

University of Denver

Digital Commons @ DU

Electronic Theses and Dissertations

Graduate Studies

2021

Development of Triangular and Tetrahedral Finite Elements for Solutions to Thermoelastic Instabilities Using Hotspotter

Cortney Samuel LeNeave
University of Denver

Follow this and additional works at: <https://digitalcommons.du.edu/etd>



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

Recommended Citation

LeNeave, Cortney Samuel, "Development of Triangular and Tetrahedral Finite Elements for Solutions to Thermoelastic Instabilities Using Hotspotter" (2021). *Electronic Theses and Dissertations*. 1877.
<https://digitalcommons.du.edu/etd/1877>

This Thesis is brought to you for free and open access by the Graduate Studies at Digital Commons @ DU. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ DU. For more information, please contact jennifer.cox@du.edu, dig-commons@du.edu.

Development of Triangular and Tetrahedral Finite Elements for Solutions to Thermoelastic Instabilities Using Hotspotter

Abstract

The objective of the work presented in this thesis is to develop first-order triangular and tetrahedral elements for solutions to Thermoelastic Instabilities (TEI) regarding sliding friction systems in Hotspotter.

Hotspotter software uses a finite element method and an eigenvalue method and is an important tool because currently no other commercial software exists which solves the TEI problem for critical velocities and wave numbers of a system. Hotspotter currently uses quadrilateral and hexahedral elements for two and three dimensional analysis, respectively. Typically, tri and tet elements are used in industry when doing static and dynamic stress analysis. Therefore, the Hotspotter user is currently required to re-mesh the system using quad or hex based elements before importing the mesh into Hotspotter, a time consuming and dispensable process. Development of triangular and tetrahedral elements for TEI analysis will eliminate the re-meshing burden on the Hotspotter user without sacrificing accuracy of results.

Presented in this investigation is the mathematical development of such tri and tet elements. Then, verification of those elements by comparing trial cases against theoretical and Abaqus results. Finally, validation by incorporating the tri elements into Hotspotter and comparing to the quad elements. Results indicate accuracy within 1 percent of the legacy elements which have been validated against experimental data.

Document Type

Thesis

Degree Name

M.S.

Department

Mechanical Engineering

First Advisor

Yun-Bo Yi

Second Advisor

Peter Laz

Third Advisor

Mohammad Matin

Keywords

Eigenvalue method, Finite element analysis, Finite element method, Fourier element, Fourier reduction, Thermoelastic instabilities

Subject Categories

Aerospace Engineering | Engineering | Mechanical Engineering | Other Aerospace Engineering

Publication Statement

Copyright is held by the author. User is responsible for all copyright compliance.

This thesis is available at Digital Commons @ DU: <https://digitalcommons.du.edu/etd/1877>

DEVELOPMENT OF TRIANGULAR AND TETRAHEDRAL FINITE ELEMENTS
FOR SOLUTIONS TO THERMOELASTIC INSTABILITIES USING HOTSPOTTER

A Thesis

Presented to

the Faculty of the Daniel Felix Ritchie School of Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Cortney Samuel LeNeave

March 2021

Advisor: Dr. Yun-Bo Yi

©Copyright by Cortney Samuel LeNeave 2021

All Rights Reserved

Author: Cortney Samuel LeNeave

Title: DEVELOPMENT OF TRIANGULAR AND TETRAHEDRAL FINITE
ELEMENTS FOR SOLUTIONS TO THERMOELASTIC INSTABILITIES USING
HOTSPOTTER

Advisor: Dr. Yun-Bo Yi

Degree Date: March 2021

Abstract

The objective of the work presented in this thesis is to develop first-order triangular and tetrahedral elements for solutions to Thermoelastic Instabilities (TEI) regarding sliding friction systems in Hotspotter.

Hotspotter software uses a finite element method and an eigenvalue method and is an important tool because currently no other commercial software exists which solves the TEI problem for critical velocities and wave numbers of a system. Hotspotter currently uses quadrilateral and hexahedral elements for two and three dimensional analysis, respectively. Typically, tri and tet elements are used in industry when doing static and dynamic stress analysis. Therefore, the Hotspotter user is currently required to re-mesh the system using quad or hex based elements before importing the mesh into Hotspotter, a time consuming and dispensable process. Development of triangular and tetrahedral elements for TEI analysis will eliminate the re-meshing burden on the Hotspotter user without sacrificing accuracy of results.

Presented in this investigation is the mathematical development of such tri and tet elements. Then, verification of those elements by comparing trial cases against theoretical and Abaqus results. Finally, validation by incorporating the tri elements into Hotspotter and comparing to the quad elements. Results indicate accuracy within 1 percent of the legacy elements which have been validated against experimental data.

Acknowledgements

I would like to thank my advisor, Dr. Yun-Bo Yi, you have been a great source of knowledge, guidance, and inspiration. I would like to thank you for providing exceptional counsel during my master's studies and research. Your thoughtful advice and calmness fostered lasting peace during inherently stressful times. Thank you for providing me with support when needed and the freedoms to explore and develop individually as a researcher.

To my classmates who have helped me grow as a scholar and individual, thank you. Your devotion to curiosity, learning, integrity, and unwavering friendship have been imperative to my successes at the University of Denver.

Lastly to my family who have been there for me throughout the hurdles of my academic career. Thank you for always believing in me and providing encouragement. I couldn't have accomplished this milestone without your continuous support and understanding.

Table of Contents

CHAPTER 1 INTRODUCTION	1
1.1 Motivation.....	1
1.2 Objectives	5
CHAPTER 2 ELEMENT DEVELOPMENT FOR THERMAL AND STRESS ANALYSIS.....	8
2.1 FEA introduction	8
2.2 First-order triangular element	12
2.3 First-order tetrahedral element.....	29
CHAPTER 3 VERIFICATION & VALIDATION	39
3.1 Element verification.....	39
3.1.1 First-order triangular element	39
3.1.1.1 Verify element in bending only	39
3.1.1.2 Verify element by adding thermal load	47
3.1.1.2.1 Fixed-free beam (fixed in X direction only).....	47
3.1.1.2.2 Fixed-fixed beam (fixed in X direction only).....	51
3.1.1.2.3 Fixed-fixed beam (fixed in X & Y)	56
3.1.1.2.4 2D generalized disk	59
3.1.2 First-order tetrahedral element.....	66
3.1.2.1 Verify 3D generalized disk	66
3.2 Element validation against legacy Hotspotter elements	70
3.2.1 Triangular element validation	70
3.2.1.1 Quadrilateral pad and rotor model	70
3.2.1.2 Triangular pad and rotor model	72
3.2.1.3 Quad pad, tri rotor model.....	74
3.2.1.4 Tri pad, quad rotor model	76
3.2.1.5 Comparing all models	78
3.2.1.6 Validation mesh refinement and convergence study	81
3.2.2 First-order tetrahedral element.....	84
CHAPTER 4 CONCLUSION AND FUTURE WORK	85
4.1 Conclusion	85
4.2 Future work.....	86
References.....	89

List of Figures

Figure 1. A Boeing 787 aircraft brake system	1
Figure 2. A clutch pressure plate	2
Figure 3. Hotspotter classic.....	5
Figure 4. Hotspotter full 3D.....	5
Figure 5. A simple two bar FEA system.....	10
Figure 6. A visual Representation of von Mises stress.....	12
Figure 7. The Constant Strain Triangle (CST) element.....	13
Figure 8. Isoparametric mapping of the CST element.....	14
Figure 9. Isoparametric mapping of the CST element.....	15
Figure 10. Isoparametric mapping.....	17
Figure 11. A cantilever beam meshed using CST elements	25
Figure 12. The 4-noded tetrahedral (tet4) element	30
Figure 13. Isoparametric mapping.....	32
Figure 14. Initial geometry of a 2D beam.....	40
Figure 15. Initial (gray) and deformed (blue) geometry of a 2D beam	41
Figure 16. Abaqus volumetric integration rough mesh	42
Figure 17. Matlab volumetric integration rough mesh	42
Figure 18. Matlab numerical integration rough mesh.....	43
Figure 19. Abaqus volumetric integration fine mesh.....	45
Figure 20. Matlab numerical integration fine mesh.....	45
Figure 21. Initial geometry of a 2D beam fixed in only the x direction on the left end ...	48
Figure 22. Initial (gray) and deformed (blue) geometry of a 2D beam fixed in only the x direction on the left end	48
Figure 23. Abaqus fixed-free beam	49
Figure 24. Matlab fixed-free beam	49
Figure 25. Initial geometry of a 2D beam fixed in only the x direction on both ends	51
Figure 26. Initial (gray) and deformed (blue) geometry of a 2D beam, fixed in only the x direction on both ends.....	52
Figure 27. Abaqus fixed-fixed beam	53
Figure 28. Matlab fixed-fixed beam	53
Figure 29. Fixed-fixed beam node numbering.....	55
Figure 30. Initial (gray) and deformed (blue) geometry of a 2D beam fixed in both directions on both ends	56
Figure 31. Abaqus fixed u1 & v1 beam.....	57
Figure 32. Matlab fixed u1 & v1 beam.....	57
Figure 33. Element numbering and minimum in-plane principal strain values.....	59
Figure 34. Initial geometry of a 2D disk.....	60
Figure 35. Initial (gray) and deformed (blue) geometry of a 2D disk	61

Figure 36. Analytical results for fixed inner diameter disk	63
Figure 37. Abaqus fixed u1 & v1 inner diameter disk.....	63
Figure 38. Fixed u1 & v1 inner diameter Matlab disk.....	64
Figure 39. Initial (gray) and deformed (green) geometries.....	67
Figure 40. Abaqus analysis for a fixed u1 & v1 inner diameter disk	67
Figure 41. Matlab analysis for a fixed u1 & v1 inner diameter disk	68
Figure 42. Mesh of an axisymmetric disk pad and rotor model using all quadrilateral elements	71
Figure 43. First mode shape results of an axisymmetric disk pad and rotor model using all quadrilateral elements	71
Figure 44. Mesh of an axisymmetric disk pad and rotor model using all triangular elements	73
Figure 45. First mode shape results of an axisymmetric disk pad and rotor model using all triangular elements.....	73
Figure 46. Mesh of an axisymmetric disk pad and rotor model using quadrilateral and triangular elements.....	75
Figure 47. First mode shape results of an axisymmetric disk pad and rotor model using quadrilateral and triangular elements.....	75
Figure 48. Mesh of an axisymmetric disk pad and rotor model using triangular and quadrilateral elements	77
Figure 49. First mode shape results of an axisymmetric disk pad and rotor model using triangular and quadrilateral elements.....	77
Figure 50. Critical velocity vs. wave number	80
Figure 51. Refined mesh of an axisymmetric disk pad and rotor model using all quadrilateral elements	81
Figure 52. Refined mesh of an axisymmetric disk pad and rotor model using all triangular elements	81
Figure 53. Refined mesh of an axisymmetric disk pad and rotor model using quadrilateral and triangular elements	82
Figure 54. Refined mesh of an axisymmetric disk pad and rotor model using triangular and quadrilateral elements	82
Figure 55. Refined mesh critical velocity vs. wave number.....	83

List of Tables

Table 1. A single CST element stiffness matrix output from Abaqus.	23
Table 2. A single CST element stiffness matrix output from Matlab.	24
Table 3. Numerical vs. volumetric integration results	44
Table 4. Numerical vs. volumetric integration results	46
Table 5. Fixed-free results	50
Table 6. Fixed-fixed results	54
Table 7. Fixed u1 & v1 results.....	58
Table 8. Fixed u1 & v1 inner diameter results.....	65
Table 9. Results for a fixed u1 & v1 inner diameter three dimensional disk	69
Table 10. Wave number and critical speed results of an axisymmetric disk using all quadrilateral elements	72
Table 11. Wave number and critical speed results of an axisymmetric disk using all triangular elements.....	74
Table 12. Wave number and critical speed results of an axisymmetric disk using quadrilateral and triangular elements.....	76
Table 13. Wave number and critical speed results of an axisymmetric disk using triangular and quadrilateral elements.....	78
Table 14 Wave number and lowest critical speed results of all element types.....	80
Table 15. Mesh refinement results for wave number and lowest critical speed results of all element types.	83

CHAPTER 1 INTRODUCTION

1.1 Motivation

When designing a sliding friction system such as aircraft brakes or automotive clutches, the engineer must consider a great deal of design elements. An example of such a system is outlined in Figure 1 [1].

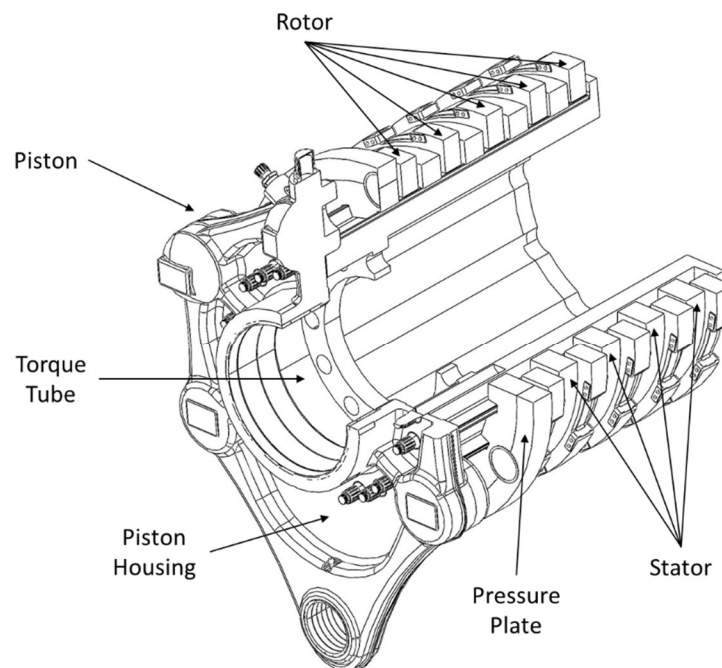


Figure 1. A Boeing 787 aircraft brake system uses many parts and involves complex geometries.

One design consideration regarding geometry and material selection is the phenomenon of thermoelastic instability (TEI) [2]. TEI is a form of thermal instability which regards two or more sliding bodies in frictional contact [3], [4]. When the bodies are contacting one another over a large area, small scale surface asperities cause a non-

uniform pressure distribution. This pressure distribution causes uneven friction which leads to uneven heating of each surface in contact. Further, thermal expansion of these parts cause non-uniform deformation and the uneven surfaces therefore hold different pressures, temperatures, and deform differently than the surrounding material [5]. Figure 2 displays antisymmetric hotspots on a clutch pressure plate from an experiment, showing support for TEI theory.

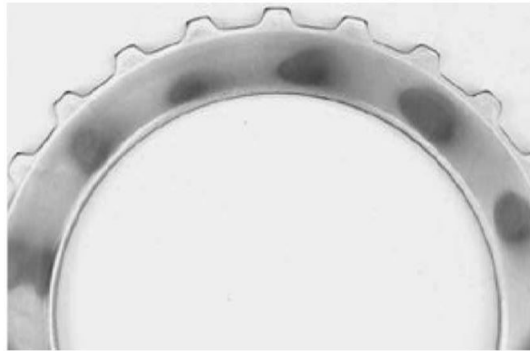


Figure 2. A clutch pressure plate after a single engagement reveals visual evidence of antisymmetric hotspots. Photo taken from [6].

“If the sliding speed is sufficiently high, the thermal mechanical feedback process is unstable, leading eventually to the localization of the load in a small region of the nominal contact area of the sliding surfaces” [3]. This localized thermal mechanical load can lead to increased vibration, localized material yielding, fatigue crack nucleation, and premature part failure which may not be accounted for using traditional failure theories [7], [8], [9].

The phenomenon was first rigorously investigated by Barber [5], however according to [10] it had been reportedly observed in railroad brakes and wheels even earlier than that. Since then, many analytical approaches to generalized cases of increasing

complexity have been investigated, from generalized half planes [11] to geometric length scales relevant for disk brake analysis [12]. However, limitations to the complexity of the problems arise with closed form analytical solutions leading to a time consuming and less than ideal method to understand if TEI could occur for any given design [13]. For example, an engineering designer may only consider a simplified geometry such as the diameter and thickness of the brake rotor when finding the analytical solution to the TEI problem, but when the final part is manufactured there may be intricacies unaccounted for in the analytical solution such as a rotor hat and cooling fins. Therefore, the analytical solution to the TEI problem is at best a time intensive approximation which suits a single simplified geometry.

An obvious alternative to the closed form analytical solution is to use the finite element method [14]. According to [15] “it is often necessary to obtain approximate numerical solutions for complex industrial problems, in which exact closed-form solutions are difficult to obtain”. Therefore, the use of the Finite Element Analysis (FEA) in either two or three dimensions to quickly analyze a complex design for vulnerabilities to TEI is a valid method. Currently no options exist to analyze TEI in the form of native or plug in applications for major FEA packages such as Abaqus or Ansys. There is a commercially available software package, Hotspotter, which evaluates brake, clutch, and other frictional sliding designs for susceptibility to thermoelastic instabilities.

Hotspotter uses an eigenvalue method to determine the growth rate of eigenmodes of the system for discrete sliding speeds. First, linear perturbations on the temperature field arising due to micro scale surface asperities are considered, at a discrete sliding speed,

which vary exponentially with time. In the governing equations and boundary conditions for thermoelasticity and heat-conduction, time cancels out leading to a complex linear eigenvalue problem for b , the growth rate, for discrete values of n , the wave number, and the sliding speed, V [6], [16]. The eigenvalue has both real and imaginary parts. The real part corresponds to the growth rate of the perturbation, and the imaginary part corresponds to the migration speed of the perturbation [14]. The problem is simplified further by the assumption that a real eigenvalue corresponds to instability because an eigenvalue of zero governs the stability boundary [14]. Essentially an eigenvalue of zero means that a steady state solution has been found, while an eigenvalue with a real number corresponds to instability. Once the growth rate of the eigenmodes is determined for a set of speeds, the critical speed is found by searching for the lowest speed corresponding to a positive growth rate in the eigenmode [17]. The eigenvalue method used by Hotspotter is complex and is covered in depth in [18], [14], [19], [20].

There are two versions of the Hotspotter code, classic and full 3D. The classic version of Hotspotter, as shown in Figure 3, uses a cross section of an axisymmetric system to be analyzed such as clutches and seals, it is not suitable for non-axisymmetric geometries such as automotive disk brake systems which have a non-axisymmetric brake pad. The classic version discretizes the cross section into first-order fully integrated quadrilateral elements and solves for critical speeds in two dimensions, then uses a Fourier series to solve for the hot spots along the circumferential dimension of the part.

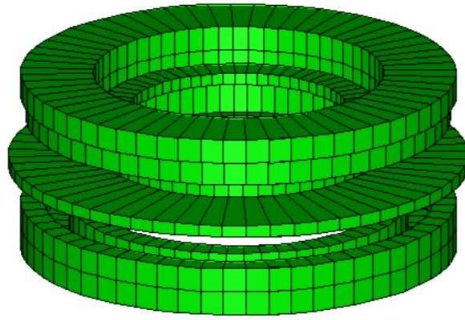


Figure 3. Hotspotter classic analyzes strictly axisymmetric geometries such as automotive clutches and airplane brakes. Image taken from [17].

The full 3D version of Hotspotter is suitable for both non-axisymmetric and axisymmetric geometries. The model is represented in three dimensions, shown by Figure 4, by first-order hexahedral finite elements, also called ‘hex’ or ‘brick’ elements, and solves for critical speeds throughout the geometry simultaneously [17].

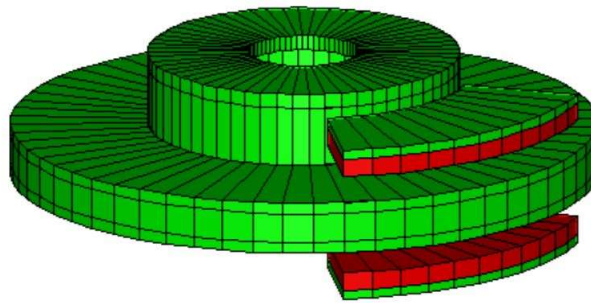


Figure 4. Hotspotter full 3D has capabilities to analyze complex non-axisymmetric geometry in three dimensions such as automotive and railroad brake systems. Image taken from [17].

1.2 Objectives

The objectives of this work are to develop triangular and tetrahedral finite elements appropriate for TEI analysis which may be incorporated into the Hotspotter software. This reduces the burden placed on the designer by making an easier model and mesh

transition from stress analysis in industry standard FEA software's such as Abaqus and Ansys to TEI analysis in Hotspotter software. Typically, users will mesh their parts and assemblies using triangular or tetrahedral elements in two dimensional or three dimensional space respectively for stress analysis in commercial FEA software's. Hotspotter currently has the capability to use quadrilateral or hexahedral elements in two dimensional or three dimensional analysis respectively for TEI analysis. Therefore, if a user wants to determine the critical velocity at which hotspots would occur in the design, a re-mesh of the geometries and further mesh convergence studies would have to be conducted due to the mis-match of element types when moving between software's.

