(i,1))^3+10.8*(ggrav(i,1)/pi)*...
    (THETA_approx(i)/ggrav(i,1))^4+5.143*...
    (THETA_approx(i)/ggrav(i,1))^5;
    THETA_max(i)=0.75*ggrav(i,1);
elseif ggrav(i,1)>100*pi/180
chi_D(i)=2.658-12.66*(ggrav(i,1)/pi)+5.379*...
    (THETA_approx(i)/pi)+38.49*(ggrav(i,1)/pi)^2-32.82*...
    (ggrav(i,1)/pi)*(THETA_approx(i)/pi)+9.967*...
    (THETA_approx(i)/pi)^2-58.19*(ggrav(i,1)/pi)^3+76.2*...
    (ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)-49.82*...
    (ggrav(i,1)/pi)*(THETA_approx(i)/pi)^2+16.61*...
    (THETA_approx(i)/pi)^3+43.74*(ggrav(i,1)/pi)^4-79.15*...
    (ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)+87.3*...
    (ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^2-79.33*...
    (ggrav(i,1)/pi)*(THETA_approx(i)/pi)^3+42.31*...
    (THETA_approx(i)/pi)^4-13.07*(ggrav(i,1)/pi)^5+29.61*...
    (ggrav(i,1)/pi)^4*(THETA_approx(i)/pi)-46.27*...
    (ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)^2+67.32*...
    (ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^3-54.47*...
    (ggrav(i,1)/pi)*(THETA_approx(i)/pi)^4+4.942*...
    (THETA_approx(i)/pi)^5;
chi_F(i)=1.608-4.706*(ggrav(i,1)/pi)+2.225*...
    (THETA_approx(i)/pi)+14.46*(ggrav(i,1)/pi)^2-15.06*...
    (ggrav(i,1)/pi)*(THETA_approx(i)/pi)+2.722*...
    (THETA_approx(i)/pi)^2-22.11*(ggrav(i,1)/pi)^3+38.49*...
    (ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)-25.48*...
    (ggrav(i,1)/pi)*(THETA_approx(i)/pi)^2+11.91*...
    (THETA_approx(i)/pi)^3+16.86*(ggrav(i,1)/pi)^4-44.84*...
    (ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)+68.46*...

```

```

(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^2-94.92*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^3+62.37*...
(THETA_approx(i)/pi)^4-5.134*(ggrav(i,1)/pi)^5+19.9*...
(ggrav(i,1)/pi)^4*(THETA_approx(i)/pi)-54.01*...
(ggrav(i,1)/pi)^3*(THETA_approx(i)/pi)^2+116.2*...
(ggrav(i,1)/pi)^2*(THETA_approx(i)/pi)^3-119.8*...
(ggrav(i,1)/pi)*(THETA_approx(i)/pi)^4+32.37*...
(THETA_approx(i)/pi)^5;
THETA_max(i)=75*pi/180;
end
end

calc_s_approx=zeros(length(sgrav),1);
syms s
if mu~=1 && mu~=0
for i=1:length(sgrav)
sinfun=sin(ggrav(i)-THETA_approx(i));
eqn_s_approx=[num2str(K_theta_PRBM(i)) '*' ...
num2str(THETA_approx(i)) '/' num2str(sinfun) '-' ...
num2str(mu) '^2*' num2str(chi_F(i)) '*s^2-0.4*(1-' ...
num2str(mu) '^3*' num2str(chi_D(i)) '*s^3=0'];
s_sol=double(solve(eqn_s_approx,s));
for j=1:length(s_sol)
if s_sol(j)>0 && s_sol(j)<sgrav(i)+1
calc_s_approx(i)=s_sol(j);
break
end
end
end
end
end

K_theta=((mu^2*sgrav.^2)+((1-mu)^3*sgrav.^3)).*sin(ggrav(:,1)-...
THETA_approx)./THETA_approx;
K_theta_eq_approx=(mu^2*sgrav.^2.*chi_F.*sin(ggrav(:,1)-...
THETA_approx)./THETA_approx) + (0.4*(1-mu)^3*sgrav.^3.*...
chi_D.*sin(ggrav(:,1)-THETA_approx)./THETA_approx);

nval=-1./tan(ggrav(:,1));
K_theta_H=zeros(length(sgrav),1);
gamma_H=zeros(length(sgrav),1);

for i=1:length(sgrav)
if nval(i)>-5 && nval(i)<=-2.5
K_theta_H(i)=3.024112+0.121290*nval(i)+0.003169*nval(i)^2;
elseif nval(i)>-2.5 && nval(i)<=-1

```



