Computational Design of Ceramic Matrix Composites for Turbine Blade Applications

by

Santiago Miret

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering-Materials Science and Engineering
and the Designated Emphasis
in
Computational and Data Science and Engineering

in the

Graduate Division
of the
University of California, Berkeley

Committee in charge:

Professor Tarek Zohdi, Co-chair
Professor Mark Asta, Co-chair
Professor Kristin Persson
Professor Per-Olof Persson

Summer 2018

Computational Design of Ceramic Matrix Composites for Turbine Blade Applications

Abstract

Computational Design of Ceramic Matrix Composites for Turbine Blade Applications

by

Santiago Miret
Doctor of Philosophy in Engineering-Materials Science and Engineering
and the Designated Emphasis
in
Computational and Data Science and Engineering

University of California, Berkeley

Professor Tarek Zohdi, Co-chair

Professor Mark Asta, Co-chair

Turbine technology is a critical part of the today's energy and transportation infrastructures, and turbine manufacturers constantly aim to increase the operating efficiencies of their products to yield better technical and economic performance. The thermodynamic efficiency of a turbine system is bounded by its operating temperature, which is often limited by the materials used to construct the turbines blades. The blade materials need to withstand very demanding conditions, including rotational loads from rotations of over 3,000 rpm and thermal loads from temperatures above 1,000 C. Informed materials design of turbine blades is therefore crucial for enhancing the performance of turbine systems. Ceramic Matrix Composite (CMC) materials are a new series of material systems that have recently received substantial interest from turbine manufacturers due to their exceptional ability to retain their mechanical properties at significantly higher temperatures than commercial superalloy compounds. Given the high cost of experimental CMC research, computation can be a very helpful tool to aid the design of novel CMC compounds by allowing the designer to test the performance of large sets of material systems quickly and effectively through targeted simulations. This study presents a computational framework that can be applied for exploratory design of CMC compounds. The design framework combines continuum mechanics based simulation tools, such as the finite element method, with evolutionary algorithms to enable computer driven design of CMC compounds. The evolutionary algorithms, driven by numerical and reduced order models, prototype various material and microstructure design choices and evaluate their performance against a set of target properties. As the evolutionary algorithms progress in their parameter choices, they create better and better material design choices until a stopping criterion is reached. Following the down-selection from the evolutionary algorithms, the computer-generated CMC design are then subjected to further virtual tests, including preliminary

structural scale finite element simulations of a turbine blade geometry composed of the algorithmically designed CMC compound.

Dedicated to Emilio Miret, a loving and inspiring grandfather, and his love for technology

Contents

Acknowledgments

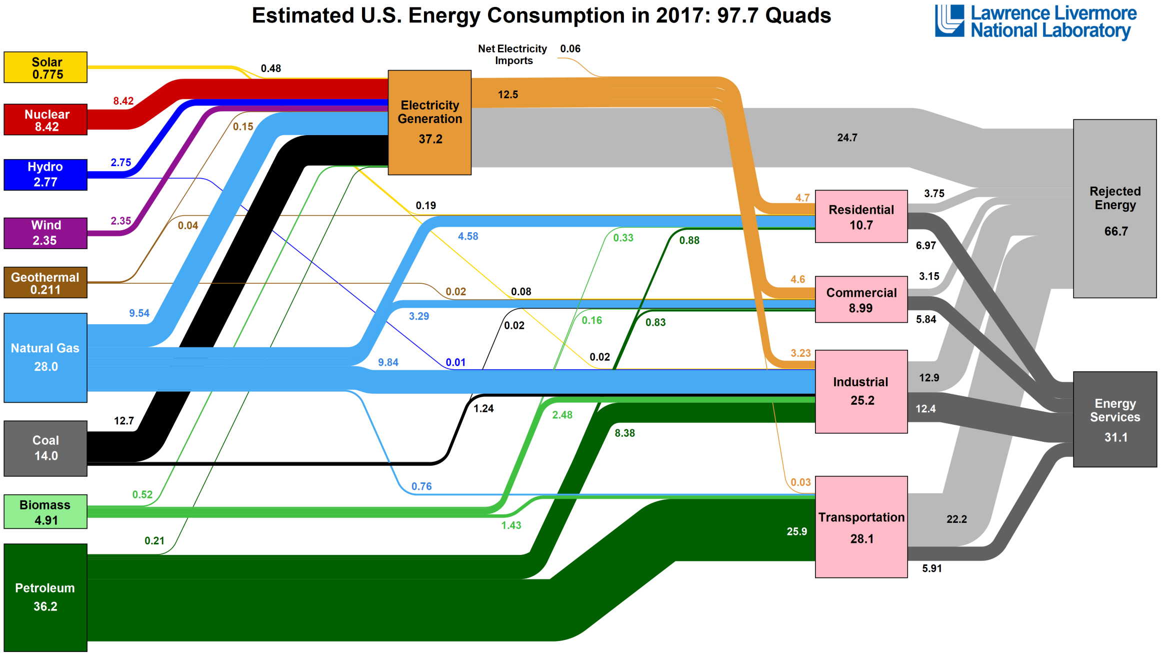
It takes a whole village to lift one person and enable their success. This page serves to acknowledge the village that has helped me reach this point of my personal and professional life. Thank you Tarek Zohdi for your support as my academic advisor in my PhD career at UC Berkeley. I would not have been able to get through the program without your guidance, support, and many memorable moments throughout this process. Thank you Mark Asta for being my co-advisor and a constant source of support and grounding in important fundamentals throughout my years at UC Berkeley. Thank you Per-Olof Persson for teaching me the mathematical foundations needed for my research in various courses, and for serving on my Qualifying Exam committee and my Dissertation Committee. Thank you Ralf Mueller from the Technical University of Kaiserslautern, Germany, for providing a space to grow through meaningful exchanges between research groups. Thank you Zeyad Zaki for always answering my in-the-weeds questions, whatever they may be. Thank you Alexander Schlueter for being my go-to person in Kaiserslautern whenever I needed help. Thank you to the members CMMRL Laboratory at UC Berkeley for their continuous support and discussions of the material. Thank you to the members of the Mechanics division at the Technical University of Kaiserslautern for discussion and support during the IRTG 2057 exchange. Thank you members of the Mark Asta Group at UC Berkeley for providing me with a forum to help shape my ideas. Thank you to all members of the IRTG 2057 to make a research exchange between UC Berkeley and the Technical University of Kaiserslautern possible. Thank you to the Berkeley Energy and Resources Collaborative for a constant source of inspiration throughout my years at UC Berkeley. Thank you to the Latinx Association of Graduate Students in Engineering and Science for providing with a community that I could feel included and supported in. Thank you professors Daryl Chrzan and Andrew Minor for serving on my Qualifying Exam committee. Thank you professor Kristin Persson for serving on my Dissertation committee. Thank you to the National Defense Science and Engineering Graduate Fellowship for substantial financial support for 3 years of my program. Thank you to the Materials Science and Engineering Department at UC Berkeley for providing me with a space to research my intellectual curiosities. Thank you UC Berkeley for significant financial support during the first years of the program and for fostering a campus of intellectual richness that enabled me to grow. Thank you to Marco Brunelli of the Siemens Corporation for allowing me to collaborate with Siemens on my research projects. Thank you to Kyle Karlson of Sandia National Laboratories for providing me with an internship opportunity that was essential to my professional growth. Thank you Joseph Bailey, Darryl Pines and Peter Kofinas for continuing to be mentors in my life, even after I left the University of Maryland. Thank you Chul Kwon, Hector Neira and Femi Pai for being great friends throughout these years and supporting me in all my crazy ideas. Thank you to all teachers, mentors and friends inside and outside UC Berkeley that have helped me grow and supported in the last five years. The collection of you is too large to name here, but rest assured that you are important and that I could not have reached this point without you. Lastly, I owe great thanks and appreciation to my father Dr. Juan Jose Miret whose support throughout my educational process has been invaluable.

Chapter 1

Introduction

1.1 Motivation

As global energy demand continues to grow, the importance of designing and maintaining efficient power systems will become increasingly important. Based on the energy flow chart created by Lawrence Livermore National Laboratories, fossil fuels, particularly natural gas, coal and petroleum, continue to be the United States' largest energy sources. Moreover, given that the energy flows generated by fossil fuel, especially natural gas and petroleum, have grown in the past three years, fossil fuels are likely to remain an important component of the United States' energy mix.[43]



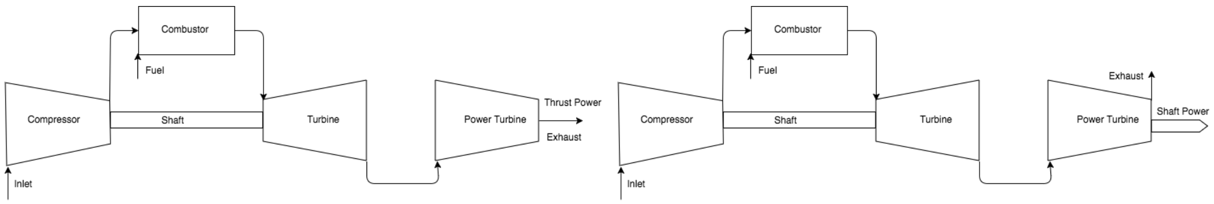
Source: LLNL April, 2018. Data is based on DOE/EIA MER (2017). If this information or a reproduction of it is used, credit must be given to the Lawrence Livermore National Laboratory and the Department of Energy, under whose auspices the work was performed. This chart was revised in 2017 to reflect changes made in mid-2016 to the Energy Information Administration's analysis methodology and reporting. The efficiency of electricity production is calculated as the total retail electricity delivered divided by the primary energy input into electricity generation. End use efficiency is estimated as 65% for the residential sector, 65% for the commercial sector, 21% for the transportation sector, and 49% for the industrial sector which was updated in 2017 to reflect DOE's analysis of manufacturing. Totals may not equal sum of components due to independent rounding. LLNL-MI-410527

Figure 1.1: United States Energy Flow 2017 shows the importance of fossil fuels for the United States' energy mix in electricity generation and further energy sector. Flowchart from Lawrence Livermore National Laboratory Energy Flowcharts, https://flowcharts.llnl.gov/commodities/energy [43]

Figure 1.1[43] above shows the continued significance of fossil fuel based energy sources in our current energy infrastructure. As can be seen from the flowchart above, the vast majority of US electricity is generated by power plants that use nuclear, natural gas and coal fuels. Each of those different fuels, however, relies on turbine technology to generate electricity [51]. In the case of nuclear power plants and coal power plants, the fuel is used to generate heat that creates steam which is then used to rotate a turbine whose rotational motion is converted to electricity by a generator.

Moreover, the recent natural gas boom in the United States has created further demand for high-quality gas turbines in the energy generation space. This trend is likely to continue globally as natural gas continues to become a more popular energy source[43]. Given that the energy generation processes in most turbine settings are driven by a temperature differential, fundamental thermodynamics dictates that a higher temperature differential, between the hot reservoir and the cold reservoir, leads to higher thermal efficiency. The ability to operate gas turbines at high temperatures therefore becomes critical to improving the efficiency of the overall energy generation process.

Increasing the temperature differential, and thereby the operating efficiency of turbine products, is also of crucial importance for the aerospace industry, another major user of turbine technology. As can be seen from Figure 1.1, transportation makes up a significant amount of the United States' energy consumption, which includes the energy consumed by the turbine technology in the aerospace sector. In 2017, the use of jet fuel constituted 12% of the fuel sources consumed for transportation purposes in the United States [4] primarily to due commercial flights. Turbine technology is a critical component of commercial aerospace engines and therefore plays a critical part in the transportation sector as well. While the energy and aerospace industries both use similar aspects of gas turbine technology, the application and consequent engineering systems surrounding the turbine varies.



(a) Jet Engine Schematic

(b) Land Energy Generation Schematic

Figure 1.2: Schematics for Major Turbine Applications adapted from Langston, L., Opdyke, G. Introduction to Gas Turbines for Non-Engineers. Global Gas Turbine News, Volume 37: 1997, No. 2. [44]

Figure 1.2 above outlines schematic differences for turbine applications in an aircraft engine (Schematic A) and a land-based energy generation gas turbine (Schematic B) [44], with the major difference being the output side of the process. In a jet engine, the power of the rotational force of the turbine is used to propel a gas/air mixture out of a nozzle to generate thrust power, whereas a land-based energy generation system translates the rotation of a gas turbine to a generator to create electrical power. A more detailed diagram of jet engine system with a gas turbine is shown in Figure 1.3 below to illustrate the complexity of the system [18].

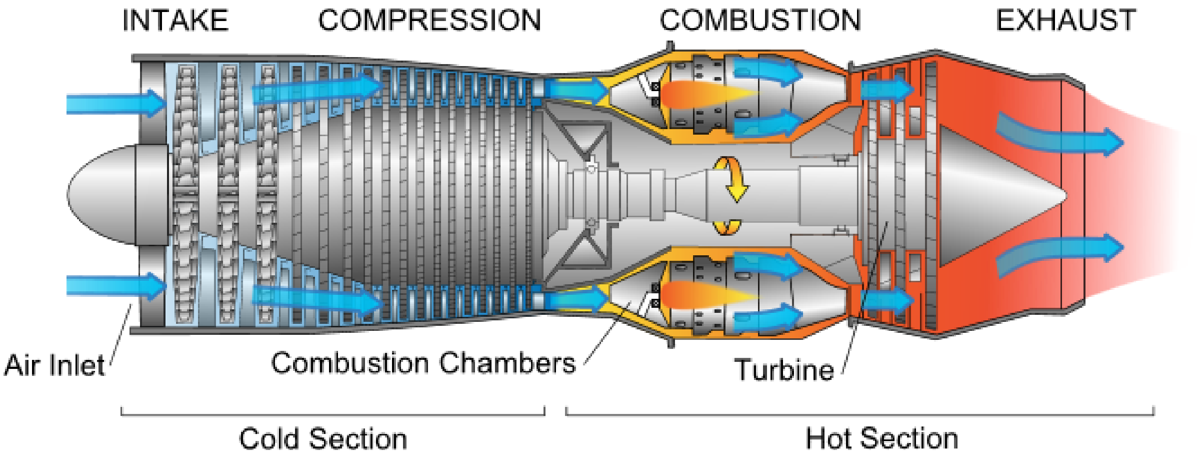


Figure 1.3: Jet Engine System Schematic with colors representing temperatures adapted from Jeff Dahl, accessed from Minnesota State University Mankato. http://cset.mnsu.edu/engagethermo/components_gasturbine.html.[18]

As a result of the distinct applications and system setups, there exist significant differences in the design requirements for turbines in aerospace applications and for turbines in utility scale energy generation:

Parameter	Aerospace	Energy Production
Weight	Very Important	Less Important
Steady State Operating Time (Hours)	25,000	>100,000
Peak Temperature Operating Time (Hours)	<1,000	>100,000
Cyclic Duty	Severe	Severe
Size	Small	Large

Table 1.1: Summary of Important Parameter for Gas Turbine Application in Aerospace and Utility-Scale Energy Generation adapted from Brij [11]

As can be seen from Table 1.1[11] above, the requirements for the turbines vary significantly depending on their application. Weight is an important design criterion for both aerospace and EGT turbines as rotational stresses inside the turbine increase with greater weight. These higher rotational stresses can then lower the lifetime of the turbine as creep becomes more prevalent. Furthermore, in the case of aerospace, a heavier airplane requires more energy to be lifted off the ground which adds significant operational costs. Both aerospace and EGT operate under severe cyclic duty, thus making fatigue and creep resistance a critical property of both gas turbine models. The most important parameters for EGTs then become high temperature mechanical and chemical reliability, as well as favorable heat conduction. The mechanical reliability is given by resistance to mechanical failure, while the chemical reliability is given by resistance to corrosion in high temperature environments. The mechanical and thermal conduction qualities of the gas turbine materials also have to last significant timespans, as long lifetimes are required for EGT to pay back the initial investment of fabricating and building the turbine system. This means that the material system of the turbine also requires high fatigue and creep resistance to deliver the desired lifetime under high cyclic duty.

The remainder of this chapter will focus on turbines for aerospace applications, as the much more demanding conditions associated with such applications has driven the most significant advances in turbine material technology. For turbine technology used in a jet engine, the power output is increased by enhancing the operating temperature of the hot section and the operating pressure of the compressor

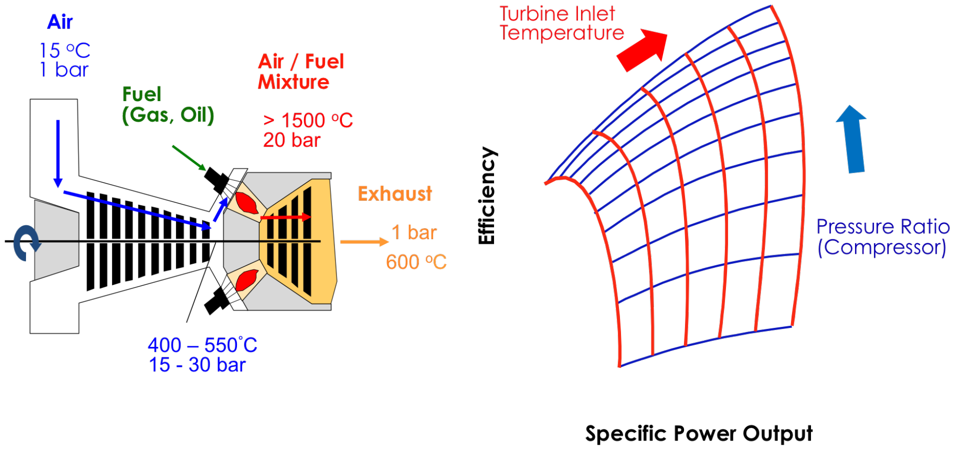


Figure 1.4: Left: Turbine Schematic describing various operating temperatures and pressures at different sections of the gas turbine; Right: Schematic diagram of the Brayton Cycle indicating that efficiency increases with higher operating temperatures and pressures - adapted from presentation given by Siemens AG at UC Berkeley.[62]

Gas turbines primarily operate in the Brayton Cycle, which consists of four distinct processes :

1. Adiabatic, quasi-static (reversible) compression in the inlet and compressor
2. Constant pressure fuel combustion (idealized as constant pressure heat addition)
3. Adiabatic, quasi-static (reversible) expansion in the turbine leading to:
   (a) work taken out of the air to drive the compressor
   (b) take the remaining work out of the system to generate thrust or power
4. Cool the air at constant pressure back to the initial condition

Following a basic analysis of the thermodynamics of the Brayton Cycle [1], the power output can be expressed as:

$$Power = \dot{m}c_pT_a[\frac{T_c}{T_a} - 2\sqrt{\frac{T_c}{T_a}} + 1] \tag{1.1}$$

where T_c and T_a are defined as the maximum turbine inlet temperature and the atmospheric temperature, respectively. This expression shows that the power increases as the operating temperature of the hot section of the jet engine, described by T_c, increases.

1.2 Raising the Operating Temperatures of Gas Turbines through Materials Development

Superalloy

Most turbines operate in extreme environments with exigent mechanical, thermal, and chemical conditions: Siemens specifies operating conditions for their 400MW SGT5-8000H heavy-duty gas turbine at rotations of 3,000 rpm and exhaust temperatures of 627 C.[3] The high thermal, mechanical, and chemical loads thus require robust materials design for the turbines to function reliably over their extensive multi-year lifetimes. In order to remain competitive, turbine manufacturers constantly aim to increase the operating efficiencies of their products. Enhanced operating efficiency will then yield better technical and economic performances, in addition to environmental benefits that are becoming more and more critical in the energy and transportation sectors.

The stringent requirements given by the operating environment of gas turbines places a high demand on the materials used to construct them. The materials that are most widely used for gas turbine applications in aerospace today are called superalloys. These superalloys maintain desirable mechanical properties to temperature ranges above 1000 K, as can be seen from Figure 1.5[23] below:

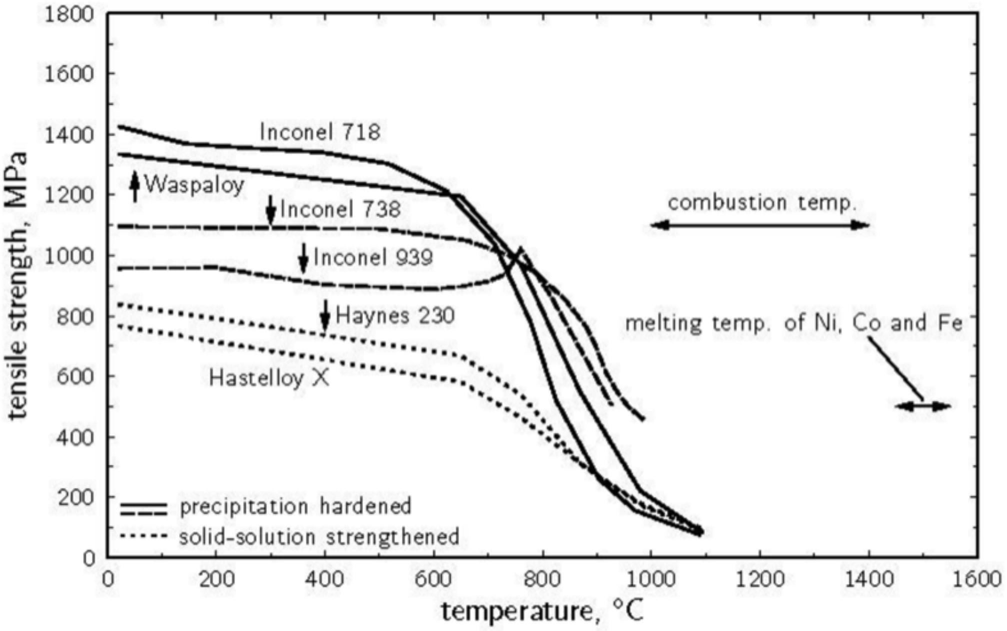


Figure 1.5: Tensile strength (MPa) vs. Temperature for common Superalloys. The curves indicate the deteriorating mechanical properties of the various alloy compounds as their surrounding temperature increases. Current combustion temperatures can only be achieved with the help of thermal barrier coatings, which will be described in a later section. Figure adapted from PhD Thesis of Eriksson, R. [23]

Many superalloys are prepared in polycrystalline form. However, when the application requires maximum creep strength these materials are usually prepared as single crystals. The higher creep strength of the single crystalline forms of these materials results from the absence of grain boundaries, whose presence provides regions for plastic flow and atomic transport of the material that reduces creep strengths. The associated grain-boundary mechanisms include grain boundary sliding and Nabarro-Herring creep, and reduction of grain boundary areas therefore limits the role of these mechanisms. Further creep mechanism, such as dislocation creep and Coble Creep which are associated by atomic transport, will still be present but the overall creep rate is more limited. Traditionally, single crystal superalloys have been fabricated using directional solidification processes, in which the material is produced in columns of the desired grain, as shown in Figure 1.6[23] below:

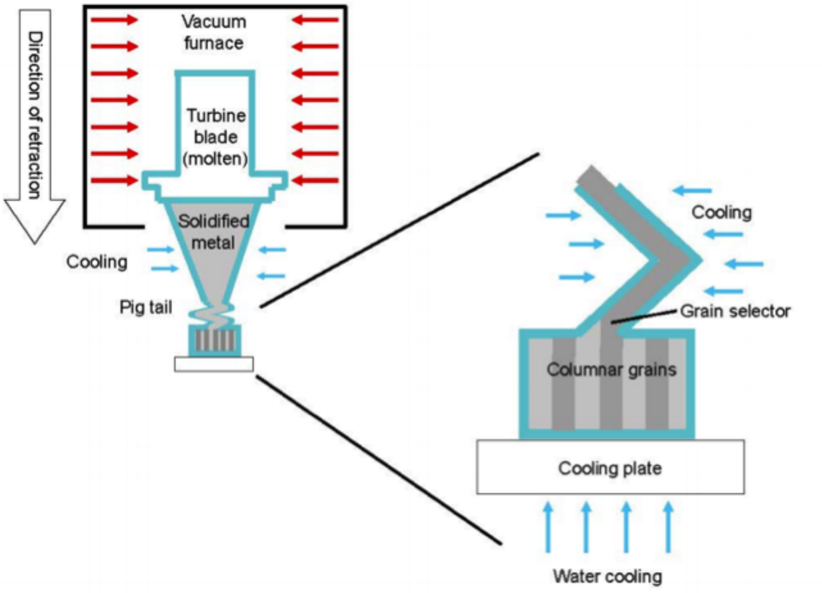


Figure 1.6: Directional solidification of a superalloy adapted from PhD Thesis by Eriksson[23]

Thermal Barrier Coatings

A common approach to increase the efficiency of turbine products is to raise the operating temperature of the overall system. The underlying superalloy material systems have an inherent limit of the temperature range that they can operate in reliably. An engineering solution to go beyond the temperature limit of the superalloy systems is to apply thermal barrier coatings (TBCs). TBCs are refractory-oxide ceramics that can be applied to the superalloys in the hottest part of the turbine system. The additional thermal resistance in the hottest section of the turbine allows the operator to raise the operating temperature beyond the limit set by the intrinsic properties of the turbine superalloy. [16]

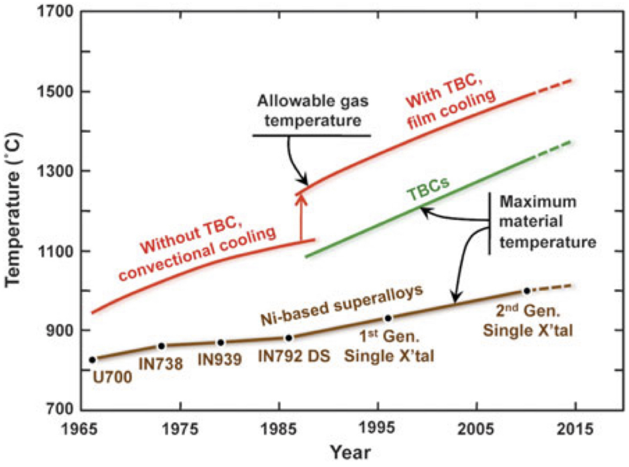
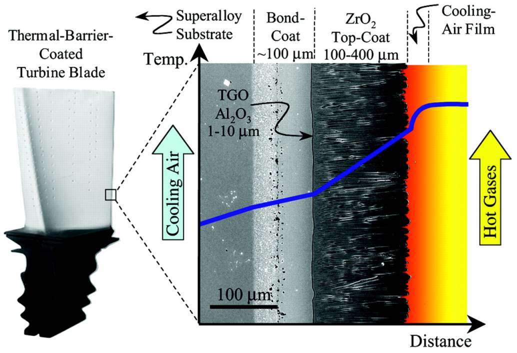


Figure 1.7: Temperature Gains for Various Turbine Materials Achieved with Thermal Barrier Coatings and Cooling Systems [16]

The application of TBCs grew to become standard in the turbine manufacturing industry over the past couple of decades, and has allowed turbine manufacturers to operate their products at significantly higher temperatures than the material alone can support. As shown in Figure 1.7 above, the application of thermal barrier coatings can raise the operating temperatures of the turbine system by over 200 degrees. Moreover, if TBCs and film cooling systems are applied jointly to cool the turbine blades from the surrounding atmosphere, the operating temperature of the overall system can be raised ~200 degrees higher, thereby adding significant efficiency gains. Figure 1.7 also shows that the application of single crystal superalloy materials, described in the above section, constitute a more stable and temperature resistant material system compared to its predecessors.