Hotspotter was developed using quad and hex elements due to the computational efficiency of those elements and availability of computers capable of running such an analysis at the time. The efficiency is realized because quad and hex elements require less elements to obtain an accurate result when compared to triangular and tetrahedral elements. A simple solution might be to use quadrilateral and hexahedral elements when doing the stress analysis. However, the limitation to using first-order quad and hex elements is that they may exhibit a shear locking behavior or be overly stiff when the model has complicated geometry, is loaded in bending, or contact is present, which is a requirement in sliding friction system analysis.

Computational advances within the last 20 years have made it feasible to run large analysis on a fine mesh discretization using triangular or tetrahedral elements with reasonable solution times. Therefore, it is preferred to use a first-order linear triangular or tetrahedral element in the stress analysis to avoid such complications from meshing

intricate geometries with quad or hex elements. Also, the use of tri and tet elements allows for auto meshing and similar adaptive mesh refinement in zones of high stress [21]. Literature presents some conflicting information regarding use of first or second order triangular and tetrahedral elements in contact problems. For example, according to literature, a linear triangular element is susceptible to shear locking behavior while higher order elements such as a 4-noded or 10-noded tetrahedron are preferred for stress analysis [22]. And on the other hand, due to the high nonlinearity of TEI analysis, a first-order element may be able to handle the contact and solution nonlinearities better than a second order element can due to the lack of mid side nodes. Therefore, in an attempt to keep computational cost low, this investigation will examine the feasibility of using first-order elements. If it is found to be not within acceptable error margins, a further study will need to be conducted which investigates the use of second order elements in Hotspotter. If such first-order element types are found to be sufficient, this investigation will provide users the ability to import the same mesh from commercial FEA packages to Hotspotter and reduce the time burden of remeshing while giving sufficiently accurate results.

CHAPTER 2 ELEMENT DEVELOPMENT FOR THERMAL AND STRESS ANALYSIS

2.1 FEA introduction

In the pursuit of modelling complex systems, defining the governing equations and physics of a problem may not be overly difficult. However, solving those equations by analytical methods is often rigorous and time consuming, or impossible [23]. The difficulty of solving the closed form analytical solution arises from irregularities and arbitrary features or geometries [15]. A simpler more flexible method of simulating real phenomenon has been developed within the last three decades which allows engineers to solve very difficult and practical problems [23]. This method which uses numerical simulation instead of closed form analytical solutions is called the Finite Element Method or Finite Element Analysis (FEM/FEA). The FEM uses many small interconnected elements and produces a “piece-wise” numerical approximation to the governing equations of the problem [15]. Instead of attempting to directly solve the complex partial differential equations governing the problem, the FEM reduces those equations to a set of simultaneous equations which can be solved with the use of most personal computers [15].

Most modern engineering problems are concerned with complex geometries, materials, loads, motion, boundary conditions, and may or may not be involved in the time domain. As such the study and use of FEA is of great interest to design engineers

looking for solutions to a wide array of variables regarding their systems performance. Having the ability to simulate such complex systems reduces the engineers overall time and computational resources required to understand how a system will perform in the field. It should be stated that use of the finite element method should not be a replacement to good understanding of the theory behind the problem.

A typical process for FEA involves five steps. Firstly, pre-processing the system which involves subdividing the problem domain into finite elements. Secondly, formulation of the elements, element definitions, and governing equations. Thirdly, assembling the elements into a mesh and element equations into a global matrix. Fourthly, solving the set of equations which represent the system for a field variable. Lastly, post-processing of the results such as stress, strain, displacement, and visual representations of the system.

The FEM uses traditional variational methods to approximate the governing equations over a series of subdomains which make up the entire domain [23]. This is done because it is easier to approximate a geometry with a series of polynomials than to find the equation which represents the domain exactly. When the domain is highly discretized, the solution to the governing equations approach exact solutions.

When discretizing the domain of the system, analysts use “finite elements” which can be a variety of shapes including bar, truss, triangular, quadrilateral, etc. in two dimensions and tetrahedral, hexahedral, etc. in three dimensions. Elements are connected by nodes at each corner. In structural stress analysis, each element is governed by Hooke’s law in the elastic region:

$$F = Kx \quad (2.1)$$

where an external force F acts on the element, causing a displacement x governed by the spring constant K .

In structural mechanics each element has a stiffness matrix which relates nodal displacements to nodal forces through:

$$[Ke]\{De\} = \{Re\} \quad (2.2)$$

where $[Ke]$ is the elemental stiffness matrix, $\{De\}$ is the elemental displacement vector, and $\{Re\}$ is the elemental reaction force vector. This step is carried out for every element in the domain and assembled into a global set of matrices and vectors.

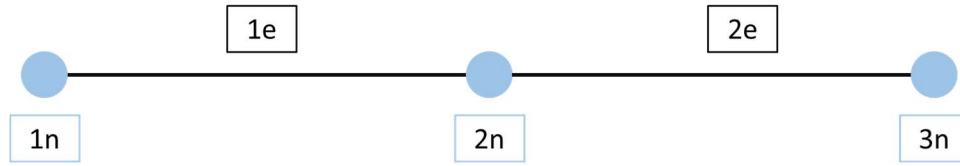


Figure 5. A simple two bar FEA system, single DOF per node, with elements (e) and nodes (n).

For a simple two bar element system with three elements who have one degree of freedom (DOF) each shown in Figure 5, the math model is represented by:

$$\begin{bmatrix} K1 & -K1 & 0 \\ -K1 & K1 + K2 & -K2 \\ 0 & -K2 & K2 \end{bmatrix} \begin{bmatrix} D1 \\ D2 \\ D3 \end{bmatrix} = \begin{bmatrix} R1 \\ R2 \\ R3 \end{bmatrix} \quad (2.3)$$

Some of the quantities of the system are known such as a zero displacement at a fixed boundary condition as well as force input into the system. Known forces are represented in a force vector R . Applying boundary conditions to the system, rearrangement of these equations to include zeros where zero displacement occurs, results in a sparser stiffness matrix, displacement vector, and force vector which are all used to solve for the unknown

displacements. This system of equations can be solved using numerical techniques such as Gauss Elimination or LU decomposition. The solution to this numerical method is the column vector D which holds the displacements for each node in each DOF.

From displacements, it is possible to post process the model to extrapolate meaningful information from the analysis. Such results can include strain which is defined for a bar element as:

$$\varepsilon = \frac{\Delta L}{L} \quad (2.4)$$

Or stress which is defined for a bar element as:

$$\delta = E\varepsilon \quad (2.5)$$

where E is Young's modulus.

Further, visual representation of the system is common to visualize deformations, stress concentrations, etc. such as in Figure 6 which shows von Mises stress in three dimensions of geometry representative of an aircraft stator subject to a uniform temperature increase with a fixed inner diameter.

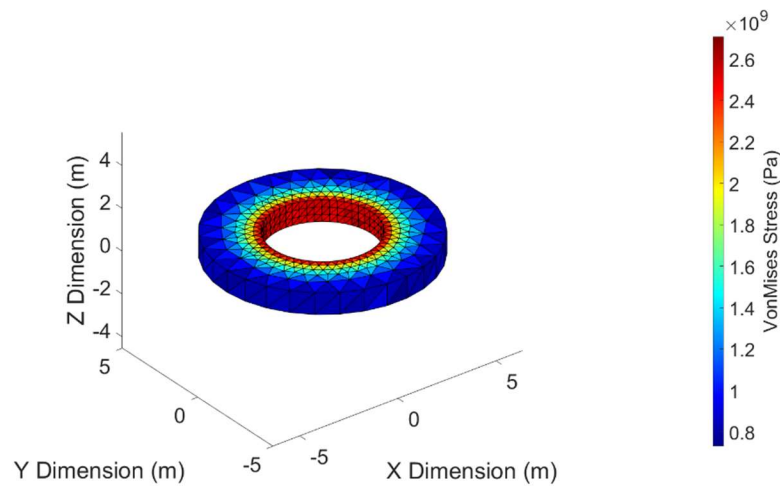


Figure 6. A visual Representation of von Mises stress of a simplified aircraft brake stator subject to uniform temperature increase.

2.2 First-order triangular element

Hotspotter currently uses an isoparametric representation of numerically integrated first-order 4-noded quadrilateral elements [17]. The first step in this investigation is to develop a program which uses iso-parametric element definition of numerically integrated first-order 3-noded triangular elements, colloquially referred to as the Constant Strain Triangle (CST). The CST has a linear displacement across the element and strain is therefore constant. A requirement for implementation to Hotspotter is that the elements may not use any symbolic math in Matlab. Reasoning for avoiding symbolic math is that Hotspotter does not have access these symbolic math libraries which are embedded in Matlab. Also, using symbolic math functions in FEA analysis greatly increases the computational cost required for a given simulation.

Development of the CST element in Matlab starts with writing code to define a single element and its corresponding global stiffness matrix. Then building a one element program in Abaqus using the CST element and comparing the stiffness matrices to ensure the element definition and integration scheme is functioning properly.

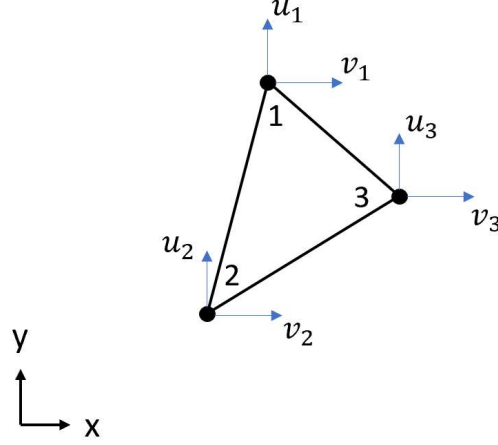


Figure 7. The Constant Strain Triangle (CST) element has three nodes, with two degrees of freedom each, leading to an element with six degrees of freedom total and a six by six stiffness matrix.

First, nodes and elements are defined in Cartesian coordinates. Then material properties required for the problem are defined, such as E , ν , & t . This analysis is an isotropic plane stress analysis meaning that stress in the z direction is equal to zero:

$$(\sigma_z = \tau_{yz} = \tau_{zx} = 0) \quad (2.6)$$

Therefore, the constitutive matrix, CM , is defined from young's modulus, E , and Poisson's ratio, ν , as:

$$CM = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1 - \nu)}{2} \end{bmatrix} \quad (2.7)$$

Lagrange interpolating polynomials, also known as shape functions, are implemented to define the displacement potential for each node. In a two dimensional problem, such as one involving the CST element, two displacement variables, u & v , are introduced into the interpolating polynomial [22]. For example, the CST uses the following interpolating polynomials to define its displacement:

$$u = a_1 + a_2x + a_3y \quad (2.8)$$

$$v = a_4 + a_5x + a_6y \quad (2.9)$$

This definition of interpolating polynomial shows a linear displacement field which further reinforces the constant value of strain across the element based on the definition of two dimensional strain:

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad (2.10)$$

$$\varepsilon_y = \frac{\partial v}{\partial y} \quad (2.11)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (2.12)$$

From the interpolating polynomial it is possible to define shape functions knowing that they are all polynomials of the same degree, the shape function is equal to one at its corresponding node, equal to zero at all other nodes, and varies linearly everywhere [24]. This shape function definition gives a value of 1 at the corresponding node and 0 at all other nodes as shown in Figure 8.

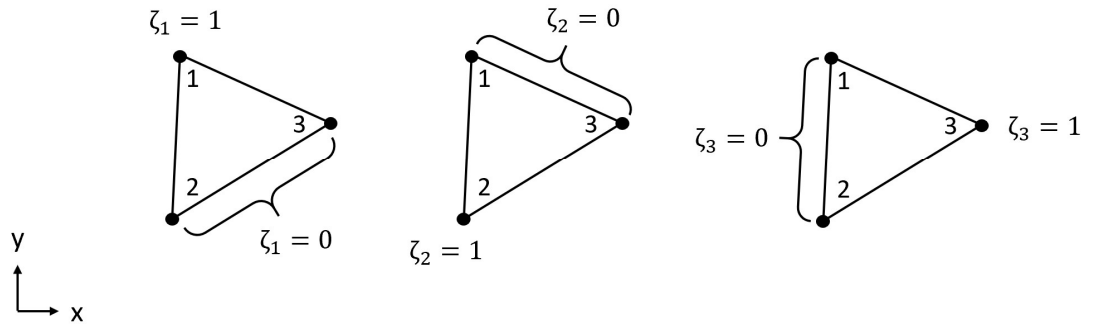


Figure 8. Isoparametric mapping of the CST element requires natural coordinates and shape functions which equal 1 at the corresponding node and 0 at all other nodes.

For example, in Figure 9, the N_1 shape function for a CST is ζ_1 in a triangular natural coordinate system.

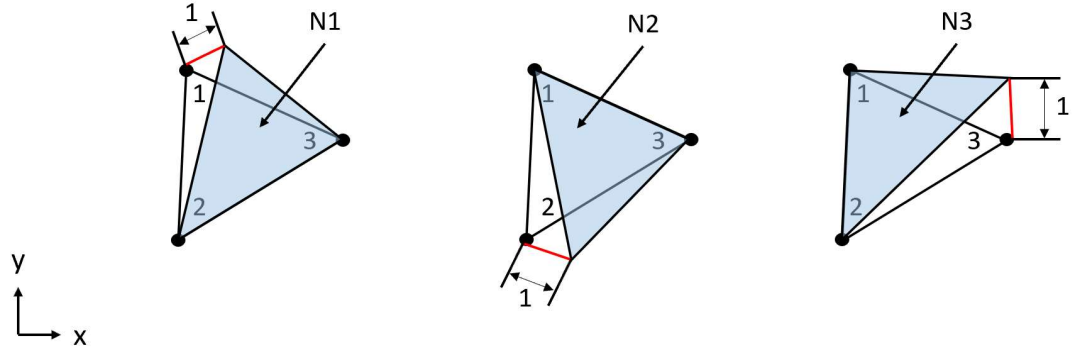


Figure 9. Isoparametric mapping of the CST element requires natural coordinates and shape functions which equal 1 at the corresponding node and 0 at all other nodes.

To conduct an analysis with many CST elements, all of which may be arbitrarily oriented and configured, it is important to use an isoparametric element definition. Isoparametric means the field variables (u, v) and shape of the element are defined by the same interpolating polynomial or shape function [22]:

$$x = \sum_i N_i x_i \quad (2.13)$$

$$y = \sum_i N_i y_i \quad (2.14)$$

$$u = \sum_i N_i u_i \quad (2.15)$$

$$v = \sum_i N_i v_i \quad (2.16)$$

where i is the range of nodes in the element.

In a classic structural analysis, the CST element has two spatial degrees of freedom corresponding to x & y displacement. However, because this element development is intended to be incorporated into Hotspotter, there needs to be a third degree of freedom in each node which represents the temperature at that node. This element type is called a Fourier element. Fourier elements are beneficial in this analysis because a three dimensional domain can be represented by a two dimensional cross section while retaining the third degree of freedom, T . This element definition allows three dimensional analysis to be meshed on a two dimensional cross section, which is a more computationally efficient process. The circumferentially periodic temperature field is defined by [6]

$$T = \sum N_i \theta_i \cos n\theta \quad (2.17)$$

where n is a wavenumber, i.e. number of hotspots along the circumference of the part. See also [25] for an application using the Fourier reduction method.

Each element has a set of Cartesian coordinates in two dimensions, x & y , which are assigned during the meshing step. In an isoparametric element definition, each CST element also contains a set of triangular natural coordinates, ζ , which stay with the element and maintain their position relative to it, even when the element deforms or otherwise changes position with respect to the Cartesian coordinate system [22].

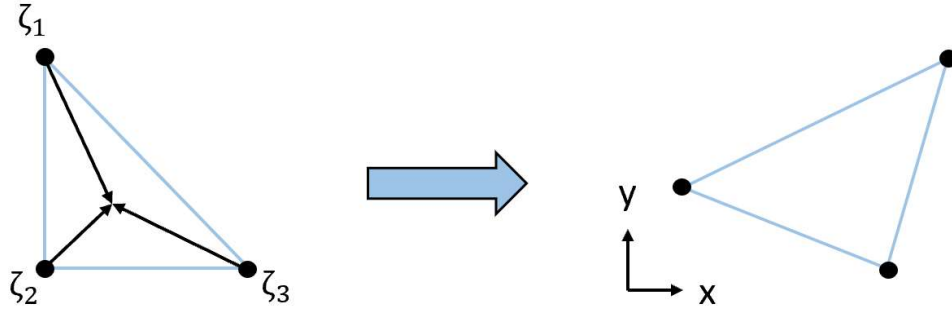


Figure 10. Isoparametric mapping allows a master element defined in natural or triangular coordinates to be mapped into a global Cartesian coordinate system.

The main task in defining an element stiffness matrix is establishing its strain-displacement matrix, $[B]$, which provides gradients in terms of each nodes DOF (u, v, T) . This strain-displacement matrix is the mapping or transformation between the natural and Cartesian coordinate system and is developed by differentiating the shape functions in Cartesian coordinates:

$$[B] = \begin{bmatrix} \partial N_i / \partial x & 0 \\ 0 & \partial N_i / \partial y \\ \partial N_i / \partial y & \partial N_i / \partial x \end{bmatrix} \quad (2.18)$$

According to [6] solutions to the frictional thermoelastic stability problem in axisymmetric geometries are most efficiently found using a Fourier reduction method. While many commercial FEA codes allow a user to transform the results into a cylindrical coordinate system, element definition is typically performed in Cartesian coordinates [26].

Hotspotter on the other hand uses a Fourier reduction, numerical perturbation method to solve for the critical sliding speed of the system. The critical sliding speed is the threshold of relative velocity of the sliding components such that thermal instability is excited. Hotspotter uses a Fourier reduction method because to solve the FEA solution to

the coupled transient thermoelastic contact problem, such as in [27], the computational resources required are too large and a numerical approach is able to approximate the solutions to a sufficiently acceptable level of accuracy according to [6]. Further, the inclusion of convective terms can be avoided in systems of geometric symmetry [28].

According to [29] and [6], the displacement and temperature fields of a Fourier element are defined as:

$$u_r = \sum_{i=1}^N N_i U_r^i \cos n\theta \quad (2.19)$$

$$u_\theta = \sum_{i=1}^N N_i U_\theta^i \sin n\theta \quad (2.20)$$

$$T = \sum_{i=1}^N N_i \Theta_r^i \cos n\theta \quad (2.21)$$

where $N_i(r, \phi)$ are the shape functions defined in the two dimensional cylindrical domain, Ω , and u_r & u_θ are components of the nodal displacement vector. The strain-displacement relationship for the Fourier element is defined by:

$$\varepsilon = \sum B_i U_i \quad (2.22)$$

For a Fourier element in cylindrical coordinates, as is the most common coordinate system according to [6], the strain-displacement matrix is defined in cylindrical coordinates as:

$$[B] = \begin{bmatrix} (\partial N_i / \partial r) \cos n\theta & 0 & (n N_i / r) \cos n\theta \\ (N_i / r) \cos n\theta & & \\ -(n N_i / 2r) \sin n\theta & \frac{1}{2} (\partial N_i / \partial r - N_i / r) \sin n\theta & \end{bmatrix} \quad (2.23)$$

A similar method can be used to define the Fourier element strain-displacement matrix in Cartesian coordinates as well [6]. The transformation between cylindrical and Cartesian coordinates is defined by [30] and [31]:

$$x = r \cos \theta \quad (2.24)$$

$$y = r \sin \theta \quad (2.25)$$

Regardless of the element type used throughout this investigation, the strain-displacement matrix remains the same. That is, it may be defined in Cartesian or cylindrical coordinates, and according to [32] defining a finite element in cylindrical coordinates for the solution of heat transfer is not well defined in literature. Regardless of how the element is defined, the values contained within the matrix are congruent from one element definition to another. The reason for defining the strain-displacement matrix in cylindrical coordinates is to allow the Hotspotter software to use values from the $[B]$ matrix at various times throughout the analysis and have compatible coordinate definitions from one element to another without the need for a coordinate transform.