```

    K_theta_H(i)=1.967647-2.616021*nval(i)-3.738166*nval(i)^2-...
        2.649437*nval(i)^3-0.891906*nval(i)^4-0.113063*nval(i)^5;
elseif nval(i)>-1 && nval(i)<=10
    K_theta_H(i)=2.654855-0.509896*10^(-1)*nval(i)+0.126749*...
        10^(-1)*nval(i)^2-0.142039*10^(-2)*nval(i)^3+0.584525*...
        10^(-4)*nval(i)^4;
else
    K_theta_H(i)=NaN;
end
end

for i=1:length(sgrav)
    if nval(i)>0.5 && nval(i)<10
        gamma_H(i)=0.841655-0.0067807*nval(i)+0.000438*nval(i)^2;
    elseif nval(i)>-1.8316 && nval(i)<0.5
        gamma_H(i)=0.852144-0.0182867*nval(i);
    elseif nval(i)>-5 && nval(i)<-1.8316
        gamma_H(i)=0.912364+0.0145928*nval(i);
    else
        gamma_H(i)=NaN;
    end
end

%THETA_H=abs(atan2(ggrav(:,7),(ggrav(:,6)-(sgrav.*(1-gamma_H)))));
THETA_H=abs(asin(ggrav(:,7)./(sgrav.*gamma_H)));
g=g+1;

if mu==1
    calc_s_approx=(K_theta_PRBM.*THETA_approx./(chi_F.*...
        sin(ggrav(:,1)-THETA_approx))).^(1/2);
    calc_s=(K_theta_PRBM.*THETA_approx./sin(ggrav(:,1)-...
        THETA_approx)).^(1/2);
    calc_s_H=(K_theta_H.*THETA_H./sin(ggrav(:,1)-THETA_H)).^(1/2);
    error_s_approx=((calc_s_approx./sgrav)-1)*100;
    error_s=((calc_s./sgrav)-1)*100;
    error_s_H=((calc_s_H./sgrav)-1)*100;
    mat_calc_s_approx(1:length(sgrav),g,h)=calc_s_approx;
    mat_calc_s(1:length(sgrav),g,h)=calc_s;
    mat_calc_s_H(1:length(sgrav),g,h)=calc_s_H;
    mat_error_s_approx(1:length(sgrav),g,h)=error_s_approx;
    mat_error_s(1:length(sgrav),g,h)=error_s;
    mat_error_s_H(1:length(sgrav),g,h)=error_s_H;
elseif mu==0
    calc_s_approx=(K_theta_PRBM.*THETA_approx./(0.4*chi_D.*...
        sin(ggrav(:,1)-THETA_approx))).^(1/3);
    calc_s=(K_theta_PRBM.*THETA_approx./(0.4*sin(ggrav(:,1)-...

```

```

    THETA_approx)).^(1/3);
    error_s_approx=((calc_s_approx./sgrav)-1)*100;
    error_s=((calc_s./sgrav)-1)*100;
    mat_calc_s_approx(1:length(sgrav),g,h)=calc_s_approx;
    mat_calc_s(1:length(sgrav),g,h)=calc_s;
    mat_error_s_approx(1:length(sgrav),g,h)=error_s_approx;
    mat_error_s(1:length(sgrav),g,h)=error_s;
elseif mu~=1 && mu~=0
    error_s_approx=((calc_s_approx./sgrav)-1)*100;
    mat_calc_s_approx(1:length(sgrav),g,h)=calc_s_approx;
    mat_error_s_approx(1:length(sgrav),g,h)=error_s_approx;
end

mat_K_theta_PRBM(1:length(sgrav),g,h)=K_theta_PRBM;
mat_K_theta_H(1:length(sgrav),g,h)=K_theta_H;
mat_phi(1:length(sgrav),g,h)=ggrav(:,1);
mat_s(1:length(sgrav),g,h)=ggrav(:,3);
mat_g(1:length(sgrav),g,h)=gamma;
mat_THETA(1:length(sgrav),g,h)=THETA;
mat_THETA_max(1:length(sgrav),g,h)=THETA_max;
mat_THETA_approx(1:length(sgrav),g,h)=THETA_approx;
mat_K_theta(1:length(sgrav),g,h)=K_theta;
mat_K_theta_eq_approx(1:length(sgrav),g,h)=K_theta_eq_approx;
mat_gamma_eq(1:length(sgrav),g,h)=gamma_eq;

muvec(h)=mu;
end
end

mat_phi_H=mat_phi;
mat_THETA_approx_H=mat_THETA_approx;
THETA_pnts_H=[7.9 9.8 12.9 18.6 23.6 31.5 44.4 58.5 64.1 67.5 65.8 ...
    69 64.6 66.4 67.5 69 69.7]*pi/180;
phi_pnts_H=[11.3 14 18.4 26.6 33.7 45 63.4 90 116.6 135 146.3 153.4 ...
    161.6 166 168.7 172.4 174.3]*pi/180;
[rownum colnum stacknum]=size(mat_phi);

for z=1:stacknum
    for x=1:rownum
        for y=1:colnum
            if mat_phi(x,y,z)==0
                mat_phi(x,y,z)=NaN;
            end
            if mat_s(x,y,z)==0
                mat_s(x,y,z)=NaN;
            end
        end
    end
end

```