TBC materials are multi-functional: While their primary purpose is to provide thermal insulation for the underlying superalloys, TBCs also provide protection from oxidation, reflect radiant heat from surrounding gases and help minimize thermal-expansion mismatch stresses of the superalloy parts throughout heating and cooling cycles.[16] TBC materials therefore have many requirements, some of which can also be conflicting. Among the most important requirements low thermal conductivity for insulation, low weight and high stress resistance to withstand thermally induced stresses. Furthermore, the TBC materials have to be chemically compatible with the underlying superalloy metal and the oxides that are exposed to the surface of the TBCs while also being able to operate in an oxidizing environment at high pressures, often greater than 10 atmospheres, surrounded by fast gases that are flowing greater than the speed of sound.[16] A thermal barrier coating is made from various layers of materials, consisting of metallic and oxide compounds, that protect the underlying metal structure. [53]



(a) Turbine Blade Schematic with SEM Cross-Section of Thermal Barrier Coating Material

Figure 1.8: Thermal Barrier Coating of a Turbine Blade adapted from Padture, N., Gell, M., and Jordan, E. Science 12 Apr 2002: Vol. 296, Issue 5566, pp. 280-284 [53]

The underlying layer is the metal substrate, which is coated by a bond coat that is ~100 μm thick oxidation resistant metallic layer. Common materials for the bond coat of NiCrAlY type alloys, and are designed to grow an $\alpha - Al_2O_3$ thermally grown oxide (TGO). The TGO is both mechanically robust and resistant to oxygen diffusion to protect the underlying substrate from oxidation that can cause substantial damage in the substrate material.[53] The ceramic is usually made of thermally insulating Y_2O_3-stabilized ZrO_2 compounds, which is often deposited in a porous matter to further reduce the thermal conduction of the layer. The top-coat layer is usually between 100-400 μm thick and is exposed to the hot circulating gases in the turbine operating environment.[53] Figure 1.8 above shows how a TBC material helps manage the temperature distribution of superalloy based turbine blades: The majority of the thermal load is borne by the ceramic top-coat, which reduces the exposure temperature of the oxidation resistant TGO that is connected to the bond coat and subsequently the superalloy substrate. The superalloy substrate is also exposed to cooling, which further reduces the temperature of the material. A properly designed cooling system is particularly effective when used in conjunction with TBCs, and can help raise the temperature of the operating environment by more than 100 degrees.[53][16]

Single-crystal superalloy compounds have an extensive history of success in the turbine industry due to their excellent mechanical properties and ability to withstand the extreme environments of turbine applications, especially when combined with TBCs and smart cooling systems. Turbine manufacturers, however, are starting to see a plateau in the improvement that the combined efforts of superalloys and TBCs can yield for operating temperatures and have started to explore a new set of material systems. A promising set of compounds that can potentially raise the operating temperatures of turbine hot sections even further are Ceramic Matrix Composites (CMCs). CMCs have the ability to retain their mechanical performance hundreds of degrees beyond current turbine operating temperatures.[32]

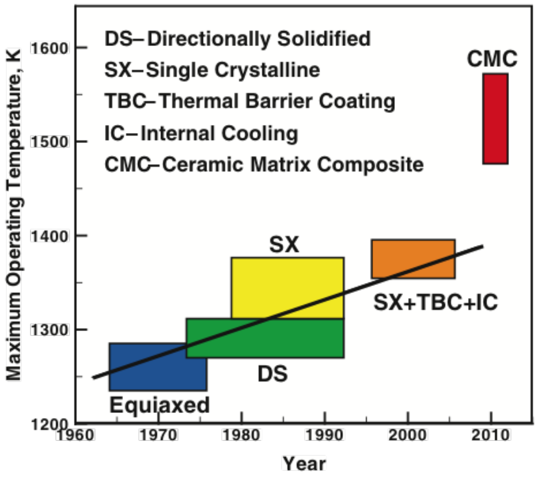


Figure 1.9: Temperature capabilities of various turbine materials showing the increase of maximum operating temperature over time. CMC materials have the potential to raise the operating temperature by hundreds of degrees beyond current levels. Adapted from Grujicic, M et al. Proc IMechE Part L. J Materials: Design and Applications 0(0) 1-20. [32]

The immense promise of operating temperature gains through the application of CMCs has sparked immense interest from turbine manufacturers, such as General Electric, Rolls Royce and Siemens, all of whom have invested significant research funds to explore the possibility of using CMCs in their turbine systems. General Electric has publicly reported various milestones of CMC integration in their turbines, including a deployment of CMCs in their latest LEAP engines in a collaboration with Oak Ridge National Laboratories in 2017 [46], as well a $200 million investment for a specialized CMC factory in Huntsville, Alabama, that the company expects to be finished by the summer of 2018 [66].

1.3 Ceramic Matrix Composites - A New Generation of Turbine Materials

Ceramic Matrix Composites (CMCs) are a class of materials that consist of a ceramic matrix material that is reinforced with another ceramic material bonded to the matrix through an intervening interfacial material. The matrix, reinforcement and interface material form an intricate composite that both retains the advantageous properties of ceramics, such as their high temperature resistance, while also alleviating major weaknesses of the materials, such as low fracture toughness, that have historically prevented the application of ceramic based materials in turbine applications. The ability of CMCs to enhance the operating temperatures of turbine systems beyond the limits of superalloy + TBC systems has been known for some time. More than 25 years ago, the Department of Energy started a program to support the development of CMC materials for aerospace applications [46]. The initial efforts from the Department of Energy was led by Oak Ridge National Laboratory, and has since expanded to various research and development collaborations between government agencies, such as the National Aeronautics and Space Association (NASA), and turbine manufacturers, such as General Electric.

In recent years, the most promising CMC system for aerospace applications have been SiC/SiC based compounds that are favored by both NASA and General Electric. Various publications from the NASA Glenn Research Center in Cleveland, OH, showcase the intricate development of SiC/SiC compounds to prepare them for aerospace applications. [35][19][50][21]

The composite developed by NASA consists of SiC fibers that are woven into a preform shape that promotes crack bridging for favorable mechanical properties. Following the arrangement of the fiber preform, the interphase material is deposited using Chemical Vapor Infiltration (CVI) followed by the deposition of the matrix SiC material, also using CVI.[19]

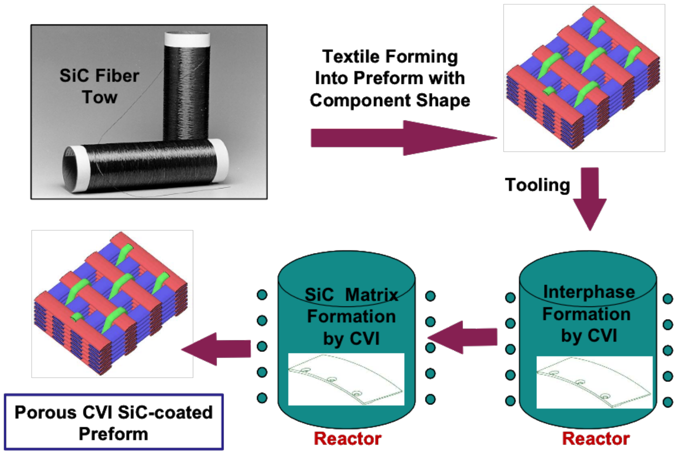

SiC Fiber Tow
Textile Forming Into Preform with Component Shape
Tooling
Interphase Formation by CVI
Reactor
SiC Matrix Formation by CVI
Reactor
Porous CVI SiC-coated Preform

Figure 1.10: NASA SiC/SiC Processing Schematic adapted from DiCarlo, J.A., et al. NASA/TM-2004-213028.[19]

NASA aims to deposit the matrix CMC material in a porous way, because the porosity of the material helps promote crack bridging by redirecting and trapping the internal stresses in the porous regions, which enhances the overall toughness of the composite material [73] by redirecting the crack energy and trapping it inside the material. While NASA, Oak Ridge National Laboratories and General Electric have historically focused on SiC/SiC composite materials, other turbine manufacturers have also considered oxide based ceramics, often referred to as Ox-Ox. The Ox-Ox CMCs often contain aluminum oxide based materials for both the fibers and the materials, but other oxide materials, such as silica, can also be used [8]. The Ox-Ox materials often have lower temperature capabilities than SiC/SiC but have better chemical stability and are usually less expensive to process. A sample micrograph of a fiber CMC shows the various layers of the CMC, and how the interphase and matrix materials surround the fiber preform.

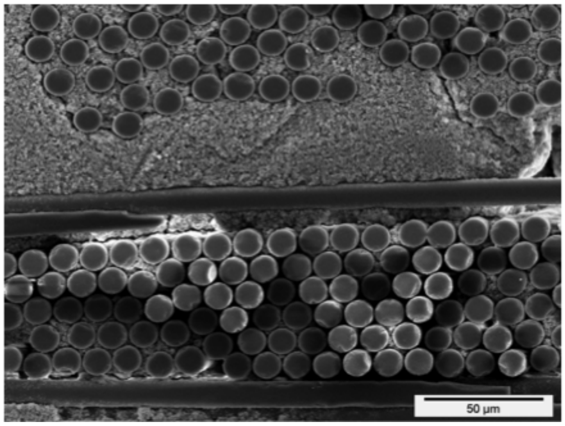


Figure 1.11: SEM Micrographs of CMC Composite showing fiber and matrix material layouts provided by the Siemens AG [62]

The micrographs also show the porosity of the matrix material, which is desired for better mechanical properties. Manufacturers aim to explore the effects of different constituent materials, as well as different geometric arrangement of the reinforcement material. When choosing the constituent materials it is important to consider the mechanical properties of both the matrix and the reinforcement material, as a high mismatch in the mechanical properties can lead to build-up of internal stress that can accelerate material failure. The internal stresses in this situation result from the stiffer material creating a high stress field in its vicinity, which can then propagate through the surrounding matrix material. The primary obstacle for experimenting with different geometric arrangement of the reinforcement materials is the processing of subsequent material layers, including the interphase material and the matrix material. In many cases, the infiltration process may not be able to reach parts of the preform leading to discontinuous coverage of the preform with both interphase and matrix material. While porosity is desired for extrinsic toughening of the material, incomplete coverage of the preform is highly unfavorable because it creates high concentrations of residual stresses in the uncovered area that lead to accelerated material failure as the matrix material can separate from the reinforcement material.

Enhancement of Mechanical Properties for Ceramic Matrix Composite

The overall mechanisms for the mechanical and thermal properties of both types of CMCs, SiC/SiC and Ox-Ox, remain the same and take advantage of similar physical phenomena. Even though CMCs have superior mechanical properties at high temperatures compared to superalloys, the application of ceramics in turbine systems has historically been limited by their low toughness and tendency for catastrophic brittle failure. Recent developments in CMCs, however, indicate that microstructural design can alleviate brittle failure, and increase the toughness of CMCs materials. Similar to steel-reinforced concrete, one can design fiber-reinforced CMC microstructures that are more resistant to brittle failure by preventing crack propagation inside the material. A crack inside the fiber-reinforced CMC matrix is stopped at the interface material and then subsequently at the fibers, which prevents the crack from spreading through the matrix material in an unstable manner and subsequently causing catastrophic failure. The mechanical energy created by the crack can be partially absorbed by the interface material, which helps alleviate the stress on the fibers. Once the stress on the fiber material becomes too high, the fiber will begin to crack and to debond from the matrix matrix, leading to pull-out failure that is shown in Figure 1.12.

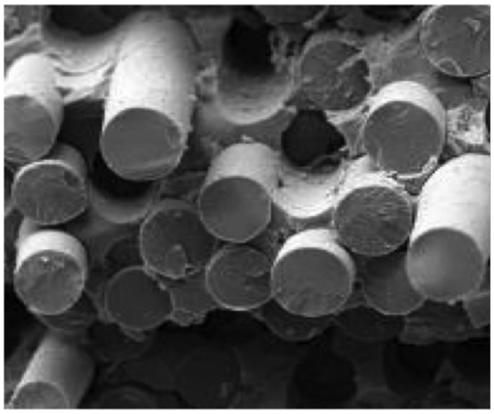


Figure 1.12: SEM Micrograph of CMC Fiber Pull-Out Failure with $\sim 10\,\mu m$ fibers provided by the Siemens AG [62]

Toughening for Ceramic Matrix Composites

In order to mitigate crack propagation, a material can be toughened through both intrinsic and extrinsic mechanisms to prevent cracks from initiating and spreading inside the material.[55]

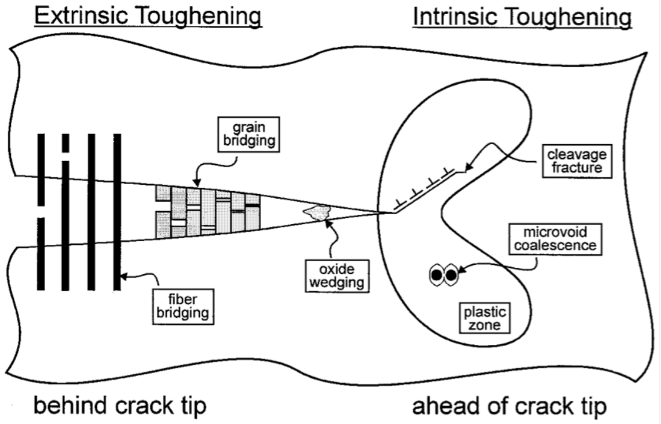


Figure 1.13: Intrinsic and Extrinsic Toughening Mechanisms against Crack Propagation taken from Ritchie, R. O. International Journal of Fracture 100: 55–83, 1999. [55]

Intrinsic toughening mechanisms are based on the inherent material properties and depend on the material's ability to create inelastic deformation zones to redirect the energy created by the crack. Since brittle materials have very limited mechanisms for diverting the energy of the crack into inelastic zones, ceramic materials mostly rely on extrinsic toughening mechanisms to prevent crack advancement. Extrinsic toughening mechanisms are active in the crack wake and become relevant during crack propagation, after the crack has already been initiated. Common extrinsic toughening mechanisms are fiber bridging and grain bridging.[55] In the case of CMC materials relevant to this study, fiber bridging is the most relevant extrinsic toughening mechanism.

Whereas crack propagation inside a single phase ceramic usually occurs in an unstable fashion where the crack spreads through the material without control or restrictions, crack propagation inside the CMC is a bit more complicated due to the fiber bridging provided by the CMC microstructure. Crack propagation inside a fiber-reinforced CMC can be divided into distinct sections:

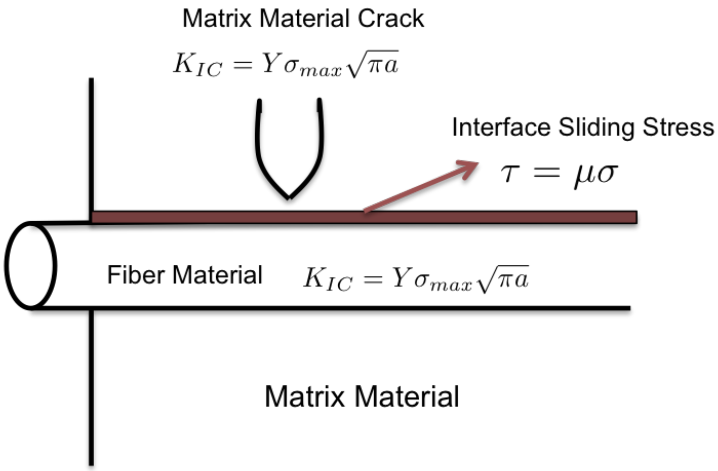


Figure 1.14: Crack Propagation Phase Inside CMC split into three distinct zones: matrix, interface, and fiber material

1. Crack propagation inside the matrix material.
2. Sliding at the fiber-matrix interface: The sliding stress at the fiber-matrix interface can be modeled as a Mode II crack using Coulombic friction for the frictional shear stress based on the work of Kuntz and Grathwohl[41] : $\tau = \zeta_{friction}\sigma$ where σ represents the stress in the matrix material that is normal to the sliding stress τ and $\zeta_{friction}$ is the friction coefficient of the interface, which is described in more detail in Chapter 4.
3. Crack propagation inside the fiber material.

The deflection of a crack along the interface and the fibers leads to the crack bridging phenomenon, where the crack is deflected inside the microstructure, thereby preventing catastrophic failure of the CMC.[55] Since the constituent ceramic materials inside the CMC are inherently brittle and often do not have the properties to promote intrinsic toughening mechanisms, the composite material relies on extrinsic mechanisms to maintain mechanical integrity.

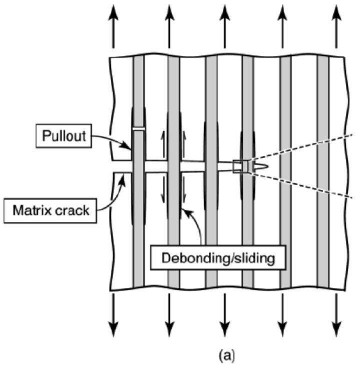


Figure 1.15: Crack Bridging in Fiber Reinforced CMC where pull-out failure happens after fiber breaking adapted from the Siemens AG [62]

1.4 Modeling Properties of Ceramic Matrix Composites

In this dissertation, I will present research focused on modeling mechanical and thermal properties of CMC materials based on a continuum mechanics approach. The fundamental equations governing the mechanical and thermal behavior of CMCs, as well as the mathematical approach used to solve them, are presented in greater detail in Chapter 2 combined with additional computational methods used in this dissertation. In the subsections that follow, I will focus on illustrating the models used to describe the mechanical behavior of CMCs.

Damage Modeling in Ceramic Matrix Composites

One possible method for modeling the propagation of cracks inside the CMC is to use a fracture mechanics approach that tracks the position, energy and stresses produced by the crack inside the CMC material. This type of modeling, however, is very computationally intensive and requires many processing steps, such as iterative re-meshing and tracking of the various crack fronts, to ensure accuracy. At the scale of interest for the work in this thesis, however, the effects of microcracks generated inside the CMC via mechanical and thermal loads can be modeled by damage evolution in the elasticity tensor [39] . The damage mechanics approach averages the effects of cracks and other defects inside the materials into an overall homogenized response of the broader material. Damage modeling assumes that the aggregate of material defects inside of representative volume element at the mesoscale can be represented by the material's mechanical response to stress. A damage evolution based model provides a good balance between computational efficiency and modeling the mesoscale effects inside the material that can lead to material failure.

A damage based modeling approach that suits itself for CMC modeling is one that models the brittle nature of the constituent ceramic materials, while keeping the overall ductile response of the broader CMC compound. Once the material fails in a certain section, damage mechanics dictates that the elastic response of that part of the material is set to 0, which leads to softening of the aggregate elastic response. Furthermore, a critical average damage value for the entire compound can be defined, beyond which point the CMC material system is considered structurally unstable. The damage model used for this study is based on the internal stresses generated inside the CMC compound, which are defined by the following stress dependent damage model[39] :

$$\mathcal{D}_{avg}^{initial} = 1.0$$

throughout the entire material system. Once the stress reaches a critical point at a given location the elastic response of that section changes

$$\|\boldsymbol{\sigma}'\| > \sigma_{critical} \rightarrow \mathcal{D}_{local} = 0 \tag{1.2}$$

By setting $\mathcal{D}_{local} = 0$, brittle failure is simulated at the local level and the overall mechanical response of the material changes. This model is implemented by using an element deletion procedure within the finite element framework used for the continuum mechanics modeling of the CMC, which is described further in Chapter 2. In the case that the internal stresses do not reach a critical value, the local damage remain at its initial value:

$$\|\boldsymbol{\sigma}'\| < \sigma_{critical} \rightarrow \mathcal{D}_{local} = 1.0 \tag{1.3}$$

where $\boldsymbol{\sigma}'$ represents the deviatoric stress inside the material is defined by

$$\boldsymbol{\sigma}' = \boldsymbol{\sigma} - \frac{tr\boldsymbol{\sigma}}{3} \cdot \mathbf{1} \tag{1.4}$$

where $\boldsymbol{\sigma}$ represent the stress tensor calculated from the thermo-mechanical equations using the finite element method. The herein described damage model can then be used as a basis for virtual tests designed to estimate the toughness of a given CMC microstructure, which will be described in Chapter 2.

Homogenization of CMC Effective Properties

The continuum mechanics modeling framework employed in this dissertation is based on homogenization of the composite material properties. The homogenization process connects the structural scale simulation of actual turbine blade geometries to the mesoscale structure of the composite material. The properties of the mesoscale structure of the composite are homogenized as the overarching material properties in the structural scale simulation. Each combination of constituent materials with each unique mesoscale structure will therefore have unique material properties that can be connected to structural scale simulation. A change in either the mesoscale or the underlying properties of the constituent materials will therefore result in a change of the effective properties of the composite material, which will be reflected in the model.

The primary aim of the computational framework is to provide a method by which fully functional virtual two-phase CMC compounds, including both unique constituents and mesoscales, can be computer generated and evaluated for automated prototyping. This automated prototyping of the materials design allows fast and efficient evaluation of CMC compounds, which can then be fed into more detailed analysis methods for further evaluation. The structural scale simulations presented in this study are one example of such higher-fidelity analyses that can be performed to further access the viability of the proposed CMC design.

1.5 Research Goals and Thesis Outline

The overarching focus of the research presented in this thesis is to use continuum mechanics based modeling techniques, primarily the Finite Element Method (FEM), combined with evolutionary algorithms for non-convex optimization to address the materials design problem of creating Ceramic Matrix Composites for gas turbine applications. Although not fully demonstrated in this dissertation, it is envisioned that the tool developed in this work can be combined with machine learning methods to enable automated design and optimization of CMC materials with targeted properties. The continuum model then draws on the homogenization described in Section 1.4 to provide insight as to how the given composite could perform in a structural scale application of a turbine blade, which is the primary component of the desired application.

The mathematical details of the models, as well as the relevant methods, are described in subsequent chapters, particularly Chapter 2, starting from a fundamental continuum mechanics approach that solves the equations related to mechanical properties and thermal conductivity. The equations are supplemented by the damage mechanics model described in Section 1.4 to provide further insight into how the CMC could fail under a given set of loading conditions, which are outlined in greater detail in Chapter 3. The future work described in Chapter 4 focuses on efforts that can be used to improve upon the modeling framework to be able to describe more complex material structures and model certain mechanical and thermal behavior more accurately.

The main focus of this chapter was to motivate the work and results presented in subsequent chapters, and to put them in a broader engineering context. The subsequent chapters will focus on the following:

- Chapter 2 - Methods: Chapter 2 focuses on outlining the mathematical details of the methods used for the modeling efforts, as well as the algorithms used for the materials design problem. This chapter outline the partial differential equations used to define the mechanical and thermal model for the material system, the finite element analysis approach used to solve those equations, the interaction of the damage model with the finite element model, as well as the evolutionary algorithms used for making the material and microstructural design choices.

- Chapter 3 - Case Study: Chapter 3 provides a comprehensive case study of how the different tools developed in Chapter 2 interact to create and analyze a virtual CMC material.
- Chapter 4 - Future Work: Chapter 4 focuses on outlining efforts that can improve the overall framework, including enhancement in the material model as well as algorithmic improvements.

Chapter 2

Computational Methods

2.1 Overall Computational Design Framework

The primary goal of this study is to propose a framework for computational design of CMCs that can be used to rapidly prototype various CMC designs for distinct applications computationally. The proposed framework focuses on a computational approach to circumvent the high time and resource costs of experimental prototyping of CMC materials, and aims to connect the material mesoscale that captures the material microstructure with the structural scale of a turbine blade using a continuum mechanics based approach. Furthermore, the material design at the mesoscale is performed in a tiered approach starting with an isotropic material model to find a set of suitable constituent materials for the composite followed by a microstructural design algorithm to further analyze the best-performing microstructures for the previously obtained material candidates. The final step of the mesoscale material design involves virtual tests of the composite with its distinct microstructure, in order to gain further insight into damage evolution of the given CMC and its potential effects on the effective toughness and fatigue behavior of the material.

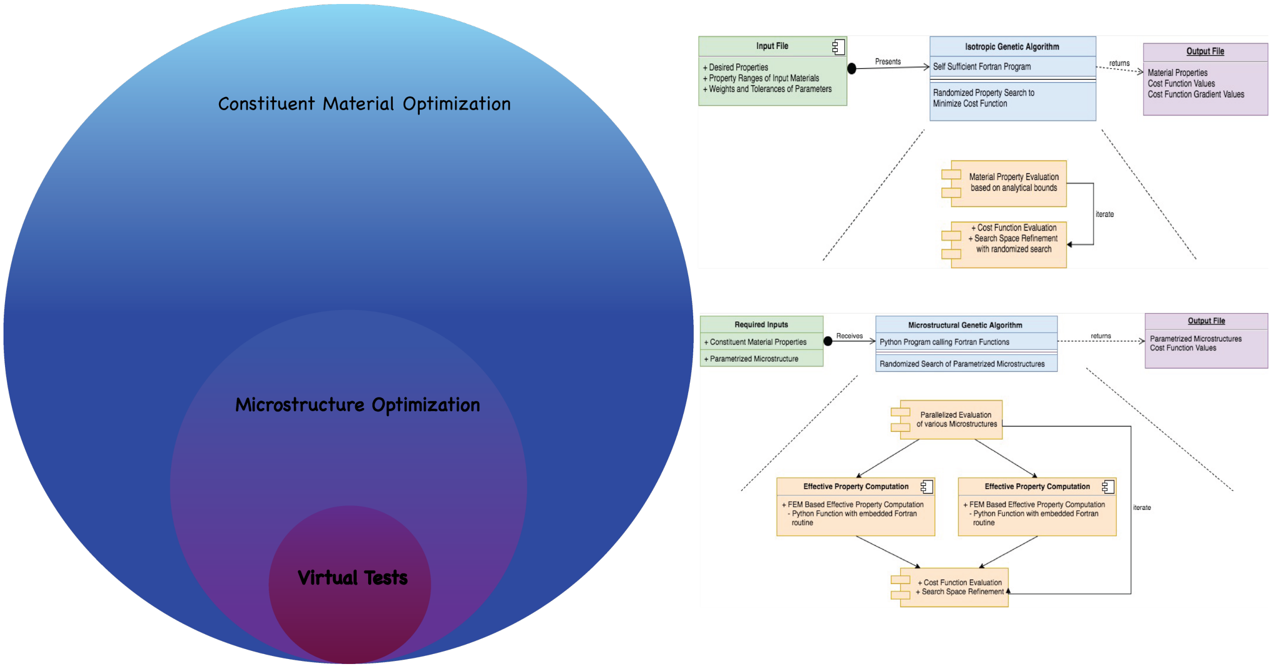
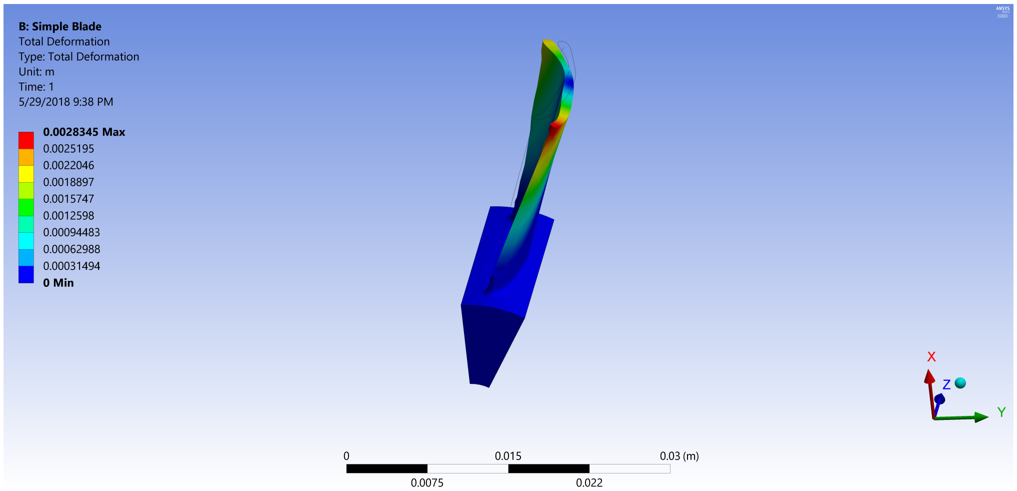

Constituent Material Optimization
Microstructure Optimization
Virtual Tests
Input File
+ Desired Properties
+ Property Ranges of Input Materials
+ Weights and Tolerances of Parameters
Presents
Isotropic Genetic Algorithm
Self Sufficient Fortran Program
Randomized Property Search to Minimize Cost Function
returns
Output File
Material Properties
Cost Function Values
Cost Function Gradient Values
Material Property Evaluation based on analytical bounds
iterate
+ Cost Function Evaluation
+ Search Space Refinement with randomized search
Required Inputs
+ Constituent Material Properties
+ Parametrized Microstructure
Receives
Microstructural Genetic Algorithm
Python Program calling Fortran Functions
Randomized Search of Parametrized Microstructures
returns
Output File
Parametrized Microstructures
Cost Function Values
Parallelized Evaluation of various Microstructures
Effective Property Computation
+ FEM Based Effective Property Computation
- Python Function with embedded Fortran routine
Effective Property Computation
+ FEM Based Effective Property Computation
- Python Function with embedded Fortran routine
iterate
+ Cost Function Evaluation
+ Search Space Refinement

Figure 2.1: CMC Computational Design Framework: Circles represent the relative size of the design spaces. The largest design space starts with the constituent materials optimization followed the microstructure optimization, which is in turn followed by the virtual tests.

The figure above summarizes the previously described approach while adding some further details into the actual program design used to perform the various steps. The size of the circles in the figure corresponds to the relative amount of the available CMC virtual material candidates, which is decreased with each step in the tiered design process. Once a given virtual CMC has progressed through the various stages of the mesoscale design framework, the effective material properties are integrated into a structural scale simulation of a turbine blade using the ANSYS simulation suite shown in the figure below.



B: Simple Blade
Total Deformation
Type: Total Deformation
Unit: m
Time: 1
5/29/2018 9:38 PM

Figure 2.2: Sample Ansys Turbine Blade Simulation showing a deformation simulation of a turbine blade part

The computational approach is based on constructing a Representative Volume Element (RVE) of the CMC material system that will be analyzed and simulated. The RVE is used to perform the relevant simulations and analysis on a characteristic sample of the larger material that is taken from the blade itself as shown in the figure below:

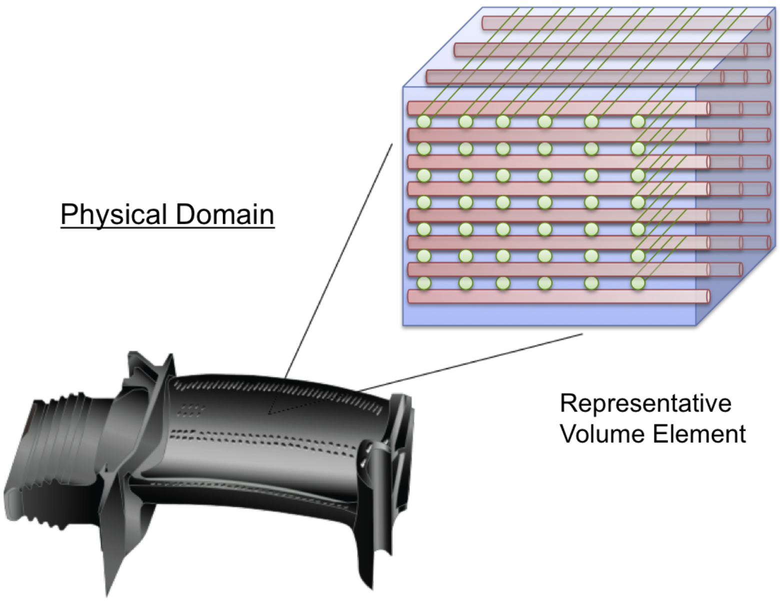


Figure 2.3: Representative Volume Element of CMC Blade

From the RVE, the proposed research aims to focus on three distinct efforts:

- CMC Compound Prototyping: Compute effective CMC properties using reduced-order models followed by higher fidelity Finite Element Method computations

The search for new CMC material systems will be performed by computing the effective properties of a given CMC compound based on the properties of the constituent matrix and fiber materials, as well as the microstructure of the CMC. The effective properties of the given CMC compound will be computed using an self-developed Finite Element Method (FEM) program, which can calculate effective thermo-mechanical properties for a variety of microstructures. The results of the effective property calculation from the FEM program can then be transferred to a Genetic Algorithm, which can then perform a non-convex optimization for the mesoscale structure of the CMC.