In Cartesian coordinates, an issue arises when attempting to differentiate the shape functions because they are expressed in the natural coordinates of the volume element. The derivatives with respect to x & y are not available directly, therefore the derivatives with respect to natural coordinates are taken first [22]. To derive the terms in the $[B]$ matrix use the chain rule to expand:

$$\partial N_i / \partial x = \partial N_i / \partial \zeta_1 * \partial \zeta_1 / \partial x + \partial N_i / \partial \zeta_2 * \partial \zeta_2 / \partial x + \partial N_i / \partial \zeta_3 * \partial \zeta_3 / \partial x \quad (2.26)$$

$$\partial N_i / \partial y = \partial N_i / \partial \zeta_1 * \partial \zeta_1 / \partial y + \partial N_i / \partial \zeta_2 * \partial \zeta_2 / \partial y + \partial N_i / \partial \zeta_3 * \partial \zeta_3 / \partial y \quad (2.27)$$

The next step in the transformation is to determine the remaining unknown terms by forming the Jacobian. Rearranging into the form:

$$[J][P] = [R] \quad (2.28)$$

$$[J] = \begin{bmatrix} 1 & 1 & 1 \\ (\partial N_i / \partial \zeta_1)x & (\partial N_i / \partial \zeta_2)x & (\partial N_i / \partial \zeta_3)x \\ (\partial N_i / \partial \zeta_1)y & (\partial N_i / \partial \zeta_2)y & (\partial N_i / \partial \zeta_3)y \end{bmatrix} \quad (2.29)$$

$$[P] = \begin{bmatrix} \partial \zeta_1 / \partial x & \partial \zeta_1 / \partial y \\ \partial \zeta_2 / \partial x & \partial \zeta_2 / \partial y \\ \partial \zeta_3 / \partial x & \partial \zeta_3 / \partial y \end{bmatrix} \quad (2.30)$$

$$[R] = \begin{bmatrix} \partial 1 / \partial x & \partial 1 / \partial y \\ \partial x / \partial x & \partial x / \partial y \\ \partial y / \partial x & \partial y / \partial y \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.31)$$

It is clear that the matrix $[P]$ includes the missing terms required to solve for $\partial N_i / \partial x$, and $\partial N_i / \partial y$ when using the chain rule. Solve for the $[P]$ matrix by taking the inverse of the Jacobian $[J]$ matrix and multiplying it by the $[R]$ matrix:

$$[P] = [J]^{-1}[R] \quad (2.32)$$

However, a more robust way to solve this system of equations in Matlab would be to use the backslash operator:

$$[P] = [J] \backslash [R] \quad (2.33)$$

This Matlab function takes advantage of any symmetries and simplifications available then applies an appropriate linear systems algorithm such as LU decomposition or similar [33].

The resulting values contained in the $[P]$ matrix can then be substituted into the equations to form the $[B]$ matrix which include partials of the shape functions with respect to Cartesian coordinates as a function of natural coordinates:

$$[B] = \begin{bmatrix} \partial N_1/\partial x & 0 & \partial N_2/\partial x & 0 & \partial N_3/\partial x & 0 \\ 0 & \partial N_1/\partial y & 0 & \partial N_2/\partial y & 0 & \partial N_3/\partial y \\ \partial N_1/\partial y & \partial N_1/\partial x & \partial N_2/\partial y & \partial N_2/\partial x & \partial N_3/\partial y & \partial N_3/\partial x \end{bmatrix} \quad (2.34)$$

All of the components for building the individual element stiffness matrix are now available. Traditionally the CST element is integrated volumetrically due to the less complex nature of the element, number of nodes, the area can be easily calculated, and thickness remains constant throughout the element. The equation for volumetric integration is:

$$[k] = \int [B]^T [CM] [B] t dA \quad (2.35)$$

However, the intent of this code is to function as a plugin for the Hotspotter software, which requires isoparametric element definition and numerical integration. Therefore, it is required to formulate the integral for numerical integration over the element at one integration point. Gaussian quadrature for a CST uses the one point rule which has a single integration point located at the center of the element. The natural coordinates of the integration point are (1/3, 1/3, 1/3). Using a one point Gauss quadrature rule the equation for numerical integration of the CST is:

$$[k] = \int 1.125 [B]^T [CM] [B] |J| t dA \quad (2.36)$$

where t is the thickness of the plain stress element, $|J|$ is the determinant of the Jacobian matrix, $[B]$ is the strain-displacement matrix, and $[CM]$ is the constitutive matrix.

Up to this point, the method covered is for a standard isoparametric 3-noded triangular element which is useful for structural analysis. For Hotspotter to function, the

elements used must be capable of handling coupled thermal stresses and strains.

Therefore, it is required to incorporate additional terms for the analysis. The equation

which will be assembled is:

$$[k]\{T\} = \{f_T\} \quad (2.37)$$

where $[k]$ is the stiffness matrix of the element, $\{T\}$ is the temperature vector for each node, and $\{f_T\}$ is the force vector due to thermal expansion at each node.

To couple the force due to thermal expansion to the structural analysis, first a thermal force vector containing the temperature distribution of the mesh is assembled for each node using the same Gaussian integration scheme as assembling the stiffness matrix over the element:

$$\{f_T\} = \int 1.125[B]^T[CM]\{\varepsilon_T\}|J|t \, dA \quad (2.38)$$

where $\{\varepsilon_T\}$ is the thermal strain vector:

$$\{\varepsilon_T\} = \{\alpha T, \alpha T, 0\} \quad (2.39)$$

where α is the coefficient of thermal expansion for the material. The last column corresponds to shear strain. The value is imposed as 0 because there is no shear strain in an isotropic material due to a temperature gradient. If an anisotropic material needs to be considered, the constitutive matrix, $[CM]$, as well as the thermal strain vector, $\{\varepsilon_T\}$, need to be updated accordingly.

Once these components have been integrated over each element, the global stiffness matrix, $[K]$, can be assembled based on each node's global degree of freedom. The first check point in development must happen here to ensure the stiffness matrix from Abaqus and Matlab code agree. Typically, during development, a single element is developed

first rather than an entire mesh to compare the stiffness matrices more easily from Abaqus and Matlab. For example, during this investigation, a single element was developed with nodal locations at $\{0, 0; 2, 0; 0, 2\}$, a Young's modulus of 2×10^{11} Pa, unity thickness, and a 0.33 value for Poisson's ratio. The single element's stiffness matrix from Abaqus and Matlab are given below.

Table 1. A single CST element stiffness matrix output from Abaqus.

1.0e+11	U1	V1	U2	V2	U3	V3
U1	1.4981	0.7462	-1.1222	-0.3759	-0.3759	-0.3703
V1	0.7462	1.4981	-0.3703	-0.3759	-0.3759	-1.1222
U2	-1.1222	-0.3703	1.1222	0.0000	0.0000	0.3703
V2	-0.3759	-0.3759	0.0000	0.3759	0.3759	0.0000
U3	-0.3759	-0.3759	0.0000	0.3759	0.3759	0.0000
V3	-0.3703	-1.1222	0.3703	0.0000	0.0000	1.1222

Table 2. A single CST element stiffness matrix output from Matlab.

1.0e+11	U1	V1	U2	V2	U3	V3
U1	1.4981	0.7463	-1.1222	-0.3759	-0.3759	-0.3703
V1	0.7463	1.4981	-0.3703	-0.3759	-0.3759	-1.1222
U2	-1.1222	-0.3703	1.1222	0.0000	0.0000	0.3703
V2	-0.3759	-0.3759	0.0000	0.3759	0.3759	0.0000
U3	-0.3759	-0.3759	0.0000	0.3759	0.3759	0.0000
V3	-0.3703	-1.1222	0.3703	0.0000	0.0000	1.1222

Developing the Matlab code this way greatly reduces the time required to debug or find problems with the code, if any are encountered. Comparing single element stiffness matrices effectively gates any code problems to pre element definition or post element definition.

Once the stiffness matrices agree for a single element, the next step is to assemble the global stiffness matrix, $[K]$, global force vector, $\{F\}$, and global displacement vector, $\{X\}$. The simplest method for global assembly is using the single element code repeated in a for loop for each element in the domain. These vectors and matrices are assembled into a global scheme based on the global degree of freedom for each node. This allows the entire system of equations to be represented by one stiffness matrix, one force vector, and one displacement vector. For example, node number 22 in Figure 11 corresponds to degrees of freedom 43 and 44 for the global x & y directions respectively. Therefore,

node 22's stiffness, force, and displacement values will be located in rows and columns 43 and 44 for the analysis.

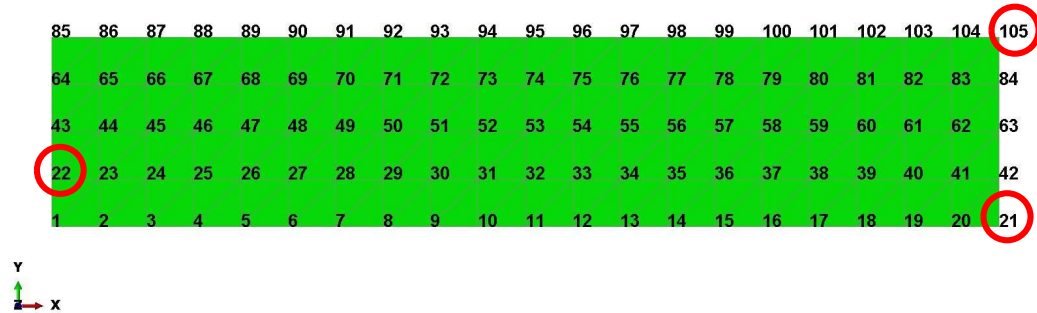


Figure 11. A cantilever beam meshed using CST elements, fixed at the left end in both x and y directions. This beam is ready to have a mechanical, thermal, or combined load imposed.

Once the system is represented in a global sense, the mechanical force vector can be assembled based on where the force is applied. If node 105, in Figure 11, has a mechanical force of 10 Newtons applied in the downward y direction, a force of -10 N will be imposed into row 210 in the Matlab force vector. If no external mechanical force is present in the system, and the resultant forces are due to a uniform temperature increase, such as in the use case in Hotspotter, the mechanical force vector will contain 0's. The previously integrated thermal force vector will include terms based on the temperature change and coefficient of thermal expansion.

Next is the application of boundary conditions. Applying boundary conditions after the assembly into global components is essential to the analysis to accurately represent the physical system. The simplest method for controlling the field variable, typically displacement, is to impose a 0 to each fixed global degree of freedom. Then, imposing a 1 in the global stiffness matrix where the row and column of the same degree of freedom intersect. For example, if node 22 was to be a fixed boundary condition in both x and y

directions, the stiffness matrix would be assigned 0's on rows and columns 43 and 44. Then at the intersection of row 43 and column 43 a 1 would be imposed, as well as the intersection of row 44 and column 44. Moving to the mechanical force vector, a 0 would be imposed on rows 43 and 44, implying no external force on node 22. This will lead to the displacement solution, $\{X\}$, having 0 displacement at node 22, effectively creating a fixed boundary condition. Typically, the stiffness matrix without boundary conditions is set aside for use in calculating the reaction forces and the sparse boundary condition stiffness matrix is renamed to $[K_{BC}]$.

Once the mechanical boundary conditions have been accounted for, the thermal force vector and mechanical force vector can be added together to create one coupled force vector:

$$\{F\} = \{f_m\} + \{f_T\} \quad (2.40)$$

where $\{f_m\}$ is the mechanical force vector with appropriate boundary conditions applied, and $\{f_T\}$ is the thermal force vector. The reason no boundary conditions are imposed on the thermal force vector is due to the entire mesh being subjected to the uniform temperature increase. If, for example there was a known temperature, material change, or a thermal boundary condition then the thermal force vector would need to be changed to represent those boundary conditions. However, for the analysis in Hotspotter it is known that those thermal boundary conditions do not exist, so they are not taken into consideration here.

After all of the prior steps are functioning correctly, and the code can create the $[K]$, $\{F\}$, and $\{X\}$ components, the analysis is ready to run. As mentioned in Chapter 1, the equation which will be solved is:

$$\{F\} = [K_{BC}]\{X\} \quad (2.41)$$

where $\{F\}$ is the coupled thermal and mechanical force vectors, $[K_{BC}]$ is the global stiffness matrix representing the material stiffness of the physical part, and $\{X\}$ is the unknown displacement at each node which occur due to the applied force. To solve this system of equations, it is necessary to use a numerical method appropriate for back substitution such as Gauss elimination, LU decomposition, or similar:

$$\{X\} = [K_{BC}]^{-1}\{F\} \quad (2.42)$$

The most efficient way of solving this system in Matlab is to take advantage of the backslash function:

$$\{X\} = [K_{BC}] \backslash \{F\} \quad (2.43)$$

Using this function is the most efficient method because Matlab calls a built in algorithm to select the best solver for the given problem set [33].

As previously discussed, the output from this operation is a vector with nodal displacement values in the x and y directions and can be located via their global degrees of freedom. For example, in Figure 11, node 21 will have a displacement with x & y components located in the $\{X\}$ vector in rows 41 and 42 respectively.

If the engineer is interested in reaction forces, it is a trivial calculation at this point because all of the components are in place:

$$\{F_R\} = [K]\{X\} \quad (2.44)$$

where $[K]$ is the stiffness matrix before any boundary conditions imposed, and $\{X\}$ is the solution to:

$$\{X\} = [K_{BC}] \setminus \{F\} \quad (2.45)$$

As previously calculated.

The last step in the FEM analysis is to post-process the results for further analysis or decision making regarding the design of the system. Typically, in structural analysis a designer is interested in stress, strain, displacement, and reaction forces.

To calculate stress and strain values for a given mesh, the $[B]$ matrix must be called again for each element. Stress is defined as:

$$\{\delta\} = [B]\{X\} \quad (2.46)$$

Strain is defined as:

$$\{\varepsilon\} = [CM]([B]\{X\} - \{\varepsilon_t\}) \quad (2.47)$$

An interesting way to visualize the results are to see the magnitude of displacement which needs to be calculated from the nodal x & y components of displacement:

$$\{D\} = \sqrt{x^2 + y^2} \quad (2.48)$$

The last check to perform when developing and FEA code in Matlab is to ensure the displacement, stress, and strain values correlate well with a commercial code and hand calculations of the analytical solution, if one exists. For the case of a simply supported beam, this is a simple comparison. However, for more advanced and coupled simulations, such as the case with this study, the analytical solution may take a long time to develop or may not be possible to develop at all. In these cases, it is important that the commercial code and development code are in close agreement. If they are not, a mesh refinement

study may show whether or not the solutions are converging or diverging. These checks for this analysis are performed and reported on in Chapter 3.

2.3 First-order tetrahedral element

For three dimensional and non-axisymmetric problems, Hotspotter currently uses an isoparametric representation of numerically integrated first-order 8-noded hexahedral elements.

The next step in this investigation is to develop a program which uses iso-parametric element definition of numerically integrated first-order 4-noded tetrahedral elements, colloquially referred to as the tet4. The tet4, much like the CST, has a linear displacement across the element and strain is therefore constant.

Just as with the CST, development of the tet4 element in Matlab starts with writing code to define a single element and its corresponding global stiffness matrix. Then building a one element program in Abaqus using the tet4 element and comparing the stiffness matrices to ensure the element definition and integration scheme is functioning properly.

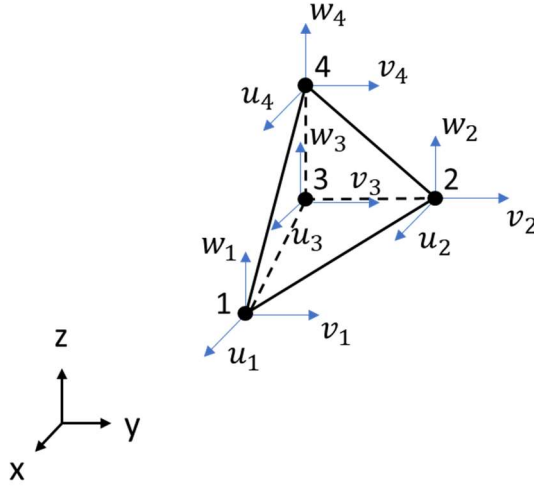


Figure 12. The 4-noded tetrahedral (tet4) element has four nodes, with three degrees of freedom each, leading to an element with twelve degrees of freedom total and a twelve by twelve stiffness matrix.

The constitutive matrix is expanded into three dimensions to:

$$CM = \frac{E}{(1-\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}-\nu & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}-\nu & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}-\nu \end{bmatrix} \quad (2.49)$$

The Lagrange interpolating polynomial for a tet4 is expanded to include the z dimension and uses the following equations to define its displacement:

$$u = a_1 + a_2x + a_3y + a_4z \quad (2.50)$$

$$v = a_5 + a_6x + a_7y + a_8z \quad (2.51)$$

$$w = a_9 + a_{10}x + a_{11}y + a_{12}z \quad (2.52)$$

Next the strain values are defined in three dimensions as:

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad (2.53)$$

$$\varepsilon_y = \frac{\partial v}{\partial y} \quad (2.54)$$

$$\varepsilon_z = \frac{\partial w}{\partial z} \quad (2.55)$$

$$\gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \quad (2.56)$$

$$\gamma_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \quad (2.57)$$

$$\gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \quad (2.58)$$

When using a tet4 element, it remains important to use an isoparametric element definition so the mesh may contain arbitrarily oriented and sized elements. The shape functions and isoparametric mapping definitions are also expanded to three dimensions:

$$x = \sum_i N_i x_i \quad (2.59)$$

$$y = \sum_i N_i y_i \quad (2.60)$$

$$z = \sum_i N_i z_i \quad (2.61)$$

$$u = \sum_i N_i u_i \quad (2.62)$$

$$v = \sum_i N_i v_i \quad (2.63)$$

$$w = \sum_i N_i w_i \quad (2.64)$$

where i is the range of nodes in the element.

The temperature field in a Fourier element is described by the same equation for temperature as in the CST element:

$$T = \sum N_i \theta_i \cos n\theta \quad (2.65)$$

where n is a wavenumber.

In an isoparametric element definition, each tet4 element also contains a set of triangular natural coordinates, ζ , which stay with the element and maintain their position relative to it, even when the element deforms or otherwise changes position with respect to the Cartesian coordinate system [22]. This is true for the CST element and tet4 elements, but the tet4 element has a fourth natural coordinate ζ_4 as shown in Figure 13.

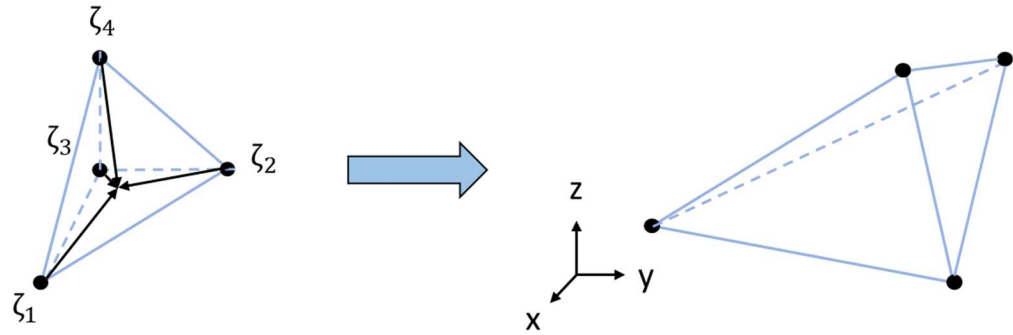


Figure 13. Isoparametric mapping allows a master element defined in natural or triangular coordinates to be mapped into a global Cartesian coordinate system.