```

if mat_g(x,y,z)==0
    mat_g(x,y,z)=NaN;
end
if mat_THETA(x,y,z)==0
    mat_THETA(x,y,z)=NaN;
end
if isreal(mat_THETA_approx(x,y,z))==0
    mat_THETA_approx(x,y,z)=NaN;
    mat_K_theta(x,y,z)=NaN;
    mat_THETA(x,y,z)=NaN;
    mat_g(x,y,z)=NaN;
    mat_s(x,y,z)=NaN;
    mat_phi(x,y,z)=NaN;
    mat_K_theta_eq_approx(x,y,z)=NaN;
    mat_gamma_eq(x,y,z)=NaN;
    mat_calc_s_approx(x,y,z)=NaN;
    mat_calc_s(x,y,z)=NaN;
    mat_calc_s_H(x,y,z)=NaN;
    mat_error_s_H(x,y,z)=NaN;
    mat_error_s_approx(x,y,z)=NaN;
    mat_error_s(x,y,z)=NaN;
    mat_THETA_max(x,y,z)=NaN;
    mat_K_theta_H(x,y,z)=NaN;
    mat_K_theta_PRBM(x,y,z)=NaN;
end
if isnan(mat_THETA_approx(x,y,z))==0 &&
    mat_THETA_approx(x,y,z)>mat_THETA_max(x,y,z)

    mat_THETA_approx(x,y,z)=NaN;
    mat_K_theta(x,y,z)=NaN;
    mat_THETA(x,y,z)=NaN;
    mat_g(x,y,z)=NaN;
    mat_s(x,y,z)=NaN;
    mat_phi(x,y,z)=NaN;
    mat_K_theta_eq_approx(x,y,z)=NaN;
    mat_gamma_eq(x,y,z)=NaN;
    mat_calc_s_approx(x,y,z)=NaN;
    mat_calc_s(x,y,z)=NaN;
    mat_calc_s_H(x,y,z)=NaN;
    mat_error_s_H(x,y,z)=NaN;
    mat_error_s_approx(x,y,z)=NaN;
    mat_error_s(x,y,z)=NaN;
    mat_THETA_max(x,y,z)=NaN;
    mat_K_theta_H(x,y,z)=NaN;
    mat_K_theta_PRBM(x,y,z)=NaN;
end

```

```

if isnan(mat_phi(x,y,z))==0 && (mat_phi(x,y,z)*180/pi<20 ||
mat_phi(x,y,z)*180/pi>170)

mat_THETA_approx(x,y,z)=NaN;
mat_K_theta(x,y,z)=NaN;
mat_THETA(x,y,z)=NaN;
mat_g(x,y,z)=NaN;
mat_s(x,y,z)=NaN;
mat_phi(x,y,z)=NaN;
mat_K_theta_eq_approx(x,y,z)=NaN;
mat_gamma_eq(x,y,z)=NaN;
mat_calc_s_approx(x,y,z)=NaN;
mat_calc_s(x,y,z)=NaN;
mat_calc_s_H(x,y,z)=NaN;
mat_error_s_H(x,y,z)=NaN;
mat_error_s_approx(x,y,z)=NaN;
mat_error_s(x,y,z)=NaN;
mat_THETA_max(x,y,z)=NaN;
mat_K_theta_H(x,y,z)=NaN;
mat_K_theta_PRBM(x,y,z)=NaN;
end
if abs(mat_error_s(x,y,z))>15 || abs(mat_error_s_H(x,y,z))>15 ||
abs(mat_error_s_approx(x,y,z))>15

mat_THETA_approx(x,y,z)=NaN;
mat_K_theta(x,y,z)=NaN;
mat_THETA(x,y,z)=NaN;
mat_g(x,y,z)=NaN;
mat_s(x,y,z)=NaN;
mat_phi(x,y,z)=NaN;
mat_K_theta_eq_approx(x,y,z)=NaN;
mat_gamma_eq(x,y,z)=NaN;
mat_calc_s_approx(x,y,z)=NaN;
mat_calc_s(x,y,z)=NaN;
mat_calc_s_H(x,y,z)=NaN;
mat_error_s_H(x,y,z)=NaN;
mat_error_s_approx(x,y,z)=NaN;
mat_error_s(x,y,z)=NaN;
mat_THETA_max(x,y,z)=NaN;
mat_K_theta_H(x,y,z)=NaN;
mat_K_theta_PRBM(x,y,z)=NaN;
end
if mat_THETA_approx_H(x,y,z)>interp1(phi_pnts_H,...
THETA_pnts_H,mat_phi_H(x,y,z))