- CMC Critical Fracture Stress Analysis

The CMC fracture toughness analysis will apply the FEM program to deliver a mechanical impulse to the given RVE and determine when the RVE reaches a state at which a critical amount of damage inside the material is exceeded.

- CMC Fatigue Analysis

The CMC fatigue analysis will focus on applying a phenomenological model of fatigue based on average stresses inside the material.

2.2 Constitutive Equations for Mechanical and Thermal Behavior

Understanding thermo-mechanical processes is critical in turbine application environments due to the substantial thermal and mechanical loads turbine materials experience in their operating environment. The balance of linear momentum and the thermodynamic energy balance govern mechanical behavior and heat conduction, respectively.

Balance of Linear Momentum

The balance of linear momentum is given by [68]

$$\nabla_x \boldsymbol{\sigma} + \mathbf{f} = \rho \frac{d^2\mathbf{u}}{dt^2} \tag{2.1}$$

where σ represents the Cauchy stress, $\mathbf{f}$ represents body forces, ρ is the material density, and $\mathbf{u}$ represents the displacement. When considering only small deformations, we can approximate $\frac{d()}{dt} \approx \frac{\partial()}{\partial t}|X$ and $\nabla_{\mathbf{x}} = \nabla_{\mathbf{X}}$ where $\mathbf{x}$ is the current configuration and $\mathbf{X}$ is the reference configuration of the system. The isotropic constitutive law for the stress-strain relationship used in this study is given by

$$\boldsymbol{\sigma} = \mathcal{D}\mathbb{E} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon_{\theta}}) \tag{2.2}$$

including the elastic strain $\boldsymbol{\epsilon}$ and the thermal strain $\boldsymbol{\epsilon_{\theta}}$. The thermal strain model used in this study is dependent on the initial temperature of the system θ_0, the current temperature θ and a constant γ:[69]

$$\boldsymbol{\epsilon_{\theta}} = \alpha \cdot (\theta - \theta_0) \cdot \mathbf{1} \tag{2.3}$$

Under this framework with $\rho \approx \rho_0$, the balance of linear momentum then becomes

$$\nabla_X \cdot (\mathcal{D}\mathbb{E} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon_{\theta}})) + \mathbf{f} = \rho_0 \frac{\partial^2\mathbf{u}}{\partial^2 t} \tag{2.4}$$

Energy Balance for Heat Conduction

Heat conduction inside a given material system is governed by the thermodynamic energy balance, which can be summarized in the following differential equation:[68]

$$\rho\dot{w} - \sigma : \nabla_x \dot{u} + \nabla_x q - \rho z = 0 \tag{2.5}$$

where

$$\rho w = W \approx \frac{1}{2}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon_{\theta}}) : \mathcal{D}\mathbb{E} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon_{\theta}}) + \rho C_p \theta \tag{2.6}$$

which leads to

$$\rho\dot{w} = \dot{W} = (\dot{\boldsymbol{\epsilon}} - \dot{\boldsymbol{\epsilon_{\theta}}}) : \mathcal{D}\mathbb{E} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon_{\theta}}) + \frac{1}{2}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon_{\theta}}) : \dot{\mathbb{E}} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon_{\theta}}) + \rho C_p \dot{\theta} \tag{2.7}$$

which then yields the following energy balance assuming only small deformations

$$\rho_0 C_p \dot{\theta} = \sigma : \dot{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} - \frac{1}{2}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}) : \mathcal{D}\dot{\mathbb{E}} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}) + \nabla_X(\mathbb{K}\nabla_X \theta) + \rho_0 z \tag{2.8}$$

assuming linear heat conduction, $q = q_0 = -\mathbb{K}\nabla_X \theta$ using Fourier's law where θ represent the temperature.

The thermo-mechanical behavior of a given material system is governed by the partial differential equations stated above. The domain for this study is a unit cube that serves as a Representative Volume Element (RVE) describing the material compounds and microstructure of the larger system. The equations are solved in a staggered scheme: First, the balance of linear momentum is solved to determine the displacements, the strains and mechanical characteristics of the system. The information from the mechanical solutions will then be used to solve the thermodynamical energy balance, which will yield the temperature distribution of the system. Since an implicit time-stepping scheme is used, the fields will often have to be solved iteratively until convergence for both fields is reached.

The computer program for solving the partial differential equations is written in a combination of Python and Fortran. The heavily used modules, such as the Gaussian integrator, will be written in compiled language, mainly Fortran, for faster computing, while the main program will be written in Python due to data management advantages. Below is a schematic of the outlining the overall architecture of the numerical solver

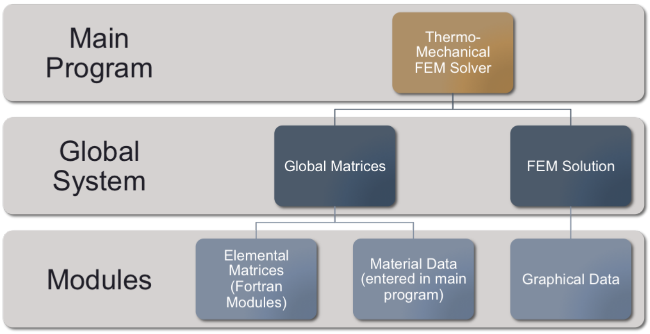

Main Program
Thermo-Mechanical FEM Solver
Global System
Global Matrices
FEM Solution
Modules
Elemental Matrices (Fortran Modules)
Material Data (entered in main program)
Graphical Data

Figure 2.4: Architecture of Numerical Solver: Main program, written in Python, manages the global matrix system and the final FEM solution output. The solution output is rendered to graphical data and post-processed for further analysis. The global matrices are dependent on the elemental matrices and the material data.

The overall solver flowchart for the staggered system of partial differential equations using implicit time-stepping is as follows:

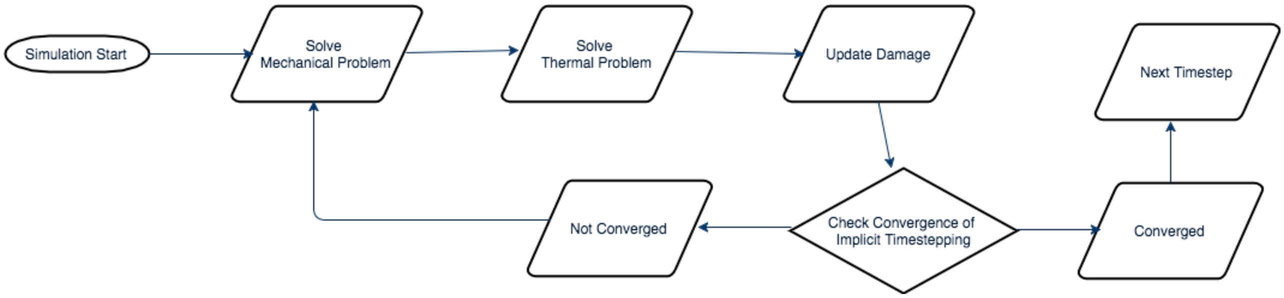


Figure 2.5: Staggered FEM Solver for Thermo-Mechanical Coupled System solved with Backward Euler Implicit Time-stepping. In the coupled solver, on problem (either mechanical or thermal) is solved while the other field is held constant.

Subsequent sections will focus on the mathematical details of the numerical methods applied to solve the aforementioned equations.

2.2.1 Balance of Linear Momentum

The first step in the finite element method is to derive the weak form of the equation by integrating the equation over the domain of the problem and multiplying the equation by a test function v:

$$\nabla_X \boldsymbol{\sigma} + \mathbf{f} = \rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} \tag{2.9}$$

$$\int_\Omega v \cdot (\nabla_X \boldsymbol{\sigma} + \mathbf{f})\, d\Omega = \int_\Omega v \cdot (\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2})\, d\Omega \quad \forall v \in H_{1(\Omega)} \tag{2.10}$$

This equation can be modified using the product rule of differentiation:

$$\int_\Omega \nabla \cdot (\boldsymbol{\sigma} \colon v)\, d\Omega = \int_\Omega v \cdot (\nabla \boldsymbol{\sigma})\, d\Omega + \int_\Omega \nabla v \colon \boldsymbol{\sigma}\, d\Omega \quad \forall v \in H_{1(\Omega)} \tag{2.11}$$

which leads to the following equation

$$\int_\Omega \nabla_X \cdot (\boldsymbol{\sigma} \colon v)\, d\Omega - \int_\Omega \nabla_X v \colon \boldsymbol{\sigma}\, d\Omega + \int_\Omega v \cdot \mathbf{f}\, d\Omega = \int_\Omega v \cdot (\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2})\, d\Omega \quad \forall v \in H_{1(\Omega)} \tag{2.12}$$

Next the divergence theorem is applied

$$\int_\Omega \nabla_X \cdot (\boldsymbol{\sigma} \colon v)\, d\Omega = \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot v\, dS \tag{2.13}$$

yielding

$$\int_\Omega \nabla_X v \colon \boldsymbol{\sigma}\, d\Omega + \int_\Omega v \cdot (\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2})\, d\Omega = \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot v\, dS + \int_\Omega v \cdot \mathbf{f}\, d\Omega \quad \forall v \in H_{1(\Omega)} \tag{2.14}$$

which applying the boundary condition becomes

$$\int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot v \, dS = \int_{\Gamma_t} \boldsymbol{t} \cdot v \, dS \tag{2.15}$$

$\forall v \textit{ such that } v = 0 \textit{ on } \Gamma_d \, where \, \Gamma_d \, represents \, surface \, with \, Dirichlet \, Boundary \, Conditions$

Now we can apply the constitutive law for the Cauchy stress

$$\boldsymbol{\sigma} = \mathcal{D}\mathbb{E} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon_\theta})$$

to obtain the equation

$$\int_\Omega \nabla_X v \colon (\mathcal{D}\mathbb{E} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon_\theta})) \, d\Omega + \int_\Omega v \cdot (\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2}) \, d\Omega = \int_{\partial\Omega} \boldsymbol{\sigma} \cdot \boldsymbol{n} \cdot v \, dS + \int_\Omega v \cdot \mathbf{f} \, d\Omega \tag{2.16}$$

which can be broken into the different strain integrals

$$\int_\Omega \nabla_X v \colon (\mathcal{D}\mathbb{E} : \boldsymbol{\epsilon}) \, d\Omega - \int_\Omega \nabla_X v \colon (\mathcal{D}\mathbb{E} : \boldsymbol{\epsilon_\theta}) \, d\Omega + \int_\Omega v \cdot (\rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2}) \, d\Omega = \int_\Omega \{v\}^T \cdot \mathbf{f} \, d\Omega + \int_{\Gamma_t} v\{\bar{t}\} \cdot \boldsymbol{t} \, d\Gamma \tag{2.17}$$

Now the finite element discretization can be applied

$$u(x,y,z) = \sum_{A=1}^{N} a_A \phi_A(x,y,z) \qquad v(x,y,z) = \sum_{A=1}^{N} b_A \phi_A(x,y,z) \tag{2.18}$$

which is equivalent to

$$u_i(x,y,z) = \sum_{A=1}^{N} a_i^A \phi^A(x,y,z) \qquad v_i(x,y,z) = \sum_{A=1}^{N} b_i^A \phi^A(x,y,z) \quad i = 1,2,3 \tag{2.19}$$

The discretization consists of two parts: $\{a_A(x,y,z)\}$ is a vector of coefficients that will be computed in the final system of matrix equations. $\{\phi_A(x,y,z)\}$ is a vector of linear shape functions that are used to approximate the solution of the differential equation. The shape functions used for the finite element procedure of this study are linear shape functions in three dimensions.

The next step in obtaining the numerical solution of the mechanical problem is to discretize the strain terms. Using Voigt notation that takes advantage of inherent symmetry in the elasticity tensor $\mathbb{E}$, one can write the elasticity equation as

$$\boldsymbol{\sigma} = \mathbb{E} : \boldsymbol{\epsilon} \tag{2.20}$$

then becomes

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{Bmatrix} = \begin{bmatrix} \mathbb{E}_{11} & \mathbb{E}_{12} & \mathbb{E}_{13} & \mathbb{E}_{14} & \mathbb{E}_{15} & \mathbb{E}_{16} \\ \mathbb{E}_{21} & \mathbb{E}_{22} & \mathbb{E}_{23} & \mathbb{E}_{24} & \mathbb{E}_{25} & \mathbb{E}_{26} \\ \mathbb{E}_{31} & \mathbb{E}_{32} & \mathbb{E}_{33} & \mathbb{E}_{34} & \mathbb{E}_{35} & \mathbb{E}_{36} \\ \mathbb{E}_{41} & \mathbb{E}_{42} & \mathbb{E}_{43} & \mathbb{E}_{44} & \mathbb{E}_{45} & \mathbb{E}_{46} \\ \mathbb{E}_{51} & \mathbb{E}_{52} & \mathbb{E}_{53} & \mathbb{E}_{54} & \mathbb{E}_{55} & \mathbb{E}_{56} \\ \mathbb{E}_{61} & \mathbb{E}_{62} & \mathbb{E}_{63} & \mathbb{E}_{64} & \mathbb{E}_{65} & \mathbb{E}_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{12} \\ 2\epsilon_{23} \\ 2\epsilon_{31} \end{Bmatrix} \tag{2.21}$$

Furthermore, the strain tensor $\{\boldsymbol{\epsilon}\}$ can be written in terms of spatial derivatives

$$\{\boldsymbol{\epsilon}\} = \begin{Bmatrix} \frac{\partial u_1}{\partial x_1} \\ \frac{\partial u_2}{\partial x_2} \\ \frac{\partial u_3}{\partial x_3} \\ \frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1} \\ \frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2} \\ \frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} \end{Bmatrix} \tag{2.22}$$

and one can define a matrix $[T]$ to represent the strains as a function of the displacements

$$[T] = \begin{Bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \end{Bmatrix} \tag{2.23}$$

When $[T]$ is multiplied by the shape functions contained in the test function v, the result is a 3x24 matrix that is combined with the other terms in the respective integral. More precisely,

$$\frac{\partial u_i}{\partial x_j} = \sum_{A=1}^{N} a_i^A \frac{\partial \phi^A}{\partial x_j} \tag{2.24}$$

which makes it useful to define the following matrix $[T]$

$$[T] = \begin{Bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \end{Bmatrix} \tag{2.25}$$

which allows the strain tensor to rewritten as

$$\{\epsilon\} = [T] \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

Now we define the matrix 3x24 $[\phi]$ as follows

$$[\phi] = \begin{array}{cccccccc} \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 & \phi_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{2.26}$$

as the first block of eight columns

$$[\phi] = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 & \phi_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \tag{2.27}$$

as the second block of eight columns

$$[\phi] = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 & \phi_8 \end{matrix} \tag{2.28}$$

as the third block of eight columns. All three blocks are put together to form the full matrix. This what not shown explicitly here due to space limitations of the text processor. A similar approach will be used to express other large matrices. Now, if we define the solution vector $\{a\}$ to be a 24x1 vector of the following form

$$\{a\} = \begin{Bmatrix} a_1^1 \\ a_2^1 \\ a_3^1 \\ a_1^2 \\ a_2^2 \\ a_3^2 \\ \vdots \\ a_3^8 \end{Bmatrix} \tag{2.29}$$

we can define the displacement vector as

$$u(x) = [\phi]\{a\} \tag{2.30}$$

and the strain tensor as

$$\{\epsilon\} = [T][\phi]\{a\} \tag{2.31}$$

where $[T][\phi]$ is

$$[T][\phi] = \begin{bmatrix} \frac{\partial\phi^1}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial\phi^1}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial\phi^1}{\partial x_3} \\ \frac{\partial\phi^1}{\partial x_2} & \frac{\partial\phi^1}{\partial x_1} & 0 \\ 0 & \frac{\partial\phi^1}{\partial x_3} & \frac{\partial\phi^1}{\partial x_2} \\ \frac{\partial\phi^1}{\partial x_1} & 0 & \frac{\partial\phi^1}{\partial x_3} \end{bmatrix} \tag{2.32}$$

for the first block

$$[T][\phi] = \begin{bmatrix} \frac{\partial \phi^2}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial \phi^2}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial \phi^2}{\partial x_3} \\ \frac{\partial \phi^2}{\partial x_2} & \frac{\partial \phi^2}{\partial x_1} & 0 \\ 0 & \frac{\partial \phi^2}{\partial x_3} & \frac{\partial \phi^2}{\partial x_2} \\ \frac{\partial \phi^2}{\partial x_1} & 0 & \frac{\partial \phi^2}{\partial x_3} \end{bmatrix} \tag{2.33}$$

for the second block

$$[T][\phi] = \begin{bmatrix} \frac{\partial \phi^3}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial \phi^3}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial \phi^3}{\partial x_3} \\ \frac{\partial \phi^3}{\partial x_2} & \frac{\partial \phi^3}{\partial x_1} & 0 \\ 0 & \frac{\partial \phi^3}{\partial x_3} & \frac{\partial \phi^3}{\partial x_2} \\ \frac{\partial \phi^3}{\partial x_1} & 0 & \frac{\partial \phi^3}{\partial x_3} \end{bmatrix} \tag{2.34}$$

for the third block, all of which assembled yield the full $[T][\phi]$. This yields the following matrix equation:

$$\underbrace{\int_\Omega ([T]\{v\})^T [\mathcal{D}\mathbb{E}]([T]\{\boldsymbol{u}\}) d\Omega}_{[S]_1} - \underbrace{\int_\Omega ([T]\{v\})^T [\mathcal{D}\mathbb{E}](\boldsymbol{\epsilon_\theta}) d\Omega}_{\{F\}_\theta} + \underbrace{\int_\Omega \{v\}^T \cdot \rho_0 \ddot{\boldsymbol{u}}\, d\Omega}_{[M]} \tag{2.35}$$

$$= \underbrace{\underbrace{\int_\Omega \{v\}^T \cdot \mathbf{f}\, d\Omega}_{\{R\}_f} + \underbrace{\int_{\Gamma_t} v\{\bar{t}\} \cdot \boldsymbol{t}\, d\Gamma}_{\{R\}_t}}_{\{R\}} \tag{2.36}$$

which completes the spatial discretization of the equation. Next, the time discretization is applied using an Implicit Backward Euler Method. The above equation contains a non-linear second order derivative, which is linearized using the following method:

$$\{vel\} = \{\dot{\boldsymbol{u}}\} \tag{2.37}$$

which leads to

$$\int_\Omega \{v\}^T \cdot \rho_0 \ddot{\boldsymbol{u}}\, d\Omega = \int_\Omega \{v\}^T \cdot \rho_0 \{\dot{vel}\}\, d\Omega \tag{2.38}$$

The derivative is then approximated using the following scheme

$$\frac{\partial \{vel\}}{\partial t}\bigg|_t \approx \frac{\{vel\}(t + \Delta t) - \{vel\}(t)}{\Delta t} \tag{2.39}$$

which rearranged in an implicit time-stepping scheme yields

$$\{vel\}(t + \Delta t) = \{vel\}(t) + \Delta t \cdot \{\dot{vel}\} \tag{2.40}$$

This equation is supplemented by

$$\{a(t+\Delta t)\} = \{a(t)\} + \Delta t \cdot \{vel(t+\Delta t)\} \tag{2.41}$$

The weak form now becomes

$$\int_\Omega ([T]\{v\})^T[\mathcal{D}\mathbb{E}]([T]\{\boldsymbol{u}(t+\Delta t)\})d\Omega - \int_\Omega ([T]\{v\})^T[\mathcal{D}\mathbb{E}]\{\boldsymbol{\epsilon_\theta}(t+\Delta t)\}d\Omega \tag{2.42}$$

$$+\int_\Omega \{v\}^T \cdot \rho_0 \frac{\{vel(t+\Delta t)\} - \{vel(t)\}}{\Delta t}\, d\Omega = \int_\Omega \{v\}^T \cdot \mathbf{f}\, d\Omega + \int_{\Gamma_t} v\{\bar{t}\} \cdot \boldsymbol{t}\, d\Gamma \tag{2.43}$$

with the stiffness matrix terms on the left-hand side:

$$\underbrace{\int_\Omega ([T]\{v\})^T[\mathcal{D}\mathbb{E}]([T]\{\boldsymbol{u}(t+\Delta t)\})d\Omega}_{[S]_1} - \underbrace{\int_\Omega ([T]\{v\})^T[\mathcal{D}\mathbb{E}]\{\boldsymbol{\epsilon_\theta}(t+\Delta t)\}d\Omega}_{\{F\}_\theta} \tag{2.44}$$

as well as the mass matrix terms on the left-hand side:

$$+\frac{1}{\Delta t}\underbrace{\int_\Omega \{v\}\rho_0\{vel(t+\Delta t)\}d\Omega}_{[M]\{\dot{a}(t+\Delta t)\}} - \frac{1}{\Delta t}\underbrace{\int_\Omega \{v\}\rho_0\{vel(t)\}d\Omega}_{[M]\{\dot{a}(t)\}} \tag{2.45}$$

and the right-hand side

$$\underbrace{\underbrace{\int_\Omega \{v\}^T \cdot \mathbf{f}\, d\Omega}_{\{R\}_f} + \underbrace{\int_{\Gamma_t} v\{\bar{t}\} \cdot \boldsymbol{t}\, d\Gamma}_{\{R\}_t}}_{\{R\}} \tag{2.46}$$

Using the same discretization scheme as above,

$$u(x,y,z) = \sum_{A=1}^{N} a_A\phi_A(x,y,z) \qquad v(x,y,z) = \sum_{A=1}^{N} b_A\phi_A(x,y,z) \qquad \dot{u}(x,y,z) = \sum_{A=1}^{N} \dot{a}_A\phi_A(x,y,z) \tag{2.47}$$

yields the following discretized weak form equation. For the left-hand-side strain terms:

$$\int_\Omega ([T]\{\sum_{A=1}^{N} b_A\phi_A\})^T[\mathcal{D}\mathbb{E}]([T]\{\sum_{A=1}^{N} a_A(t{+}\Delta t)\phi_A\})d\Omega + \int_\Omega ([T]\{\sum_{A=1}^{N} b_A\phi_A\})^T[\mathcal{D}\mathbb{E}](\{\boldsymbol{\epsilon_\theta}(t{+}\Delta t)\})d\Omega \tag{2.48}$$

as well as the mass matrix terms on the left-hand side:

$$+\frac{1}{\Delta t}\int_\Omega \sum_{A=1}^{N} b_A\phi_A\rho \sum_{B=1}^{N} \dot{a_B}(t+\Delta t)\phi_B d\Omega - \frac{1}{\Delta t}\int_\Omega \sum_{A=1}^{N} b_A\phi_A\rho_0 \sum_{B=1}^{N} \dot{a_B}(t)\phi_B d\Omega \tag{2.49}$$

and the right-hand-side

$$\int_{\Omega} \sum_{A=1}^{N} b_A \phi_A \cdot \mathbf{f}\, d\Omega + \int_{\Gamma_t} \sum_{A=1}^{N} b_A \phi_A \cdot \boldsymbol{t}\, d\Gamma \tag{2.50}$$

The b_A's are arbitrary and can be canceled out of the equation leading to the following stiffness matrix terms on the left-hand side

$$\sum_{A=1}^{N}\sum_{B=1}^{N} \underbrace{\int_{\Omega} ([T]\{\phi_A\})^T [\mathcal{D}\mathbb{E}]([T]\{\phi_B\}) d\Omega}_{[S_1]} \underbrace{a_B(t+\Delta t)}_{\{a(t+\Delta t)\}} + \underbrace{\int_{\Omega} ([T]\{\phi_A\})^T [\mathcal{D}\mathbb{E}](\{\boldsymbol{\epsilon_\theta}(t+\Delta t)\}) d\Omega}_{\{F\}_\theta} \tag{2.51}$$

and the mass matrix terms:

$$+ \sum_{A=1}^{N}\sum_{B=1}^{N} \frac{1}{\Delta t} \underbrace{\int_{\Omega} \phi_A \rho_0 \phi_B d\Omega}_{[\mathbf{M}]} \underbrace{\dot{a_B}(t+\Delta t)}_{\{\dot{a}(t+\Delta t)\}} - \frac{1}{\Delta t} \underbrace{\int_{\Omega} \phi_A \rho_0 \phi_B d\Omega}_{[\mathbf{M}]} \underbrace{\dot{a_B}(t)}_{\{\dot{a}(t)\}} \tag{2.52}$$

and the following right-hand side

$$\underbrace{\sum_{A=1}^{N} \underbrace{\int_{\Omega} \phi_A \cdot \mathbf{f}\, d\Omega}_{\{R\}_f} + \underbrace{\int_{\Gamma_t} \phi_A \cdot \boldsymbol{t}\, d\Gamma}_{\{R\}_t}}_{\{R\}} \tag{2.53}$$

The weak then becomes the following matrix system of equations

$$[S_1]\{a(t+\Delta t)\} - \{F_\theta(t+\Delta t)\} + \frac{1}{\Delta t}[M](\{\dot{a}\{t+\Delta t)\} - \{\dot{a}(t)\}) = \{R\} \tag{2.54}$$

$$\{\dot{a}(t+\Delta t)\} = (\frac{1}{\Delta t}[M])^{-1}(\{R\} - [S_1]\{a(t+\Delta t)\} + \{F_\theta(t+\Delta t)\} + \frac{1}{\Delta t}[M]\{\dot{a}(t)\}) \tag{2.55}$$

which is supplemented by

$$\{a(t+\Delta t)\} = \{a(t)\} + \Delta t \cdot \{\dot{a}(t+\Delta t)\} \tag{2.56}$$

where $\{a(t)\}$ and $\{\dot{a}(t)\}$ are both given as initial conditions of the problem to solve for the first set of time-dependent solutions. The equation can then be solved using the following process:

1. Solve the above equation for $\{\dot{a}(t+\Delta t)\}^L$ by guessing values for $\{a(t+\Delta t)\}^L$ and $\{F_\theta(t+\Delta t)\}$

2. Solve for $\{a(t+\Delta t)\}^{L+1}$ using $\{a(t+\Delta t)\}^{L+1} = \{a(t)\} + \Delta t \cdot \{\dot{a}(t+\Delta t)\}^L$

3. Compute an error estimate by inserting the computed solutions into the original equations using

$$error = \frac{\left\|\{a(t+\Delta t)\}^{L+1} - \{a(t+\Delta t)\}^{L}\right\|}{\left\|\{a(t+\Delta t)\}^{L+1}\right\|} \leqq tol \tag{2.57}$$

4. Repeat steps 1-3 if $error \geqq tol$

2.2.1.1 Evaluation of the Weak Form

The mapping from the real space to the master element in three dimensions is defined as follows

$$x_1 = \sum_{i=1}^{8} X_{1i}\hat{\phi}_i = Map_{x_1}(\zeta_1, \zeta_2, \zeta_3) \tag{2.58}$$

$$x_2 = \sum_{i=1}^{8} X_{2i}\hat{\phi}_i = Map_{x_2}(\zeta_1, \zeta_2, \zeta_3) \tag{2.59}$$

$$x_3 = \sum_{i=1}^{8} X_{3i}\hat{\phi}_i = Map_{x_3}(\zeta_1, \zeta_2, \zeta_3) \tag{2.60}$$

In 3-D the shape function of the master element take on a different form. First, there are 8 distinct linear shape functions that define the isoparametric space

$$\hat{\phi}_1(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1-\zeta_1)(1-\zeta_2)(1-\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.61}$$

$$\hat{\phi}_2(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1+\zeta_1)(1-\zeta_2)(1-\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.62}$$

$$\hat{\phi}_3(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1+\zeta_1)(1+\zeta_2)(1-\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.63}$$

$$\hat{\phi}_4(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1-\zeta_1)(1+\zeta)(1-\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.64}$$

$$\hat{\phi}_5(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1-\zeta_1)(1-\zeta_2)(1+\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.65}$$

$$\hat{\phi}_6(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1+\zeta_1)(1-\zeta_2)(1+\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.66}$$

$$\hat{\phi}_7(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1+\zeta_1)(1+\zeta_2)(1+\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.67}$$

$$\hat{\phi}_8(\zeta_1, \zeta_2, \zeta_3) = \frac{1}{8}(1-\zeta_1)(1+\zeta_2)(1+\zeta_3) \qquad for\, \zeta_1, \zeta_2, \zeta_3 \in [-1, 1] \tag{2.68}$$

Now we define the the $[F]$ matrix

$$[F] = \begin{bmatrix} \frac{\partial x_1}{\partial \zeta_1} & \frac{\partial x_1}{\partial \zeta_2} & \frac{\partial x_1}{\partial \zeta_3} \\ \frac{\partial x_2}{\partial \zeta_1} & \frac{\partial x_2}{\partial \zeta_2} & \frac{\partial x_3}{\partial \zeta_3} \\ \frac{\partial x_3}{\partial \zeta_1} & \frac{\partial x_3}{\partial \zeta_2} & \frac{\partial x_3}{\partial \zeta_3} \end{bmatrix} \tag{2.69}$$