Establishing and defining the strain-displacement matrix for a tet4 element remains crucial in utilizing an isoparametric element definition. Therefore the $[B]$ matrix is expanded to three dimensions:

$$[B] = \begin{bmatrix} \partial N_i / \partial x & 0 & 0 \\ 0 & \partial N_i / \partial y & 0 \\ 0 & 0 & \partial N_i / \partial z \\ \partial N_i / \partial y & \partial N_i / \partial x & 0 \\ 0 & \partial N_i / \partial z & \partial N_i / \partial y \\ \partial N_i / \partial z & 0 & \partial N_i / \partial x \end{bmatrix} \quad (2.66)$$

Defining the Fourier element in three dimensions is accomplished using the same method as for the CST but expanded to include the third dimension. The temperature and displacement fields of the Fourier tet4 element are defined as:

$$u_r = \sum_{i=1}^N N_i U_r^i \cos n\theta \quad (2.67)$$

$$u_\theta = \sum_{i=1}^N N_i U_\theta^i \sin n\theta \quad (2.68)$$

$$u_z = \sum_{i=1}^N N_i U_z^i \sin n\theta \quad (2.69)$$

$$T = \sum_{i=1}^N N_i \Theta_r^i \cos n\theta \quad (2.70)$$

where $N_i(r, \theta, z)$ are the shape functions defined in the three dimensional cylindrical domain, Ω , and u_r , u_θ , & u_z are components of the nodal displacement vector. The strain-displacement relationship for the Fourier element is defined by:

$$\varepsilon = \sum B_i U_i \quad (2.71)$$

Therefore, the strain-displacement matrix for a Fourier element in three dimensions is:

$$[B] \quad (2.72)$$

$$= \begin{bmatrix} (\partial N_i / \partial r) \cos n\theta & 0 & 0 \\ (N_i / r) \cos n\theta & (n N_i / r) \cos n\theta & 0 \\ 0 & 0 & (\partial N_i / \partial z) \cos n\theta \\ -(n N_i / 2r) \sin n\theta & \frac{1}{2} (\partial N_i / \partial r - N_i / r) \sin n\theta & 0 \\ 0 & \frac{1}{2} (\partial N_i / \partial z) \sin n\theta & -(n N_i / 2r) \sin n\theta \\ \frac{1}{2} (\partial N_i / \partial z) \cos n\theta & 0 & \frac{1}{2} (\partial N_i / \partial r) \cos n\theta \end{bmatrix}$$

In Cartesian coordinates, the same issue arises when attempting to differentiate the shape functions because they are again expressed in the natural coordinates of the volume element. The derivatives are found using the chain rule by taking the derivatives with respect to the natural coordinates first. To find these terms in the $[B]$ matrix use the chain rule to expand:

$$\partial N_i / \partial x = \partial N_i / \partial \zeta_1 * \partial \zeta_1 / \partial x + \partial N_i / \partial \zeta_2 * \partial \zeta_2 / \partial x + \partial N_i / \partial \zeta_3 * \partial \zeta_3 / \partial x \quad (2.73)$$

$$+ + \partial N_i / \partial \zeta_4 * \partial \zeta_4 / \partial x$$

$$\partial N_i / \partial y = \partial N_i / \partial \zeta_1 * \partial \zeta_1 / \partial y + \partial N_i / \partial \zeta_2 * \partial \zeta_2 / \partial y + \partial N_i / \partial \zeta_3 * \partial \zeta_3 / \partial y \quad (2.74)$$

$$+ + \partial N_i / \partial \zeta_4 * \partial \zeta_4 / \partial y$$

$$\partial N_i / \partial z = \partial N_i / \partial \zeta_1 * \partial \zeta_1 / \partial z + \partial N_i / \partial \zeta_2 * \partial \zeta_2 / \partial z + \partial N_i / \partial \zeta_3 * \partial \zeta_3 / \partial z \quad (2.75)$$

$$+ + \partial N_i / \partial \zeta_4 * \partial \zeta_4 / \partial z$$

In the next step, the same procedure is applied to the tet4 element to form the Jacobian and then rearrange into the form:

$$[J][P] = [R] \quad (2.76)$$

$$[J] = \begin{bmatrix} 1 & 1 & 1 & 1 \\ (\partial N_i/\partial \zeta_1)x & (\partial N_i/\partial \zeta_2)x & (\partial N_i/\partial \zeta_3)x & (\partial N_i/\partial \zeta_4)x \\ (\partial N_i/\partial \zeta_1)y & (\partial N_i/\partial \zeta_2)y & (\partial N_i/\partial \zeta_3)y & (\partial N_i/\partial \zeta_4)y \\ (\partial N_i/\partial \zeta_1)z & (\partial N_i/\partial \zeta_2)z & (\partial N_i/\partial \zeta_3)z & (\partial N_i/\partial \zeta_4)z \end{bmatrix} \quad (2.77)$$

$$[P] = \begin{bmatrix} \partial \zeta_1/\partial x & \partial \zeta_1/\partial y & \partial \zeta_1/\partial z \\ \partial \zeta_2/\partial x & \partial \zeta_2/\partial y & \partial \zeta_2/\partial z \\ \partial \zeta_3/\partial x & \partial \zeta_3/\partial y & \partial \zeta_3/\partial z \\ \partial \zeta_4/\partial x & \partial \zeta_4/\partial y & \partial \zeta_4/\partial z \end{bmatrix} \quad (2.78)$$

$$[R] = \begin{bmatrix} \partial 1/\partial x & \partial 1/\partial y & \partial 1/\partial z \\ \partial x/\partial x & \partial x/\partial y & \partial x/\partial z \\ \partial y/\partial x & \partial y/\partial y & \partial y/\partial z \\ \partial z/\partial x & \partial z/\partial y & \partial z/\partial z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.79)$$

Once again, the $[P]$ matrix includes the missing terms required to solve for $\partial N_i/\partial x$, $\partial N_i/\partial y$, and $\partial N_i/\partial z$ when using the chain rule

Solve for the $[P]$ matrix by taking the inverse of the Jacobian $[J]$ matrix and multiplying it by the $[R]$ matrix. In Matlab it is recommended to take advantage of the backslash operator:

$$[P] = [J] \backslash [R] \quad (2.80)$$

After the $[P]$ matrix has been solved for, the values can be substituted into the equations to form the $[B]$ matrix:

$$[B] = \begin{bmatrix} \partial N_i/\partial x & 0 & 0 \\ 0 & \partial N_i/\partial y & 0 \\ 0 & 0 & \partial N_i/\partial z \\ \partial N_i/\partial y & \partial N_i/\partial x & 0 \\ 0 & \partial N_i/\partial z & \partial N_i/\partial y \\ \partial N_i/\partial z & 0 & \partial N_i/\partial x \end{bmatrix} \quad (2.81)$$

Once the $[B]$ matrix is fully assembled, all of the components are available to integrate the individual element stiffness matrix. The tet4 element is numerically integrated using Gaussian quadrature which uses a one point rule which has a single

integration point located at the center of the element. The natural coordinates of the integration point are $(1/3, 1/3, 1/3, 1/3)$. Using a one point Gauss quadrature rule the equation for numerical integration of the tet4 is:

$$[k] = \int 1.125[B]^T[CM][B]|J| dA \quad (2.82)$$

where $|J|$ is the determinant of the Jacobian matrix, $[B]$ is the strain-displacement matrix, and $[CM]$ is the constitutive matrix.

The next step is to incorporate the additional terms required to run the coupled thermal stress analysis required for incorporation into Hotspotter. First the thermal force vector is integrated using the same Gaussian integration scheme as assembling the stiffness matrix over the element:

$$\{f_T\} = \int 1.125[B]^T[CM]\{\varepsilon_T\}|J| dA \quad (2.83)$$

where $\{\varepsilon_T\}$ is the thermal strain vector:

$$\{\varepsilon_T\} = \{\alpha T, \alpha T, \alpha T, 0, 0, 0\} \quad (2.84)$$

Once these components have been integrated over each element, the global stiffness matrix, $[K]$, can be assembled based on each node's global degree of freedom. Just as with the CST element, the tet4 will be developed using a single element before an entire mesh so that the stiffness matrices may be compared more easily to one another to ensure the element definitions are the same.

Once the stiffness matrices agree for a single element, the next step is to assemble the global stiffness matrix, $[K]$, global force vector, $\{F\}$, and global displacement vector,

$\{X\}$. The simplest method for global assembly is using the single element code repeated in a for loop for each element in the domain.

Once the system is represented in a global sense, the mechanical force vector can be assembled based on where the force is applied in three dimensions.

Next is the application of boundary conditions in the same manner as the CST, but expanded to three dimensions. Then the stiffness matrix is renamed to $[K_{BC}]$, and the pre-boundary conditions stiffness matrix is set aside for use later in calculating reaction forces.

Once the mechanical boundary conditions have been accounted for, the thermal force vector and mechanical force vector can be added together to create one coupled force vector:

$$\{F\} = \{f_m\} + \{f_T\} \quad (2.85)$$

After all of the prior steps are functioning correctly, and the code can create the $[K]$, $\{F\}$, and $\{X\}$ components, the analysis is ready to run. As mentioned in Chapter 1, the equation which will be solved is:

$$\{F\} = [K_{BC}]\{X\} \quad (2.86)$$

To solve this system of equations, it is necessary to use a numerical method appropriate for back substitution such as Gauss elimination, LU decomposition, backslash operator in Matlab, or similar:

$$\{X\} = [K_{BC}] \backslash \{F\} \quad (2.87)$$

As with the CST the output from this operation is a vector with nodal displacement values in the x , y , and z directions and can be located via their global degrees of freedom.

Any post processing of results including stresses, strains, displacement magnitudes, and reaction forces are the same equations for the tet4 as for the CST.

As with any FEA code in Matlab, the last check to perform is to ensure the displacement, stress, and strain values correlate well with a commercial code and hand calculations of the analytical solution, if one exists. These checks for this analysis are performed and reported on in Chapter 3.

Once the element types have been developed and compared against commercial code and where possible, theoretical calculations, they must be implemented into Hotspotter for analysis of TEI. The constitutive law, loading, and boundary conditions are the same in Hotspotter as in classical FEM analysis. The only further consideration for adapting Matlab code to Hotspotter is ensuring the strain-displacement and stiffness matrices are compatible. As previously discussed, the strain-displacement matrix and stiffness matrix are developed using a Fourier reduction method.

Further, to run a Hotspotter analysis with the new element types, the 3D version of Hotspotter is used which requires an input file much like that of an Abaqus input file. The definition of the new element type is called by the input file, but the calculation of the new strain-displacement and stiffness matrices is computed in the main Hotspotter algorithm which is in the form of an executable program.

CHAPTER 3 VERIFICATION & VALIDATION

3.1 Element verification

Element verification must be performed prior to implementing new code into Hotspotter to ensure the elements are robust. Verification is performed on simplified geometry and loading to prove the new elements give sufficiently accurate results when compared with results from a commercial code such as Abaqus as well as any attainable analytical results. The verification cases start out overly simplified in attempt to compartmentalize any potential problems which may occur with increased complexity. With each subsequent verification case, the analysis becomes more and more generalized, ending in a three dimensional analysis of geometry representative of an aircraft stator with a fixed inner diameter subject to a uniform temperature increase.

Material properties for the following element verification analyses are:

$$E = 2 \times 10^{11} \text{ Pa} \quad (3.1)$$

$$\nu = 0.33 \quad (3.2)$$

$$\alpha = 1.15 \times 10^{-6} \cdot ^\circ\text{C}^{-1} \quad (3.3)$$

$$\Delta T = 1000 \text{ }^\circ\text{C} \quad (3.4)$$

3.1.1 First-order triangular element

3.1.1.1 Verify element in bending only

The first case to compare analytical, Abaqus, and Matlab results is a simple cantilever beam with a fixed end subject to a 100 N mechanical load at the free end; no thermal load

was added to this analysis. The beam is fixed in both the x & y directions to prevent rigid body translations. This comparison verifies that the CST code is functioning when a mechanical load is present. Because the element integration schemes are different from the Matlab code to Abaqus, it is important to compare these results to ensure the code is accurate when compared to a traditional volumetrically integrated element. Figure 14 shows the geometry used in the analysis and Figure 15 shows the anticipated deformed shape once the load has been applied. Initial geometric values for the beam are given by:

$$L_o = 10 \text{ m} \quad (3.5)$$

$$h = 2 \text{ m} \quad (3.6)$$

$$t = 1 \text{ m} \quad (3.7)$$

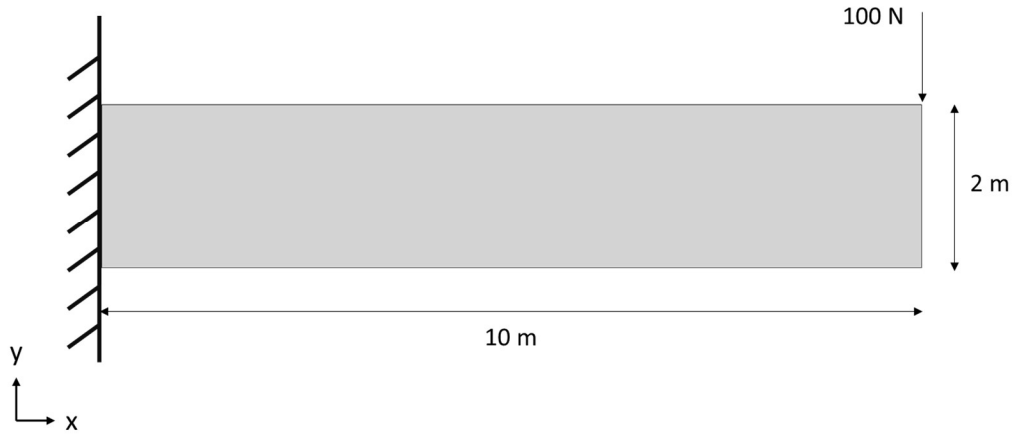


Figure 14. Initial geometry of a 2D beam fixed in both directions on the left end.

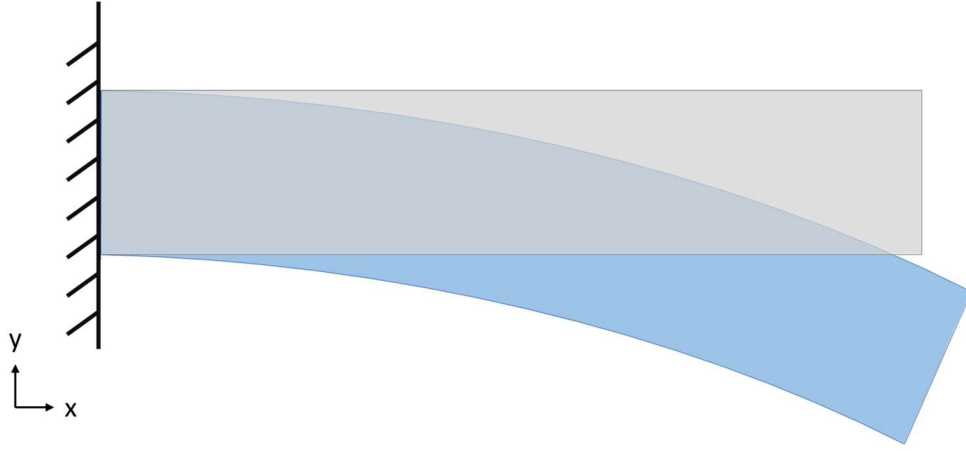


Figure 15. Initial (gray) and deformed (blue) geometry of a 2D beam fixed in both directions on the left end allowing deformation along the beam subject to a mechanical load of 100 N.

The maximum stress in the beam is defined by:

$$\sigma = \frac{My}{I} \quad (3.8)$$

$$M = FL_o \quad (3.9)$$

$$I = \frac{th_o^3}{12} \quad (3.10)$$

The maximum strain in the beam is defined by:

$$E = \frac{\sigma}{\varepsilon} \quad (3.11)$$

Therefore:

$$\varepsilon = \frac{\sigma}{E} \quad (3.12)$$

The maximum deformation in the beam is defined by:

$$\delta = \frac{FL^3}{3EI} \quad (3.13)$$

$$I = \frac{th_o^3}{12} \quad (3.14)$$

The first case is a rough mesh where both Matlab and Abaqus use a volumetric integration scheme as shown in Figure 16 and Figure 17. These results are then compared to a numerically integrated CST element in Matlab, Figure 18, to ensure all three cases are sufficiently close to one another. All three results are then compared to the analytical solution in Table 3.

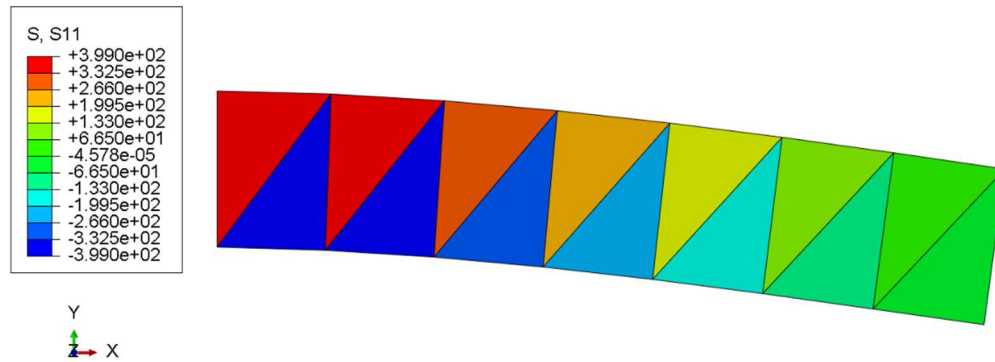


Figure 16. Abaqus volumetric integration rough mesh beam results for stress in the x direction.

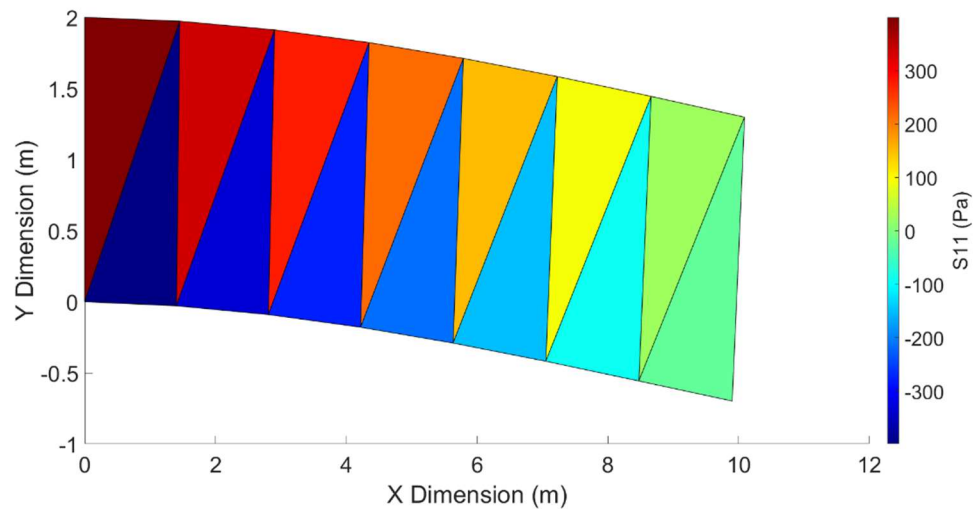


Figure 17. Matlab volumetric integration rough mesh beam results for stress in the x direction.

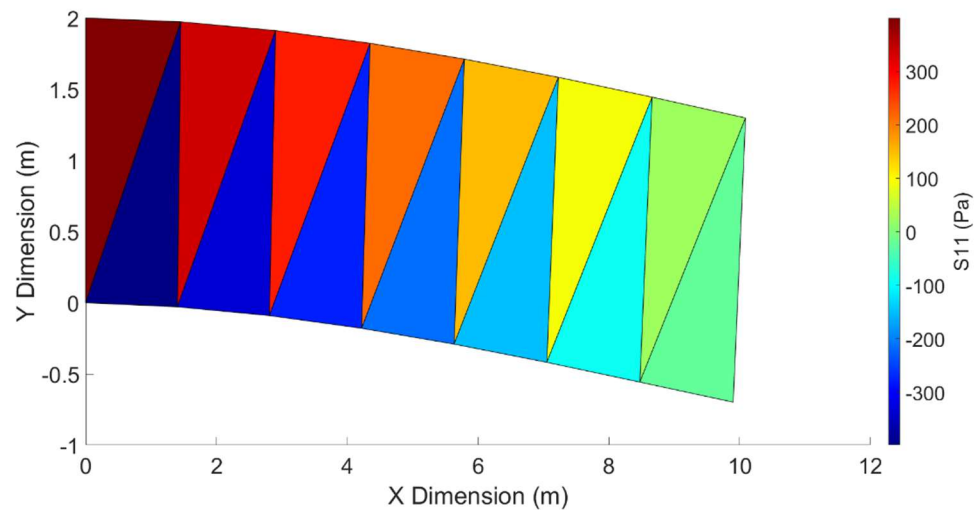


Figure 18. Matlab numerical integration rough mesh beam results for stress in the x direction.

Table 3. Numerical vs. volumetric integration results for mechanically loaded rough mesh beam.

Value		Volumetric Integration		Numerical Integration
	Theory	Abaqus	Matlab	Matlab
Max Stress (Pa)	1500	398.9780	398.9780	398.9780
Location (element)		13	13	13
Min Stress (Pa)	-1500	-398.9780	-398.9780	-398.9780
Location (element)		14	14	14
Max Strain	7.5×10^{-9}	1.7776×10^{-9}	1.7776×10^{-9}	1.7776×10^{-9}
Location (element)		13	13	13
Min Strain	-7.5×10^{-9}	-1.8081×10^{-9}	-1.8081×10^{-9}	-1.8081×10^{-9}
Location (element)		14	14	14
Max Disp. (m)	2.5×10^{-7}	0.7081×10^{-7}	0.7081×10^{-7}	0.7081×10^{-7}
Location (node)		1	1	1

As is evident in Table 3, the results from all three FEA codes match each other well, therefore, it is safe to say the numerically integrated CST element is sufficiently close to the volumetrically integrated CST element and will be used throughout this investigation moving forward.

Also as is evident in Table 3, the FEA results are not reasonably close to the analytical solution, therefore, it is required to perform a mesh refinement to ensure the results from the FEA code are converging on the analytical solution. Below in Figure 19 and Figure 20, a more finely discretized model has been analyzed. In Table 4 the results of the mesh refinement are compared to the analytical solution.