mat_THETA_approx_H(x,y,z)=NaN;

```

```

        mat_phi_H(x,y,z)=NaN;
        mat_error_s_H(x,y,z)=NaN;
    end
end
end
end

figure(1)
hold on
surf(mat_phi(:,:,h)*180/pi,mat_THETA_approx(:,:,h)*180/pi,mat_error_s_approx(:,:,h))
surf(mat_phi(:,:,h)*180/pi,mat_THETA_approx(:,:,h)*180/pi,mat_error_s(:,:,h))
hold off
shading flat
grid on
title('Error surfaces in s for the Force Only Case')
xlabel('\phi (degrees)')
ylabel('\Theta_{approx} (degrees)')
zlabel('Relative percent error')
legend('Load index with correction','Load index alone')

figure(2)
hold on
surf(mat_phi(:,:,1)*180/pi,mat_THETA_approx(:,:,1)*180/pi,mat_error_s_approx(:,:,1))
surf(mat_phi(:,:,1)*180/pi,mat_THETA_approx(:,:,1)*180/pi,mat_error_s(:,:,1))
hold off
shading flat
grid on
title('Error surfaces in s for Distributed Load Only Case')
xlabel('\phi (degrees)')
ylabel('\Theta_{approx} (degrees)')
zlabel('Relative percent error')
legend('Load index with correction','Load index alone')

```

B.19 Example Problem

```

clc
clear all
close all
clf

%% Inputs and initializations

load_angle_deg=90;
E_exact=200*10^9;
I_exact=5.4752*10^-13;
F_exact=0.24;

```

```

w_exact=1.062355767;
L_exact=0.44174;
lambda1=(E_exact*I_exact/F_exact)^(1/2);
lambda2=(E_exact*I_exact/w_exact)^(1/3);
lambda3=(lambda1*lambda2)/(lambda1+lambda2);
mu_exact1=lambda2/(lambda1+lambda2)
b=0.10363;
phi=load_angle_deg*pi/180;

%% Code for mu=0.4097, phi=90
syms mu

gamma_eq=(0.7987+(0.1758/log(5.647*tan(phi/2)^3-5.774*tan(phi/2)^2+26.31*...
tan(phi/2)+1.1)))+((1.272/(1+exp(9.889*mu^2-0.4737*mu-1.296)))+...
0.0003257)*(0.04462+(-0.01106/log(8.296*tan(phi/2)^3-11.33*...
tan(phi/2)^2+26.11*tan(phi/2)+1.336)));
THETA=asin(b/(L_exact*gamma_eq));
K_theta=(2.398+(0.5074/log(5.062*tan(phi/2)^3-5.485*tan(phi/2)^2+24.11*...
tan(phi/2)+1.18)))+(-0.248*mu^2-0.2399*mu+0.3882)/(1+exp(-11.21*mu^3+...
22.52*mu^2-4.814*mu-1.258));