The Jacobian of the transformation to the isoparametric space is

$$J = det\,|F| = \frac{\partial x_1}{\partial \zeta_1}(\frac{\partial x_2}{\partial \zeta_2}\frac{\partial x_3}{\partial \zeta_3} - \frac{\partial x_3}{\partial \zeta_2}\frac{\partial x_2}{\partial \zeta_3}) - \frac{\partial x_1}{\partial \zeta_2}(\frac{\partial x_2}{\partial \zeta_1}\frac{\partial x_3}{\partial \zeta_3} - \frac{\partial x_2}{\partial \zeta_1}\frac{\partial x_3}{\partial \zeta_3}) + \frac{\partial x_1}{\partial \zeta_3}(\frac{\partial x_2}{\partial \zeta_1}\frac{\partial x_2}{\partial \zeta_2} - \frac{\partial x_3}{\partial \zeta_1}\frac{\partial x_2}{\partial \zeta_2}) \tag{2.70}$$

The differential relationships for the isoparametric mapping are

$$\frac{\partial}{\partial \zeta_1} = \frac{\partial}{\partial x_1}\frac{\partial x_1}{\partial \zeta_1} + \frac{\partial}{\partial x_2}\frac{\partial x_2}{\partial \zeta_1} + \frac{\partial}{\partial x_3}\frac{\partial x_3}{\partial \zeta_1} \tag{2.71}$$

$$\frac{\partial}{\partial \zeta_2} = \frac{\partial}{\partial x_1}\frac{\partial x_1}{\partial \zeta_2} + \frac{\partial}{\partial x_2}\frac{\partial x_2}{\partial \zeta_2} + \frac{\partial}{\partial x_3}\frac{\partial x_3}{\partial \zeta_2} \tag{2.72}$$

$$\frac{\partial}{\partial \zeta_3} = \frac{\partial}{\partial x_1}\frac{\partial x_1}{\partial \zeta_3} + \frac{\partial}{\partial x_2}\frac{\partial x_2}{\partial \zeta_3} + \frac{\partial}{\partial x_3}\frac{\partial x_3}{\partial \zeta_3} \tag{2.73}$$

and the inverse differential relationships are

$$\frac{\partial}{\partial x_1} = \frac{\partial}{\partial \zeta_1}\frac{\partial \zeta_1}{\partial x_1} + \frac{\partial}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial x_1} + \frac{\partial}{\partial \zeta_3}\frac{\partial \zeta_3}{\partial x_1} \tag{2.74}$$

$$\frac{\partial}{\partial x_2} = \frac{\partial}{\partial \zeta_1}\frac{\partial \zeta_1}{\partial x_2} + \frac{\partial}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial x_2} + \frac{\partial}{\partial \zeta_3}\frac{\partial \zeta_3}{\partial x_2} \tag{2.75}$$

$$\frac{\partial}{\partial x_3} = \frac{\partial}{\partial \zeta_1}\frac{\partial \zeta_1}{\partial x_3} + \frac{\partial}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial x_3} + \frac{\partial}{\partial \zeta_3}\frac{\partial \zeta_3}{\partial x_3} \tag{2.76}$$

Therefore

$$\begin{Bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{Bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial \zeta_1} & \frac{\partial x_1}{\partial \zeta_2} & \frac{\partial x_1}{\partial \zeta_3} \\ \frac{\partial x_2}{\partial \zeta_1} & \frac{\partial x_2}{\partial \zeta_2} & \frac{\partial x_2}{\partial \zeta_3} \\ \frac{\partial x_3}{\partial \zeta_1} & \frac{\partial x_3}{\partial \zeta_2} & \frac{\partial x_3}{\partial \zeta_3} \end{bmatrix}}_{F} \begin{Bmatrix} d\zeta_1 \\ d\zeta_2 \\ d\zeta_3 \end{Bmatrix} \tag{2.77}$$

and

$$\begin{Bmatrix} d\zeta_1 \\ d\zeta_2 \\ d\zeta_3 \end{Bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial \zeta_1}{\partial x_1} & \frac{\partial \zeta_1}{\partial x_2} & \frac{\partial \zeta_1}{\partial x_3} \\ \frac{\partial \zeta_2}{\partial x_1} & \frac{\partial \zeta_2}{\partial x_2} & \frac{\partial \zeta_2}{\partial x_3} \\ \frac{\partial \zeta_3}{\partial x_1} & \frac{\partial \zeta_3}{\partial x_2} & \frac{\partial \zeta_3}{\partial x_3} \end{bmatrix}}_{F^{-1}} \begin{Bmatrix} dx_1 \\ dx_2 \\ dx_3 \end{Bmatrix} \tag{2.78}$$

Now the matrices in the weak form need to be expressed in terms of the master domain coordinates $\zeta_1, \zeta_2, \zeta_3$

$$[T(\phi(x_1, x_2, x_3)] = [\hat{T}(\hat{\phi}(Map_{x_1}(\zeta_1, \zeta_2, \zeta_3), Map_{x_2}(\zeta_1, \zeta_2\zeta_3), Map_{x_3}(\zeta_1, \zeta_2, \zeta_3)))] \tag{2.79}$$

Now we define $[\hat{T}]$ to be

$$[\hat{T}] = \begin{bmatrix} \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} \\ 0 \\ 0 \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} \\ 0 \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} \end{bmatrix} \tag{2.80}$$

for the first column

$$[\hat{T}] = \begin{bmatrix} 0 \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} \\ 0 \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} \\ 0 \end{bmatrix} \tag{2.81}$$

for the second column

$$[\hat{T}] = \begin{bmatrix} 0 \\ 0 \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} \\ 0 \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} \\ \frac{\partial}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1} + \frac{\partial}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1} + \frac{\partial}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} \end{bmatrix} \tag{2.82}$$

The elemental shape function matrix $[\hat{\phi}]$ has the following form

$$[\hat{\phi}] = \begin{matrix} \hat{\phi}_1 & \hat{\phi}_2 & \hat{\phi}_3 & \hat{\phi}_4 & \hat{\phi}_5 & \hat{\phi}_6 & \hat{\phi}_7 & \hat{\phi}_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \tag{2.83}$$

for the first block of eight columns

$$[\hat{\phi}] = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{\phi}_1 & \hat{\phi}_2 & \hat{\phi}_3 & \hat{\phi}_4 & \hat{\phi}_5 & \hat{\phi}_6 & \hat{\phi}_7 & \hat{\phi}_8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \tag{2.84}$$

for the second block of eight columns

$$[\hat{\phi}] = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{\phi}_1 & \hat{\phi}_2 & \hat{\phi}_3 & \hat{\phi}_4 & \hat{\phi}_5 & \hat{\phi}_6 & \hat{\phi}_7 & \hat{\phi}_8 \end{matrix} \tag{2.85}$$

for the third block of eight columns, resulting in a 3x24 matrix. The product $[\hat{T}][\hat{\phi}]$ then becomes a 6x24 matrix

$$[\hat{T}][\hat{\phi}] = \begin{bmatrix}
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} & \dots & \dots & \dots & \dots & \dots
\end{bmatrix}$$

for the first block of eight columns

$$[\hat{T}][\hat{\phi}] = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} & \dots & \dots & \dots & \dots & \dots \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

for the second block of eight columns

$$[\hat{T}][\hat{\phi}] = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_3}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_3}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_3} & \dots & \dots & \dots & \dots & \dots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_2}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_2}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_2} & \dots & \dots & \dots & \dots & \dots \\
\frac{\partial\hat{\phi}_1}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1}+\frac{\partial\hat{\phi}_1}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1}+\frac{\partial\hat{\phi}_1}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} & \frac{\partial\hat{\phi}_2}{\partial\zeta_1}\frac{\partial\zeta_1}{\partial x_1}+\frac{\partial\hat{\phi}_2}{\partial\zeta_2}\frac{\partial\zeta_2}{\partial x_1}+\frac{\partial\hat{\phi}_2}{\partial\zeta_3}\frac{\partial\zeta_3}{\partial x_1} & \dots & \dots & \dots & \dots & \dots
\end{bmatrix}$$

2.2.1.2 Applying Gaussian Quadrature

Now that all the matrices are defined, quadrature is applied to evaluate the expressions numerically. This leads to the following expressions

$$S^e_{ij} = \underbrace{\sum_{q=1}^{g}\sum_{r=1}^{g}\sum_{s=1}^{g} w_q w_r w_s ([\hat{T}]\{\hat{\phi}_i\})^T[\hat{\mathbb{E}}]([\hat{T}]\{\phi_j\})\,|F|}_{standard\,[S]\,term} \tag{2.86}$$

for the stiffness matrix, where g is the number of Gauss Points and $|F_s|$ is the Jacobian of the surface integral transformation. Elements that are not subject to the Dirichlet boundary conditions will only have the standard contribution to the stiffness matrix and elements with Dirichlet boundary conditions will be implemented by modifying the corresponding stiffness matrix accordingly. The standard mass matrix terms is

$$M^e_{ij} = \underbrace{\sum_{q=1}^{g}\sum_{r=1}^{g}\sum_{s=1}^{g} w_q w_r w_s (\frac{1}{\Delta t}\{\hat{\phi}_i\}^T \rho \{\hat{\phi}_j\}^T)\,|F|}_{standard\,[M]\,term} \tag{2.87}$$

The load vector then becomes

$$R^e_i = \underbrace{\sum_{q=1}^{g}\sum_{r=1}^{g}\sum_{s=1}^{g} w_q w_r w_s [\hat{\phi}_i]^T \{f\}\,|F|}_{standard\,\{R\}\,term} \tag{2.88}$$

$$+\underbrace{\sum_{q=1}^{g}\sum_{r=1}^{g} w_q w_r [\hat{\phi}_i]^T \{t\}\,|F_s|}_{for\,\Gamma_t \cap \Omega_e \neq 0}$$

Elements that are not subject to the Dirichlet or the traction boundary conditions, meaning elements that are not on the surface of Γ_t or Γ_d, are only subject to the standard contribution to the load vector. Elements that are subject to the traction boundary condition, meaning on the surface defined by Γ_t will have a contribution from the second sum. Elements that are subject to the Dirichlet boundary conditions, meaning on the surface defined by Γ_d will be modified to account for the Dirichlet boundary terms.

The transformation factor for the surface integral can be found using Nanson's formula

$$\mathbf{n} d\mathbf{\Gamma} = J F^{-T} \mathbf{N}\, d\Gamma_\zeta \tag{2.89}$$

which for an expression inside a surface integral yields

$$\mathbf{n} \cdot \mathbf{n}\, d\Gamma = J\sqrt{\mathbf{N} \cdot F^{-1} F^{-T} \cdot \mathbf{N}} d\Gamma_\zeta \tag{2.90}$$

as the transformation factor for surface integrals. The finite element approximation of the weak form is defined in the space of $H_1(\Omega)$.

2.2.1.3 Integrals for Balance of Linear Momentum

Given the procedure outlined, the balance of linear differential equation requires the following integrals:

Integral 1

$$[S_1] = \int_\Omega ([T]\{\phi_A\})^T [\mathcal{D}\,\mathbb{E}_0]([T]\{\phi_B\}) d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} ([T]\{\phi_A\})^T [\mathcal{D}\,\mathbb{E}_0]([T]\{\phi_B\}) d\Omega_e \tag{2.91}$$

$$[S_1]^e_{AB} = \iiint_{-1}^{1} \underbrace{([\hat{T}]\{\hat{\phi}_i\})^T [\mathcal{D}\,\mathbb{E}_0]([\hat{T}]\{\phi_j\}) \, \cdot |F| \; d\zeta_1 \, d\zeta_2 \, d\zeta_3}_{f_1(\zeta_1,\zeta_2,\zeta_3)} \tag{2.92}$$

which can using Gaussian Quadrature becomes

$$[S_1]^e_{AB} = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} \sum_{K=1}^{NP_3} w_I w_J w_K f_1(\zeta_1,\zeta_2,\zeta_3) \tag{2.93}$$

If the material properties change with time or temperature, then this integral has to be computed at each time-step. If they remain constant, then it has to be computed only once.

Integral 2

$$F_\theta = \int_\Omega ([T]\{\phi_A\})^T [\mathcal{D}\,\mathbb{E}_0](\{\boldsymbol{\epsilon_\theta}(t+\Delta t)\}) d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} ([T]\{\phi_A\})^T [\mathcal{D}\,\mathbb{E}_0](\{\boldsymbol{\epsilon_\theta}(t+\Delta t)\}) d\Omega_e \tag{2.94}$$

$$[F_\theta]^e_A = \iiint_{-1}^{1} \underbrace{([\hat{T}]\{\hat{\phi}_i\})^T [\mathcal{D}\,\mathbb{E}_0](\{\boldsymbol{\epsilon_\theta}(t+\Delta t)\}) \, \cdot |F| \; d\zeta_1 \, d\zeta_2 \, d\zeta_3}_{f_2(\zeta_1,\zeta_2,\zeta_3)} \tag{2.95}$$

which can using Gaussian Quadrature becomes

$$[F_\theta]^e_A = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} \sum_{K=1}^{NP_3} w_I w_J w_K f_2(\zeta_1,\zeta_2,\zeta_3) \tag{2.96}$$

If the material properties change with time or temperature, then this integral has to be computed at each time-step. If they remain constant, then it has to be computed only once.

Integral 3

$$[M] = \int_\Omega \phi_A \, \rho \, \phi_B d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} \phi_A \, \rho \, \phi_B d\Omega_e \tag{2.97}$$

$$[M]^e_{AB} = \iiint_{-1}^{1} \underbrace{(\{\hat{\phi}_i\}^T \, \rho \, \{\hat{\phi}_j\}) \, \cdot |F| \; d\zeta_1 \, d\zeta_2 \, d\zeta_3}_{f_4(\zeta_1,\zeta_2,\zeta_3)} \tag{2.98}$$

which can using Gaussian Quadrature becomes

$$[M]^e_{AB} = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} \sum_{K=1}^{NP_3} w_I w_J w_K f_4(\zeta_1,\zeta_2,\zeta_3) \tag{2.99}$$

If the material properties change with time or temperature, then this integral has to be computed at each time-step. If they remain constant, then it has to be computed only once.

Integral 4

$$\{R\}_f = \int_{\Omega} \phi_A \cdot \mathbf{f} \, d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} \phi_A \cdot \mathbf{f} \, d\Omega_e \tag{2.100}$$

$$\{R_f\}^e_A = \iiint_{-1}^{1} \underbrace{(\{\hat{\phi}_i\}^T \, \mathbf{f}) \; \cdot |F| \; d\zeta_1 \, d\zeta_2 \, d\zeta_3}_{f_5(\zeta_1, \zeta_2, \zeta_3)} \tag{2.101}$$

which can using Gaussian Quadrature becomes

$$\{R_f\}^e_A = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} \sum_{K=1}^{NP_3} w_I w_J w_K f_5(\zeta_1, \zeta_2, \zeta_3) \tag{2.102}$$

This integral has to be computed only once.

Integral 5

$$\{R\}_t = \int_{\Omega} \phi_A \cdot \boldsymbol{t} \, d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} \phi_A \cdot \boldsymbol{t} \, d\Omega_e \tag{2.103}$$

$$\{R_t\}^e_A = \iint_{-1}^{1} \underbrace{(\{\hat{\phi}_i\}^T \, \boldsymbol{t}) \; \cdot |F| \; \sqrt{\mathbf{N} \cdot F^{-1} F^{-T} \cdot \mathbf{N}} \; |_{\zeta_i = -1,1} \; d\zeta_2 \, d\zeta_3}_{f_6(a, \zeta_2, \zeta_3)} \tag{2.104}$$

which can using Gaussian Quadrature becomes

$$\{R_t\}^e_A = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} w_I w_J f_6(a, \zeta_2, \zeta_3) \tag{2.105}$$

where a is a constant that represents the surface normal. The surface normal can be placed on any ζ_i depending on the specific boundary conditions. This integral has to be computed only once.

Integral 6

$$\{R\}_t = \int_{\Omega} \phi_A \cdot \boldsymbol{t} \, d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} \phi_A \cdot \boldsymbol{t} \, d\Omega_e \tag{2.106}$$

$$\{R_t\}^e_A = \iint_{-1}^{1} \underbrace{(\{\hat{\phi}_i\}^T \, \boldsymbol{t}) \; \cdot |F| \; \sqrt{\mathbf{N} \cdot F^{-1} F^{-T} \cdot \mathbf{N}} \; |_{\zeta_i = -1,1} \; d\zeta_2 \, d\zeta_3}_{f_6(a, \zeta_2, \zeta_3)} \tag{2.107}$$

which can using Gaussian Quadrature becomes

$$\{R_t\}^e_A = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} w_I w_J f_6(a, \zeta_2, \zeta_3) \tag{2.108}$$

where a is a constant that represents the surface normal. The surface normal can be placed on any ζ_i depending on the specific boundary conditions. This integral has to be computed only once.

2.2.1.4 Applying the Boundary Conditions

Since this problem is a linear problem, one can enforce the boundary conditions by forcing $v = 0 \quad on\, \Gamma_d$, which was applied as follows in the program:

Dirichlet Boundary Conditions

1. Find the rows of $[S]$, $[M]$, corresponding to the boundary nodes using the connectivity table
   (a) Make the rows of the boundary nodes in $[S]$, $[M]$ equal to 0
   (b) Make the diagonal element corresponding to that row equal to 1.
2. Find the rows of $\{R\}$ corresponding to the same boundary nodes using the connectivity table
   (a) Make the rows of the interior nodes in $\{R\}$ equal to $\bar{u}$.

This procedure ensures that each of the interior nodes always has the solution $u = \bar{u}$. Another method of ensuring this is to make the interior nodes of $\{a(t)\}$ equal to $u = \bar{u}$ for each time-step.

Neumann Boundary Conditions

The flux condition was applied by using the surface integral of the flux term and adding it to the load vector.

2.2.2 Thermodynamic Energy Balance

The weak form of the thermodynamic energy balance is derived by integrating the equation over the domain of the problem and multiplying the equation by a test function v:

$$\int_\Omega v\rho_0 C_p\dot{\theta} d\Omega = \int_\Omega v(\nabla_X\cdot(\mathbb{K}\nabla_X\theta)+(\sigma : \dot{\epsilon_\theta})-\frac{1}{2}(\boldsymbol{\epsilon}-\boldsymbol{\epsilon_\theta}) : \mathcal{D}\dot{\mathbb{E}} : (\boldsymbol{\epsilon}-\boldsymbol{\epsilon_\theta})+\rho_0 z)d\Omega \qquad \forall v \in H_{1(\Omega)} \tag{2.109}$$

This equation can be modified using the product rule of differentiation

$$\int_\Omega \nabla_X(v\cdot(\mathbb{K}\nabla_X\theta))\, d\Omega = \int_\Omega v\cdot(\nabla_X(\nabla_X\mathbb{K}\theta))\, d\Omega + \int_\Omega (\nabla_X v)\cdot(\mathbb{K}\nabla_X\theta)\, d\Omega \qquad \forall v \in H_{1(\Omega)} \tag{2.110}$$

which leads to

$$\int_\Omega v\rho_0 C_p\dot{\theta} d\Omega = \int_\Omega \nabla_X(v\cdot(\mathbb{K}\nabla_X\theta)) - \int_\Omega (\nabla_X v)\cdot(\mathbb{K}\nabla_X\theta)\, d\Omega \tag{2.111}$$

$$+\int_\Omega (\sigma : \dot{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} - \frac{1}{2}(\boldsymbol{\epsilon}-\boldsymbol{\epsilon_\theta}) : \mathcal{D}\dot{\mathbb{E}} : (\boldsymbol{\epsilon}-\boldsymbol{\epsilon_\theta}) + \rho_0 z)d\Omega \qquad \forall v \in H_{1(\Omega)} \tag{2.112}$$

next we can use the divergence theorem

$$\int_{\Omega} \nabla u \, d\Omega = \int_{\partial\Omega} u \cdot n \, dS \tag{2.113}$$

which leads to the following weak form with right-hand side:

$$\underbrace{\int_{\Omega} (\nabla_X v(\mathbb{K}\nabla_X \theta))}_{[S]} + \underbrace{\int_{\Omega} v\rho_0 C_p \dot{\theta} d\Omega}_{[M]} = \qquad \forall v \; such \; that \; v = 0 \; on \; \Gamma_d \tag{2.114}$$

$\forall v \; such \; that \; v = 0 \; on \; \Gamma_d \, where \, \Gamma_d \, represents \, surface \, with \, Dirichlet \, Boundary \, Conditions$

and left-hand side:

$$\underbrace{\underbrace{\int_{\Omega} v \cdot \rho_0 z \, d\Omega}_{\{R\}_z} + \underbrace{\int_{\Gamma_q} v \cdot \bar{q}_n d\Gamma}_{\{R\}_q} + \underbrace{\int_{\Omega} v \cdot (\sigma : \dot{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}) \, d\Omega}_{\{R\}_{\dot{\epsilon}}} - \underbrace{\int_{\Omega} v \cdot (\frac{1}{2}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}) : \mathcal{D}\dot{\mathbb{E}} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}})) \, d\Omega}_{\{R\}_{\mathbb{E}}}}_{\{R\}} \tag{2.115}$$

which completes the spatial discretization of the equation. The time discretization follows by applying an Implicit Backward Euler Method.

Starting with the previously derived weak form

$$\underbrace{\int_{\Omega} (\nabla_X v(\mathbb{K}\nabla_X \theta))}_{[S]} + \underbrace{\int_{\Omega} v\rho_0 C_p \dot{\theta} d\Omega}_{[M]} \tag{2.116}$$

on the right-hand side, and on the left-hand side:

$$\underbrace{\underbrace{\int_{\Omega} v \cdot \rho_0 z \, d\Omega}_{\{R\}_z} + \underbrace{\int_{\Gamma_q} v \cdot \bar{q}_n d\Gamma}_{\{R\}_q} + \underbrace{\int_{\Omega} v \cdot (\sigma : \dot{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}) \, d\Omega}_{\{R\}_{\dot{\epsilon}}} - \underbrace{\int_{\Omega} v \cdot (\frac{1}{2}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}) : \mathcal{D}\dot{\mathbb{E}} : (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}})) \, d\Omega}_{\{R\}_{\mathbb{E}}}}_{\{R\}} \tag{2.117}$$

The derivative is approximated using the following scheme

$$\frac{\partial\theta}{\partial t}\bigg|_{t+\Delta t} \approx \frac{\theta(t+\Delta t) - \theta(t)}{\Delta t} \tag{2.118}$$

which leads to all temperature terms being evaluated at time $t = t + \Delta t$. The weak form now becomes

$$\int_{\Omega} (\nabla_X v \cdot (\nabla_X \mathbb{K}\theta(t+\Delta t))) + \int_{\Omega} v\rho_0 C_p \frac{\theta(t+\Delta t) - \theta(t)}{\Delta t} d\Omega \tag{2.119}$$

on the left-hand side and

$$\int_\Omega zv d\Omega + \int_{\Gamma_q} v\bar{q}_n d\Gamma + \int_\Omega v \cdot (\sigma : \dot{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t+\Delta t))\, d\Omega \tag{2.120}$$

$$+ \int_\Omega v \cdot (\frac{1}{2}(\boldsymbol{\epsilon}(t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(t+\Delta t)) : \mathcal{D}\dot{\mathbb{E}}(t) : (\boldsymbol{\epsilon}(t+\Delta t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(t+\Delta t)))\, d\Omega \tag{2.121}$$

on the right-hand side. This yields

$$\underbrace{\int_\Omega (\nabla_X v \cdot (\nabla_X \mathbb{K}\theta(t+\Delta t)))}_{[S]\{a(t)\}} + \underbrace{\int_\Omega \frac{1}{\Delta t} v\rho_0 C_p \theta(t+\Delta t) d\Omega}_{[M]\{a(t+\Delta t)\}} - \underbrace{\int_\Omega \frac{1}{\Delta t} v\rho_0 C_p \theta(t) d\Omega}_{[M]\{a(t)\}} \tag{2.122}$$

on the left-hand side, and

$$\underbrace{\int_\Omega zv d\Omega}_{\{R_z\}} + \underbrace{\int_{\Gamma_q} v\bar{q}_n d\Gamma}_{\{R_q\}} + \underbrace{\int_\Omega v \cdot (\sigma : \dot{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t+\Delta t))\, d\Omega}_{\{R\}_{\dot{\epsilon}}} \tag{2.123}$$

$$-\underbrace{\int_\Omega v \cdot (\frac{1}{2}(\boldsymbol{\epsilon}(t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(t+\Delta t)) : \mathcal{D}\dot{\mathbb{E}}(t) : (\boldsymbol{\epsilon}(t+\Delta t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(t+\Delta t)))\, d\Omega}_{\{R\}_{\mathbb{E}}} \tag{2.124}$$

on the right-hand side.

Applying the following finite element discretization scheme with linear shape functions,

$$\theta \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \sum_{A=1}^{N} a_A \phi_A \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad v \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \sum_{A=1}^{N} b_A \phi_A \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{2.125}$$

then yields the following discretized weak form equation. For the left-hand side:

$$\int_\Omega (\nabla_X \sum_{A=1}^{N} b_A \phi_A (\nabla_X \mathbb{K} \sum_{B=1}^{N} a_B(t+\Delta t)\phi_B)) + \int_\Omega \frac{1}{\Delta t} \sum_{A=1}^{N} b_A \phi_A \rho_0 C_p \sum_{B=1}^{N} a_B(t+\Delta t)\phi_B d\Omega$$
$$- \int_\Omega \frac{1}{\Delta t} \sum_{A=1}^{N} b_A \phi_A \rho_0 C_p \sum_{B=1}^{N} a_B \phi_B d\Omega \tag{2.126}$$

and the right-hand-side

$$\int_\Omega \sum_{A=1}^{N} b_A \phi_A\, z d\Omega + \int_{\Gamma_q} \sum_{A=1}^{N} b_A \phi_A\, \bar{q}_n d\Gamma + \int_\Omega \sum_{A=1}^{N} b_A \phi_A\, (\sigma : \dot{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(t+\Delta t))\, d\Omega \tag{2.127}$$

$$+ \int_\Omega \sum_{A=1}^{N} b_A \phi_A\, (\frac{1}{2}(\boldsymbol{\epsilon}(t+\Delta t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(t+\Delta t)) : \mathcal{D}\dot{\mathbb{E}}(t) : (\boldsymbol{\epsilon}(t+\Delta t) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(t+\Delta t)))\, d\Omega$$

The b_A's are arbitrary and can be canceled out of the equation leading to the following left-hand side

$$\sum_{A=1}^{N}\sum_{B=1}^{N}\underbrace{(\int_{\Omega}(\nabla_X\phi_A(\nabla_X\mathbb{K}\,\phi_B))d\Omega}_{[\mathbf{S}]}\underbrace{a_B(t+\Delta t)}_{\{a(t+\Delta t)\}}+\underbrace{\frac{1}{\Delta t}(\int_{\Omega}(\phi_A(\rho_0C_p\,\phi_B))d\Omega}_{[\mathbf{M}]}\underbrace{a_B(t+\Delta t)}_{\{a(t+\Delta t)\}}$$
$$\underbrace{-\frac{1}{\Delta t}(\int_{\Omega}(\phi_A(\rho_0C_p\,\phi_B))d\Omega}_{[\mathbf{M}]}\underbrace{a_B(t)}_{\{a(t)\}} \tag{2.128}$$

with the previously defined vectors $\{R_z+R_q+R_{\dot{\epsilon}}(t+\Delta t)+R_{\mathbb{E}}(t+\Delta t)\}$ on the right hand side.

The weak then becomes the following matrix equation

$$([S]+\frac{1}{\Delta t}[M])\{a(t+\Delta t)\}-\frac{1}{\Delta t}[M]\{a(t)\}=\{R_z+R_q+R_{\dot{\epsilon}}(t+\Delta t)+R_{\mathbb{E}}(t+\Delta t)\} \tag{2.129}$$

$$\{a(t+\Delta t)\}=([S]+\frac{1}{\Delta t}[M])^{-1}\cdot(\{R_z+R_q+R_{\dot{\epsilon}}(t+\Delta t)+R_{\mathbb{E}}(t+\Delta t)\}+\frac{1}{\Delta t}[M]\{a(t)\} \tag{2.130}$$

where $\{a(t)\}$ is given as an initial condition of the problem. The equation can then be solved using the following process:

1. Solve the above equation for $\{a(t+\Delta t)\}^K$ by guessing values for $R_{\dot{\epsilon}}(t+\Delta t)^K$ and $R_{\mathbb{E}}(t+\Delta t)^K$

2. Compute $R_{\dot{\epsilon}}(t+\Delta t)^{K+1}$ and $R_{\mathbb{E}}(t+\Delta t)^{K+1}$ using $\{a(t+\Delta t)\}^K$

3. Compute $\{a(t+\Delta t)\}^{K+1}$ using $R_{\dot{\epsilon}}(t+\Delta t)^{K+1}$ and $R_{\mathbb{E}}(t+\Delta t)^{K+1}$

4. Compute an error estimate by inserting the computed solutions into the original equations using

$$error=\frac{\left\|\{a(t+\Delta t)\}^{K+1}-\{a(t+\Delta t)\}^K\right\|}{\|\{a(t+\Delta t)\}^{K+1}\|}\leqq tol \tag{2.131}$$

5. Repeat steps 1-4 if $error \geqq tol$

The thermodynamic energy balance weak form therefore contains the following integrals:

Integral 1

$$[S]=\int_{\Omega}((\nabla_X\phi_B\cdot(\nabla_X\mathbb{K}\,\phi_A)))\,d\Omega=\sum_{e=1}^{N_e}\int_{\Omega_e}((\nabla_X\phi_B\cdot(\nabla_X\mathbb{K}\,\phi_A)))\,d\Omega_e \tag{2.132}$$

$$[S]^e_{AB} = \iiint_{-1}^{1} \underbrace{\left([F^{-T}] \begin{Bmatrix} \frac{\partial \hat{\phi}_A}{\partial \zeta_1} \\ \frac{\partial \hat{\phi}_A}{\partial \zeta_2} \\ \frac{\partial \hat{\phi}_A}{\partial \zeta_3} \end{Bmatrix} \mathbb{K}\, [F^{-T}] \begin{Bmatrix} \frac{\partial \hat{\phi}_A}{\partial \zeta_1} \\ \frac{\partial \hat{\phi}_A}{\partial \zeta_2} \\ \frac{\partial \hat{\phi}_A}{\partial \zeta_3} \end{Bmatrix}\right) \cdot J\, d\zeta_1\, d\zeta_2\, d\zeta_3}_{f_1(\zeta_1, \zeta_2 \zeta_3)} \tag{2.133}$$

which can using Gaussian Quadrature becomes

$$[S]^e_{AB} = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} \sum_{K=1}^{NP_3} w_I w_J w_K f_1(\zeta_1, \zeta_2 \zeta_3) \tag{2.134}$$

If the material properties change with time or temperature, then this integral has to be computed at each time-step. If they remain constant, then it has to be computed only once.