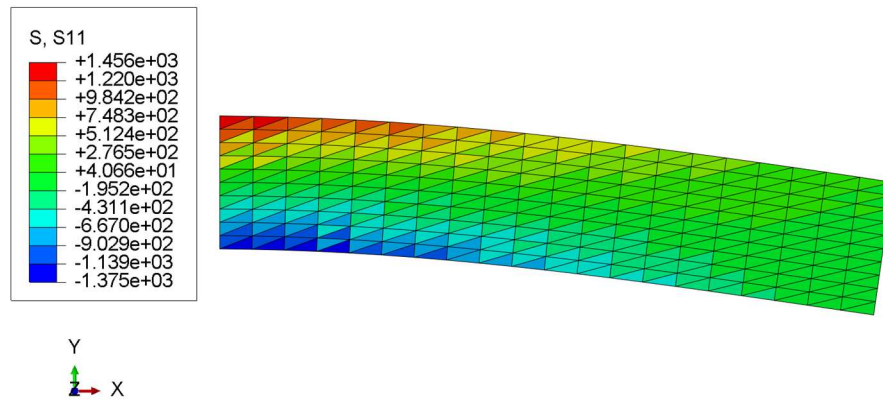


Figure 19. Abaqus volumetric integration fine mesh beam results for stress in the x direction.

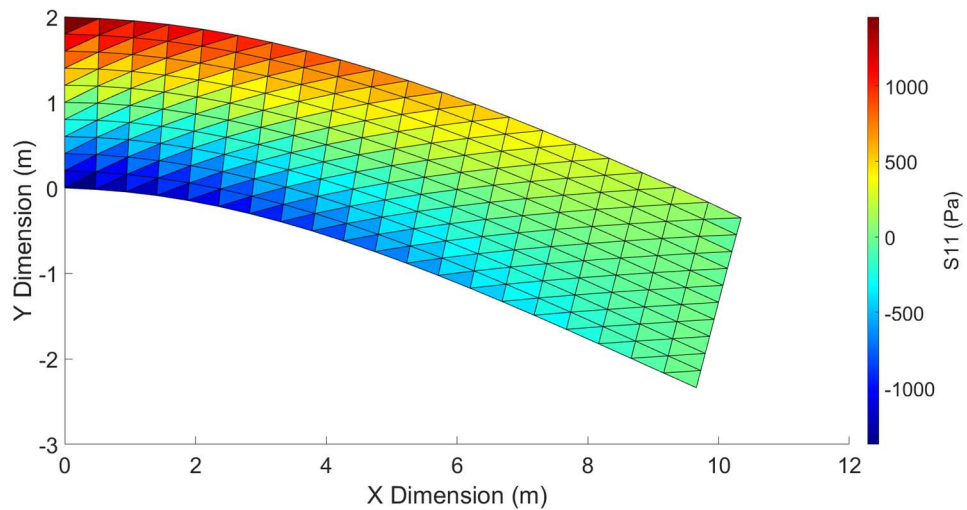


Figure 20. Matlab numerical integration fine mesh beam results for stress in the x direction.

Table 4. Numerical vs. volumetric integration results for mechanically loaded fine mesh beam.

Value		Volumetric Integration	Numerical Integration
	Theory	Abaqus	Matlab
Max Stress (Pa)	1500	1455.92	1455.9244
Location (element)		39	39
Min Stress (Pa)	-1500	-1374.61	-1374.6134
Location (element)		400	400
Max Strain	7.5×10^{-9}	6.4868×10^{-9}	6.4868×10^{-9}
Location (element)		39	39
Min Strain	-7.5×10^{-9}	-6.7550×10^{-9}	-6.7550×10^{-9}
Location (element)		400	400
Max Disp. (m)	2.5×10^{-7}	2.3759×10^{-7}	2.3759×10^{-7}
Location (node)		1	1

The results in Table 4 show that the mesh refinement provided results much closer to that of the analytical results. It is safe to say that the numerically integrated CST element code is converging towards the analytical solution and is sufficiently close to Abaqus and the analytical solution such that the investigation will move forward with verification of thermal stress and strain in the CST element formulation.

3.1.1.2 Verify element by adding thermal load

3.1.1.2.1 Fixed-free beam (fixed in X direction only)

The next case to compare analytical, Abaqus, and Matlab results is an eigenstrain problem using a simple cantilever beam with a fixed end subject to a uniform 1000 °C thermal load. The beam is fixed in the x direction only and is allowed to expand in the y direction on both top and bottom of the beam. To eliminate any rigid body translations, a single node in the middle of the beam at the left end has also been fixed in the y direction. This comparison ensures that the CST code is functioning when a thermal load is present, and strains are developed throughout the beam. However, due to the boundary conditions, no stresses are present. Figure 21 shows the geometry used in the analysis and Figure 22 shows the anticipated deformed shape once the load has been applied. Initial geometric values for the beam are given by:

$$L_o = 10 \text{ m} \quad (3.15)$$

$$h = 2 \text{ m} \quad (3.16)$$

$$t = 1 \text{ m} \quad (3.17)$$

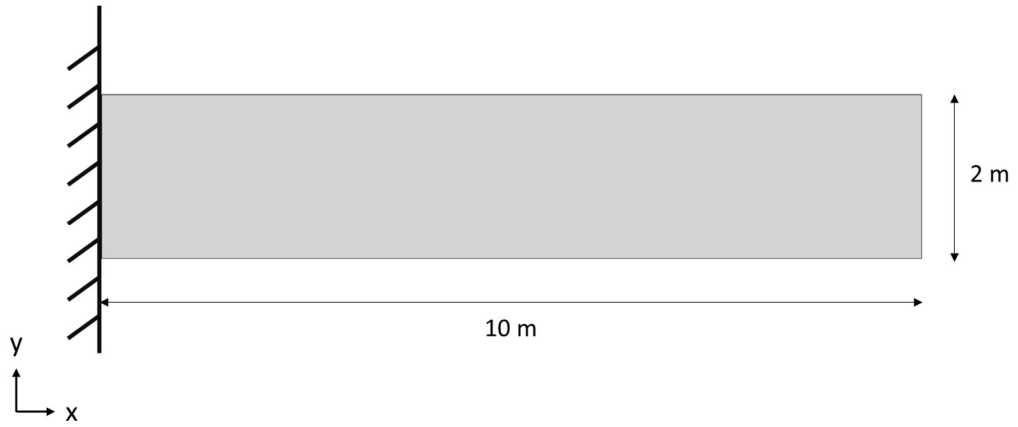


Figure 21. Initial geometry of a 2D beam fixed in only the x direction on the left end, allowing expansion in both the x and y directions.

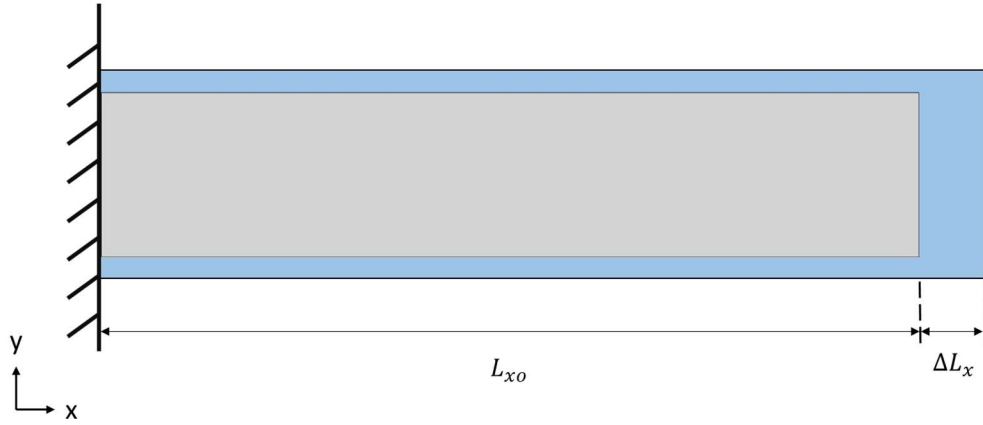


Figure 22. Initial (gray) and deformed (blue) geometry of a 2D beam fixed in only the x direction on the left end, subject to a uniform temperature increase of 1000°C .

The change in length for the beam is defined by:

$$\varepsilon = \frac{\Delta L_x}{L_o} \quad (3.18)$$

$$\varepsilon = \alpha \Delta T \quad (3.19)$$

Therefore:

$$\Delta L_x = \alpha \Delta T L_o \quad (3.20)$$

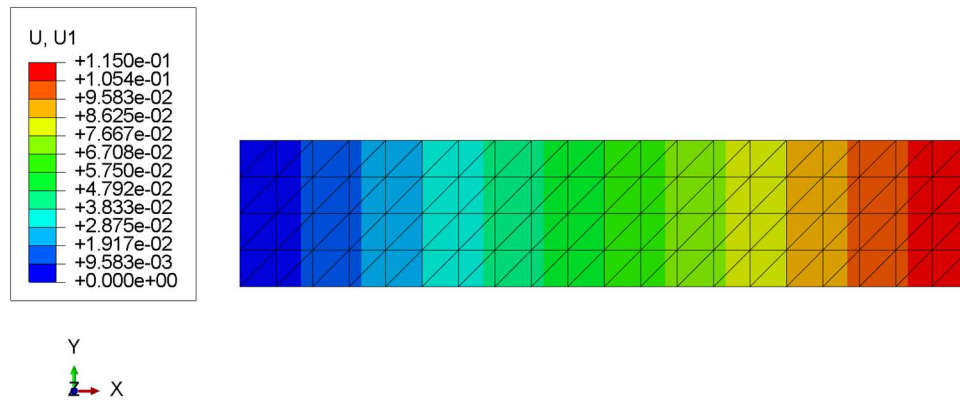


Figure 23. Abaqus fixed-free beam with uniformly distributed thermal load results for displacement in the x direction.

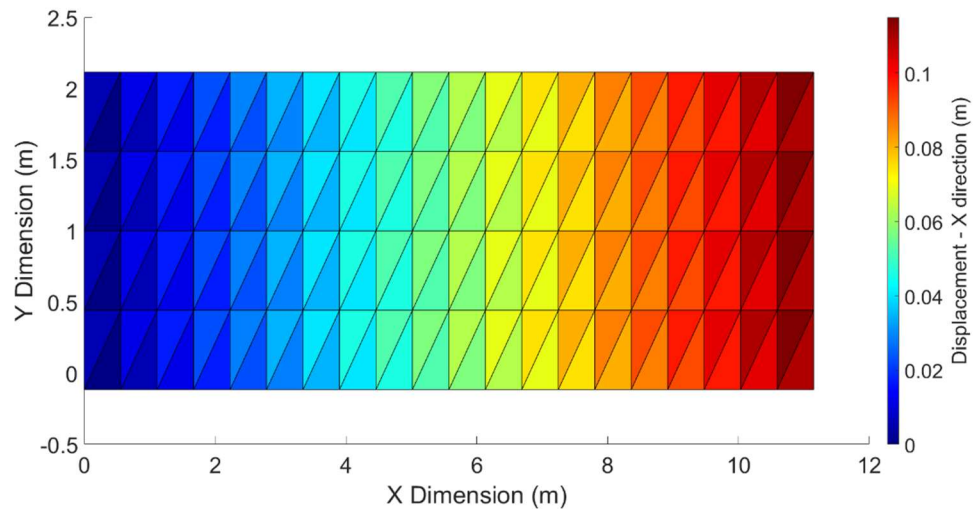


Figure 24. Matlab fixed-free beam with uniformly distributed thermal load results for displacement in the x direction.

Table 5. Fixed-free results for uniformly distributed thermal load fine mesh beam.

Value	Theory	Abaqus	Matlab
Max Stress (Pa)	0	8.4195×10^{-7}	4.6×10^{-5}
Location (element)		119	122
Min Stress (Pa)	0	-9.7238×10^{-7}	-9.4×10^{-5}
Location (element)		39	1
Max Strain	0.0115	0.0115	0.0115
Location (element)		160	122
Min Strain	0.0115	0.0115	0.0115
Location (element)		160	1
Max Disp. (m)	0.115	0.1168	0.1155
Location (node)		21	21

As is evident in Table 5, the results from both FEA codes match each other well, and also match the analytical solution closely. Stress values (maximum and minimum) are effectively zero and the discrepancies in element location between Matlab and Abaqus results are due to differences in numerical noise arising from many elements having the same value.

3.1.1.2.2 Fixed-fixed beam (fixed in X direction only)

The next case to compare analytical, Abaqus, and Matlab results is a beam with both ends fixed subject to a uniform $1000\text{ }^{\circ}\text{C}$ thermal load. The beam is fixed in the x direction only and is allowed to expand in the y direction on both top and bottom of the beam. This comparison ensures that the CST code is functioning when a thermal load is present, and stresses and strains are developed due to the boundary conditions. Figure 25 shows the geometry used in the analysis and Figure 26 shows the anticipated deformed shape once the load has been applied. Initial geometric values for the beam are given by:

$$L_o = 10\text{ m} \quad (3.21)$$

$$h = 2\text{ m} \quad (3.22)$$

$$t = 1\text{ m} \quad (3.23)$$

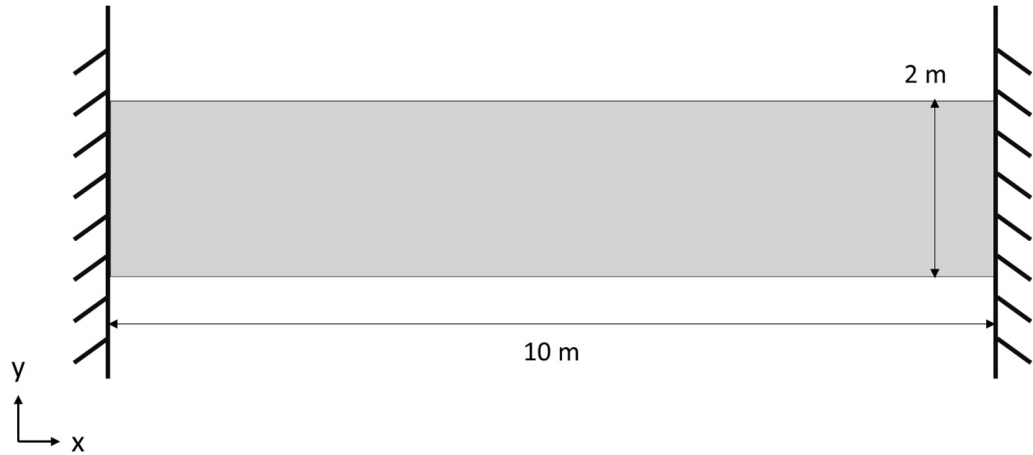


Figure 25. Initial geometry of a 2D beam fixed in only the x direction on both ends .

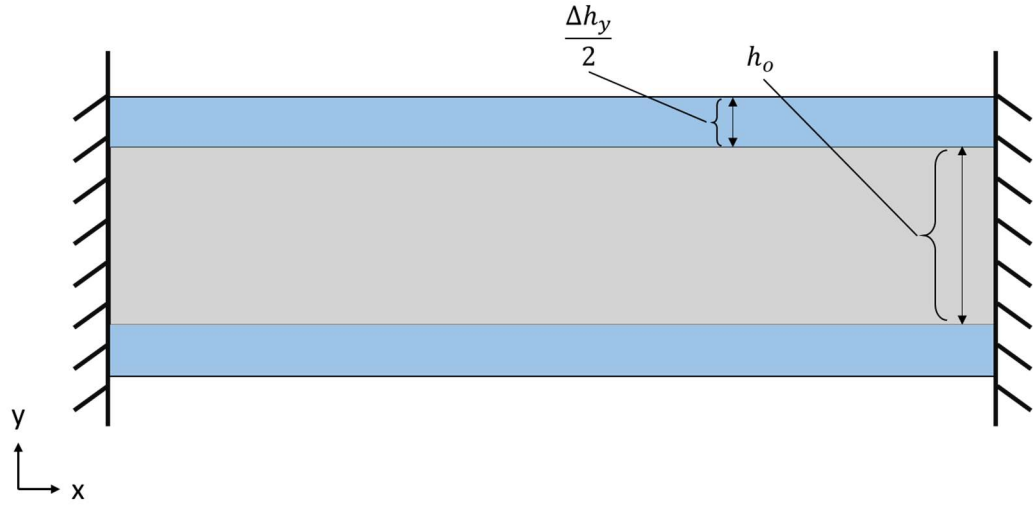


Figure 26. Initial (gray) and deformed (blue) geometry of a 2D beam, fixed in only the x direction on both ends, subject to a uniform temperature increase of 1000°C .

The stress developed in the beam due to being restricted in the x direction is defined by:

$$\sigma = E\varepsilon \quad (3.24)$$

$$\sigma = E \frac{\Delta L_x}{L_o} \quad (3.25)$$

$$\Delta L_x = \alpha \Delta T L_o \quad (3.26)$$

where ΔL_x is the change in length the beam would undergo if it weren't restricted in the x direction. Therefore:

$$\sigma = E\alpha\Delta T \quad (3.27)$$

The change in height for the beam is defined by:

$$\varepsilon = \frac{\Delta h_y}{h_o} \quad (3.28)$$

$$\varepsilon = \alpha\Delta T \quad (3.29)$$

therefore:

$$\Delta h_y = \alpha \Delta T h_o \quad (3.30)$$

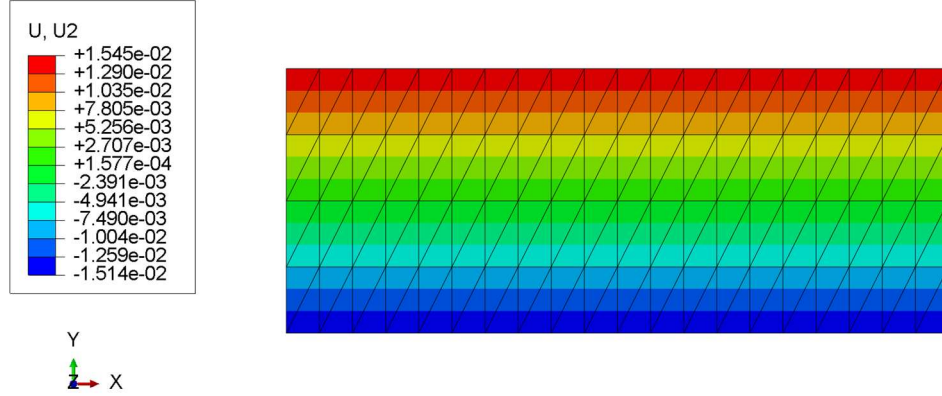


Figure 27. Abaqus fixed-fixed beam with uniformly distributed thermal load results for displacement in the y direction.

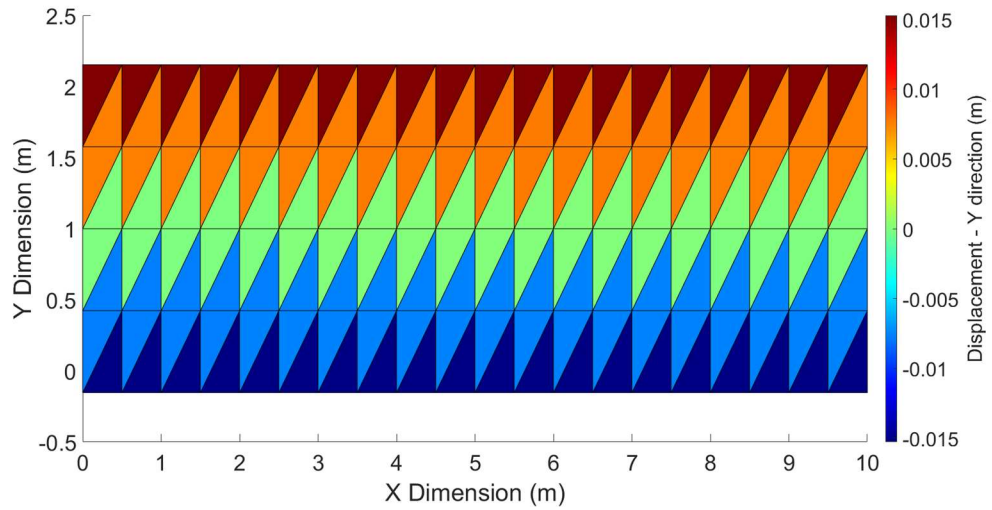


Figure 28. Matlab fixed-fixed beam with uniformly distributed thermal load results for displacement in the y direction.

Table 6. Fixed-fixed results for uniformly distributed thermal load fine mesh beam.

Value	Theory	Abaqus	Matlab
Max Stress (Pa)	-2.3×10^9	-2.3×10^9	-2.299×10^9
Location (element)		160	39
Min Stress (Pa)	-2.3×10^9	-2.3×10^9	-2.3×10^9
Location (element)		160	126
Max Strain	0.0115	0.0152	0.0152
Location (element)		160	110
Min Strain	0.0115	0.0152	0.0152
Location (element)		160	79
Max Disp. (m)	0.0115	0.0154	0.0152
Location (node)		105	97

Maximum displacement happens in the Y-direction along the top and bottom of the beam, and is equal to 0.0115 m. As is evident in Figure 29 (node numbering in black) the location of node 105 and node 97 are both on the top edge of the beam and have effectively the same displacement. Node numbering is congruent from Abaqus to Matlab and in this analysis the stress and strain gradients are both zero, therefore the differences between nodes of max & min: displacement, stress, and strain are due to differences in numerical noise.