if load_angle_deg<=100
chi_D=0.977+0.3589*(phi/pi)+0.3*(THETA/phi)-2.355*(phi/pi)^2-0.4842*...
(phi/pi)*(THETA/phi)-2.522*(THETA/phi)^2+7.796*(phi/pi)^3-1.915*...
(phi/pi)^2*(THETA/phi)+7.867*(phi/pi)*(THETA/phi)^2+7.207*...
(THETA/phi)^3-12.6*(phi/pi)^4+7.218*(phi/pi)^3*(THETA/phi)-10.76*...
(phi/pi)^2*(THETA/phi)^2-10.62*(phi/pi)*(THETA/phi)^3-9.493*...
(THETA/phi)^4+7.87*(phi/pi)^5-7.536*(phi/pi)^4*(THETA/phi)+8.156*...
(phi/pi)^3*(THETA/phi)^2+1.761*(phi/pi)^2*(THETA/phi)^3+7.767*...
(phi/pi)*(THETA/phi)^4+4.517*(THETA/phi)^5;
chi_F=0.9572+0.5625*(phi/pi)+0.4362*(THETA/phi)-3.358*(phi/pi)^2-1.796*...
(phi/pi)*(THETA/phi)-3.918*(THETA/phi)^2+10.6*(phi/pi)^3+0.6124*...
(phi/pi)^2*(THETA/phi)+13.34*(phi/pi)*(THETA/phi)^2+9.698*...
(THETA/phi)^3-16.79*(phi/pi)^4+6.454*(phi/pi)^3*(THETA/phi)-18.3*...
(phi/pi)^2*(THETA/phi)^2-17.73*(phi/pi)*(THETA/phi)^3-11.79*...
(THETA/phi)^4+10.41*(phi/pi)^5-7.822*(phi/pi)^4*(THETA/phi)+10.28*...
(phi/pi)^3*(THETA/phi)^2+6.785*(phi/pi)^2*(THETA/phi)^3+10.8*...
(phi/pi)*(THETA/phi)^4+5.143*(THETA/phi)^5;
elseif load_angle_deg>100
chi_D=2.658-12.66*(phi/pi)+5.379*(THETA/phi)+38.49*(phi/pi)^2-32.82*...
(phi/pi)*(THETA/phi)+9.967*(THETA/phi)^2-58.19*(phi/pi)^3+76.2*...
(phi/pi)^2*(THETA/phi)-49.82*(phi/pi)*(THETA/phi)^2+16.61*...
(THETA/phi)^3+43.74*(phi/pi)^4-79.15*(phi/pi)^3*(THETA/phi)+87.3*...
(phi/pi)^2*(THETA/phi)^2-79.33*(phi/pi)*(THETA/phi)^3+42.31*...
(THETA/phi)^4-13.07*(phi/pi)^5+29.61*(phi/pi)^4*(THETA/phi)-46.27*...
(phi/pi)^3*(THETA/phi)^2+67.32*(phi/pi)^2*(THETA/phi)^3-54.47*...

```

```

(phi/pi)*(THETA/pi)^4+4.942*(THETA/pi)^5;
chi_F=1.608-4.706*(phi/pi)+2.225*(THETA/pi)+14.46*(phi/pi)^2-15.06*...
(phi/pi)*(THETA/pi)+2.722*(THETA/pi)^2-22.11*(phi/pi)^3+38.49*...
(phi/pi)^2*(THETA/pi)-25.48*(phi/pi)*(THETA/pi)^2+11.91*...
(THETA/pi)^3+16.86*(phi/pi)^4-44.84*(phi/pi)^3*(THETA/pi)+68.46*...
(phi/pi)^2*(THETA/pi)^2-94.92*(phi/pi)*(THETA/pi)^3+62.37*...
(THETA/pi)^4-5.134*(phi/pi)^5+19.9*(phi/pi)^4*(THETA/pi)-54.01*...
(phi/pi)^3*(THETA/pi)^2+116.2*(phi/pi)^2*(THETA/pi)^3-119.8*...
(phi/pi)*(THETA/pi)^4+32.37*(THETA/pi)^5;
end

eqn_w_unk=(K_theta*THETA/sin(phi-THETA)) - ((L_exact/lambda1)^2*chi_F) - ...
(0.4*(L_exact/lambda1)^3*chi_D*(1/mu-1)^3);
eqn_f_unk=(K_theta*THETA/sin(phi-THETA)) - ((L_exact/lambda2)^2*chi_F*...
(mu/(1-mu))^2) - (0.4*(L_exact/lambda2)^3*chi_D);
muvec=0.01:0.0001:0.99;
eqwvec=subs(eqn_w_unk,mu,muvec);
eqfvec=subs(eqn_f_unk,mu,muvec);

mu_w_approx1=interp1(eqwvec,muvec,0)
mu_f_approx1=interp1(eqfvec,muvec,0)
w_approx1=E_exact*I_exact*(1/mu_w_approx1-1)^3/lambda1^3
f_approx1=E_exact*I_exact/(lambda2*(1/mu_f_approx1-1))^2
error_w1=(w_approx1-w_exact)/w_exact*100
error_f1=(f_approx1-F_exact)/F_exact*100
error_mu_w1=(mu_w_approx1-mu_exact1)/mu_exact1*100
error_mu_f1=(mu_f_approx1-mu_exact1)/mu_exact1*100

figure(1)
hold on
plot(muvec,eqwvec,'k')
plot(muvec,eqfvec,'k-.')
plot(mu_exact1,0,'r*')
hold off
grid on
title('Plot of PRBM equation of \mu for an intermediate case')
xlabel('\mu values')
ylabel('PRBM equation of \mu')
legend('w unknown solution','F unknown solution','Exact value of \mu')
axis([0 1 -1 1])

```

B.20 Error Analysis for ANSYS Results of Intermediate μ Value

```

clc
clear all
close all

```

clf

%% Constants (combined loading)