Integral 2

$$[M] = \frac{1}{\Delta t} \int_{\Omega} (\phi_A \cdot (\rho_0 C_p) \cdot \phi_B)\, d\Omega = \frac{1}{\Delta t} \sum_{e=1}^{N_e} \int_{\Omega_e} (\phi_A \cdot (\rho_0 C_p) \cdot \phi_B)\, d\Omega_e \tag{2.135}$$

$$[M]^e_{AB} = \iiint_{-1}^{1} \underbrace{(\phi_A \cdot (\rho_0 c_p) \cdot \phi_B) \cdot J\, d\zeta_1\, d\zeta_2\, d\zeta_3}_{f_2(\zeta_1, \zeta_2 \zeta_3)} \tag{2.136}$$

which can using Gaussian Quadrature becomes

$$[M]^e_{AB} = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} \sum_{K=1}^{NP_3} w_I w_J w_K f_3(\zeta_1, \zeta_2 \zeta_3) \tag{2.137}$$

If the material properties change with time or temperature, then this integral has to be computed at each time-step. If they remain constant, then it has to be computed only once.

Integral 3

$$\{R\}_z = \int_{\Omega} (\phi_A\, \rho_0 z)\, d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} (\phi_A\, \rho_0 z)\, d\Omega_e \tag{2.138}$$

$$\{R\}^{z,e}_{AB} = \iiint_{-1}^{1} \underbrace{\hat{\phi}_A\, \rho_0 z) \cdot J\, d\zeta_1\, d\zeta_2\, d\zeta_3}_{f_3(\zeta_1, \zeta_2 \zeta_3)} \tag{2.139}$$

which can using Gaussian Quadrature becomes

$$\{R\}^e_z = \sum_{I=1}^{NP_1} \sum_{J=1}^{NP_2} \sum_{K=1}^{NP_3} w_I w_J w_K f_3(\zeta_1, \zeta_2 \zeta_3) \tag{2.140}$$

If the material properties change with time or temperature, then this integral has to be computed at each time-step. If they remain constant, then it has to be computed only once.

Integral 4

$$\{R\}_q = \int_\Omega (\phi_A\, \bar{q}_n)\, d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} (\phi_A\, \bar{q}_n)\, d\Omega_e \tag{2.141}$$

$$\{R\}^{q,e}_{AB} = \iint_{-1}^{1} \underbrace{\hat{\phi}_A\, \bar{q}_n \;\cdot\; J\sqrt{\mathbf{N}\cdot F^{-1}F^{-T}\cdot \mathbf{N}}\;|_{\zeta_i=-1,1}\;\, d\zeta_2\, d\zeta_3}_{f_4(a,\,\zeta_2\zeta_3)} \tag{2.142}$$

which can using Gaussian Quadrature becomes

$$\{R\}^e_q = \sum_{I=1}^{NP_1}\sum_{J=1}^{NP_2} w_I w_J f_4(a, \zeta_2\zeta_3) \tag{2.143}$$

where a is a constant that represents the surface normal. The surface normal can be placed on any ζ_i depending on the specific boundary conditions. This integral has to be computed only once.

Integral 5

$$\{R\}_{\dot{\epsilon}} = \int_\Omega (\phi_A\, (\sigma : (\dot{\epsilon_\theta})))\, d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} (\phi_A\, (\sigma : (\dot{\epsilon_\theta})))\, d\Omega_e \tag{2.144}$$

$$\{R\}^{\dot{\epsilon},e}_{AB} = \iiint_{-1}^{1} \underbrace{(\hat{\phi}_A\, (\sigma : (\dot{\epsilon_\theta}))) \;\cdot\; J\, d\zeta_1\, d\zeta_2\, d\zeta_3}_{f_5(\zeta_1,\,\zeta_2\zeta_3)} \tag{2.145}$$

which can using Gaussian Quadrature becomes

$$\{R\}^e_{\dot{\epsilon}} = \sum_{I=1}^{NP_1}\sum_{J=1}^{NP_2}\sum_{K=1}^{NP_3} w_I w_J w_K f_5(\zeta_1, \zeta_2\zeta_3) \tag{2.146}$$

If the stress states changes with time or temperature, then this integral has to be computed at each time-step. If the stress state remains constant, then it has to be computed only once.

Integral 6

$$\{R\}_{\mathbb{E}} = \int_\Omega (\phi_A\, (\frac{1}{2}(\epsilon - \epsilon_\theta) : \dot{\mathbb{E}} : (\epsilon - \epsilon_\theta)))\, d\Omega = \sum_{e=1}^{N_e} \int_{\Omega_e} (\phi_A\, (\frac{1}{2}(\epsilon - \epsilon_\theta) : \dot{\mathbb{E}} : (\epsilon - \epsilon_\theta)))\, d\Omega_e \tag{2.147}$$

$$\{R\}^{\mathbb{E},e}_{AB} = \iiint_{-1}^{1} \underbrace{(\hat{\phi}_A\, (\frac{1}{2}(\epsilon - \epsilon_\theta) : \dot{\mathbb{E}} : (\epsilon - \epsilon_\theta))) \;\cdot\; J\, d\zeta_1\, d\zeta_2\, d\zeta_3}_{f_6(\zeta_1,\,\zeta_2\zeta_3)} \tag{2.148}$$

which can using Gaussian Quadrature becomes

$$\{R\}^e_{\mathbb{E}} = \sum_{I=1}^{NP_1}\sum_{J=1}^{NP_2}\sum_{K=1}^{NP_3} w_I w_J w_K f_6(\zeta_1, \zeta_2\zeta_3) \tag{2.149}$$

If the material properties change with time or temperature, then this integral has to be computed at each time-step. If they remain constant, then the integral becomes 0.

2.2.2.1 Applying the Boundary Conditions

Since this problem is a linear problem, one can enforce the boundary conditions by forcing $v = 0 \quad on\, \Gamma_d$, which was applied as follows in the program:

Dirichlet Boundary Conditions

1. Find the rows of $[S]$, $[M]$, corresponding to the boundary nodes using the connectivity table
   (a) Make the rows of the boundary nodes in $[S]$, $[M]$ equal to 0
   (b) Make the diagonal element corresponding to that row equal to 1.
2. Find the rows of $\{R\}$ corresponding to the same boundary nodes using the connectivity table
   (a) Make the rows of the interior nodes in $\{R\}$ equal to $\bar{\theta}$.

This procedure ensures that each of the interior nodes always has the solution $\theta = \bar{\theta}$. Another method of ensuring this is to make the interior nodes of $\{a(t)\}$ equal to $\theta = \bar{\theta}$ for each time-step.

Neumann Boundary Conditions

The flux condition was applied by using the surface integral of the flux term and adding it to the load vector.

2.3 Calculating Material Properties using the Finite Element Method

The finite element method can to calculate the effective properties of a composite material by applying the following procedure:

1. Construct the desired material microstructure in the RVE domain used for the finite element simulation
2. Apply a load state as a Dirichlet boundary condition to the RVE and the solve the time-independent finite element problem for the entire body
3. Set up a system of equations using the known load state and the finite element solution to solve for the desired material property.

2.3.1 Elasticity Tensor

To obtain the elasticity using the above method, six different load states have to be applied

$$\mathcal{E} = \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta \end{bmatrix}, \begin{bmatrix} 0 & \beta & 0 \\ \beta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \beta \\ 0 & \beta & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \beta \\ 0 & 0 & 0 \\ \beta & 0 & 0 \end{bmatrix} \tag{2.150}$$

where the load state is applied as a strain to the boundary of the RVE using displacement boundary conditions for better stability

$$\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \tag{2.151}$$

to yield Dirichlet Boundary Conditions on a specified boundary surface. Following post-processing of the FEM solution, each of the load states will set up the following system of equations

$$\begin{Bmatrix} \langle\sigma_{11}\rangle_\Omega \\ \langle\sigma_{22}\rangle_\Omega \\ \langle\sigma_{33}\rangle_\Omega \\ \langle\sigma_{12}\rangle_\Omega \\ \langle\sigma_{23}\rangle_\Omega \\ \langle\sigma_{13}\rangle_\Omega \end{Bmatrix} = \begin{bmatrix} E^*_{1111} & E^*_{1122} & E^*_{1133} & E^*_{1112} & E^*_{1123} & E^*_{1113} \\ E^*_{2211} & E^*_{2222} & E^*_{2233} & E^*_{2212} & E^*_{2223} & E^*_{2213} \\ E^*_{3311} & E^*_{3322} & E^*_{3333} & E^*_{3312} & E^*_{3323} & E^*_{3313} \\ E^*_{1211} & E^*_{1222} & E^*_{1233} & E^*_{1212} & E^*_{1223} & E^*_{1213} \\ E^*_{2311} & E^*_{2322} & E^*_{2333} & E^*_{2312} & E^*_{2323} & E^*_{2313} \\ E^*_{1311} & E^*_{1322} & E^*_{1333} & E^*_{1312} & E^*_{1323} & E^*_{1313} \end{bmatrix} \begin{Bmatrix} \langle\epsilon_{11}\rangle_\Omega \\ \langle\epsilon_{22}\rangle_\Omega \\ \langle\epsilon_{33}\rangle_\Omega \\ 2\langle\epsilon_{12}\rangle_\Omega \\ 2\langle\epsilon_{23}\rangle_\Omega \\ 2\langle\epsilon_{13}\rangle_\Omega \end{Bmatrix} \tag{2.152}$$

where the left-hand side of the equation is the imposed load state and the right-hand side of the equation can be obtained using the average strain theorem $\langle\epsilon\rangle = \frac{1}{2|\Omega|}\int_\Omega (\nabla u + (\nabla u)^T) d\Omega$.

This will set up a system of 6 equations for each load state, leading to 36 total equations for the 36 unknowns in the elasticity tensor, which can be rearranged to the following matrix system in the form $\mathbf{A}x = b$. [1]

$$\begin{bmatrix} \langle\epsilon_{11}\rangle^{11}_\Omega & \langle\epsilon_{22}\rangle^{11}_\Omega & \langle\epsilon_{33}\rangle^{11}_\Omega & \langle\epsilon_{12}\rangle^{11}_\Omega & \langle\epsilon_{23}\rangle^{11}_\Omega & \langle\epsilon_{13}\rangle^{11}_\Omega & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \langle\epsilon_{11}\rangle^{11}_\Omega & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ \langle\epsilon_{11}\rangle^{22}_\Omega & \langle\epsilon_{22}\rangle^{22}_\Omega & \langle\epsilon_{33}\rangle^{22}_\Omega & \langle\epsilon_{12}\rangle^{22}_\Omega & \langle\epsilon_{23}\rangle^{22}_\Omega & \langle\epsilon_{13}\rangle^{22}_\Omega & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \begin{Bmatrix} E^*_{1111} \\ E^*_{1122} \\ E^*_{1133} \\ E^*_{1112} \\ \vdots \\ E^*_{1313} \end{Bmatrix} = \begin{Bmatrix} \langle\sigma_{11}\rangle^{11}_\Omega \\ \langle\sigma_{22}\rangle^{11}_\Omega \\ \langle\sigma_{33}\rangle^{11}_\Omega \\ \langle\sigma_{12}\rangle^{11}_\Omega \\ \vdots \\ \langle\sigma_{13}\rangle^{13}_\Omega \end{Bmatrix} \tag{2.153}$$

[1] Appendix A A.2

2.3.2 Thermal Conductivity Matrix

The thermal conductivity matrix can be obtained in a similar fashion. In this case, three distinct load states with the spatial temperature derivative are imposed:

$$\mathcal{Q} = \left\{ \begin{array}{ccc} \nabla_x T & 0 & 0 \end{array} \right\}, \left\{ \begin{array}{ccc} 0 & \nabla_x T & 0 \end{array} \right\}, \left\{ \begin{array}{ccc} 0 & 0 & \nabla_x T \end{array} \right\} \tag{2.154}$$

which can be applied to the boundary using displacement boundary conditions for better stability as follows

$$T = \left\{ \begin{array}{ccc} \nabla_x T & 0 & 0 \end{array} \right\} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} \tag{2.155}$$

leading to the following system of equation

$$\begin{Bmatrix} \langle q_1 \rangle_\Omega \\ \langle q_2 \rangle_\Omega \\ \langle q_3 \rangle_\Omega \end{Bmatrix} = \begin{Bmatrix} k_{11} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_3 \end{Bmatrix} \begin{Bmatrix} \frac{\langle \partial T_\Omega \rangle}{\partial x_1} \\ \frac{\langle \partial T_\Omega \rangle}{\partial x_2} \\ \frac{\langle \partial T_\Omega \rangle}{\partial x_3} \end{Bmatrix} \tag{2.156}$$

where the left hand side is the imposed load state and the right-hand side can be computed using the finite element solution for the derivate the temperature $\frac{\langle \partial T_{11\Omega} \rangle}{\partial x} = \left\langle \frac{\partial \phi}{\partial x} T \right\rangle_\Omega$. This yields a system of 9 equations for the conductivity matrix that looks as follows, which can be rearranged to the following system of matrix equations in the form of $\mathbf{A}x = b$.

$$\begin{bmatrix}
\frac{\partial \langle T \rangle_\Omega^1}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^1}{\partial x_3} \\
\frac{\partial \langle T \rangle_\Omega^2}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^2}{\partial x_3} \\
\frac{\partial \langle T \rangle_\Omega^3}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_1} & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_2} & \frac{\partial \langle T \rangle_\Omega^3}{\partial x_3}
\end{bmatrix}
\begin{Bmatrix} k_{11} \\ k_{12} \\ k_{13} \\ k_{21} \\ k_{22} \\ k_{23} \\ k_{31} \\ k_{32} \\ k_{33} \end{Bmatrix}
=
\begin{Bmatrix} \langle q_1 \rangle_\Omega^1 \\ \langle q_2 \rangle_\Omega^1 \\ \langle q_3 \rangle_\Omega^1 \\ \langle q_1 \rangle_\Omega^2 \\ \langle q_2 \rangle_\Omega^2 \\ \langle q_3 \rangle_\Omega^2 \\ \langle q_1 \rangle_\Omega^3 \\ \langle q_2 \rangle_\Omega^3 \\ \langle q_3 \rangle_\Omega^3 \end{Bmatrix} \tag{2.157}$$

2.4 Genetic Algorithm Optimization of Isotropic Microstructures

Genetic algorithms are a family of optimization procedures which are used to search for global minima in non-convex problems. The majority of genetic algorithms employ probabilistic search methods using random number distributions to sample a wide space of genetic strings. Genetic algorithms are generally more robust than classical gradient methods due to the inability of classical gradient methods to differentiate between a local minimum and a global minimum. The fundamental principles of genetic algorithms are based on the principles of natural selection, including species evolution, reproduction, mutation and crossover. These natural selection concepts can be applied to microstructural systems by converting the material properties of interest into sets of genetic strings used in the algorithm. [67] Following the initialization of the genetic strings, the evolutionary concepts of the particular genetic algorithm are applied to find a set of favorable candidates for the desired optimization problem.

In this thesis, a genetic algorithm framework was applied to create a computer generated CMC. In the first tier of the automated design process, the constituent materials of CMC, meaning the matrix and fiber materials, are assumed to be isotropic. This assumption allows the simplification of the elasticity matrix and the thermal conductivity matrix as follows:

- Since the optimization involves isotropic materials, the thermal conductivity tensor $\mathbb{K}$ can be reduced to a single representative value of λ

  The conductivity matrix can be written as

$$\mathbb{K} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{2.158}$$

  which in the isotropic case reduces to

$$\mathbb{K} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{bmatrix} = \lambda \mathbb{I} \tag{2.159}$$

- The elasticity tensor $\mathbb{E}$, for which a high value is desired to ensure high material strength

  - Since the optimization involves isotropic materials, the elasticity tensor can be reduced to the bulk modulus κ and the shear modulus μ

  The constitutive law for the stress can be written more compactly using Voigt notation

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{Bmatrix} = \begin{bmatrix} \mathbb{E}_{11} & \mathbb{E}_{12} & \mathbb{E}_{13} & \mathbb{E}_{14} & \mathbb{E}_{15} & \mathbb{E}_{16} \\ \mathbb{E}_{21} & \mathbb{E}_{22} & \mathbb{E}_{23} & \mathbb{E}_{24} & \mathbb{E}_{25} & \mathbb{E}_{26} \\ \mathbb{E}_{31} & \mathbb{E}_{32} & \mathbb{E}_{33} & \mathbb{E}_{34} & \mathbb{E}_{35} & \mathbb{E}_{36} \\ \mathbb{E}_{41} & \mathbb{E}_{42} & \mathbb{E}_{43} & \mathbb{E}_{44} & \mathbb{E}_{45} & \mathbb{E}_{46} \\ \mathbb{E}_{51} & \mathbb{E}_{52} & \mathbb{E}_{53} & \mathbb{E}_{54} & \mathbb{E}_{55} & \mathbb{E}_{56} \\ \mathbb{E}_{61} & \mathbb{E}_{62} & \mathbb{E}_{63} & \mathbb{E}_{64} & \mathbb{E}_{65} & \mathbb{E}_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{12} \\ 2\epsilon_{23} \\ 2\epsilon_{31} \end{Bmatrix} \tag{2.160}$$

which can be rewritten in terms of the effective bulk modulus $\kappa*$ and the effective shear modulus $\mu*$ for isotropic materials

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{23} \\ \sigma_{31} \end{Bmatrix} = \begin{bmatrix} \kappa^* + \frac{4}{3}\mu^* & \kappa^* - \frac{2}{3}\mu^* & \kappa^* - \frac{2}{3}\mu^* & 0 & 0 & 0 \\ \kappa^* - \frac{2}{3}\mu^* & \kappa^* + \frac{4}{3}\mu^* & \kappa^* - \frac{2}{3}\mu^* & 0 & 0 & 0 \\ \kappa^* - \frac{2}{3}\mu^* & \kappa^* - \frac{2}{3}\mu^* & \kappa^* + \frac{4}{3}\mu^* & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu^* & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu^* & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu^* \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{12} \\ 2\epsilon_{23} \\ 2\epsilon_{31} \end{Bmatrix} \tag{2.161}$$

The genetic algorithm optimization for CMC design requires a procedure by which to compute the effective property of the composite material can be computed from the property of the two constituent materials, the matrix and the dopant material. In the case of isotropic materials doped with spherical particles, the Hashin & Shtrikman (1983) bounds are among the tightest known bounds for the mechanical and thermal properties. The Hashin & Shtrikman bounds for the properties of interest are defined as follows:

- For the bulk modulus κ, the lower and upper bounds, $\kappa^*_{(-)}(\kappa_1, \kappa_2, \mu_1, \mu_2, v_2)$ and $\kappa^*_{(+)}(\kappa_1, \kappa_2, \mu_1, \mu_2, v_2)$ are respectively

$$\kappa^*_{(-)} = \kappa_1 + v_2(\frac{1}{\kappa_2 - \kappa_1} + \frac{3(1 - v_2)}{3\kappa_1 + 4\mu_1})^{-1} \tag{2.162}$$

$$\kappa^*_{(+)} = \kappa_2 + (1 - v_2)(\frac{1}{\kappa_1 - \kappa_2} + \frac{3v_2}{3\kappa_2 + 4\mu_2})^{-1} \tag{2.163}$$

- The shear modulus μ, $\mu^*_{(-)}(\kappa_1, \kappa_2, \mu_1, \mu_2, v_2)$ and $\mu^*_{(+)}(\kappa_1, \kappa_2, \mu_1, \mu_2, v_2)$, are respectively

$$\mu^*_{(-)} = \mu_1 + v_2(\frac{1}{\mu_2 - \mu_1} + \frac{6(1 - v_2)(\kappa_1 + 2\mu_1)}{5\mu_1(3\kappa_1 + 4\mu_1)})^{-1} \tag{2.164}$$

$$\mu^*_{(+)} = \mu_2 + (1 - v_2)(\frac{1}{\mu_1 - \mu_2} + \frac{6v_2(\kappa_2 + 2\mu_2)}{5\mu_2(3\kappa_2 + 4\mu_2)})^{-1} \tag{2.165}$$

- For the thermal conductivity λ, $\lambda^*_{(-)}(\lambda_1, \lambda_2, v_2)$ and $\lambda^*_{(+)}(\lambda_1, \lambda_2, v_2)$, are respectively

$$\lambda^*_{(-)} = \lambda_1 + \frac{v_2}{\frac{1}{\lambda_2 - \lambda_1} + \frac{1 - v_2}{\lambda_1}} \tag{2.166}$$

$$\lambda^*_{(+)} = \lambda_2 + \frac{1 - v_2}{\frac{1}{\lambda_1 - \lambda_2} + \frac{v_2}{\lambda_2}} \tag{2.167}$$

- The effective density ρ^* can be obtained by dividing the collective mass of the RVE by the collective volume of the RVE

$$\rho^* = \frac{m_{total}}{V_{total}} = \frac{(\rho_{matrix} V_{total}) - (\rho_{matrix} V_{dopant}) + (\rho_{dopant} V_{dopant})}{V_{total}} \tag{2.168}$$

the above shows that if $\rho_{matrix} > \rho_{dopant}$ the material density will decrease, which will result in effective down-weighting.

When constructing a composite material, the different phases in the composite can cause stress fields in the microstructure of the material. Ideally, the microscopic stress fields inside the composite should be smooth to achieve better performance under external stresses. One approach to characterize this internal stress field behavior is to use concentration tensors. Concentration tensors provide a measure of the deviation of the stress fields from the mean stress field inside the material. The mean strain and stress inside a composite material can be expressed as[67]

$$\langle \boldsymbol{\epsilon} \rangle_{\Omega} = \frac{1}{|\Omega|}(\int_{\Omega_1} \boldsymbol{\epsilon}\, d\Omega + \int_{\Omega_2} \boldsymbol{\epsilon}\, d\Omega) = v_1 \langle \boldsymbol{\epsilon} \rangle_{\Omega_1} + v_2 \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \tag{2.169}$$

$$\langle \boldsymbol{\sigma} \rangle_{\Omega} = \frac{1}{|\Omega|}(\int_{\Omega_1} \boldsymbol{\sigma}\, d\Omega + \int_{\Omega_2} \boldsymbol{\sigma}\, d\Omega) = v_1 \langle \boldsymbol{\sigma} \rangle_{\Omega_1} + v_2 \langle \boldsymbol{\sigma} \rangle_{\Omega_2} \tag{2.170}$$

which can be rewritten to

$$\langle \boldsymbol{\sigma} \rangle_{\Omega} = v_1 \langle \boldsymbol{\sigma} \rangle_{\Omega_1} + v_2 \langle \boldsymbol{\sigma} \rangle_{\Omega_2} = v_1 \mathbb{E}_1 : \langle \boldsymbol{\epsilon} \rangle_{\Omega_1} + v_2 \mathbb{E}_2 : \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \tag{2.171}$$

$$\langle \boldsymbol{\sigma} \rangle_{\Omega} = \mathbb{E}_1 : (\langle \boldsymbol{\epsilon} \rangle_{\Omega} - v_2 \langle \boldsymbol{\epsilon} \rangle_{\Omega_2}) + v_2 \mathbb{E}_2 : \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \tag{2.172}$$

$$\langle \boldsymbol{\sigma} \rangle_{\Omega} = ((\mathbb{E}_1 + v_2(\mathbb{E}_2 - \mathbb{E}_1)) : \mathbf{C}) : \langle \boldsymbol{\epsilon} \rangle_{\Omega} \tag{2.173}$$

where

$$\underbrace{(\frac{1}{v_2}(\mathbb{E}_2 - \mathbb{E}_1)^{-1} : (\mathbb{E}^* - \mathbb{E}_1))}_{\equiv \mathbf{C}} : \langle \boldsymbol{\epsilon} \rangle_{\Omega} = \langle \boldsymbol{\epsilon} \rangle_{\Omega_2} \tag{2.174}$$

The variation in the stress then becomes

$$\mathbf{C} : \mathbb{E}^{*-1} : \langle \boldsymbol{\sigma} \rangle_{\Omega} = \mathbb{E}_2^{-1} : \langle \boldsymbol{\sigma} \rangle_{\Omega_2} \tag{2.175}$$

which is equal to

$$\mathbb{E}_2 : \mathbf{C} : \mathbb{E}^{*-1} : \langle \boldsymbol{\sigma} \rangle_{\Omega} \equiv \bar{\mathbf{C}} : \langle \boldsymbol{\sigma} \rangle_{\Omega} = \langle \boldsymbol{\sigma} \rangle_{\Omega_2} \tag{2.176}$$

where $\bar{\mathbf{C}}$ is a stress concentration tensor. Given the relationship between $\bar{\mathbf{C}}$ and $\mathbb{E}^*$, one can be determined when the other is known. In the case of isotropy the relationship is as follows

$$\bar{C}_{\kappa} \equiv \frac{1}{v_2} \frac{\kappa_2}{\kappa^*} \frac{\kappa^* - \kappa_1}{\kappa_2 - \kappa_1} \quad and \quad \bar{C}_{\mu} \equiv \frac{1}{v_2} \frac{\mu_2}{\mu^*} \frac{\mu^* - \mu_1}{\mu_2 - \mu_1} \tag{2.177}$$

with $\bar{C}_{\kappa} \left\langle \frac{1}{3} tr\boldsymbol{\sigma} \right\rangle_{\Omega} = \frac{1}{3} \langle tr\boldsymbol{\sigma} \rangle_{\Omega_2}$ and $\bar{C}_{\mu} \langle \boldsymbol{\sigma'} \rangle_{\Omega} = \langle \boldsymbol{\sigma'} \rangle_{\Omega_2}$. The microstress fields are smoothest, with minimal distortion, when $\bar{C}_{\kappa} = \bar{C}_{\mu} = 1$. Since

$$\langle \boldsymbol{\sigma} \rangle_{\Omega_1} = \frac{\langle \boldsymbol{\sigma} \rangle_{\Omega} - v_2 \langle \boldsymbol{\sigma} \rangle_{\Omega_2}}{v_1} \tag{2.178}$$

therefore

$$\langle\boldsymbol{\sigma}\rangle_{\Omega_1} = \frac{\langle\boldsymbol{\sigma}\rangle_{\Omega} - v_2\bar{\mathbf{C}} : \langle\boldsymbol{\sigma}\rangle_{\Omega}}{v_1} = \frac{(1 - v_2\bar{\mathbf{C}}) : \langle\boldsymbol{\sigma}\rangle_{\Omega}}{v_1} = \bar{\bar{\mathbf{C}}} : \langle\boldsymbol{\sigma}\rangle_{\Omega} \tag{2.179}$$

and in the case of isotropy

$$\bar{\bar{C}}_{\kappa} \equiv \frac{1}{v_1}(1 - v_2\bar{C}_{\kappa}) \quad and \quad \bar{\bar{C}}_{\mu} \equiv \frac{1}{v_1}(1 - v_2\bar{C}_{\mu}) \tag{2.180}$$

The deviation of the microstructural stress from the mean in the particulates is then given by

$$\left|\frac{\langle tr\boldsymbol{\sigma}\rangle_{\Omega_2} - \langle tr\boldsymbol{\sigma}\rangle_{\Omega}}{\langle tr\boldsymbol{\sigma}\rangle_{\Omega_2}}\right| = \left|\frac{\bar{C}_{\kappa} - 1}{\bar{C}_{\kappa}}\right| \tag{2.181}$$

and

$$\sqrt{\frac{(\langle\boldsymbol{\sigma}'\rangle_{\Omega_2} - \langle\boldsymbol{\sigma}'\rangle_{\Omega}) : (\langle\boldsymbol{\sigma}'\rangle_{\Omega_2} - \langle\boldsymbol{\sigma}\rangle_{\Omega})}{\langle\boldsymbol{\sigma}'\rangle_{\Omega_2} : \langle\boldsymbol{\sigma}'\rangle_{\Omega_2}}} = \left|\frac{\bar{C}_{\mu} - 1}{\bar{C}_{\mu}}\right| \tag{2.182}$$

For the matrix we have

$$\left|\frac{\langle tr\boldsymbol{\sigma}\rangle_{\Omega_1} - \langle tr\boldsymbol{\sigma}\rangle_{\Omega}}{\langle tr\boldsymbol{\sigma}\rangle_{\Omega_1}}\right| = \left|\frac{\bar{\bar{C}}_{\kappa} - 1}{\bar{\bar{C}}_{\kappa}}\right| \tag{2.183}$$

and

$$\sqrt{\frac{(\langle\boldsymbol{\sigma}'\rangle_{\Omega_1} - \langle\boldsymbol{\sigma}'\rangle_{\Omega}) : (\langle\boldsymbol{\sigma}'\rangle_{\Omega_1} - \langle\boldsymbol{\sigma}\rangle_{\Omega})}{\langle\boldsymbol{\sigma}'\rangle_{\Omega_1} : \langle\boldsymbol{\sigma}'\rangle_{\Omega_1}}} = \left|\frac{\bar{\bar{C}}_{\mu} - 1}{\bar{\bar{C}}_{\mu}}\right| \tag{2.184}$$

These stresses can then by incorporated into the overall cost function by introducing a tolerance for them

$$\left|\frac{\bar{C}_{\kappa} - 1}{\bar{C}_{\kappa}}\right| \leq tol_{\kappa} \quad and \quad \left|\frac{\bar{C}_{\mu} - 1}{\bar{C}_{\mu}}\right| \leq tol_{\mu} \tag{2.185}$$

$$\left|\frac{\bar{\bar{C}}_{\kappa} - 1}{\bar{\bar{C}}_{\kappa}}\right| \leq tol_{\kappa} \quad and \quad \left|\frac{\bar{\bar{C}}_{\mu} - 1}{\bar{\bar{C}}_{\mu}}\right| \leq tol_{\mu} \tag{2.186}$$

When the microstructural stress exceeds the given tolerance, the weight term in the cost function corresponding to the microstructural stress can penalize the given genetic string.