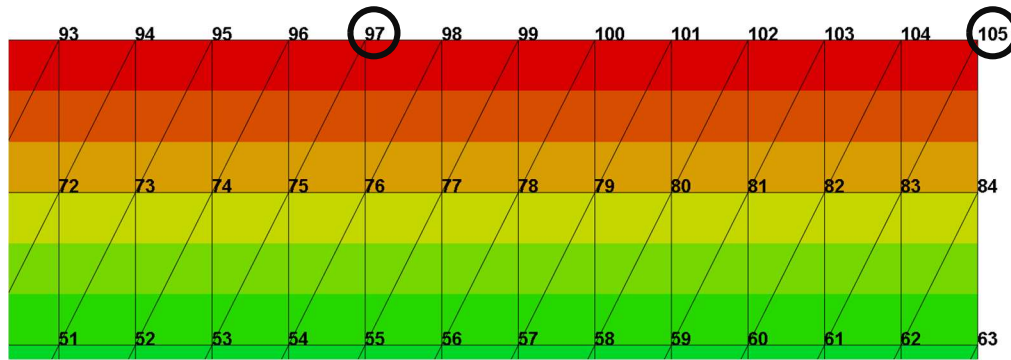


Figure 29. Fixed-fixed beam node numbering shows equivalent displacements along the top edge.

The FEA displacement results are very close, but not exact when compared to theoretical calculations which is due to a lack of elements through the height of the beam, a known issue with beam geometries in FEA. However, the stress values from Abaqus, Matlab, and analytical solutions are sufficiently close to move on to verification using more generalized cases.

3.1.1.2.3 Fixed-fixed beam (fixed in X & Y)

The next case to compare analytical, Abaqus, and Matlab results is a beam with both ends fixed subject to a uniform $1000\text{ }^{\circ}\text{C}$ thermal load. The beam is fixed in the x & y directions. This comparison ensures that the CST code is functioning when a thermal load is present, and stresses and strains are developed due to the boundary conditions. Figure 25 shows the geometry used in the analysis and Figure 30 shows the anticipated deformed shape once the load has been applied. Initial geometric values for the beam are given by:

$$L_o = 10\text{ m} \quad (3.31)$$

$$h = 2\text{ m} \quad (3.32)$$

$$t = 1\text{ m} \quad (3.33)$$

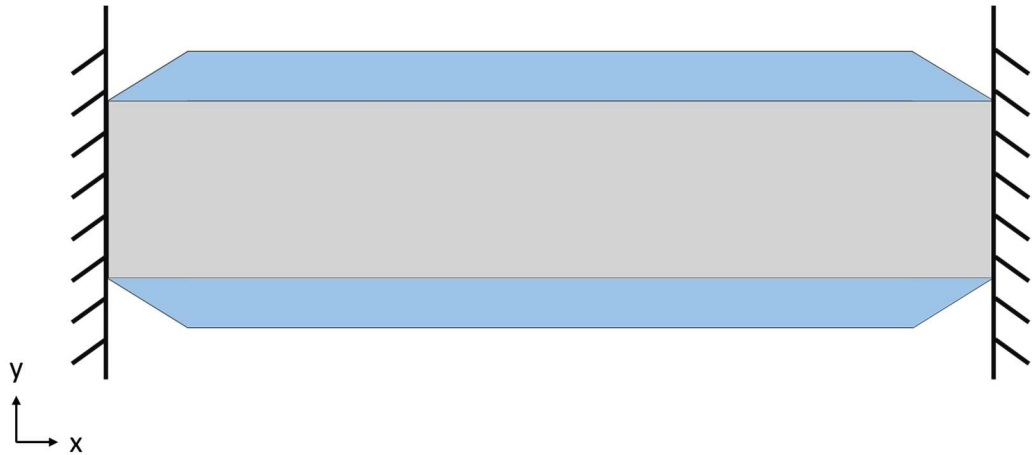


Figure 30. Initial (gray) and deformed (blue) geometry of a 2D beam fixed in both directions on both ends, subject to a uniform temperature increase of $1000\text{ }^{\circ}\text{C}$.

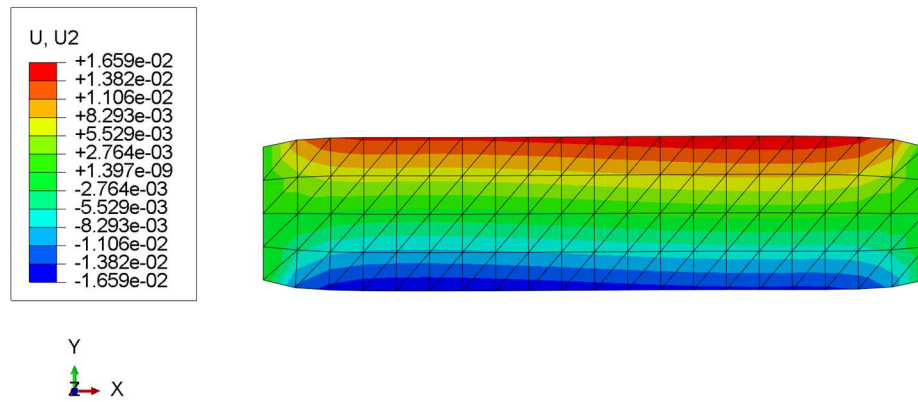


Figure 31. Abaqus fixed u1 & v1 beam with uniformly distributed thermal load results for displacement in the y direction.

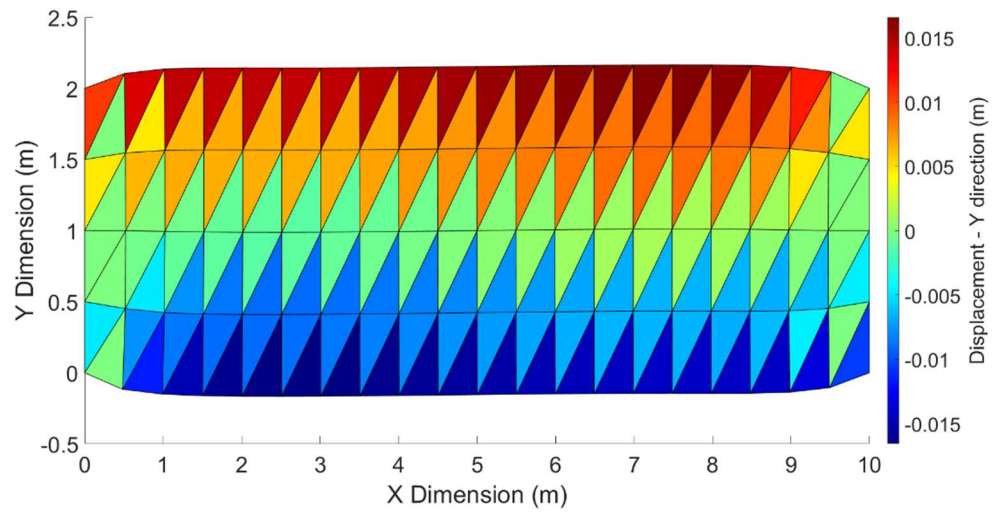


Figure 32. Matlab fixed u1 & v1 beam with uniformly distributed thermal load results for displacement in the y direction.

Table 7. Fixed $u1$ & $v1$ results for uniformly distributed thermal load fine mesh beam.

Value	Abaqus	Matlab
Max Mises Stress (Pa)	4.22959×10^9	4.22959×10^9
Location (element)	122	122
Min Mises Stress (Pa)	1.4818×10^9	1.4818×10^9
Location (element)	81	80
Max Strain	0.0156	0.0156
Location (element)	90	74
Min Strain	0	0
Location (element)	39	122
Max Disp. (m)	0.0166	0.0166
Location (node)	100	100

As shown in Table 7, Abaqus and Matlab code are in close agreement for a true fixed-fixed beam geometry and boundary conditions. The discrepancies in element location between Abaqus and Matlab are shown in Figure 33 with element numbering in white. The stress, strain, and displacement values have symmetry which explains why element 39 and 122 are given as minimum strain locations but have the same value, for example.

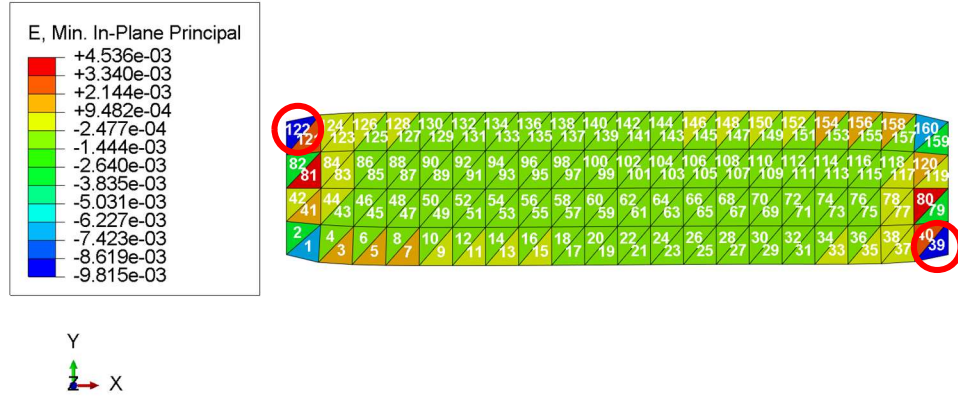


Figure 33. Element numbering and minimum in-plane principal strain values show the lines of symmetry for a fixed-fixed beam subject to a uniform thermal load.

The next verification is to use this code to analyze a generalized case such as a disk with a fixed inner diameter subject to a uniform temperature increase . Then compare the results with Abaqus and theoretical calculations.

3.1.1.2.4 2D generalized disk

The two dimensional generalized case to compare analytical, Abaqus, and Matlab results is a disk with a fixed inner diameter subject to a 1000 °C uniform temperature increase. The disk is fixed in both the x & y directions to prevent rigid body translations. This comparison verifies that the CST code is functioning when a thermal load is present in a generalized, non-orthogonal mesh. Figure 34 shows the geometry used in the analysis and Figure 35 shows the anticipated deformed shape once the load has been applied. Initial geometric values for the disk are given by:

$$r_1 = 2.5 \text{ m} \quad (3.34)$$

$$r_2 = 5 \text{ m} \quad (3.35)$$

$$t = 1 \text{ m} \quad (3.36)$$

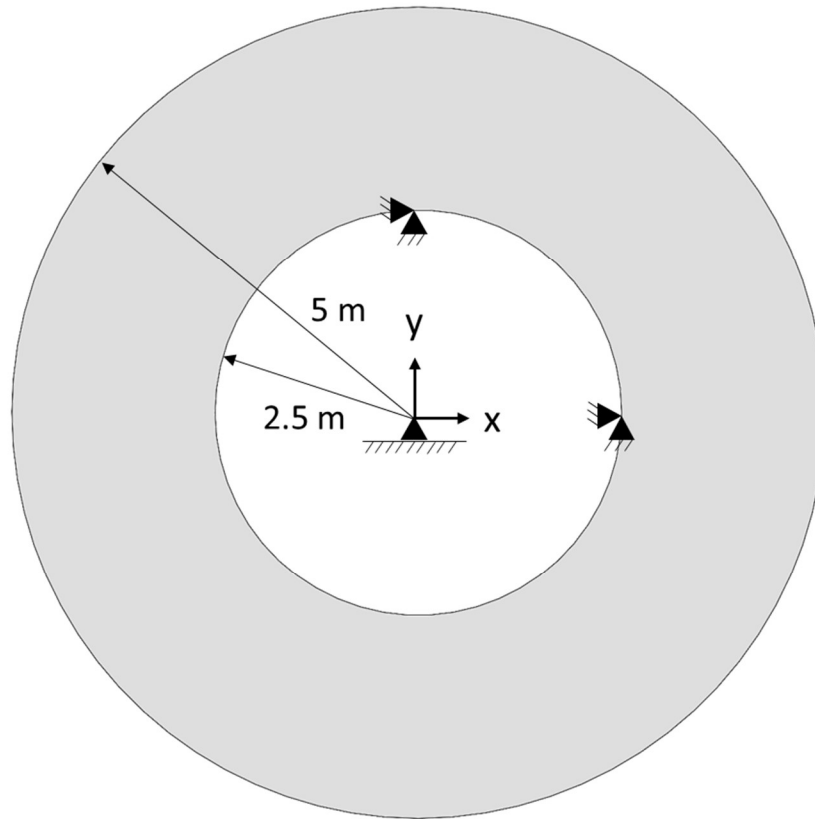


Figure 34. Initial geometry of a 2D disk fixed in both directions on the inner diameter.

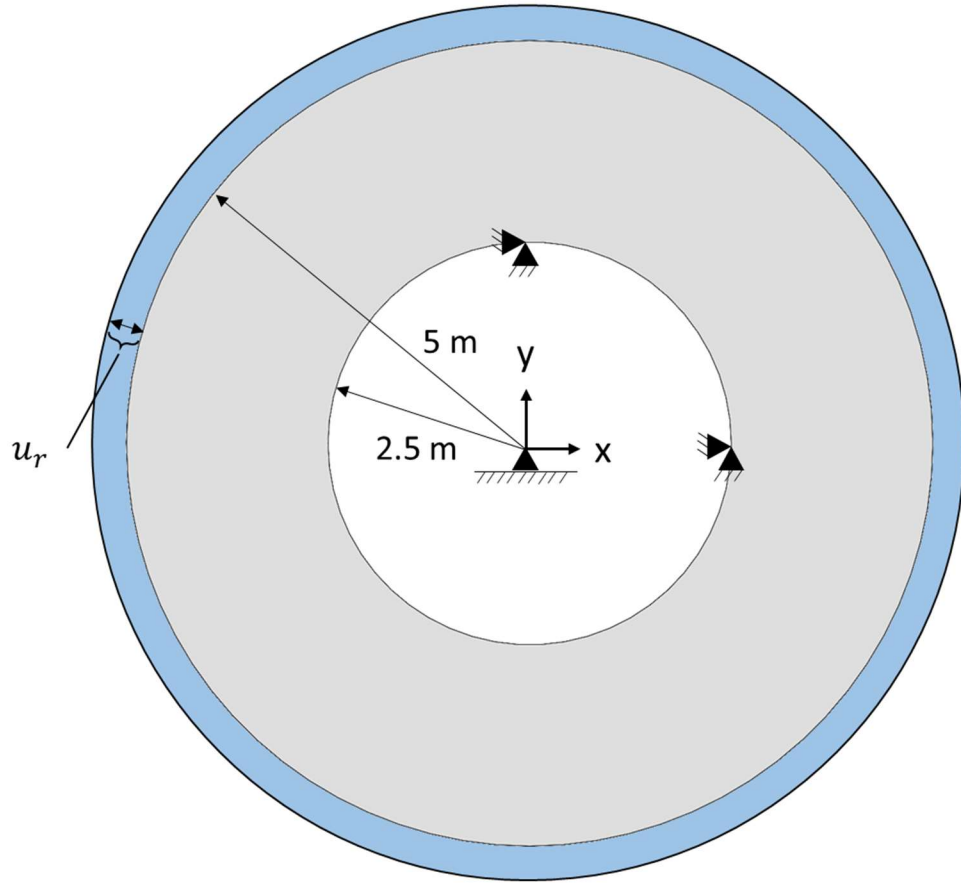


Figure 35. Initial (gray) and deformed (blue) geometry of a 2D disk fixed in both directions on the inner diameter, subject to a uniform temperature increase of 1000 °C.

The two dimensional analysis of this geometry is considered plane stress because the disk is very short in the z direction ($t = 1$) and the surfaces in the z direction are traction free [34]. Therefore:

$$\sigma_{zz} = 0 \quad (3.37)$$

The radial stress values are defined by:

$$\sigma_{rr} = -\frac{E\alpha}{r^2} \int rT dr + A + \frac{B}{r^2} \quad (3.38)$$

The circumferential stress values are defined by:

$$\sigma_{\theta\theta} = \frac{E\alpha}{r^2} \int rT dr - E\alpha T + A - \frac{B}{r^2} \quad (3.39)$$

The radial strain values are defined by:

$$\varepsilon_{rr} = \frac{\sigma_{rr}}{E} - \frac{\nu\sigma_{\theta\theta}}{E} + \alpha\Delta T \quad (3.40)$$

The circumferential strain values are defined by:

$$\varepsilon_{\theta\theta} = \frac{\sigma_{\theta\theta}}{E} - \frac{\nu\sigma_{rr}}{E} + \alpha\Delta T \quad (3.41)$$

The radial displacement values are defined by:

$$u_r = \frac{\alpha(1+\nu)}{r} \int Tr dr + \frac{A(1-\nu)r}{E} - \frac{(1+\nu)B}{Er} \quad (3.42)$$

And circumferential displacement is considered to be zero.

A & B are constants of integration and are typically found using known conditions at an inner or outer radii of the disk [34]. To find A & B , use the known boundary conditions to solve for A & B simultaneously.

$$\sigma_{rr}(r = 5) = 0 \quad (3.43)$$

$$u_r(r = 2.5) = 0 \quad (3.44)$$

Then substitute both A & B into either stress equation to solve for unknown stress values.

A Matlab code was developed to solve the analytical solutions using discrete radial values with a step size of:

$$\Delta r = \frac{r_1 + r_2}{50} \quad (3.45)$$

Results are shown in Figure 36 which display stress, strain, and displacement values along the entire radius of the disk.

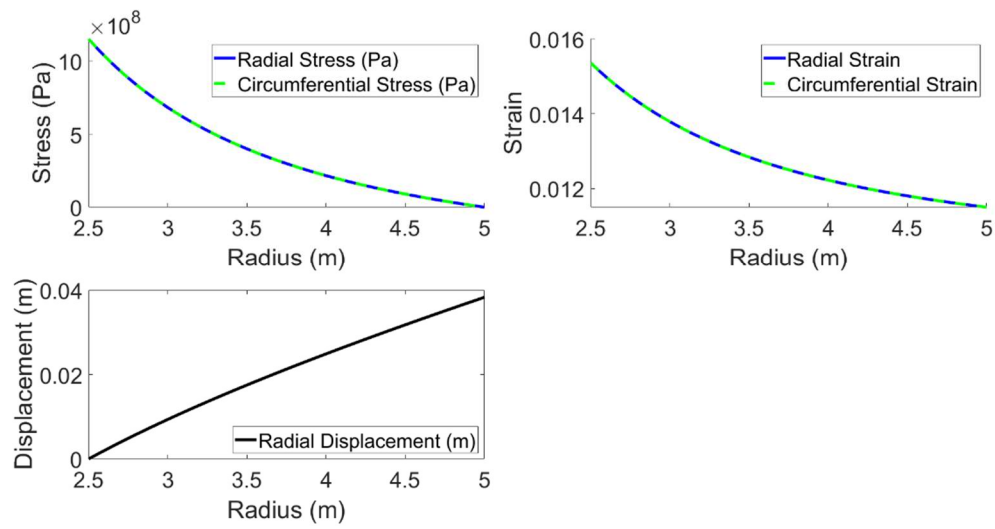


Figure 36. Analytical results for fixed inner diameter disk subject to uniformly distributed thermal load.

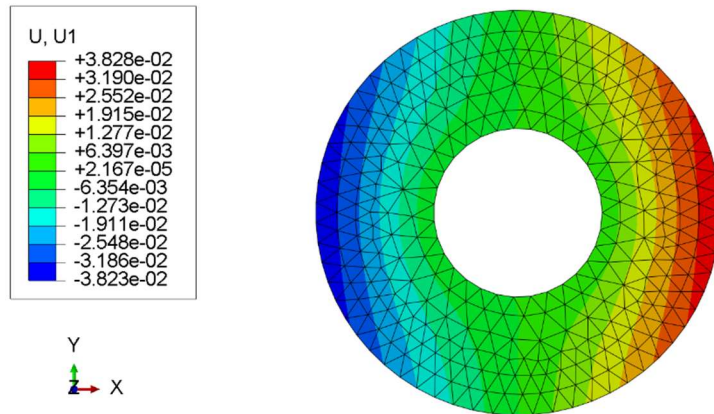


Figure 37. Abaqus fixed u1 & v1 inner diameter disk with uniformly distributed thermal load showing results for displacement in the x direction.