```
E=200*10^9;
I=5.4752*10^-13;
F=0.24;
w=1.062355767;
lambda1=(E*I/F)^(1/2);
lambda2=(E*I/w)^(1/3);
lambda3=(lambda1*lambda2)/(lambda1+lambda2);
magratio=lambda2/(lambda1+lambda2);
mu_exact1=lambda2/(lambda1+lambda2);
```

%% Ansys Data (combined loading)

```
phi_deg=(165:-15:15)*(pi/180);
dataloop_max=length(phi_deg);
```

```
%data entry
% for hide=1:1
%end data entry
```

%% Data Manipulation

```
options=odeset('MaxStep',0.01);
```

```
for a=1:dataloop_max
```

```
eval(['phi_meas' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
eval(['K_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
eval(['S_tilde' num2str(a) '=zeros(1,length(xdisp' num2str(a) ')'-1);'])
eval(['xvec_length(a)=length(xdisp' num2str(a) ')'-1;'])
eval(['Mvec_length(a)=length(Mvec' num2str(a) ');'])
```

```
for j=2:xvec_length(a)
```

```
eval(['tempx=(svec(j+1) + xdisp' num2str(a) '(j+1)) - (svec(j-1) + ...
xdisp' num2str(a) '(j-1));'])
eval(['tempy=ydisp' num2str(a) '(j+1)-ydisp' num2str(a) '(j-1);'])
tempmag=sqrt(tempx^2+tempy^2);
tempalpha=abs(acos(tempx/tempmag));
eval(['phi_meas' num2str(a) '(j)=phi_deg(a)-tempalpha;'])
eval(['S_tilde' num2str(a) '(j)=svec(length(svec)+1-j)/lambda3;'])
eval(['temp_coor=[cos(tempalpha) sin(tempalpha);-sin(tempalpha) ...
cos(tempalpha)]*[abs((svec(length(svec))+xdisp' num2str(a) ...
'(length(svec)))-(svec(j)+xdisp' num2str(a) '(j)));abs(ydisp' ...
num2str(a) '(length(svec))-ydisp' num2str(a) '(j));'])
```



```

    bmat(a,j)=temp_coor(2);
end
for n=1:2:Mvec_length(a)
    eval(['K_tilde' num2str(a) '((n+1)/2)=(lambda3*-Mvec' num2str(a)...
        '(n)) / (E*I);'])
end

eval(['test(a)=phi_meas' num2str(a) '(j);'])
eval(['PM=phi_meas' num2str(a) ';'])
eval(['KT=K_tilde' num2str(a) ';'])
eval(['ST=S_tilde' num2str(a) ';'])
PM(1)=[];
KT(1)=[];
ST(1)=[];
phimat(a,1:length(PM))=PM;
kmat(a,1:length(KT))=KT;
smat(a,1:length(ST))=ST*lambda3;
[sgrav,ggrav]=ode45('beam_all_cases',[0:0.01:svec(69)/lambda3],...
    [test(a) 0 0 magratio test(a) 0 0],options);

figure(1)
hold on
plot3(ggrav(:,1)*180/pi,ggrav(:,2),ggrav(:,3),'b-','Linewidth',2)
plot3(PM*180/pi,KT,ST,'r.')
hold off

end

figure(1)
grid on
xlabel('\phi')
ylabel('\kappa\lambda_3')
zlabel('s/\lambda_3')
legend('Model','Ansys Data')
title(['Phase Space Comparisons Between ANSYS and ODE45 (\mu = ' num2str(magratio) ')'])

bmat(:,1)=[];
[rowb,colb]=size(bmat);
mu_w_approx1=zeros(rowb,colb);
mu_f_approx1=zeros(rowb,colb);
w_approx1=zeros(rowb,colb);
f_approx1=zeros(rowb,colb);
error_w1=zeros(rowb,colb);
error_f1=zeros(rowb,colb);
error_mu_w1=zeros(rowb,colb);
error_mu_f1=zeros(rowb,colb);

```

syms mu

for i=1:rowb

for j=1:colb

```
gamma_eq=(0.7987+(0.1758/log(5.647*tan(phimat(i,j)/2)^3-5.774*...
tan(phimat(i,j)/2)^2+26.31*tan(phimat(i,j)/2+1.1)))+...
((1.272/(1+exp(9.889*mu^2-0.4737*mu-1.296)))+0.0003257)*...
(0.04462+(-0.01106/log(8.296*tan(phimat(i,j)/2)^3-11.33*...
tan(phimat(i,j)/2)^2+26.11*tan(phimat(i,j)/2+1.336)));
THETA=asin(bmat(i,j)/(smat(i,j)*gamma_eq));
K_theta=(2.398+(0.5074/log(5.062*tan(phimat(i,j)/2)^3-5.485*...
tan(phimat(i,j)/2)^2+24.11*tan(phimat(i,j)/2+1.18)))+...
(-0.248*mu^2-0.2399*mu+0.3882)/(1+exp(-11.21*mu^3+22.52*...
mu^2-4.814*mu-1.258));
```

if phimat(i,j)*180/pi<=100