The overall cost function is then defined by taking a value between the Hashin-Shtrikman bounds for the desired properties, and subsequently adding the weights for the microstructural stress considerations. The value for the effective properties can be determined using the following approximation $\kappa^* \approx \theta\kappa^*_+ + (1-\theta)\kappa^*_-$ where θ represents a random number.

The isotropic materials design problem has five design variables in its genetic string $\mathbf{\Lambda} = \{\kappa_2, \mu_2, \lambda_2, \rho_2, v_2\}$ with

$$\kappa_2^{(-)} \leq \kappa_2 \leq \kappa_2^{(+)} \quad \mu_2^{(-)} \leq \mu_2 \leq \mu_2^{(+)} \quad \lambda_2^{(-)} \leq \lambda_2 \leq \lambda_2^{(+)} \quad \rho_2^{(-)} \leq \rho_2 \leq \rho_2^{(+)} \quad v_2^{(-)} \leq v_2 \leq v_2^{(+)}$$

The algorithm for the isotropic genetic algorithm using the Hashin-Shtrikman bounds is summarized below

Algorithm 2.1 Genetic Algorithm for Isotropic Microstructures

1. Randomly create n initial genetic strings within a given design range that contain the information of the properties to be optimized: $\mathbf{\Lambda}^i(i = 1,\ldots,N) \equiv \{\Lambda_1^i, \Lambda_2^i, \Lambda_3^i, \ldots\} = \{\kappa^i, \mu^i, \rho^i, \lambda^i, \ldots\}$. For this study the genetic string will be

$$\mathbf{\Lambda} = \{\kappa_1, \mu_1, \rho_1, \lambda_1, \kappa_2, \mu_2, \lambda_2, \rho_2, v_2\}$$

2. Compute the fitness of each genetic string in the set using the cost function $\Pi(\mathbf{\Lambda}^i)$ defined for the desired optimization.

$$\Pi(k^*, \mu^*, \lambda^*, \rho^*) = w_1 \left| \frac{k^*}{k^{*,D}} - 1 \right| + w_2 \left| \frac{\mu^*}{\mu^{*,D}} - 1 \right| + w_3 \left| \frac{\lambda^*}{\lambda^{*,D}} - 1 \right| + w_4 \left| \frac{\rho^*}{\rho^{*,D}} - 1 \right|$$

$$+\hat{w}_5(\left| \frac{((\bar{C}_\kappa - 1)/\bar{C}_\kappa)}{tol_\kappa} - 1 \right|) + \hat{w}_6(\left| \frac{((\bar{C}_\mu - 1)/\bar{C}_\mu)}{tol_\mu} - 1 \right|)$$

$$+\hat{w}_7(\left| \frac{((\bar{\bar{C}}_\kappa - 1)/\bar{\bar{C}}_\kappa)}{tol_\kappa} - 1 \right|) + \hat{w}_8(\left| \frac{((\bar{\bar{C}}_\mu - 1)/\bar{\bar{C}}_\mu)}{tol_\mu} - 1 \right|)$$

   where

   (a) if $\left|(\bar{C}_\kappa - 1)/\bar{C}_\kappa\right| \leq tol_\kappa$, then $\hat{w}_5 = 0$, if $\left|(\bar{C}_\kappa - 1)/\bar{C}_\kappa\right| > tol_\kappa$, then $\hat{w}_5 = w_5$

   (b) if $\left|(\bar{C}_\mu - 1)/\bar{C}_\mu\right| \leq tol_\mu$, then $\hat{w}_6 = 0$, if $\left|(\bar{C}_\mu - 1)/\bar{C}_\mu\right| > tol_\mu$, then $\hat{w}_6 = w_6$

   (c) if $\left|(\bar{\bar{C}}_\kappa - 1)/\bar{\bar{C}}_\kappa\right| \leq tol_\kappa$, then $\hat{w}_7 = 0$, if $\left|(\bar{\bar{C}}_\kappa - 1)/\bar{\bar{C}}_\kappa\right| > tol_\kappa$, then $\hat{w}_7 = w_7$

   (d) if $\left|(\bar{\bar{C}}_\mu - 1)/\bar{\bar{C}}_\mu\right| \leq tol_\mu$, then $\hat{w}_8 = 0$, if $\left|(\bar{\bar{C}}_\mu - 1)/\bar{\bar{C}}_\mu\right| > tol_\mu$, then $\hat{w}_8 = w_8$

3. Rank the genetic strings by their fitness, meaning by the value of the cost function they produce. A lower cost function yields a higher rank for the string.

4. Make the nearest pairs in the ranked set produce offspring using: $\lambda^i \equiv \Phi^{(I)}\mathbf{\Lambda}^i + (1 - \phi^{(I)})\mathbf{\Lambda}^{i+1}$, $\lambda^{i+1} \equiv \Phi^{(II)}\mathbf{\Lambda}^i + (1 - \phi^{(II)})\mathbf{\Lambda}^{i+1}$, where $0 \leq \Phi^{(I)}, \Phi^{(II)} = Random \leq 1$ for each component.

5. Enforce the design constraints of the problem and discard unviable candidates; for example $\kappa_1^{(-)} \leq \kappa_1^i \leq \kappa_1^{(+)}$, $\kappa_2^{(-)} \leq \kappa_2^i \leq \kappa_2^{(+)}$, etc.

6. Discard the bottom $m < n$ strings and keep the top k parents, where m, n, k are selected by the user

7. Repeat steps 2 through 6 with the new set of genetic strings

2.5 Microstructural Genetic Algorithm for Mesoscale Design

Following the down-selection of constituent materials for the CMC using the above-described genetic algorithm for isotropic microstructures, another genetic algorithm based optimization is applied to obtain a more detailed image of the mesoscale structure of the CMC. The microstructural genetic algorithm adds inclusions to the RVE consisting of the matrix material of the CMC to form a composite material structure based on the matrix material and the inclusion of different geometries. The inclusion for this study focuses on the spherical inclusions.

2.5.1 Spherical Inclusions

Spherical inclusions in the CMC were mathematically modeled by selecting a center point of the spherical inclusion and a radius for the sphere. Inside the finite element computation, the following equation was to determine whether the integration point was located inside the inclusion material or the matrix material:

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 \leq r^2 \tag{2.187}$$

where $\{x_0, y_0, z_0\}$ represent the center of the spherical inclusion and $\{r\}$ represents the radius of the given spherical inclusion. If the integration was located inside the spherical inclusion, the material properties of the inclusion material were used in the finite element computation, otherwise the matrix material properties were used.

Each mesoscale structure consisting of matrix material and spherical inclusions can then be represented by a list that contains the relevant information for the inclusions, such as

	x_0	y_0	z_0	r
Inclusion 1	0.1	0.1	0.1	0.04
Inclusion 2	0.5	0.2	0.3	0.1
Inclusion 3	0.1	0.1	0.5	0.01
etc				

Table 2.1: Sample Spherical Inclusion List with center coordinates $\{x_0, y_0, z_0\}$ and radius $\{r\}$

Next, the following algorithm is applied:

Algorithm 2.2 Genetic Algorithm for Microstructural Design

```
1. Randomly create $n$ initial spherical inclusion lists
2. Compute the effective material properties of each microstructure using the finite element method as described earlier
3. Evaluate the fitness of each microstructure using the following cost function
       $$\Pi = \frac{||\mathbb{E}^*_{desired} - \mathbb{E}^*||}{||\mathbb{E}^*_{desired}||} + \frac{||\mathbb{K}^*_{desired} - \mathbb{K}^*||}{||\mathbb{K}^*_{desired}||}$$
   representing the effective properties of the entire microstructure
4. Rank the genetic strings by their fitness, meaning by the value of the cost function they produce. A lower cost function yields a higher rank for the string.
5. Make the nearest pairs in the ranked set produce offspring using: $\lambda^i \equiv \Phi^{(I)}\mathbf{\Lambda}^i + (1 - \phi^{(I)})\mathbf{\Lambda}^{i+1}$, $\lambda^{i+1} \equiv \Phi^{(II)}\mathbf{\Lambda}^i + (1 - \phi^{(II)})\mathbf{\Lambda}^{i+1}$ , where $0 \leq \Phi^{(I)}, \Phi^{(II)} = Random \leq 1$ for each component.
6. Enforce the design constraints of the problem, which in this case includes making sure no spherical inclusions intersect
7. Discard the bottom $m < n$ strings and keep the top $k$ parents, where $m, n, k$ are selected by the user
8. Repeat steps 2 through 6 with the new set of genetic strings
```

The above algorithm then gives a set of best performing microstructures with spherical inclusions.

2.6 Virtual Tests for Mesoscale CMCs

Following the selection of the best-performing mesoscale structures, two virtual experiments are performed:

- Material Fracture Stress Computation

The computation of the material toughness is based on a variation of the previously described damage model. For this computation, the effective stress inside an element is calculated, and when that stress exceeds a critical level, the element is effectively "deleted," meaning that its elasticity matrix component of that element is set to 0. This element deletion model is used to approximate the brittle behavior of the constituent ceramic material, and assumes that once the material inside an element is damaged it can no longer sustain mechanical loads, thereby forcing the rest of the material to handle greater loads. In other words if

$$\|\boldsymbol{\sigma}'\| > \sigma_{critical} \to \mathcal{D} = 0 \tag{2.188}$$

where $\boldsymbol{\sigma}'$ is the deviator stress inside the material is defined by

$$\boldsymbol{\sigma}' = \boldsymbol{\sigma} - \frac{tr\boldsymbol{\sigma}}{3} \cdot \mathbf{1} \tag{2.189}$$

In this virtual test, the following search algorithm is applied to find the critical stress at which a given fraction of the material fractures, represented in the damage variable $\mathcal{D}_{avg}^{crit}$.

Algorithm 2.3 Toughness Stress Search Algorithm

```
1. Initialize a starting value for the loading stress and compute the average damage for the RVE $\mathcal{D}_{avg}$
2. Find a suitable range for the critical loading stress where the lower stress yields $\mathcal{D}_{avg} > \mathcal{D}_{avg}^{crit}$ and the higher stress yields $\mathcal{D}_{avg} < \mathcal{D}_{avg}^{crit}$
    (a) If the initial guess $\sigma_{i=0}$ yields $\mathcal{D}_{avg} > \mathcal{D}_{avg}^{crit}$ then increase $\sigma_{i=0}$ to $\sigma_{i=1} = 10 * \sigma_{i=0}$
    (b) If the initial guess $\sigma_{i=0}$ yields $\mathcal{D}_{avg} < \mathcal{D}_{avg}^{crit}$ then increase $\sigma_{i=0}$ to $\sigma_{i=1} = 0.1 * \sigma_{i=0}$
3. Take the midpoint $\sigma_{mid} = \frac{\sigma_{high} - \sigma_{low}}{2}$ of the previously determined stress of $\sigma_{high}$ with $\mathcal{D}_{avg} < \mathcal{D}_{avg}^{crit}$ and $\sigma_{low}$ with $\mathcal{D}_{avg} > \mathcal{D}_{avg}^{crit}$
    (a) if $\mathcal{D}_{avg}^{\sigma_{mid}} > \mathcal{D}_{avg}^{crit}$ then $\sigma_{mid} = \sigma_{low}$
    (b) if $\mathcal{D}_{avg}^{\sigma_{mid}} < \mathcal{D}_{avg}^{crit}$ then $\sigma_{mid} = \sigma_{high}$
4. Repeat step 3 until to a certain number of iterations or until the stress change is within a given tolerance
```

- Phenomenological Fatigue Virtual Test

The stress dependent damage model shown in the above section can then also be applied for fatigue considerations to calculate relevant stresses for a phenomenological fatigue model based on the Basquin relation for cycles to failure N_f

$$\sigma_a = \sigma_f'(2N_f)^b \tag{2.190}$$

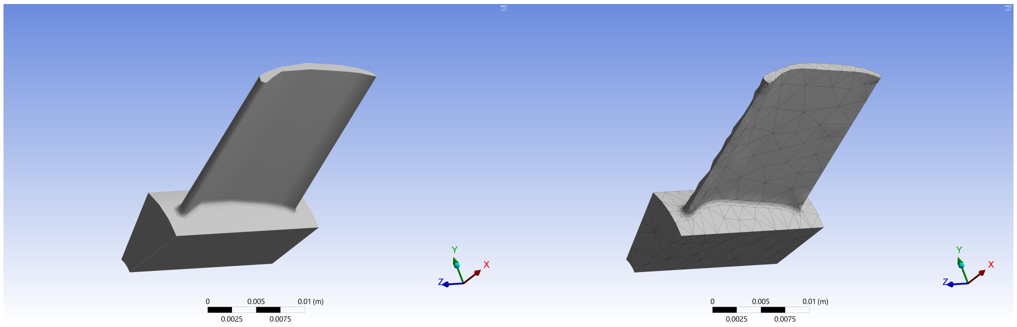
$$N_f = \frac{1}{2} \cdot (\frac{\sigma_a}{\sigma_f'})^{1/b} \tag{2.191}$$

where σ_a represent the amplitude of the cyclic stress load , σ_f is a fatigue stress coefficient that can be determined from experimental data for the constituent materials, and b is Basquin's exponent, which usually ranges between -0.05 and -0.12.

The stress amplitude σ_a is calculated by alternating between tension and compaction stresses applied to the boundary of the RVE. For this computation, the stress was applied on the top surface of the RVE with all other surfaces having fixed displacements. These conditions simulate an environment where a stress is imposed on the RVE coming from the surface of the greater material and traveling through the bulk material inwards, where the bulk material is represented by the RVE. The impact stress on top surface, representing either tension or compaction, is varied according to the search algorithm shown in Algorithm 2.3 above.

2.7 Structural Level Simulations

The simulations at the structural level were performed with the ANSYS Mechanical solver. The mechanical solver takes in the geometry of a single turbine blade, which represents the full three-dimensional structure of the final product. The material properties for the different CMC structures calculated at the mesoscale can be imported via the ANSYS 'Engineering Data' library that allows the user to create new materials. The turbine blade is then meshed with a regular triangular mesh from the ANSYS library, and imported into ANSYS Mechanical to apply the boundary conditions and solve the structural level problem.



(a) Turbine Blade Geometry

(b) Turbine Blade Mesh

Figure 2.6: ANSYS Turbine Blade

Chapter 3

Case Study

3.1 General Workflow

The following section summarizes a case study applying the design framework discussed in previous chapters. The aim of this case study was to apply the automated design framework to find a CMC material that approximates the mechanical properties, specifically the elasticity tensor, and the thermal conductivity tensor of a given target material. Subsequently, the compound chosen from the design framework undergoes virtual tests related to gaining some preliminary insights into the toughness and fatigue properties of the given compound. Ideally, the virtual tests could be integrated into the overall design framework, as described in Chapter 4, which was not completed in this work.

The properties of the superalloy compounds described in Chapter 1, as well as many other commercial materials are not publicly available given the proprietary nature of their commercial use. As such, the case was carried with a sample material that can be found in the ANSYS Engineering Data library, namely the material defined in the ANSYS library as structural steel. It should be emphasized that the choice of structural steel just serves as an example to demonstrate the framework that was described in Chapter 2, and the material properties can be easily replaced by material properties of choice within the given framework. This flexibility of the framework to handle various inputs enables it to be used to identify a preliminary design for a variety of target materials and target material properties.

For the specific study presented in this chapter, the material properties from ANSYS structural steel were used as the target property in the cost functions of the design algorithm, which then prototyped various designs of CMC mesoscale structures. The material properties from the best performing CMC mesoscale structures are then imported into the ANSYS materials library to perform the structural simulations. The overall workflow of the design framework using the tools described in Chapter 2, is as follows:

Algorithm 3.1 CMC Design General Workflow

1. Choose the target material properties

2. Apply the isotropic genetic algorithm described in Section 2.4 to obtain the constituent materials for the matrix and the inclusion material

3. Apply the microstructural genetic algorithm described in Section 2.5 to obtain the microstructural arrangement of the composite material

4. Perform the virtual tests described in Section 2.6, including the toughness and the fatigue virtual tests

5. Import the resulting material properties into ANSYS by creating a new entry in the ANSYS Engineering Data library to perform structural scale simulations with the properties of the virtually created CMC material

3.1.1 Parallel Computing Framework

The calculations required to perform the mesoscale design of CMC structures in Step 3 of Algorithm 3.1, are computationally intensive given that each design requires solving at least one complete three-dimensional finite element simulation. Since each of the CMC designs has to undergo the same virtual tests, with each design being completely independent and not interacting with other design, the computations required for each design can be performed in parallel to reduce the wall-clock time required for the overall calculation. While the CPU hours for the overall remains the same, the overall wall-clock time can be reduced significantly by parallelizing the required computations for each individual mesoscale CMC design.

Message-Passing-Interface (MPI) was used to divide the various CMC mesoscale designs among the available processors for parallel computations of the necessary finite element calculations, and the required information was then transferred back to the root processor for the necessary serial computations. The overall framework for a parallelized mesoscale design genetic algorithm is illustrated in the figure below:

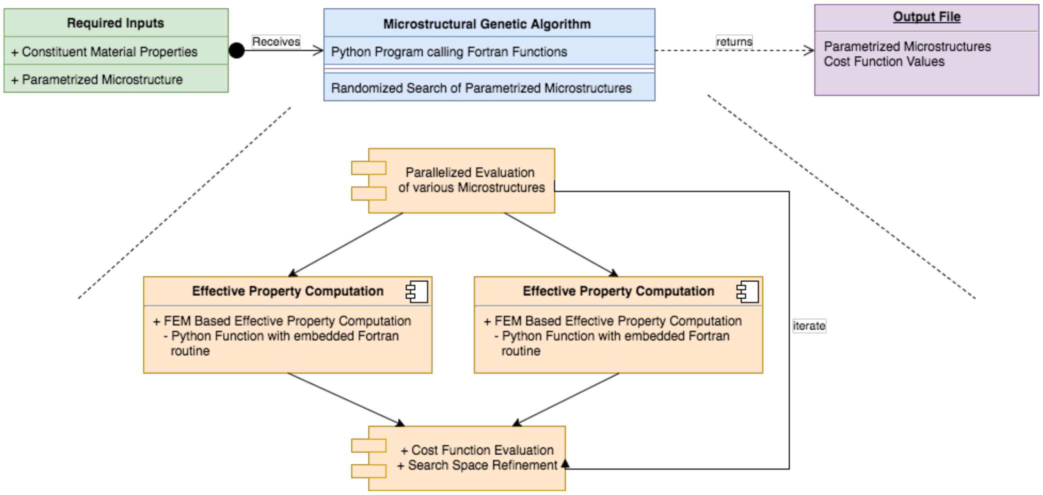

Required Inputs
+ Constituent Material Properties
+ Parametrized Microstructure
Receives
Microstructural Genetic Algorithm
Python Program calling Fortran Functions
Randomized Search of Parametrized Microstructures
returns
Output File
Parametrized Microstructures
Cost Function Values
Parallelized Evaluation of various Microstructures
Effective Property Computation
+ FEM Based Effective Property Computation
- Python Function with embedded Fortran routine
Effective Property Computation
+ FEM Based Effective Property Computation
- Python Function with embedded Fortran routine
iterate
+ Cost Function Evaluation
+ Search Space Refinement

Figure 3.1: Parallelized Mesoscale CMC Design

Figure 3.1 illustrates the workflow of the parallel computation for mesoscale design. On the left are the required inputs for computation, which are the material properties of the constituent matrix and inclusion materials, as well as a parametrized microstructural arrangement of the two materials. The inputs are fed into the Microstructural Genetic Algorithm, which evaluates the various microstructures in parallel as schematically illustrated by the Effective Property Computation boxes. The result of the evaluation yields a value for the cost function described in Section 2.5 that can then be used to further refine the microstructural search. The output of this step are the best-performing microstructures in their parametrized form along with their cost function values.

In the virtual test used to compute the critical fracture stress for a mesoscale CMC defined by the procedure outlined in Section 2.6, parallel computing was used to search various stresses at the same time. This allows one to find the desired stress range more quickly, which will then also allow the user to narrow down a more precise for the critical fracture stress more effectively.

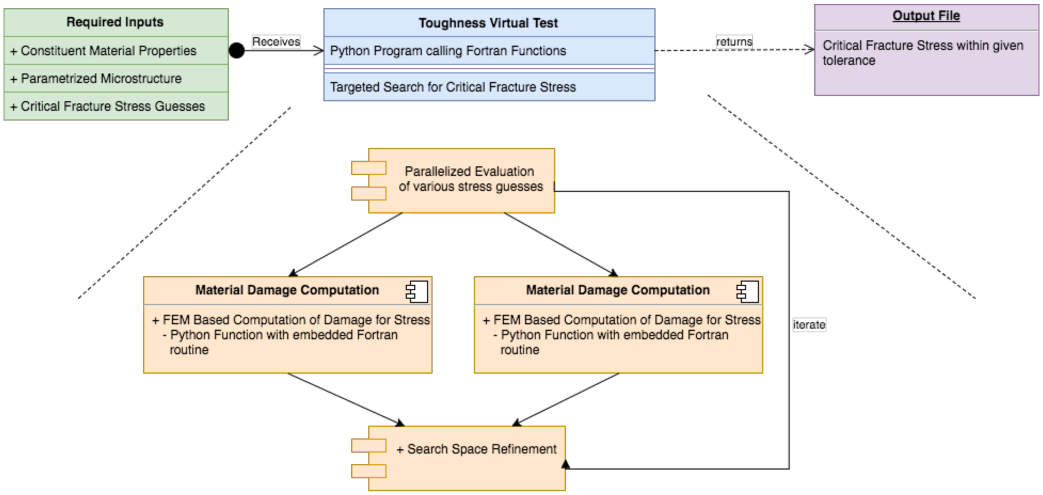

Required Inputs
+ Constituent Material Properties
+ Parametrized Microstructure
+ Critical Fracture Stress Guesses
Receives
Toughness Virtual Test
Python Program calling Fortran Functions
Targeted Search for Critical Fracture Stress
returns
Output File
Critical Fracture Stress within given tolerance
Parallelized Evaluation of various stress guesses
Material Damage Computation
+ FEM Based Computation of Damage for Stress
- Python Function with embedded Fortran routine
Material Damage Computation
+ FEM Based Computation of Damage for Stress
- Python Function with embedded Fortran routine
iterate
+ Search Space Refinement

Figure 3.2: Toughness Virtual Test

The toughness virtual tests requires the inputs outlined in the left panel: relevant material properties, composite microstructure, initial guess for the critical fracture stress. The program then performs a search for the critical fracture stress, where each stress state involves the computation of a finite element problem to determine the overall damage state of the material. The resulting damage state allows the program to refine the search space for the critical stress and repeat the procedure until the stress is found within a given tolerance.

3.2 Verification of the Genetic Algorithm

The cost functions that are used in both Genetic Algorithms in Section 2.4 and Section 2.5 (Algorithm 2.1 and Algorithm 2.2 respectively) are highly complex and the algorithms are not guaranteed to converge to a minimum value for the design problem formulated in this case study. This is because the trivial solutions for an unconstrained Genetic Algorithm have been removed by the problem setup, both in the isotropic algorithm that uses the Hashin-Shtrikman bounds and the mesoscale algorithm that computes the effective properties using the finite element method. As such, it is very difficult, if not impossible, to know whether the solutions found by the Genetic Algorithms are global minima and it is also very difficult to know if better solutions could be found in later iterations. Therefore, the designer must choose when to stop the algorithm, which is generally done by

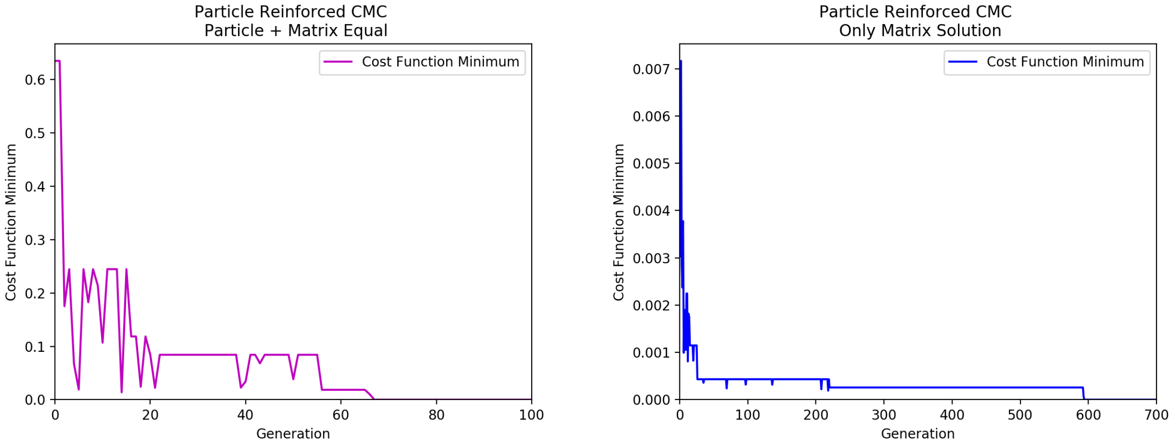
1. Stopping when the cost function falls below a given tolerance
2. Stopping when the cost function gradient falls below a given tolerance
3. Stopping when a maximum number of iterations is reached

Generally speaking, the preference for stopping goes from being most preferable in the first case to least preferable in the third. However, given that computational resources are often limited, especially in the case of numerically intensive calculations, the program is often stopped after reaching a maximum number of iterations.

In the situations for which a known trivial solution exists, the algorithm should converge to that given solution. In the case of the Genetic Algorithm presented here there are two trivial solutions:

- The matrix and inclusion properties are both equal to the desired property
- The composite is made entirely of matrix material which is equal to the desired property

When the conditions of the Genetic Algorithm are not constrained, meaning that when the ranges of the matrix and inclusion material are set such that those trivial solutions are included in the solution space, the algorithm does converge to one of the possible trivial solutions as shown in Figure 3.3 below:



(a) Matrix and Inclusion Both Equal to Target

(b) Matrix Equal to Target with No Inclusion Material

Figure 3.3: Trivial Solutions for Genetic Algorithm

3.3 ANSYS Structural Steel Example

3.3.1 Isotropic Genetic Algorithm

The isotropic genetic algorithm, which is Algorithm 2.1 outlined in Section 2.4, uses the Hashin-Shtrikman bounds to aid the relevant effective material property computations. The summary of the algorithm is shown below with more relevant details, including the definition of the design variables, outlined in Section 2.4. Algorithm 3.2 was applied using the following target variables (Table 3.1) and constraints (Table 3.2):

Property	Target Value
Bulk Modulus	166.67 GPa
Shear Modulus	79.923 GPa
Thermal Conductivity	60.5 W/mK

Table 3.1: Target Design Variables

Matrix Variable	Range	Reinforcement Variable	Range
Bulk Modulus	0.001- 100 GPa	Bulk Modulus	10^3 - 10^7 GPa
Shear Modulus	0.007- 70 GPa	Shear Modulus	700- $7 * 10^5$ GPa
Thermal Conductivity	1- 40 W/mK	Thermal Conductivity	100- 4000 W/mK

Table 3.2: Isotropic Genetic Algorithm Constraints

The weight for the algorithm were chosen as follows: $w_1 = 10.0$, $w_2 = 10.0\, w_3 = 1.0$, $w_4 = 1.0$, $w_5 = 3.0$, $w_6 = 3.0$, $w_7 = 3.0$, $w_8 = 3.0$. The weight above emphasize better performance on the mechanical properties (bulk modulus k and shear modulus μ) over the thermal conductivity λ. The weights also penalize a mismatch in the moduli between the matrix and the reinforcement material (expressed in w_5, w_6, w_7, w_8).