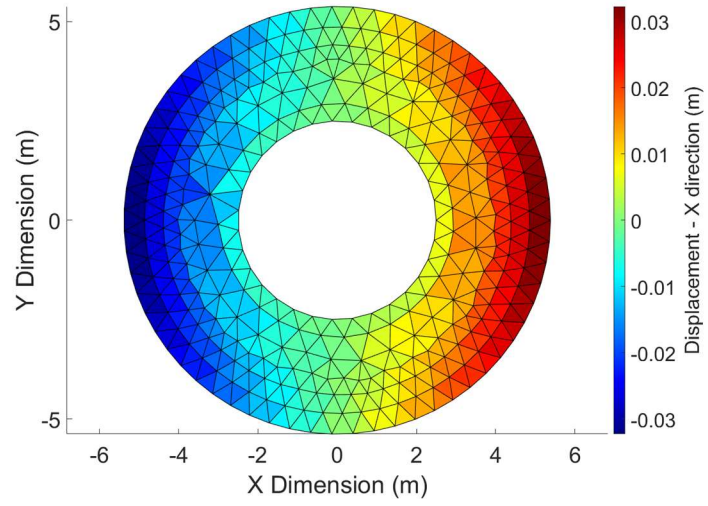


Figure 38. Fixed u_1 & v_1 inner diameter Matlab disk with uniformly distributed thermal load showing results for displacement in the x direction.

Table 8. Fixed u1 & v1 inner diameter results for uniformly distributed thermal load.

Value	Theory	Abaqus	Matlab
Max Radial Stress (Pa)	1.1519×10^9	1.1735×10^9	1.1735×10^9
Location (element)		51	51
Min Radial Stress (Pa)	0	-2.0615×10^9	-2.0615×10^9
Location (element)		107	107
Max Radial Strain	0.01535	0.01967	0.01967
Location (element)		202	202
Min Radial Strain	0.0115	0.01291	0.01291
Location (element)		137	137
Max Disp. (m)	0.0383	0.0383	0.0383
Location (node)		68	68

As displayed in Table 8, Abaqus, Matlab, and Theory are all in close agreement for a generalized non-orthogonal mesh representative of an aircraft stator which has fixed inner diameter and is subject to a uniform thermal load. Because the CST element verification was successful in all of the previously discussed cases, the investigation will continue on with expanding the CST element to three dimensions to verify a tet4 element. Once element verification has been completed and sufficiently accurate results have been

documented using both element types, the investigation will move to implementing the new element definitions in Hotspotter and validating a few trial cases against the legacy Hotspotter elements.

3.1.2 First-order tetrahedral element

3.1.2.1 Verify 3D generalized disk

The three dimensional analysis considers the case of a disk with a fixed inner diameter (fixed in x & y directions), where the z direction is allowed to expand due to the plane stress assumption. The three dimensional analysis of this geometry is considered plane stress because the disk is very short in the z direction ($t = 1$) and the surfaces in the z direction are traction free [34]. Therefore, the analytical method and solutions to the two dimensional analysis are the same for three dimensional case here as well.

Figure 39 displays the initial geometry (gray) and deformed geometry (green) in Abaqus and Figure 40 shows the von Mises stress in each element for the Abaqus analysis in Cartesian coordinates.

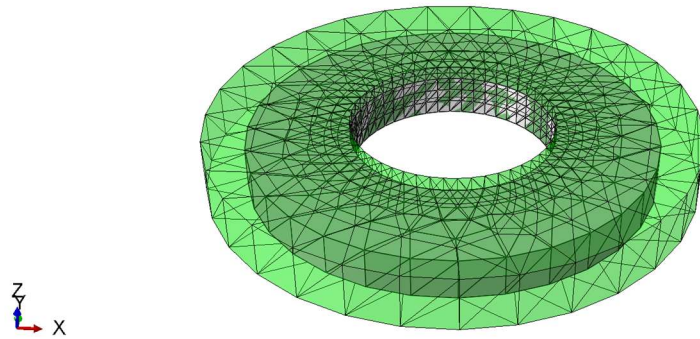


Figure 39. Initial (gray) and deformed (green) geometries of a fixed u_I & v_I inner diameter disk subject to uniform thermal load.

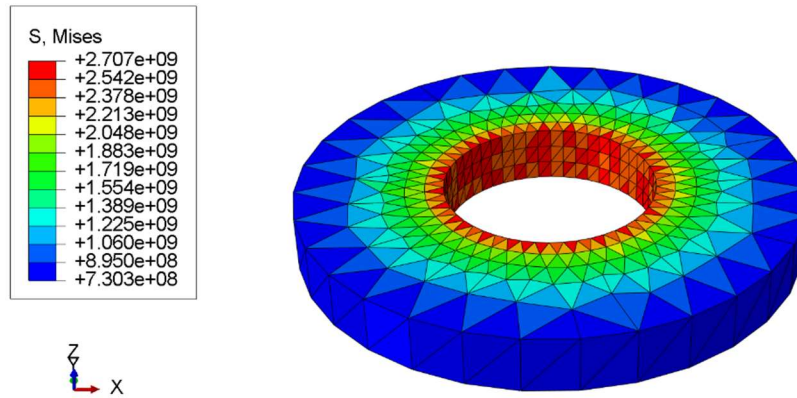


Figure 40. Abaqus analysis for a fixed u_I & v_I inner diameter disk in Cartesian coordinates subject to uniform thermal load showing results for von Mises stress.

Figure 41 shows the von Mises stress when using the Matlab code tet4 element in Cartesian coordinates and Table 9 displays the results of all three analysis methods.

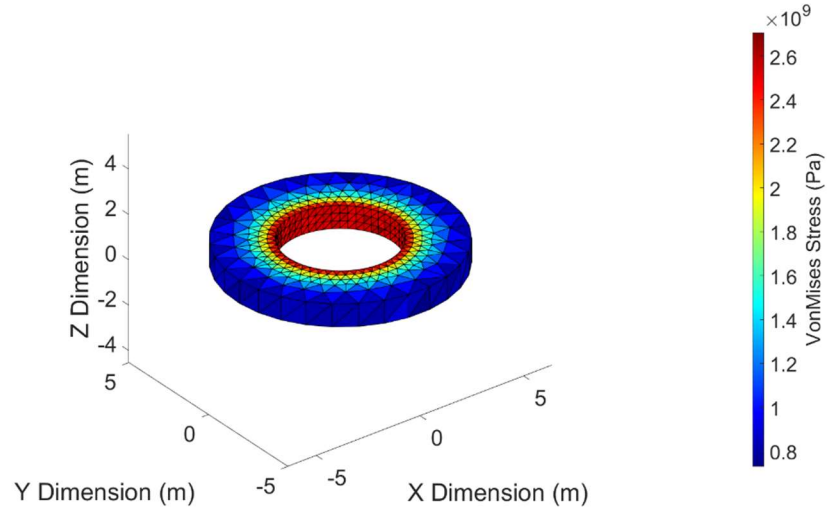


Figure 41. Matlab analysis for a fixed u1 & v1 inner diameter disk in Cartesian coordinates subject to uniform thermal load showing results for von Mises stress.

Table 9. Results for a fixed u1 & v1 inner diameter three dimensional disk with uniformly distributed thermal load.

Value	Theory	Abaqus	Matlab
Max Radial Stress (Pa)	1.1519×10^9	1.2849×10^9	1.2849×10^9
Location (element)		2117	2117
Min Radial Stress (Pa)	0	-2.1559×10^9	-2.1559×10^9
Location (element)		2664	2664
Max Radial Strain	0.01535	0.02009	0.02009
Location (element)		2117	2117
Min Radial Strain	0.0115	2.3162×10^{-6}	2.3162×10^{-6}
Location (element)		1640	1640
Max Disp. (m)	0.0383	0.0403	0.0403
Location (node)		140	140

Abaqus and Matlab are in reasonably close agreement to each other, as well as to the analytical solution. The investigation will continue on to element incorporation into Hotspotter and further validation against legacy Hotspotter elements.

3.2 Element validation against legacy Hotspotter elements

As simulation across many fields of engineering is growing, model verification and validation are becoming increasingly more important [35]. One question on the mind of every simulation user is whether or not the model adequately depicts reality [36]. Model validation is the essential process of determining how closely a model represents the real world from the perspective of the user [35]. According to [37], it is relatively easy to develop a numerical model, but validation that the model is an accurate representation of the problem and is useful for real world problem solving is more difficult. The legacy hex and quad elements in Hotspotter have been validated and shown to be sufficiently close to real world data. Therefore, the next step in this investigation is to compare the newly developed triangular and tetrahedral elements to the validated legacy quad and hex elements in Hotspotter.

3.2.1 Triangular element validation

All of the following simulations use the same stopping criteria in Hotspotter. Stopping criteria used in the bisection search method employed by Hotspotter is specified in the input file; and for this investigation is set at 0.5 percent. The stopping criteria in a bisection search method dictates how close to a solution the algorithm must be before stopping. This means that the reported results for critical velocity are within 0.5 percent of the actual result from the function being numerically estimated.

3.2.1.1 Quadrilateral pad and rotor model

This model validation run uses quadrilateral elements for the pad and the rotor. This model has been validated with real world experiments and data. Mesh and TEI mode

shapes are shown in Figure 42 and Figure 43, respectively. Wave number, i.e. number of hotspots, and critical speed results are shown in Table 10.

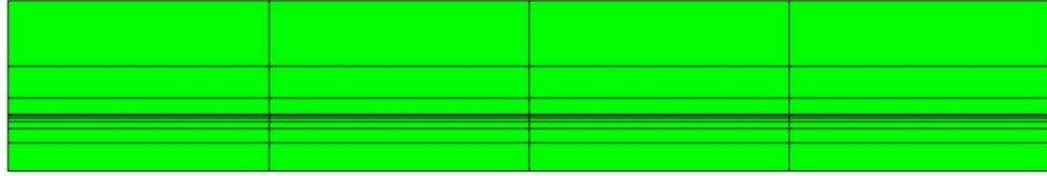


Figure 42. Mesh of an axisymmetric disk pad and rotor model using all quadrilateral elements in Hotspotter 3D.

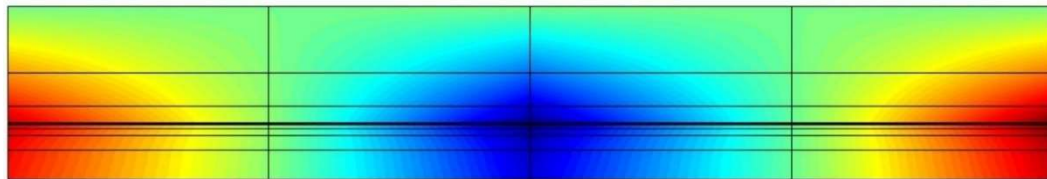


Figure 43. First mode shape results of an axisymmetric disk pad and rotor model using all quadrilateral elements in Hotspotter 3D.

The visual representation of the mode shapes, output by Hotspotter and shown in Figure 43, Figure 45, Figure 47, & Figure 49, are contour plots of the eigenmode with the largest growth rate of the analysis [17]. In an eigenvalue analysis, relative values or eigenvectors are important instead of absolute values and therefore a legend is not required on the output plots. The eigenvector shows the direction in which instability occurs and the eigenvalue is the factor by which the eigenvector is scaled. With this in mind, the plots in Figure 43, Figure 45, Figure 47, & Figure 49 are relative values representing the first mode shape of the system. Mode shapes are important as they show how a system responds to inputs. In the case of Hotspotter, the mode shapes show the systems spatial distribution of the dominant eigenmode in the cross-sectional plane of the geometry [17]; the colors represent relative perturbation temperature distribution.

Table 10. Wave number and critical speed results of an axisymmetric disk using all quadrilateral elements in Hotspotter 3D.

Wave Number	Critical Speed (radians/second)
0	351.6
2	693.4
4	580.1
6	311.5
8	205.1
10	168.9
12	167.5
14	188.0
16	224.6
18	272.5
20	331.1
22	404.3
24	505.9

3.2.1.2 Triangular pad and rotor model

This model validation run uses CST elements for the pad and the rotor. Mesh and TEI mode shapes are shown in Figure 44 and Figure 45, respectively. Wave number and critical speed results are shown in Table 11.

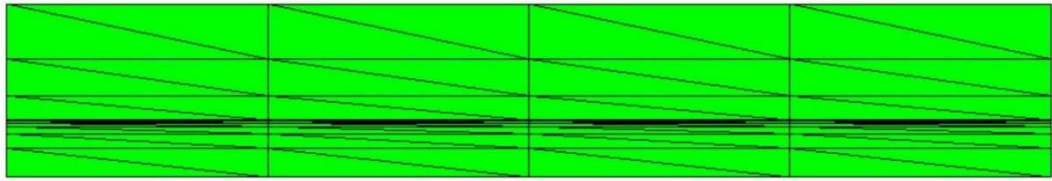


Figure 44. Mesh of an axisymmetric disk pad and rotor model using all triangular elements in Hotspotter 3D.

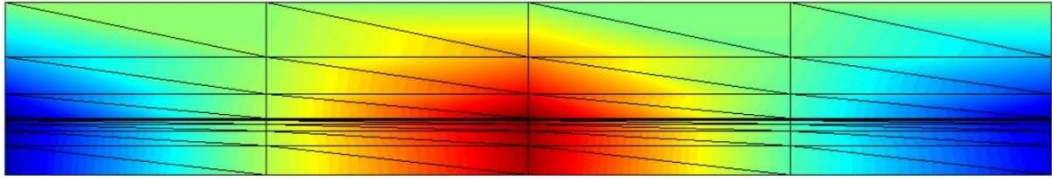


Figure 45. First mode shape results of an axisymmetric disk pad and rotor model using all triangular elements in Hotspotter 3D.

Table 11. Wave number and critical speed results of an axisymmetric disk using all triangular elements in Hotspotter 3D.

Wave Number	Critical Speed (radians/second)
0	494.1
2	953.1
4	662.1
6	320.3
8	204.1
10	167.0
12	165.5
14	186.0
16	225.6
18	282.2
20	354.5
22	437.5
24	533.2

3.2.1.3 Quad pad, tri rotor model

This model validation run uses quadrilateral elements for the pad and CST elements for the rotor. Mesh and TEI mode shapes are shown in Figure 46 and Figure 47, respectively. Wave number and critical speed results are shown in Table 12.

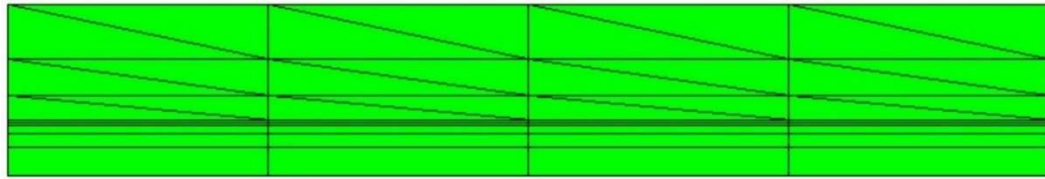


Figure 46. Mesh of an axisymmetric disk pad and rotor model using quadrilateral and triangular elements, respectively, in Hotspotter 3D.

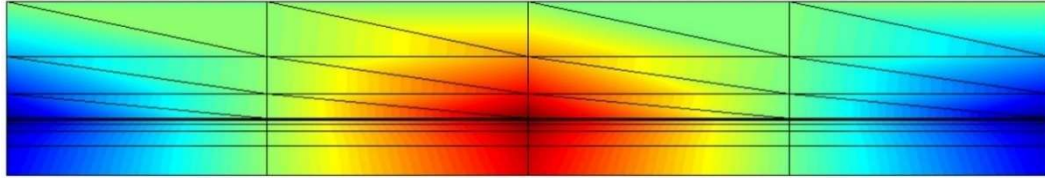


Figure 47. First mode shape results of an axisymmetric disk pad and rotor model using quadrilateral and triangular elements, respectively, in Hotspotter 3D.

Table 12. Wave number and critical speed results of an axisymmetric disk using quadrilateral and triangular elements, respectively, in Hotspotter 3D.

Wave Number	Critical Speed (radians/second)
0	634.8
2	1422.0
4	693.4
6	323.2
8	204.1
10	167.0
12	166.0
14	187.0
16	224.6
18	274.4
20	335.0
22	412.1
24	515.6

3.2.1.4 Tri pad, quad rotor model

This model validation run uses CST elements for the pad and quadrilateral elements for the rotor. Mesh and TEI mode shapes are shown in Figure 48 and Figure 49, respectively. Wave number and critical speed results are shown in Table 13.

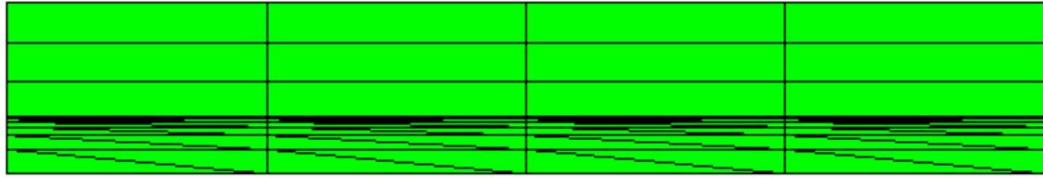


Figure 48. Mesh of an axisymmetric disk pad and rotor model using triangular and quadrilateral elements, respectively, in Hotspotter 3D.

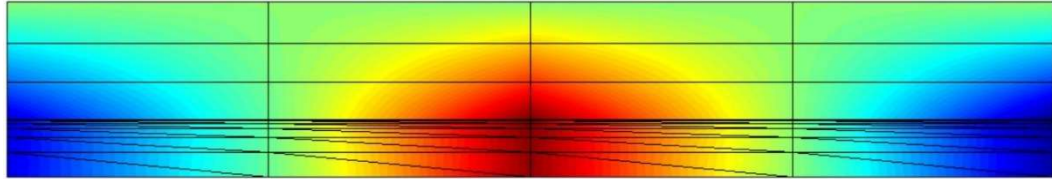


Figure 49. First mode shape results of an axisymmetric disk pad and rotor model using triangular and quadrilateral elements, respectively, in Hotspotter 3D.

Table 13. Wave number and critical speed results of an axisymmetric disk using triangular and quadrilateral elements, respectively, in Hotspotter 3D.

Wave Number	Critical Speed (radians/second)
0	265.6
2	464.8
4	488.3
6	304.7
8	205.1
10	169.4
12	168.0
14	185.5
16	216.8
18	259.8
20	314.5
22	388.7
24	509.8

3.2.1.5 Comparing all models

Lowest critical speed, associated wave number, and percent error are reported in Table 14. In designing a system, engineers are primarily concerned with the lowest critical sliding speed. Concern only for the lowest critical speed is because regardless of other faster critical speeds, TEI has been excited at the lowest critical speed during normal operation of the system. Therefore, comparison of different wave numbers is not

typically performed. The results of all validation simulations indicate that TEI will be excited for this geometry around a velocity of 167.5 radians per second, corresponding to a mode which presents 12 periodic hot spots around the circumference of the disk. For this design, an engineer would need to consider the speed at which the system may operate. If the system is likely to see such speeds, then the design needs to be rebuilt in such a way as to increase the critical velocity of the system. However, if the system is not likely to see such speeds during use, then the design process may move on to subsequent steps.

Percent error in the result compared to the validated result (all quad model) is calculated using:

$$\delta = \left| \frac{v_A - v_E}{v_E} \right| * 100 \quad (3.46)$$

where v_E is the expected value, i.e. the result when using all quad elements, v_A is the result obtained from the new simulation using tri elements, and δ is the percent error in the critical velocities. Percent error as reported in Table 14 compares the differences between each simulation and the expected result from the all quad model. This percent error is different than the stopping criteria used in each simulation.

Results show that no matter the element choice, singular or in combination, and selection of elements relative to each part, stator or rotor, the results for lowest critical sliding speed and wave number are within 1.5 percent of the expected value.

Table 14 Wave number and lowest critical speed results of all element types.

Model	Dominant Mode Wave Number	Lowest Critical Speed (rad/sec)	Percent Error (%)
All quad model	12	167.5	0.00
All tri model	12	165.5	1.19
Quad pad, tri rotor model	12	166.0	.895
Tri pad, quad rotor model	12	168.0	.298

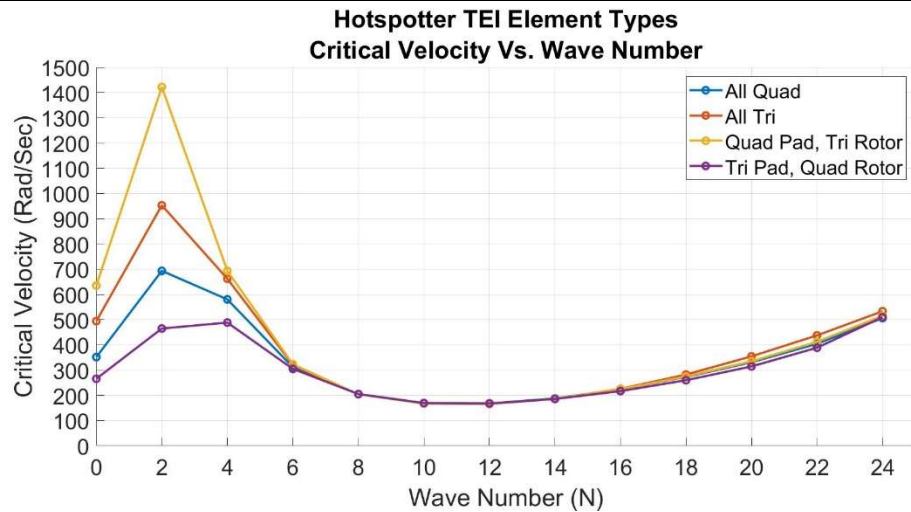


Figure 50. Critical velocity vs. wave number results for element validation simulations.