```
chi_D=0.977+0.3589*(phimat(i,j)/pi)+0.3*(THETA/phimat(i,j))-...
2.355*(phimat(i,j)/pi)^2-0.4842*(phimat(i,j)/pi)*...
(THETA/phimat(i,j))-2.522*(THETA/phimat(i,j))^2+7.796*...
(phimat(i,j)/pi)^3-1.915*(phimat(i,j)/pi)^2*...
(THETA/phimat(i,j))+7.867*(phimat(i,j)/pi)*...
(THETA/phimat(i,j))^2+7.207*(THETA/phimat(i,j))^3-12.6*...
(phimat(i,j)/pi)^4+7.218*(phimat(i,j)/pi)^3*...
(THETA/phimat(i,j))-10.76*(phimat(i,j)/pi)^2*...
(THETA/phimat(i,j))^2-10.62*(phimat(i,j)/pi)*...
(THETA/phimat(i,j))^3-9.493*(THETA/phimat(i,j))^4+7.87*...
(phimat(i,j)/pi)^5-7.536*(phimat(i,j)/pi)^4*...
(THETA/phimat(i,j))+8.156*(phimat(i,j)/pi)^3*...
(THETA/phimat(i,j))^2+1.761*(phimat(i,j)/pi)^2*...
(THETA/phimat(i,j))^3+7.767*(phimat(i,j)/pi)*...
(THETA/phimat(i,j))^4+4.517*(THETA/phimat(i,j))^5;
chi_F=0.9572+0.5625*(phimat(i,j)/pi)+0.4362*(THETA/phimat(i,j))-...
3.358*(phimat(i,j)/pi)^2-1.796*(phimat(i,j)/pi)*...
(THETA/phimat(i,j))-3.918*(THETA/phimat(i,j))^2+10.6*...
(phimat(i,j)/pi)^3+0.6124*(phimat(i,j)/pi)^2*...
(THETA/phimat(i,j))+13.34*(phimat(i,j)/pi)*...
(THETA/phimat(i,j))^2+9.698*(THETA/phimat(i,j))^3-16.79*...
(phimat(i,j)/pi)^4+6.454*(phimat(i,j)/pi)^3*...
(THETA/phimat(i,j))-18.3*(phimat(i,j)/pi)^2*...
(THETA/phimat(i,j))^2-17.73*(phimat(i,j)/pi)*...
(THETA/phimat(i,j))^3-11.79*(THETA/phimat(i,j))^4+10.41*...
(phimat(i,j)/pi)^5-7.822*(phimat(i,j)/pi)^4*...
(THETA/phimat(i,j))+10.28*(phimat(i,j)/pi)^3*...
(THETA/phimat(i,j))^2+6.785*(phimat(i,j)/pi)^2*...
(THETA/phimat(i,j))^3+10.8*(phimat(i,j)/pi)*...
```

```

        (THETA/phimat(i,j))^4+5.143*(THETA/phimat(i,j))^5;
elseif phimat(i,j)*180/pi>100
    chi_D=2.658-12.66*(phimat(i,j)/pi)+5.379*(THETA/pi)+38.49*...
        (phimat(i,j)/pi)^2-32.82*(phimat(i,j)/pi)*(THETA/pi)+9.967*...
        (THETA/pi)^2-58.19*(phimat(i,j)/pi)^3+76.2*...
        (phimat(i,j)/pi)^2*(THETA/pi)-49.82*(phimat(i,j)/pi)*...
        (THETA/pi)^2+16.61*(THETA/pi)^3+43.74*(phimat(i,j)/pi)^4-...
        79.15*(phimat(i,j)/pi)^3*(THETA/pi)+87.3*(phimat(i,j)/pi)^2*...
        (THETA/pi)^2-79.33*(phimat(i,j)/pi)*(THETA/pi)^3+42.31*...
        (THETA/pi)^4-13.07*(phimat(i,j)/pi)^5+29.61*...
        (phimat(i,j)/pi)^4*(THETA/pi)-46.27*(phimat(i,j)/pi)^3*...
        (THETA/pi)^2+67.32*(phimat(i,j)/pi)^2*(THETA/pi)^3-54.47*...
        (phimat(i,j)/pi)*(THETA/pi)^4+4.942*(THETA/pi)^5;
    chi_F=1.608-4.706*(phimat(i,j)/pi)+2.225*(THETA/pi)+14.46*...
        (phimat(i,j)/pi)^2-15.06*(phimat(i,j)/pi)*(THETA/pi)+2.722*...
        (THETA/pi)^2-22.11*(phimat(i,j)/pi)^3+38.49*...
        (phimat(i,j)/pi)^2*(THETA/pi)-25.48*(phimat(i,j)/pi)*...
        (THETA/pi)^2+11.91*(THETA/pi)^3+16.86*(phimat(i,j)/pi)^4-...
        44.84*(phimat(i,j)/pi)^3*(THETA/pi)+68.46*(phimat(i,j)/pi)^2*...
        (THETA/pi)^2-94.92*(phimat(i,j)/pi)*(THETA/pi)^3+62.37*...
        (THETA/pi)^4-5.134*(phimat(i,j)/pi)^5+19.9*...
        (phimat(i,j)/pi)^4*(THETA/pi)-54.01*(phimat(i,j)/pi)^3*...
        (THETA/pi)^2+116.2*(phimat(i,j)/pi)^2*(THETA/pi)^3-119.8*...
        (phimat(i,j)/pi)*(THETA/pi)^4+32.37*(THETA/pi)^5;
end

```