Algorithm 3.2 Genetic Algorithm for Isotropic Microstructures

1. Randomly create n initial genetic strings within a given design range that contain the information of the properties to be optimized: $\mathbf{\Lambda}^i (i = 1, \ldots, N) \equiv \{\Lambda^i_1, \Lambda^i_2, \Lambda^i_3, \ldots\} = \{\kappa^i, \mu^i, \rho^i, \lambda^i, \ldots\}$. For this study the genetic string will be

$$\mathbf{\Lambda} = \{\kappa_1, \mu_1, \rho_1, \lambda_1, \kappa_2, \mu_2, \lambda_2, \rho_2, v_2\}$$

2. Compute the fitness of each genetic string in the set using the cost function $\Pi(\mathbf{\Lambda}^i)$ defined for the desired optimization.

$$\Pi(k^*, \mu^*, \lambda^*, \rho^*) = w_1 \left| \frac{k^*}{k^{*,D}} - 1 \right| + w_2 \left| \frac{\mu^*}{\mu^{*,D}} - 1 \right| + w_3 \left| \frac{\lambda^*}{\lambda^{*,D}} - 1 \right| + w_4 \left| \frac{\rho^*}{\rho^{*,D}} - 1 \right|$$

$$+\hat{w}_5 (\left| \frac{((\bar{C}_\kappa - 1)/\bar{C}_\kappa)}{tol_\kappa} - 1 \right|) + \hat{w}_6 (\left| \frac{((\bar{C}_\mu - 1)/\bar{C}_\mu)}{tol_\mu} - 1 \right|)$$

$$+\hat{w}_7 (\left| \frac{((\bar{\bar{C}}_\kappa - 1)/\bar{\bar{C}}_\kappa)}{tol_\kappa} - 1 \right|) + \hat{w}_8 (\left| \frac{((\bar{\bar{C}}_\mu - 1)/\bar{\bar{C}}_\mu)}{tol_\mu} - 1 \right|)$$

   where

   (a) if $\left|(\bar{C}_\kappa - 1)/\bar{C}_\kappa\right| \le tol_\kappa$, then $\hat{w}_5 = 0$, if $\left|(\bar{C}_\kappa - 1)/\bar{C}_\kappa\right| > tol_\kappa$, then $\hat{w}_5 = w_5$

   (b) if $\left|(\bar{C}_\mu - 1)/\bar{C}_\mu\right| \le tol_\mu$, then $\hat{w}_6 = 0$, if $\left|(\bar{C}_\mu - 1)/\bar{C}_\mu\right| > tol_\mu$, then $\hat{w}_6 = w_6$

   (c) if $\left|(\bar{\bar{C}}_\kappa - 1)/\bar{\bar{C}}_\kappa\right| \le tol_\kappa$, then $\hat{w}_7 = 0$, if $\left|(\bar{\bar{C}}_\kappa - 1)/\bar{\bar{C}}_\kappa\right| > tol_\kappa$, then $\hat{w}_7 = w_7$

   (d) if $\left|(\bar{\bar{C}}_\mu - 1)/\bar{\bar{C}}_\mu\right| \le tol_\mu$, then $\hat{w}_8 = 0$, if $\left|(\bar{\bar{C}}_\mu - 1)/\bar{\bar{C}}_\mu\right| > tol_\mu$, then $\hat{w}_8 = w_8$

3. Rank the genetic strings by their fitness, meaning by the value of the cost function they produce. A lower cost function yields a higher rank for the string.

4. Make the nearest pairs in the ranked set produce offspring using: $\lambda^i \equiv \Phi^{(I)}\mathbf{\Lambda}^i + (1 - \phi^{(I)})\mathbf{\Lambda}^{i+1}$, $\lambda^{i+1} \equiv \Phi^{(II)}\mathbf{\Lambda}^i + (1 - \phi^{(II)})\mathbf{\Lambda}^{i+1}$, where $0 \le \Phi^{(I)}, \Phi^{(II)} = Random \le 1$ for each component.

5. Enforce the design constraints of the problem and discard unviable candidates that do not fit into the design range

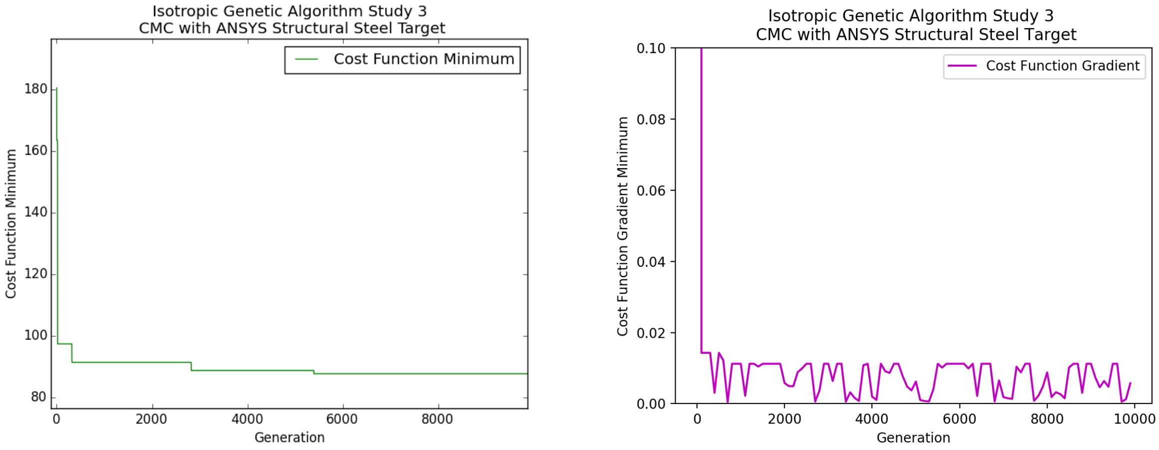
6. Discard the bottom $m < n$ strings and keep the top k parents, where m, n, k are selected by the user

7. Repeat steps 2 through 6 with the new set of genetic strings

As seen from the algorithm above, the cost is unit-less and measures the performance of a composite material compared to a set of target material properties. As outlined in Section 3.1, for the illustrative purposes of this study the ANSYS structural steel properties used as the desired properties. The stopping criteria for the algorithm were as follows:

1. Cost function falls below 10^{-5}
2. Cost function gradient falls below 10^{-5}
3. Reach a maximum number of iterations at $10,000$

The algorithm did not reach the cost function or cost function gradient criteria, and terminated at 10,000 iterations. The properties from the best-performing string were then transferred to algorithm for mesoscale design. As shown in the figure below, the cost function decreased until reaching a plateau around 6,000 iterations.



(a) Cost Function Minimum Over Various Iterations (b) Cost Function Minimum Over Various Iterations

Figure 3.4: Isotropic Genetic Algorithm Results: a) Cost Function; b) Cost Function Gradient

The figure above indicates that the algorithm reaches a series of local extremum in the exploration of the cost function, and that the best-performing string remains at the given local extremum until a lower extremum is found to replace it. The gradient of the cost function indicates that the neighborhood of each extremum reached by the cost function fluctuates between 10^{-2} and 10^{-4}, which suggests that the algorithm may have reached a shallow local minimum or a local saddle point. Since the algorithm keeps the best-performing parents of previous generations, the overall minimum of the cost function function decreases monotonically as the lowest extremum is only replaced when a better candidate is found.

The properties for the best performing string after 10,000 iterations in this case were:

Property	Matrix Material	Reinforcement Material	Target Properties
Bulk Modulus	$234 \cdot 10^9\, Pa$	$103 \cdot 10^9\, Pa$	$166.67 \cdot 10^9\, Pa$
Shear Modulus	$164 \cdot 10^9\, Pa$	$72 \cdot 10^9\, Pa$	$79.923 \cdot 10^9\, Pa$
Thermal Conductivity	$94.3\, \frac{W}{mK}$	$100\, \frac{W}{mK}$	$60.5\, \frac{W}{mK}$
Volume Fraction	0.95	0.05	

Table 3.3: Properties of Best-Performing Isotropic Genetic Algorithm String

The algorithm above can only determine the properties of the constituent matrix and reinforcement materials. The properties of the composite were computed using a weighted average of the Hashin-Shtrikman bounds as described in Chapter 2. These material properties were then used to compute the effective elasticity ($\mathbb{E}^*$) and thermal conductivity ($\mathbb{K}^*$) tensors in the mesoscale design algorithm that is described in Section 3.3.2 below. The mesoscale structure resulting from generated from the algorithm was then used for the virtual tests related to fracture stress, fatigue and structural scale behavior.

3.3.2 Mesoscale Design Genetic Algorithm

The material properties from the isotropic genetic algorithm were then used as the starting point for the mesoscale design algorithm. The mesoscale design algorithm is summarized below with more details described in Section 2.5:

Algorithm 3.3 Genetic Algorithm for Mesoscale Design

1. Randomly create n initial spherical inclusion lists

2. Compute the effective material properties of each microstructure using the finite element method as described earlier

3. Evaluate the fitness of each microstructure using the following cost function using the effective elasticity ($\mathbb{E}^*$) and thermal conductivity ($\mathbb{K}^*$) tensors

$$\Pi = \frac{||\mathbb{E}^*_{desired} - \mathbb{E}^*||}{||\mathbb{E}^*_{desired}||} + \frac{||\mathbb{K}^*_{desired} - \mathbb{K}^*||}{||\mathbb{K}^*_{desired}||}$$

   representing the effective properties of the entire microstructure

4. Rank the genetic strings by their fitness, meaning by the value of the cost function they produce. A lower cost function yields a higher rank for the string.

5. Make the nearest pairs in the ranked set produce offspring using: $\lambda^i \equiv \Phi^{(I)}\mathbf{\Lambda}^i + (1 - \phi^{(I)})\mathbf{\Lambda}^{i+1}$, $\lambda^{i+1} \equiv \Phi^{(II)}\mathbf{\Lambda}^i + (1 - \phi^{(II)})\mathbf{\Lambda}^{i+1}$, where $0 \leq \Phi^{(I)}, \Phi^{(II)} = Random \leq 1$ for each component.

6. Enforce the design constraints of the problem, which in this case includes making sure no spherical inclusions intersect

7. Discard the bottom $m < n$ strings and keep the top k parents, where m, n, k are selected by the user

8. Repeat steps 2 through 6 with the new set of genetic strings

The stopping criteria for the algorithm were as follows:

1. Cost function falls below 10^{-5}

2. Cost function gradient falls below 10^{-5}

3. Reach a maximum number of iterations at 500

The stopping criteria for 500 iteration was set due to the high computational intensity of the calculation. Each string in the algorithm requires solving the finite element problem outlined in Section 2.3, which is a series of 6 FEM computations for the elasticity tensor and 9 FEM computations for the thermal conductivity tensor. The computation was parallelized over 20 cores at the Savio supercomputing cluster at the University of California, Berkeley with a total runtime of about 1,000 CPU hours.

The mesoscale design algorithm started by initiating a random particle based mesoscale structure and then yielded improvements in the design as seen in the decrease in the cost function, until the simulations was stopped at 500 iterations. Gradient computations for this algorithm were not performed given that those computations would also require solving the series of FEM computations required for the cost function computations.

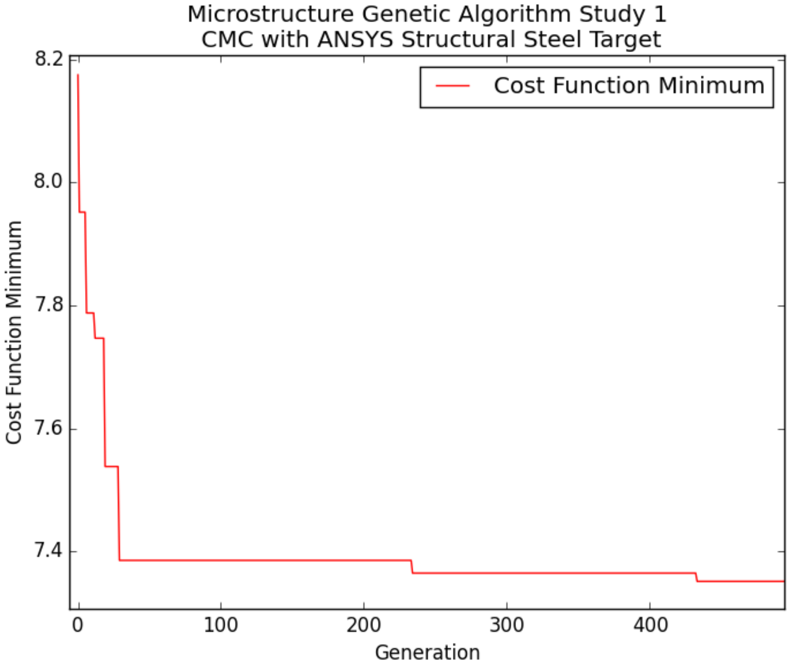
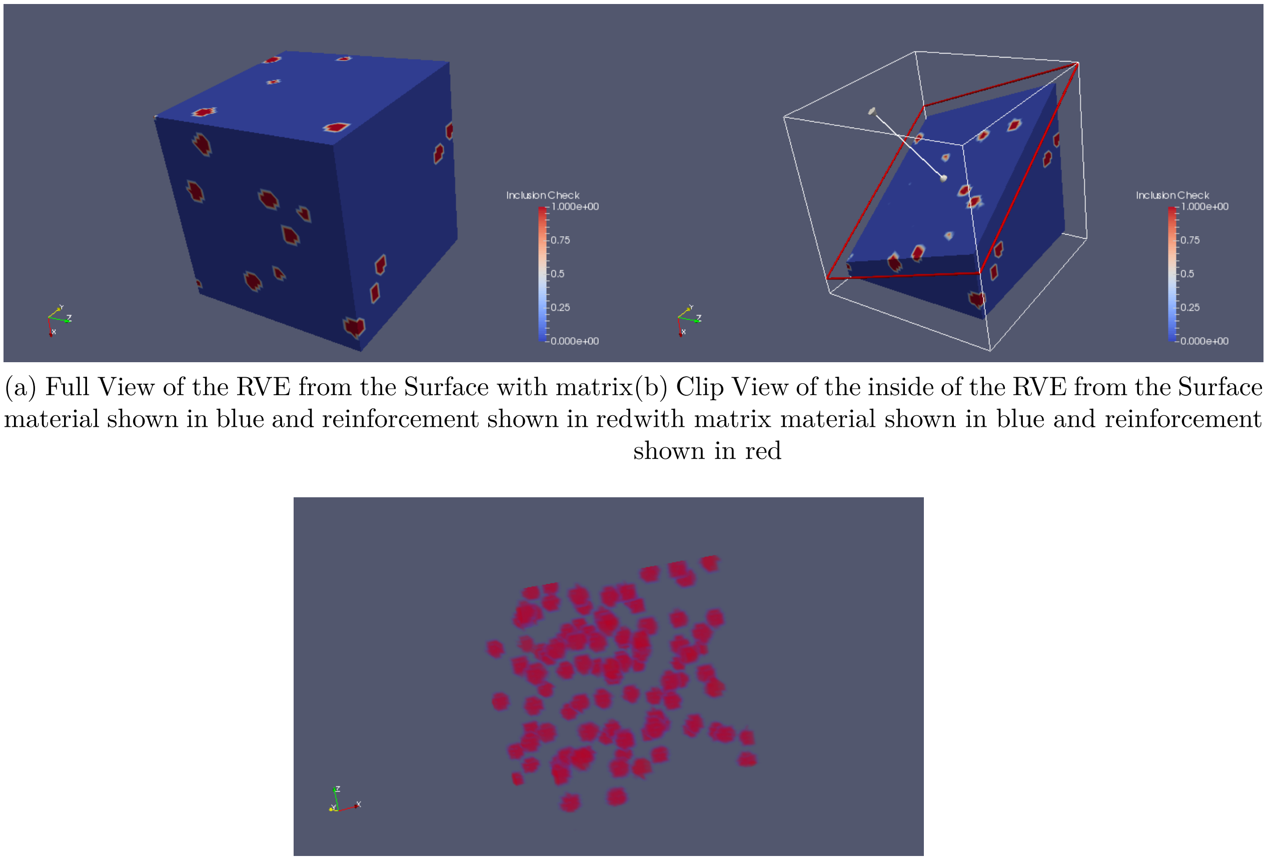


Figure 3.5: Cost Function Performance of Mesoscale Design Algorithm

The best-performing mesoscale structure of the algorithm is shown below



(a) Full View of the RVE from the Surface with matrix material shown in blue and reinforcement shown in red

(b) Clip View of the inside of the RVE from the Surface with matrix material shown in blue and reinforcement shown in red

(c) Volume View of the RVE showing the reinforcement structure

Figure 3.6: Mesoscale Structure Generated by Genetic Algorithm. The blue color indicates the presence of matrix material while the red color indicates the presence of reinforcement material. The variable inclusion check variable illustrates the amount of inclusion present at each location. The material is either matrix or reinforcement at a given location; variations between 0 and 1 can be attributed to the imprecision of the rending. Panel a) shows a full view of the RVE microstructure as seen from the surface; Panel b) shows a clip view inside the RVE; and Panel c) shows a Volume View that has the matrix material removed so that the inclusions can be clearly seen.

3.3.3 CMC Virtual Tests

3.3.3.1 Fracture Stress Computation

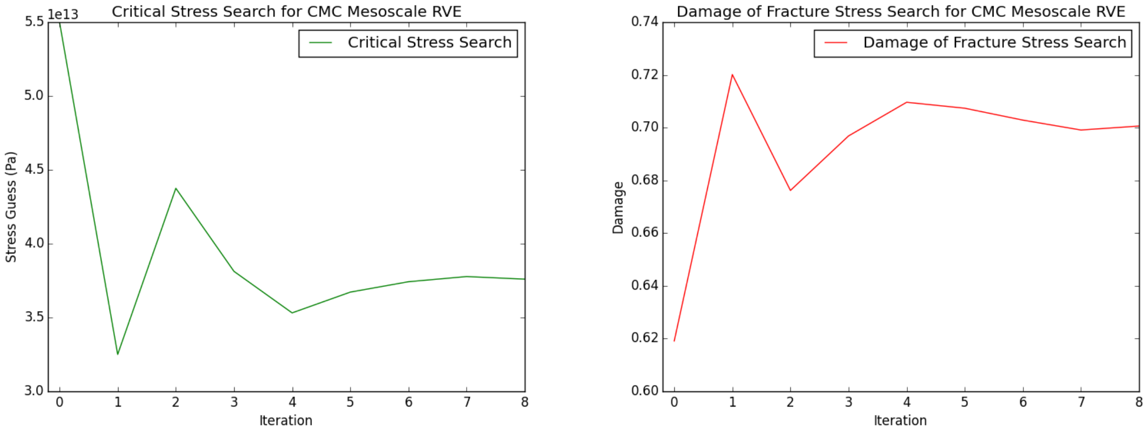
The fracture stress computation employed Algorithm 2.3 (described in Section 2.5) based on the previously described damage model to find a critical fracture stress for which $\mathcal{D}_{avg}^{crit} = 0.7$, as an illustrative example. The $\mathcal{D}_{avg}^{crit}$ is the parameter that determines when the RVE is considered to have failed, meaning that critical failure occurs if damage exceeds that critical value. Since the undamaged state is $\mathcal{D}_{avg} = 1$, $\mathcal{D}_{avg} > \mathcal{D}_{avg}^{crit}$ for the RVE to still be intact. A summary of the algorithm for finding $\mathcal{D}_{avg}^{crit}$ is reproduced from Section 2.5:

Algorithm 3.4 Toughness Stress Search Algorithm

```
1. Initialize a starting value for the loading stress and compute the average damage for the RVE $\mathcal{D}_{avg}$
2. Find a suitable range for the critical loading stress where the lower stress yields $\mathcal{D}_{avg} > \mathcal{D}_{avg}^{crit}$ and the higher stress yields $\mathcal{D}_{avg} < \mathcal{D}_{avg}^{crit}$
    (a) If the initial guess $\sigma_{i=0}$ yields $\mathcal{D}_{avg} > \mathcal{D}_{avg}^{crit}$ then increase $\sigma_{i=0}$ to $\sigma_{i=1} = 10*\sigma_{i=0}$
    (b) If the initial guess $\sigma_{i=0}$ yields $\mathcal{D}_{avg} < \mathcal{D}_{avg}^{crit}$ then increase $\sigma_{i=0}$ to $\sigma_{i=1} = 0.1*\sigma_{i=0}$
3. Take the midpoint $\sigma_{mid} = \frac{\sigma_{high} - \sigma_{low}}{2}$ of the previously determined stress of $\sigma_{high}$ with $\mathcal{D}_{avg} < \mathcal{D}_{avg}^{crit}$ and $\sigma_{low}$ with $\mathcal{D}_{avg} > \mathcal{D}_{avg}^{crit}$
    (a) if $\mathcal{D}_{avg}^{\sigma_{mid}} > \mathcal{D}_{avg}^{crit}$ then $\sigma_{mid} = \sigma_{low}$
    (b) if $\mathcal{D}_{avg}^{\sigma_{mid}} < \mathcal{D}_{avg}^{crit}$ then $\sigma_{mid} = \sigma_{high}$
4. Repeat step 3 until to a certain number of iterations or until the stress change is within a given tolerance
```

The computation of $\mathcal{D}_{avg}$ involves a finite element computation. For this computation, the stress was applied on the top surface of the RVE with all other surfaces having fixed displacements. These conditions simulate an environment where a stress is imposed on the RVE coming from the surface of the greater material and traveling through the bulk material inwards, where the bulk material is represented by the RVE. The impact stress on top surface of the RVE is varied according to the search algorithm shown in Algorithm 3.4 above.

For the RVE developed through the sequence of genetic algorithm, the critical loading stress was found to be $\sim 377 \cdot 10^{11}\ Pa$, which yielded material damage within a tolerance of ~ 0.00001 of overall RVE damage of $\mathcal{D}_{avg}^{crit} = 0.7$.

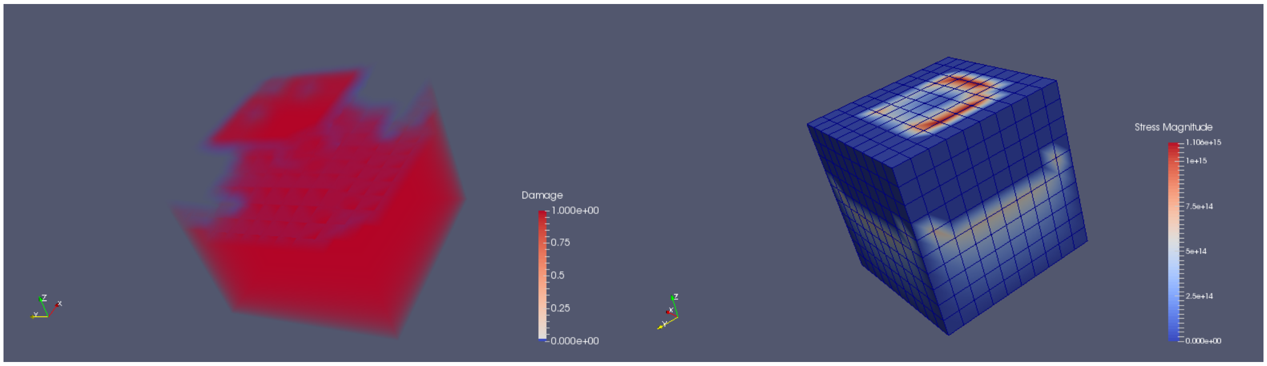


(a) Stress vs Binary Search Iterations

(b) $\mathcal{D}_{avg}$ vs Binary Search Iterations

Figure 3.7: Fracture Stress Search for the Material Found from the Sequential Genetic Algorithms

The virtual test showed that the material damaged near the boundary where the stress was applied, as shown in Figure 3.8 images below. In Figure 3.8 below, the volume damage shown in red corresponds to the undamaged material with $\mathcal{D} = 1$, while the material that is not shown has $\mathcal{D} = 0$. A stress profile of the surface of the material indicates zero stress across certain areas of the material that correspond to failed material that has a damage factor of $\mathcal{D} = 0$, which is to be expected from failed material. As a result of the failed material, the remaining material has to endure higher loads leading to high stress values at the remaining materials, which is unfavorable and likely to lead to further material failure.



(a) Volume Damage of the RVE

(b) Stress (in Pa) of the RVE shown on the Surface View

Figure 3.8: Volume Damage and Surface Stress of RVE: Panel a) describes the material that was damaged during the virtual stress test. Since that has damage $\mathcal{D} = 1.0$ (shown in red) remains undamaged while anything is in the background color has been fractured away. Panel b) shows a stress profile of the RVE at the end of the virtual test. The large areas of blue with zero stress indicate material that has been removed and can no longer bear any stress.

A slice through the RVE reveals the contrasting stress profile inside the RVE at the critical loading stress, where the damaged elements carry no stress and the remaining material has to carry substantial stress loads

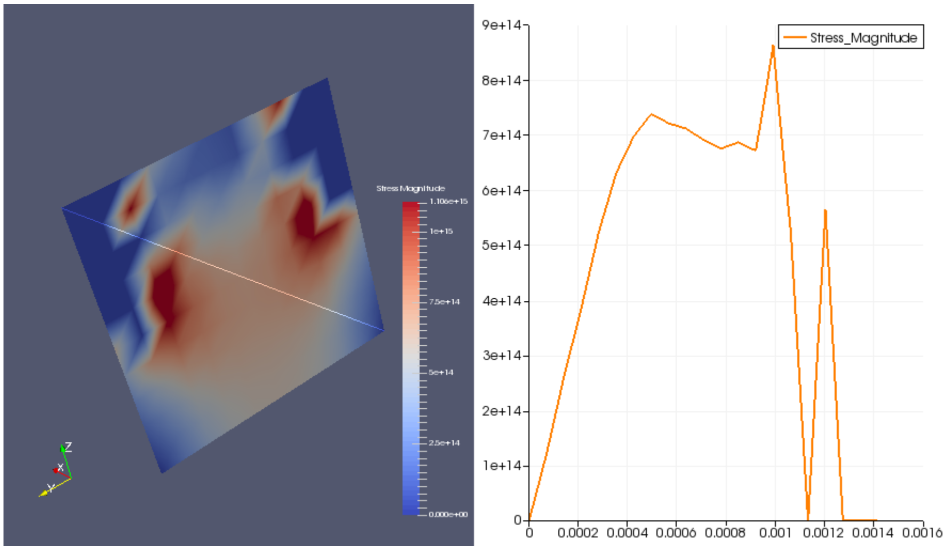


Figure 3.9: The left panel shows a plane inside the RVE after the virtual test. The blue areas indicate the damaged areas that can no longer bear any stresses. The right side shows a line profile of the white line shown in the left pane with the y-axis being stress in Pa and the x-axis being the position of the line in mm compared to the starting point in the corner of the surface. The line profile shows large concentrations of stress where the material is undamaged. The peaks in the stress distributions are locations of the reinforcement material, which can bear the largest loads. The damaged material bear no stress, hence the stress profile falls to zero.

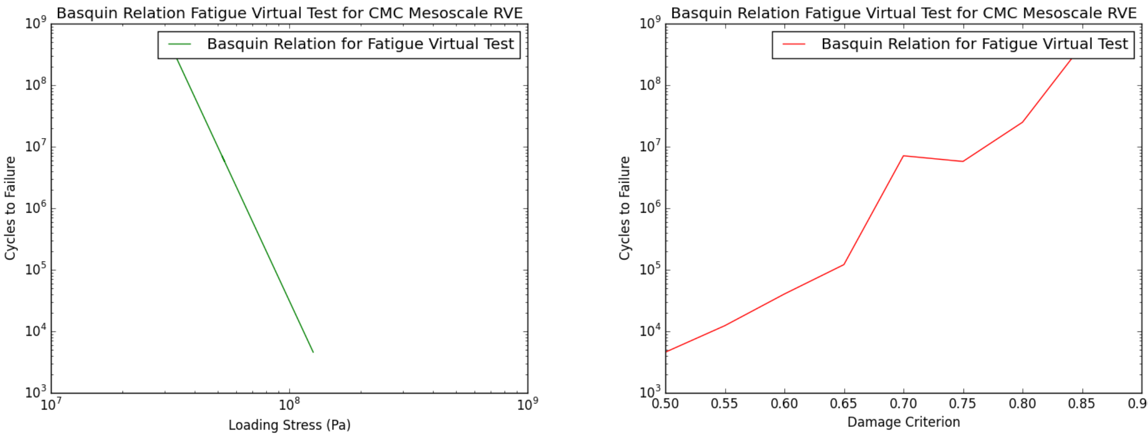
3.3.3.2 Phenomenological Fatigue Virtual Test

The fatigue virtual test was performed based on the Basquin relation described in Section 2.5 in the previous chapter:

$$\sigma_a = \sigma'_f (2N_f)^b \tag{3.1}$$

$$N_f = \frac{1}{2} \cdot (\frac{\sigma_a}{\sigma'_f})^{1/b} \tag{3.2}$$

The stress amplitude σ_a was computed by using a process similar to Algorithm 3.4 where a critical average damage $\mathcal{D}^{crit}_{avg}$ is specified followed by a search for the corresponding loading stress. In Equation 3.2 above, there are two variables that can affect the number of cycles to failure N_f: the stress amplitude σ_a and σ'_f. Figure 3.10 below shows two trends using that model: In Figure 3.10 a) σ'_f is fixed and taken from the fracture stress virtual test, which in this case was $\sim 377 \cdot 10^{11}\, Pa$ and the amplitude stress σ_a is varied; in Figure 3.10 b) the amplitude stress was fixed and σ'_f was varied, which is shown as variation in the critical damage criterion shown on the x-axis. In Figure 3.10 a) a fixed σ'_f , the relation between N_f and σ_a is linear on a log-log scale, which is expected given the Basquin equation. In Figure 3.10 b) the variation in the damage criterion $\mathcal{D}_{avg}^{crit}$ leads to different σ_f' that have a somewhat linear relationship to the cycles to failure. The dent in the graph is probably due to the material response during the virtual test that lead to values of σ_f' that were not entirely linear but follow the expected trend of higher cycles with less damaged material.