Figure 50 displays the critical velocity versus the wave number of all the models used in validation. It is clear from the graph that at lower critical velocities the results converge on the expected value, and as the critical velocity increases the results diverge from the expected results with larger divergence happening at lower wave numbers. A

mesh refinement is required to understand if convergence on the expected value will happen at lower wave numbers.

3.2.1.6 Validation mesh refinement and convergence study

All of the models from section 3.2.1.5 were re-meshed to increase the number of nodes and elements through the entire model in an attempt to understand if convergence on the expected results from Table 14 would occur.

The all quadrilateral model increased the number of elements from 36 to 180, and increased the number of nodes from 55 to 220. The refined mesh is shown in Figure 51.

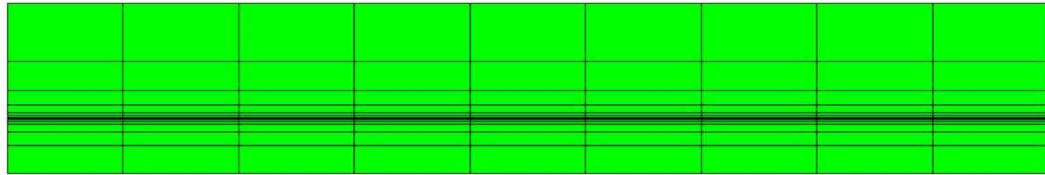


Figure 51. Refined mesh of an axisymmetric disk pad and rotor model using all quadrilateral elements in Hotspotter 3D.

The all triangular model increased the number of elements from 72 to 360, and increased the number of nodes from 55 to 220. The refined mesh is shown in Figure 52.

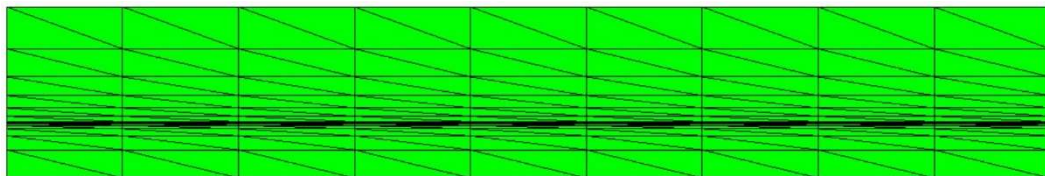


Figure 52. Refined mesh of an axisymmetric disk pad and rotor model using all triangular elements in Hotspotter 3D.

The quad pad, tri rotor model increased the number of elements from 48 to 234, and increased the number of nodes from 55 to 220. The refined mesh is shown in Figure 53.

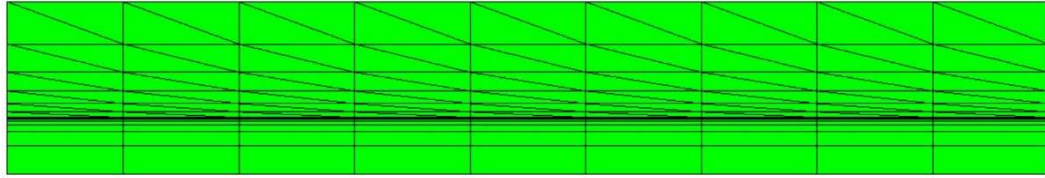


Figure 53. Refined mesh of an axisymmetric disk pad and rotor model using quadrilateral and triangular elements, respectively, in Hotspotter 3D.

The tri pad, quad rotor model increased the number of elements from 60 to 306, and increased the number of nodes from 55 to 220. The refined mesh is shown in Figure 54.

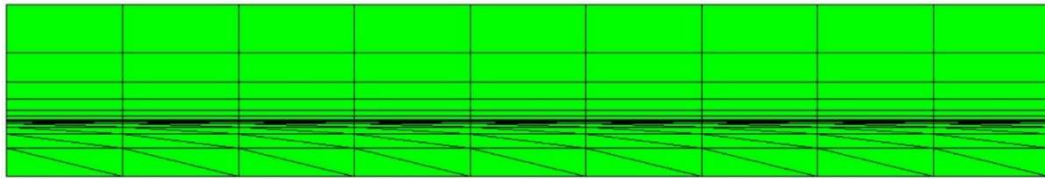


Figure 54. Refined mesh of an axisymmetric disk pad and rotor model using triangular and quadrilateral elements, respectively, in Hotspotter 3D.

Results in Table 15 show that no matter the element choice, singular or in combination, and selection of elements relative to each part, stator or rotor, the results for lowest critical sliding speed and wave number are within 0.85 percent of the expected value.

Table 15. Mesh refinement results for wave number and lowest critical speed results of all element types.

Model	Dominant Mode Wave Number	Lowest Critical Speed (rad/sec)	Percent Error (%)
All quad model	12	177.7	0.00
All tri model	12	178.2	.281
Quad pad, tri rotor model	12	179.2	.844
Tri pad, quad rotor model	12	178.2	.281

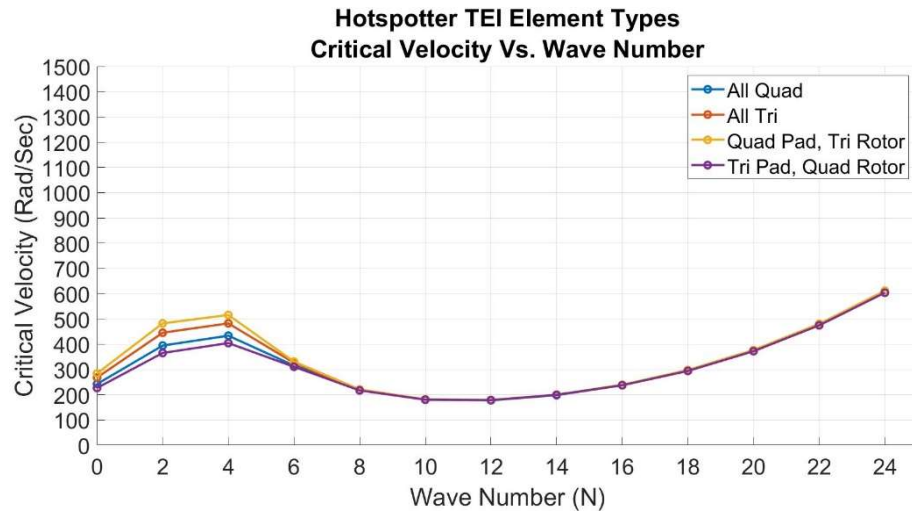


Figure 55. Refined mesh critical velocity vs. wave number results for element validation simulations.

In comparison to the results from the rough meshes used in the previous section, displayed in Figure 50, the results of the more finely meshed models are much closer to the expected value, as shown in Figure 55. Refined mesh critical velocity vs. wave number. Therefore, because the lowest critical velocity results are within 0.85

percent error of the expected value and occurs at the same wave number, it can be concluded that the CST element, used in any configuration in Hotspotter, is a validated element.

Further, because the all quad model has been validated against real world experimental data and because the CST element models match the critical speed of the all quad model within 0.85 percent error, it is reasonable to consider the CST element also validated against that same experimental data.

3.2.2 First-order tetrahedral element

Due to the difficulties and time commitment in developing the tetrahedral element in three dimensional Cartesian space and verifying its proper working order via comparison to Abaqus and analytical solutions, the tetrahedral element has yet to be incorporated into Hotspotter software. The element has been verified to be functional and sufficiently accurate, but needs to be defined in cylindrical coordinates before it can be implemented into Hotspotter for validation. However, this is a source of ongoing work and will be incorporated into Hotspotter in the future.

CHAPTER 4 CONCLUSION AND FUTURE WORK

4.1 Conclusion

While developing a system vulnerable to thermoelastic instabilities, an engineer must consider many factors. From the type of materials and material properties to be used, geometric restrictions imposed by other connected systems, to mechanical and thermal stresses which may develop in the system from normal use. When dealing with thermal and structural analysis of the system, special attention must be placed on regions of high stress concentrations, thermal loads, and temperature gradients. The designer and analyst spend a considerable amount of time meshing the system accurately enough such that the results from any finite element analysis is sufficiently close to real world results. Therefore, the time taken to re-mesh that system for compatibility with Hotspotter is time that could have been spent further developing the design or moving along in the design process.

Development and implementation of triangular and tetrahedral elements into Hotspotter has been performed and shown to be a worthwhile endeavor for reducing a user's redundancies in remeshing a system from commercial code to Hotspotter.

Triangular (CST) and tetrahedral (tet4) elements have been developed with regard to compatibility with Hotspotter. Element verification based on a few trial cases of increasing complexity, with accompanying analytical solutions, has been performed and shown to provide sufficiently accurate results when compared to commercial code and

analytical solutions. In a few cases, mesh refinements were performed and provided convergence toward the analytical solution. In all the element verification cases the results from a commercial code, Abaqus, and the results of the elements developed for Hotspotter agreed closely. Such results provided the basis to move onto element implementation into Hotspotter and validation based on legacy elements.

Implementation of the triangular elements into Hotspotter occurred and results of a few trial validation cases were presented. As discussed previously, the Hotspotter user now has the ability to use the elements developed here along with any legacy elements, individually or in any combination, and remain confident that the results are sufficiently accurate. To prove this, the trial cases compared all tri elements and all possible combinations of element types to the legacy all quadrilateral elements in Hotspotter, and showed results within 0.85 percent error.

Because of the element verification with commercial code and analytical solution, paired with the element validation in Hotspotter, comparison to legacy solutions, and convergence study, it is safe to say the Hotspotter user may no longer be required to re-mesh a system from commercial code to Hotspotter software. The user may import an existing mesh using triangular, quadrilateral, tetrahedral, or hexahedral elements while remaining confident that results for TEI analysis are accurate.

4.2 Future work

Further convergence studies in Hotspotter should be considered to understand if the values reported have reached a plateau. For example, the mesh refinement in section 3.2.1.6 indicate that the results for the finely discretized mesh using quad elements is

different than the results using a rough mesh. To ensure the model is fully converged, one or two more rounds of mesh refinement may be required to see where the critical velocity values cease to change. This investigation determined that the new element types would give sufficiently accurate results compared to legacy elements, but further work needs to be done to ensure the legacy elements are truly converged.

The tetrahedral element implementation into Hotspotter is a source of ongoing work. As developed here, the tet4 code was defined and verified using Cartesian coordinates. Compatibility with Hotspotter requires the element definition take place in cylindrical coordinates so some intermediate values such as relative velocity may be calculated from the strain-displacement matrix. Therefore, the work to develop the tet4 element in cylindrical coordinates is currently ongoing and will take place as the next step after this investigation.

While developing the Matlab code and implementing into Hotspotter, some difficulties arose in the form of incorrect, and confusing solutions. It was thought that perhaps a first-order element definition was insufficient in capturing the levels of nonlinearity in the problem. While it was proven here that in fact first-order elements are sufficient, it is still a source of curiosity and unanswered questions as to whether or not second order and higher element definitions may be a better fit for this type of analysis. Therefore, the intent is to continue this work and develop and implement second order triangular and tetrahedral elements for use in Hotspotter. It is anticipated from literature that these element types will perform better in nonlinear analysis where contact is present.

Further it is the intent of the author to continue research in the area of FEA, TEI, and wear phenomenon. In regard to TEI, it was discovered during the course of this investigation that there is no viable solution for or commercially available software which analyses TEI in wet clutches, limit slip differentials, or any sliding friction system coupled with a fluid or lubricant. According to literature, recent developments in manufacturing and electronic controls have made widespread adoption of wet clutches and limited slip differentials possible [38]. The work in [39] could be expanded to further solve for thermoelastic instabilities similar to [40]. In such systems, development of similar design tools to Hotspotter or a plug-in code for Hotspotter could be developed to consider sliding friction systems with a fluid present.

Also, the effects of wear in relation to TEI have been investigated by [41], [10], [42], and [43] among others and has yet to be rigorously investigated for realistic geometries. There is no commercially available software which models wear in relation to TEI. However, such tools could be developed to consider the interactions between all or a combination of TEI, wear, and thermal fluids to further simplify the design process.

References

- [1] M. H. Faidh-Allah, "Numerical And Finite Element Contact Temperature Analysis Of Friction Material's Type Effect On A Thermal Transient Behavior Of A Single-Disc Dry Clutch," *Tribologia*, pp. 35-43, 2017.
- [2] Y.-B. Yi, "Finite Element Analysis Of Thermoelastodynamic Instability Involving Frictional Heating," *Journal of Tribology*, pp. 718-724, 2006.
- [3] Y.-B. Yi, "Pertrubation Methods In Thermoelastic Instability (TEI) With Finite Element Implementation," *Richard B. Jetnarski Encyclopedia of Thermal Stresses*, pp. 3635-3641, 2014.
- [4] J. Y. Jang and M. Khonsari, "A Generalized Thermoelastic Instability Analysis," *Proceedings of the Royal Society*, pp. 309-329, 2003.
- [5] J. R. Barber, "Thermoelastic Instabilities In The Sliding Of Conforming Solids," *Proceedings of the Royal Society*, pp. 381-394, 1969.
- [6] J. Barber and Y.-B. Yi, "Eigenvalue Solution Of Thermoelastic Instability Problems Using Fourier Reduction," *Proceedings Of The Royal Society*, pp. 2799-2821, 2000.
- [7] A. E. Anderson and R. A. Knapp, "Hotspotting In Automotive Friction Systems," *Wear*, pp. 319-337, 1990.
- [8] P. Zagrodzki, "Analysis Of Thermomechanical Phenomena In Multidisc Clutches And Brakes," *Wear*, pp. 291-308, 1990.

- [9] J. R. Barber and M. Ciavarella, "Contact Mechanics," *International Journal of Solids and Structures*, pp. 29-43, 2000.
- [10] A. Papangelo and M. Ciavarella, "The Effect Of Wear On Thermoelastic Instabilities (TEI) In Bimaterial Interfaces," *Tribology International*, 2019.
- [11] R. Burton, V. Nerlikar and S. Kilaparti, "Thermoelastic Instability In A Seal Like Configuration," *Wear*, pp. 177-188, December 1972.
- [12] J. R. Barber and K. Lee, "Frictionally Excited Thermoelastic Instability In Automotive Disk Brakes," *Journal of Tribology*, pp. 607-614, 1993.
- [13] Y. Liu, Y. H. Jang and J. R. Barber, "Finite Element Implementation Of An Eigenfunction Solution For The Contact Pressure Variation Due To Wear," *Wear*, pp. 134-138, 2013.
- [14] S. Du, P. Zagrodzki, J. R. Barber and G. M. Hulbert, "Finite Element Anlysis Of Frictionally Excited Thermoelastic Instability," *Journal Of Thermal Stress*, pp. 185-201, 1997.
- [15] R. W. Lewis, P. Nithiarasu and K. N. Seetharamu, *Fundamentals Of The Finite Element Method For Heat And Fluid Flow*, Hoboken: John Wiley & Sons, 2004.
- [16] S. Du, *Thermoelastic Effects In Auotomotive Brakes*, Ann Arbor: University of Michigan, 1997.
- [17] J. R. Barber and Y.-B. Yi, *Hotspotter User's Manual*, University of Michigan, 2006.

- [18] T. Yeo and J. R. Barber, "Finite Element Analysis Of Thermoelastic Contact Stability," *Journal of Applied Mechanics*, pp. 919-922, 1994.
- [19] Y.-B. Yi, S. Du, J. R. Barber and J. W. Fash, "Effect Of Geometry On Thermoelastic Instability In Disk Brakes And Clutches," *Journal of Tribology*, pp. 661-666, 1999.
- [20] Z. Chen, Y.-B. Yi, K. Bao and J. Zhao, "Numerical Analysis Of The Coupling Between Frictionally Excited Thermoelastic Instability And Thermal Buckling In Automotive Clutches," *Journal Of Engineering Tribology*, vol. 223, no. 1, pp. 178-187, 2019.
- [21] C. K. Lee and S. H. Lo, "An Automatic Adaptive Refinement Procedure Using Triangular And Quadrilateral Meshes," *Engineering Fracture Mechanics*, pp. 671-686, 1995.
- [22] R. D. Cook, D. S. Malkus, M. E. Plesha and R. J. Witt, Concepts And Applications Of Finite Element Analysis, Hoboken: John Wiley & Sons, 2002.
- [23] J. Reddy, An Introduction To The Finite Element Method, New York: Mcgraw-Hill, 2006.
- [24] O. Zienkiewicz, R. Taylor and J. Zhu, The Finite Element Method: Its Basis And Fundamentals, Burlington: Butterworth-Heinemann, 2005.
- [25] Z. Chen, Y.-B. Yi and J. Zhao, "Fourier Finite Element Model For Prediction Of Thermal Buckling In Disc Clutches And Brakes," *Journal Of Thermal Stress*, pp. 1241-1251, 2016.

- [26] K. T. Danielson and A. K. Noor, "Three-Dimensional Finite Element Analysis In Cylindrical Coordinates For Nonlinear Solid Mechanics Problems," *Finite Elements in Analysis and Design*, pp. 225-249, 1997.
- [27] P. Zagrodzki, K. B. Lam, E. Al-Bahkali and J. R. Barber, "Simulation Of A Sliding System With Frictionally-Excited Thermoelastic Instability," *Proceedings of the 3rd International Congress on Thermal Stresses*, pp. 617-620, 1999.
- [28] J. R. Barber, "Instability Of Thermoelastic Contact," in *Friction And Instabilities*, Springer-Verlag Wien, 2002, pp. 1-37.
- [29] O. Zienkiewicz, *The Finite Element Method*, 4th edition, New York: McGraw-Hill , 1989.
- [30] A. F. Bower, *Applied Mechanics Of Solids*, Boca Raton: CRC Press, 2010.
- [31] I. Doghri, *Mechanics Of Deformable Solids: Linear, Nonlinear, Analytical And Computational Aspects*, New York: Springer, 2000.
- [32] P. Duda, "Finite Element Method Formulation In Polar Coordinates for Transient Heat Conduction Problems," *Journal of Thermal Sciences*, pp. 188-194, 2016.
- [33] The MathWorks, Inc., "Matlab Users Manual," Natick, 2020.
- [34] J. Barber, *Intermediate Mechanics Of Materials*, Second Edition, New York: McGraw-Hill, 2000.

- [35] S. Riedmaier, B. Danquah, B. Schick and F. Diermeyer, "Unified Framework and Survey for Model Verification, Validation and Uncertainty Quantification," *Archives of Computational Methods in Engineering*, 2020.
- [36] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin and J. Tu, "A Framework for Validation of Computer Models," *Technometrics*, pp. 138-154, 2007.
- [37] C. Yin and A. McKay, "Model Verification and Validation Strategies and Methods: An Application Case Study," in *The 8th International Symposium on Computational Intelligence and Industrial Applications; The 12th China-Japan International Workshop on Information Technology and Control Applications*, Tengzhou, 2018.
- [38] R. Maki, *Wet Clutch Tribology - Friction Characteristics In Limited Slip Differentials*, Lulea University of Technology, 2005.
- [39] P. Zagrodzki, "Numerical Analysis Of Temperature Fields And Thermal Stresses In The Friction Discs Of A Multidisc Wet Clutch," *Wear*, pp. 255-271, 1985.
- [40] J. Y. Jang and M. M. Khonsari, "On The Formation Of Hot Spots In Wet Clutch Systems," in *Transactions of the ASME*, San Fransisco, 2002.
- [41] T. A. Dow and R. A. Burton, "The Role Of Wear In The Initiation Of Thermoelastic Instabilities Of Rubbing Contact," *Journal Of Lubrication Technology*, vol. 95, no. 1, pp. 71-75, 1973.

- [42] L. Johansson, "Model And Numerical Algorithm For Sliding Contact Between Two Elastic Half-Planes With Frictional Heat Generation And Wear," *Wear*, vol. 160, pp. 77-93, 1993.
- [43] R. R. Phiri, O. P. Oladijo and E. T. Akinlabi, "A Finite Element Approach To Modelling Wear: Review," *Journal Of Tribology*.
- [44] "The Thermo-Mechanical Behavior in Automotive Brake and Clutch Systems," in *New Trends and Developments in Automotive System Engineering*, InTech, 2011, pp. 207-230.