```

eqn_w_unk=(K_theta*THETA/sin(phimat(i,j)-THETA)) - ...
    ((smat(i,j)/lambda1)^2*chi_F) - ...
    (0.4*(smat(i,j)/lambda1)^3*chi_D*(1/mu-1)^3);
eqn_f_unk=(K_theta*THETA/sin(phimat(i,j)-THETA)) - ...
    ((smat(i,j)/lambda2)^2*chi_F*(mu/(1-mu))^2) - ...
    (0.4*(smat(i,j)/lambda2)^3*chi_D);
muvec=0:0.001:1;
eqwvec=subs(eqn_w_unk,mu,muvec);
eqfvec=subs(eqn_f_unk,mu,muvec);

mu_w_approx1(i,j)=interp1(eqwvec,muvec,0);
mu_f_approx1(i,j)=interp1(eqfvec,muvec,0);
w_approx1(i,j)=E*I*(1/mu_w_approx1(i,j)-1)^3/lambda1^3;
f_approx1(i,j)=E*I/(lambda2*(1/mu_f_approx1(i,j)-1))^2;
error_w1(i,j)=(w_approx1(i,j)-w)/w*100;
error_f1(i,j)=(f_approx1(i,j)-F)/F*100;
error_mu_w1(i,j)=(mu_w_approx1(i,j)-mu_exact1)/mu_exact1*100;
error_mu_f1(i,j)=(mu_f_approx1(i,j)-mu_exact1)/mu_exact1*100;

```

```

end
end

```

```

for i=1:dataloop_max
[s,g]=ode45('beam_nograv',(0:0.01:20),[test(i) 0 0],options);
for mm=1:length(s)
if g(mm,1)>pi || g(mm,2)<0
g(mm:length(s),:)=[];
s(mm:length(s),:)=[];
break
end
end
figure(1)
hold on
plot3(g(:,1)*180/pi,g(:,2),g(:,3),'k-')
hold off
end

figure(2)
grid on
hold on
for i=1:rowb
for j=1:colb
plot3(phimat(i,j)*180/pi,kmat(i,j),error_w1(i,j),'b*')
end
end
hold off
title(['Error in predicted distributed load for \mu = ' num2str(magratio)])
xlabel('\phi (degrees)')
ylabel('\kappa\lambda_3')
zlabel('% Error in w')
zlim([-10 10])

figure(3)
grid on
hold on
for i=1:rowb
for j=1:colb
plot3(phimat(i,j)*180/pi,kmat(i,j),error_f1(i,j),'b*')
end
end
hold off
title(['Error in predicted endpoint force for \mu = ' num2str(magratio)])
xlabel('\phi (degrees)')
ylabel('\kappa\lambda_3')
zlabel('% Error in F')
zlim([-10 10])

```

```

figure(4)
grid on
hold on
for i=1:rowb
    for j=1:colb
        plot3(phimat(i,j)*180/pi,kmat(i,j),error_mu_w1(i,j),'k*')
    end
end
hold off
title(['Error in predicted \mu when w is unknown for \mu = ' num2str(magratio)])
xlabel('\phi (degrees)')
ylabel('\kappa\lambda_3')
zlabel('% Error in \mu')
zlim([-10 10])

figure(5)
grid on
hold on
for i=1:rowb
    for j=1:colb
        plot3(phimat(i,j)*180/pi,kmat(i,j),error_mu_f1(i,j),'r*')
    end
end
hold off
title(['Error in predicted \mu when F is unknown for \mu = ' num2str(magratio)])
xlabel('\phi (degrees)')
ylabel('\kappa\lambda_3')
zlabel('% Error in \mu')
zlim([-10 10])

```