(a) σ_a vs N_f

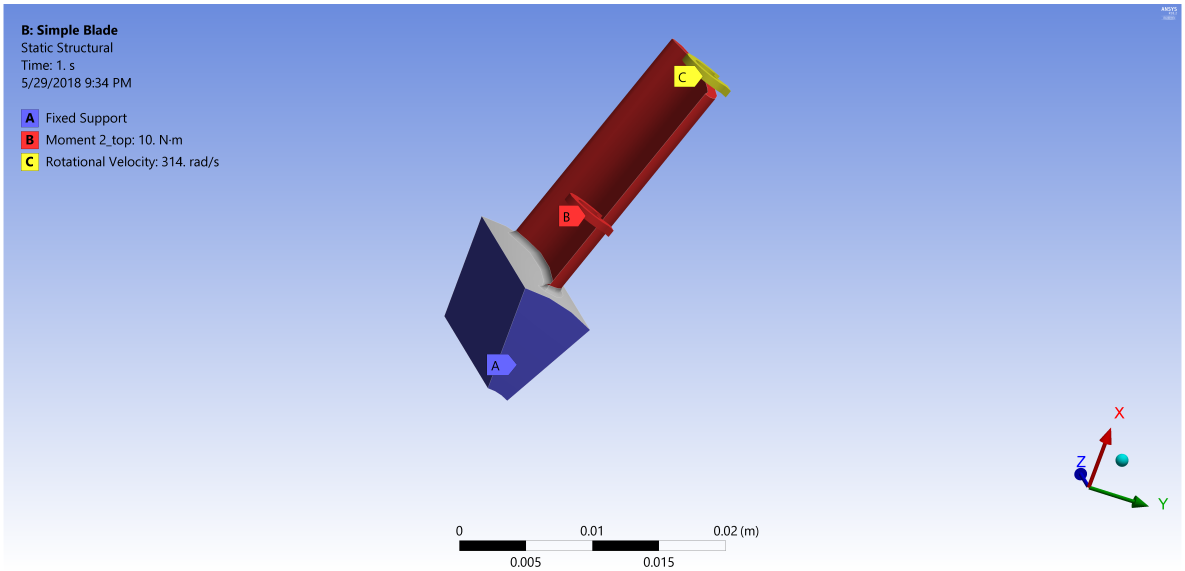
(b) $\mathcal{D}_{avg}^{crit}$ vs N_f

Figure 3.10: Fatigue Virtual Test - Left Figure a) shows relation between stress amplitude σ_a vs N_f given a fixed σ_f' ; Right Figure b) shows relation between $\mathcal{D}_{avg}^{crit}$ (which influences σ_f') vs N_f given a fixed σ_a

3.3.4 Structural Level Simulation

Once the mesoscale properties of the RVE were determined, a new material was created with the effective mechanical and thermal properties in the ANSYS Engineering Data library. The new material properties were imported into the ANSYS Engineering Data library, and then used to construct a turbine blade part with the properties of the CMC designed from the previously described algorithms. The boundary conditions imposed consisted of fixed supports at the base of the blade, and moment boundary conditions at the upper part of the blade to simulate a rotational load the turbine would experience during application, which include:

1. A fixed body at the base of the turbine blade which is attached to the broader turbine system
2. A moment applied at the long part of the blade stemming from the rotational motion of the turbine
3. A rotational velocity applied to the overall blade stemming from the rotational motion of the turbine

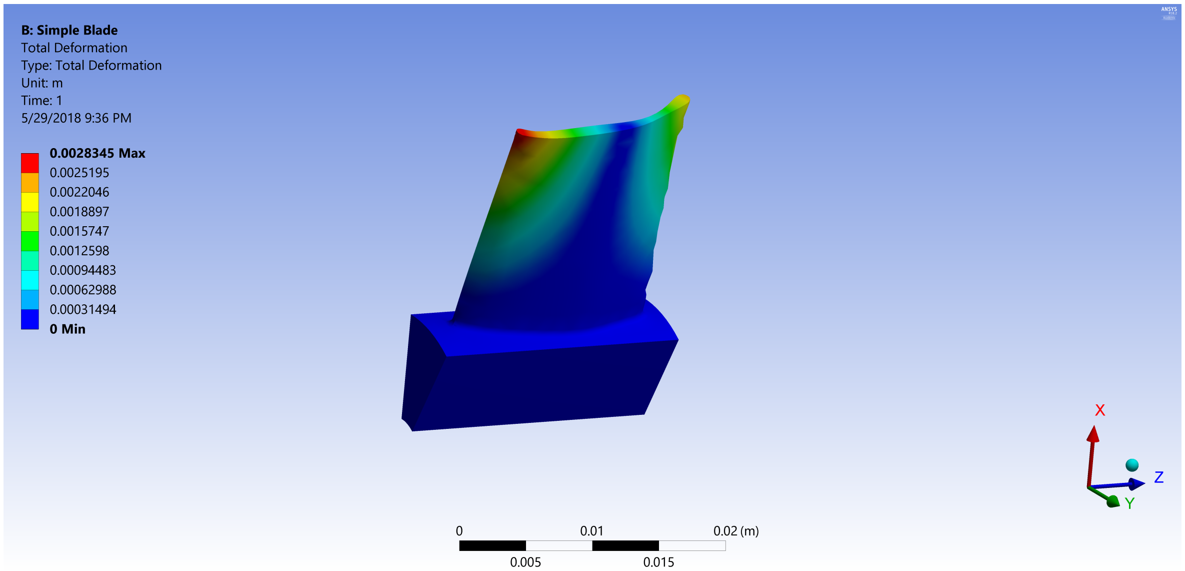


B: Simple Blade
Static Structural
Time: 1. s
5/29/2018 9:34 PM

A Fixed Support
B Moment 2_top: 10. N·m
C Rotational Velocity: 314. rad/s

Figure 3.11: Structural Simulation Boundary Conditions

The simulation yielded the following results for the deformation, strain and stress respectively.



B: Simple Blade
Total Deformation
Type: Total Deformation
Unit: m
Time: 1
5/29/2018 9:36 PM

Figure 3.12: ANSYS Structural Simulation Results - Deformation of the turbine blade geometry from simulation performed using boundary conditions from Figure 3.11

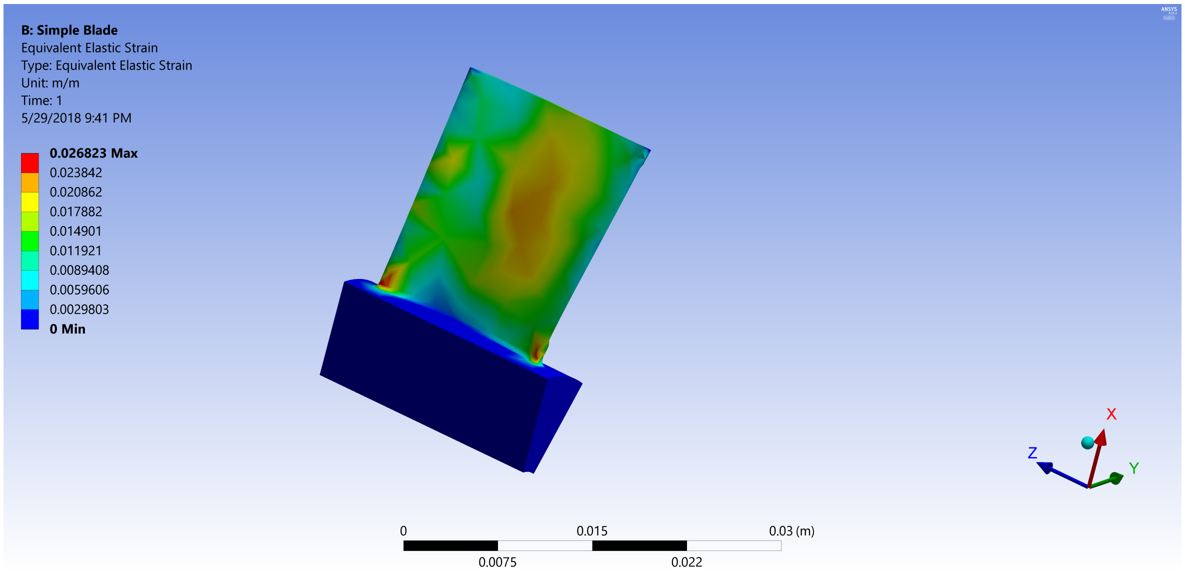


Figure 3.13: ANSYS Structural Simulation Results - Equivalent Strain of the turbine blade geometry from simulation performed using boundary conditions from Figure 3.11

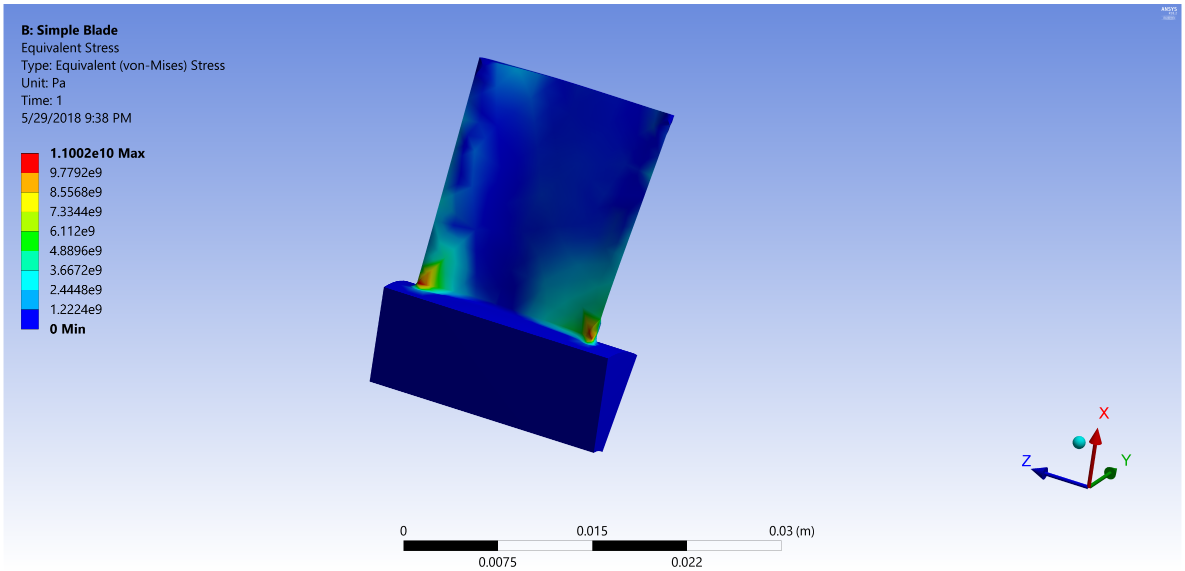


Figure 3.14: ANSYS Structural Simulation Results - Equivalent Stress of the turbine blade geometry from simulation performed using boundary conditions from Figure 3.11

The ANSYS simulation above was performed with triangular meshing of a relatively coarse mesh of $\sim 1,500$ elements. The primary motivation behind the study below is to show that finite element analysis on the structural level can be performed for the material generated with the automated design framework. The results for the deformation, equivalent strain and equivalent stress are within what is generally expected for a simulation with the previously described boundary conditions. The strain and stress accumulate at the connection point between the base and the wing of the blade, while the greatest deformation happens at the edge of the wing. This is expected because the connections points between the based on the wing have less area to distribute and relax the stresses generated by the rotation than other parts of the blade.

3.4 Conclusion

The results of the case study presented herein showcase that the algorithm can create computer generated designs for particle reinforced CMCs. The design process presented spans multiple tiers of materials design, including material selection and mesoscale structure design, as well as various scales of simulation - the mesoscale and the structural scale. The integration of the various simulation scales provides further insight not only into the relevant mechanical and thermal material properties, but also about their behavior in the actual part geometry.

The design framework that was showcased combines attributes from many research fields to create the foundation of a design tool for CMCs. The framework draws from continuum mechanics to simulate the mesoscale behavior, from numerical methods for partial differential equations to resolve the relevant constitutive models using the finite element method, from evolutionary algorithm to provide an automated design framework for material compounds and mesoscale structures. Moreover, the framework was largely built from the first line in a programming sense, meaning that because a lot of the required tools did not exist in the required form for the project, many of them were developed as part of the research. This is particularly true of the mesoscale design tools, where the finite element method based solver was developed from self-developed code, so that it could be integrated into the relevant mesoscale models and the automated design framework. The general framework of the Genetic Algorithm applied was based on previous work by Zohdi [67] but was expanded and adapted for the present purposes.

Given that many of the computational tools of this study were self-developed, significant effort was also put in to improve the computational efficiency of the tools as much as possible. The main improvements of computation times were observed by using a mixed-language program that effectively used the speed of compiled language for fast computation of common operation. Furthermore, time-intensive computations were parallelized using the MPI framework to yield significant improvement in wall-clock time of the computation. This was particularly important in the mesoscale design, as the properties of many mesoscale structures could be computed in parallel to yield significant wall-clock time savings.

For the materials modeling aspect, damage modeling was implemented to simulate the aggregate mechanical response of defects for a given material section. While the ceramic materials making up the matrix and inclusion were simulated to fail in a brittle manner by locally adjusting the damage variable, the overall ductile response of the CMC representative volume element was preserved for the overall simulation. The simulated CMCs constitute two-phase composite materials of two distinct ceramic materials. The computer generated design produced by this study can be used as a basis to guide the design of a set of materials that are extremely costly to fabricate and whose properties and behavior are still largely unknown.

Chapter 4

Future Work

The work presented here lays the foundation for a more integrated framework for computational design of CMCs. The continuum mechanics based approach enabled the connection between mesoscale modeling of a representative volume element of the composite material and the structural scale simulation of a turbine blade geometry. The results of this study showcase that computer generated designs for CMCs can be used to guide the development of future CMC compounds and thereby contribute to rapid and inexpensive prototyping of materials design using computational models.

While this study lays a foundation for computational design of CMCs, there still remains a significant amount of future work to improve the overarching framework as well the modeling methods within the framework. On the materials modeling efforts, there is a need to enhance the model with higher fidelity models of the interphase between the matrix material and the reinforcement material. Furthermore, within the mesoscale modeling efforts there is room to prototype more complex geometries of the reinforcement materials, including various combinations of fibers and particles. These more complex geometries may exhibit more favorable effective properties for the greater composite and can be integrated into the overarching design framework to create more complex computer generated mesoscale structures. The structural scale simulations can be extended to dynamically based simulation that show the evolution of various thermal and mechanical fields over time and allow for a more detailed analysis of the behavior of the material on the structural and engineering part scale. From a software engineering point of view, the overall design framework can be integrated into a closed loop structure, where the mesoscale tools, both the isotropic and microstructural genetic algorithms, and the structural simulation tools are interacting in a dynamic way to share data and improve the computer generated CMC designs.

4.1 Interphase Modeling

The current interphase model of the design framework provides a range of material properties, where perfect bonding between the matrix and reinforcement materials constitute one bound and no bonding between the matrix and reinforcement material provide the other bound. While this computationally efficient method for rapid prototyping of CMC microstructures and computing a range of desired properties, it does not provide information about the effects of the interphase on the greater composite.

Interphase modeling can add tremendous value to the overall modeling efforts of the CMC designs, yet high-fidelity modeling of the interphase mechanics and chemistry may be unfeasible given the rapid prototyping emphasis of the proposed design framework. Therefore, a continuum based model of the interphase is proposed to compromise on the computational requirements required for the model to be reasonably fast while also providing insights into the effects of the interphase on the broader CMC properties. The proposed continuum model of the interphase is based on the Coloumbic friction model introduced by Kuntz and Grathwohl [41]. Here, the interphase is treated a third material component that exists between the matrix and the reinforcement material. The interface material has its own set of mechanical and thermal properties, as well as its own failure criterion for damage modeling. Mechanically, interphase materials of CMC are generally substantially weaker than the matrix or reinforcement materials so that they can trap the crack energy coming from the matrix and thereby enhance the crack deflection to the interphase, which increases the overall toughness of the CMC compound. The damage model proposed for the interphase is similar to the damage model for the matrix and reinforcement material, with the major difference being the method by which the stress at the interface is calculated. The stress at the interface is calculated using a Coloumbic friction coefficient that relates the stress of the neighboring material to the stress at the interface

$$\tau = \zeta_{friction}\sigma \tag{4.1}$$

where σ represents the stress in the matrix material that is normal to the sliding stress τ and $\zeta_{friction}$ is the friction coefficient of the interphase material. A similar damage model can now be applied based on the sliding stress τ. If

$$\|\boldsymbol{\tau}'\| > \tau_{critical} \rightarrow \mathcal{D}_{local} = 0 \tag{4.2}$$

and if

$$\|\boldsymbol{\tau}'\| < \tau_{critical} \rightarrow \mathcal{D}_{local} = 1.0 \tag{4.3}$$

where τ' represents the deviatoric stress inside the material is defined by

$$\tau' = \boldsymbol{\tau} - \frac{tr\boldsymbol{\tau}}{3} \cdot \mathbf{1} \tag{4.4}$$

The friction coefficient $\zeta_{friction}$ can be determined with higher fidelity simulations of the interphase effects that can give a range of values to use for the continuum based model.

4.2 Reinforcement Geometries

Currently, the microstructures that can be generated within the CMC design framework of this study are limited to particle reinforced composites. This creates a limitation of the mesoscale structures that can be explored by the Genetic Algorithms that enable the design framework. Furthermore, many CMC mesoscale structures empty fiber geometries for the reinforcement material, as can be seen in the SEM micrograph in the introduction. The ability to model and generate fiber based composites is therefore a natural extension to the current framework, which also would bring the computer generated designs closer to the reality of the actual CMC mesoscale applied by industrial manufacturers.

The fiber based microstructures can be implemented by modeling the fiber inclusions as cylinders. In that case, the inclusion table has to track the following metrics that define a given cylinder:

	Axis	Center-point	Radius	Positive Height
Cylinder 1	$vector_1^{axis}$	$center_1$	$radius_1$	$height_1^{positive}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Cylinder n	$vector_n^{axis}$	$center_n$	$radius_n$	$height_n^{positive}$

Table 4.1: Cylinder Tracking

where the axis refers to the vector defining the central axis of the cylinder pointing away from the center-point, the center-point is the starting point of the cylinder creation, the radius is radius of the circle perpendicular to the axis vector making up the base of the cylinder, the positive height is the height of the cylinder in the direction of the axis. Cylinder based geometries then allow the creation of CMC microstructures that have various layers and directions of fibers.

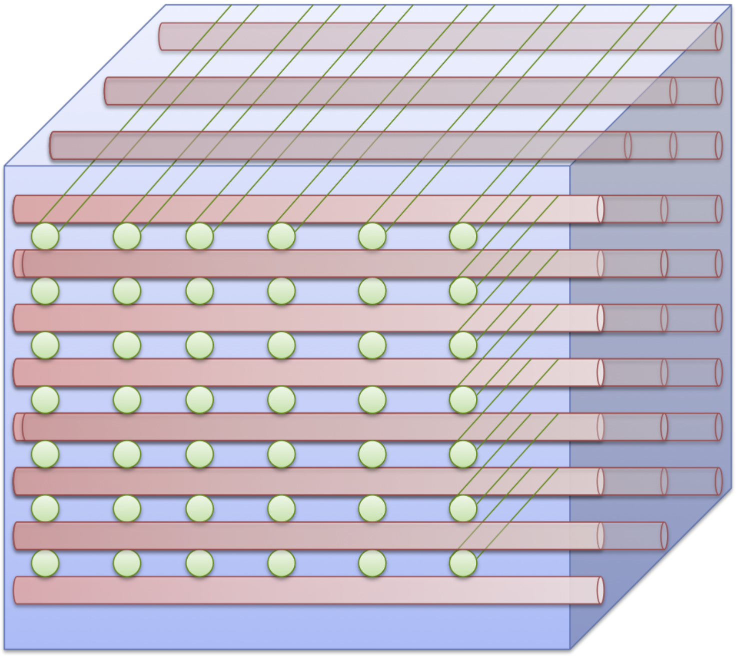


Figure 4.1: Fiber Microstructures Modeled with Cylinders

These mesoscale structures enable the design of CMCs with anisotropic mechanical and thermal properties that may be beneficial for the desired application. Anisotropic heat conduction, for example allows for more effective thermal management, while anisotropic mechanical properties allow the material designer to tune the strength of the material towards the high load areas. The tuning of these anisotropic properties can be performed in conjunction with the structural scale that gives detailed information about the locations and directions of the largest mechanical and thermal stresses.

Another interesting exploration may be to create mesoscale structures that combine both fiber and particle reinforcements, and thereby create hybrid structures that could have very beneficial properties. A sample mesoscale structure that combines fiber and particle reinforcements is shown below [38]

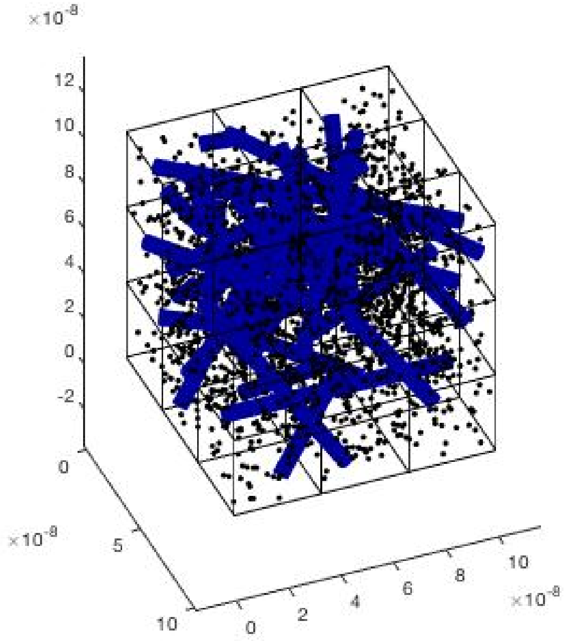


Figure 4.2: Fiber and particle microstructure combination of a representative volume element including both fibers (blue) and particles (black) as reinforcements. The figure aboves shows dense packing of fibers and particles without intersecting bodies. The axes are shown in meters.

This geometric flexibility would allow both the algorithm and the human designers to explore new sets of geometries and find the mesoscale structures that fit the properties and needs that they are looking for. This would be particularly useful for designing fiber based microstructures, since the fiber preform can be set before the infiltration processes create the surrounding matrix. While creating a combination of fiber and particle reinforcements shown in Figure 4.2 above may be very challenging given current manufacturing ability, it might provide helpful guidance for which mesoscale structures the designers should be aiming for.

4.3 Integrated Simulations

While the current framework has the ability to automatically generate CMC compounds for rapid prototyping, many of the simulation steps are still disjointed and can be tied together to develop a more integrated and richer model of the mechanical and thermal properties and behavior of CMC both at the mesoscale and the structural scale. In the ideal framework, all the computations performed at the mesoscale, including the mesoscale virtual tests, would be interacting with the structural scale simulation dynamically. This dynamic interaction would allow the flow of material property information from the homogenized mesoscale to the structural scale simulations, while the structural scale simulation can feed deformation, strain, mechanical and thermal stress data to the mesoscale simulations to better model the properties of the CMC at the conditions of the structural simulations.

The dynamic flow of data can then be integrated into the broader optimization framework to enable richer and more detailed prototyping and computer-generated design of CMC compounds at the mesoscale that behave how the structural scale requires them to. Moreover, under the dynamically integrated computation framework the materials designer could adjust their requirement in real time and observe the resulting CMC constituent material choices and mesoscale structures for the desired design parameters. The real time adjustment would then allow the material designer not only to see various options for different conditions, but also compare further factors that have not been integrated into the simulation of this study, such as the feasibility and cost of fabricating a given CMC compound that is proposed by the algorithm.

4.4 Algorithmic Improvements

The design framework described in this study has many hyperparameters that can be tuned to improve the performance of the optimization of both the isotropic and the mesoscale genetic algorithms. A carefully conducted design of experiments study can investigate many of these hyperparameters to determine the best-performing set for the desired optimization. The set of hyperparameters that can be modified, include the sample size that is used for the algorithm, the mutation rate for each iteration, and the process by which the mutation is performed. Furthermore, dynamic algorithmic improvements can change the mutation rate and mutation parameters to encourage more exploration when the algorithm appears stuck at a local minimum and to drive the arrival at a local minimum with greater momentum once the algorithm is in the vicinity of a local minimum. In addition to improving upon the already existing design framework, different sets of non-convex optimization algorithms can be tested to compare the computer generated designs of various schemes and compare their advantages and disadvantages.

Bibliography

[1] Brayton Cycle - Thermodynamics Notes MIT Open Courseware.

[2] *An Improved Long Life Duration CMC For Jet Aircraft Engine Applications*, volume 4. ASME Turbo Expo 2002: Power for Land, Sea, and Air, 2002.

[3] Siemens Gas Turbine Portfolio, 2017.

[4] U.S. Energy Information Administration. Energy Use for Transportation. Technical report, U.S Department of Energy, 2018.

[5] V. S. Bakunov. High-Temperature Creep in Ceramics. *Refractories and Industrial Ceramics*, 38(11-12):2–6, 1997.

[6] G. Bao and Y Song. Crack Bridging Models for Fiber Composites with Slip-Dependent Interfaces. *J. Mech. Phys. Solids*, 41(9):1425–1444, 1993.

[7] G Bao and Z Suo. Remarks on crack-bridging concepts. *Appl Mech Rev*, 45(8):1992, 1992.

[8] Larry Visser Amy Barnes John Lincoln Barrett Jackson, Aaron Beaber. Oxide-Oxide Ceramic Matrix Composites - Enabling Widespread Industry Adoption. Technical report, 3M, 2016.

[9] M. Basista and W. Weglewski. Modelling of Damage and Fracture in Ceramic Matrix Composites - An Overview. *Journal of Theoretical and Applied Mechanics*, 44(3):455–484, 2006.

[10] P. F. Becher. Crack Bridging Processes in Toughened Ceramics. Technical report, Oak Ridge National Laboratory, 1991.

[11] Seth B. Brij. Superalloys- the Utility Gas Turbine Perspective in Superalloys. *The Minerals, Metals and Materials Society*, page 3, 2000.

[12] William D. Jr. Callister and David G. Rethwisch. *Fundamentals of Materials Science and Engineering.* John Wiley and Sons, 3 edition, 2008.

[13] Carmen Calvo-Jurado and William Parnell. Hashin-Shtrikman bounds on the effective thermal conductivity of a transversely isotropic two-phase composite material. *J Math Chem*, 53:828–843, 2015.

[14] W. Roger Cannon and Terence G. Langdon. Review of Creep of Ceramics. *Journal of Materials Science*, 18, 1983.

[15] C. Chateau, L. Gélébart, M. Bornert, et al. Modeling of damage in unidirectional ceramic matrix composites and multi-scale experimental validation on third generation SiC/SiC minicomposites. *Journal of the Mechanics and Physics of Solids*, 63:298–319, 2014.

[16] David R. Clarke, Matthias Oechsner, and Nitin P. Padture. Thermal-barrier coatings for more efficient gas-turbine engines. *MRS Bulletin*, 37, 2012.

[17] B. N. et al Cox. Stochastic Virtual Tests High-Temperature Ceramic Matrix Composites. *Annu. Rev. Mater. Res.*, 44:479–529, 2014.

[18] Jeff Dahl. Jet Engine System. Online, 2018.

[19] J.A. DiCarlo. SiC/SiC Composites for 1200 C and Above. Technical Report NASA/TM-2004-213048, NASA, 2004.

[20] James A. DiCarlo. Advances in SiC/SiC for Aero-Propulsion. Technical Report TM-2013-217889, NASA, 2013.

[21] et al. Douglas J Kiser. Overview of CMC (Ceramic Matrix Composite) Research at the NASA Glenn Research Center. Technical Report Report Number: GRC-E-DAA-TN31466., National Aeronautics and Space Association (NASA), 2016.

[22] S. Dutta. Fracture Toughness and Reliability in High-Temperature Structural Ceramics and Composites: Prospects and Challenges for the 21st Century. Technical report, NASA.

[23] Robert Eriksson. *Thermal Barrier Coatings; Durability Assessment and Life Prediction.* PhD thesis, Linköping Studies in Science and Technology, 2013.

[24] W. Liu et al. An efficient augmented finite element method for arbitrary cracking and crack interaction in solids. *Int. J. Numer. Meth. Engng*, 99:438–468, 2014.

[25] A. G. Evans, F. W. Zok, and R. M. McMeeking. Fatigue of Ceramic Matrix Composites. *Acta metall. mater*, 43(3):859–875, 1995.

[26] A. G. Evans, F. W. Zok, and R. M. McMeeking. Overview No. 118: Fatigue of Ceramic Matrix Composites. *Acta metall. mater*, 43(3):859–875, 1995.

[27] A.G. Evans and D. B. Marshall. The Mechanical Behavior of Ceramic Matrix Composites. *Aca metall*, 37(10):2567–2583, 1989.

[28] Anthony C. Fischer-Cripps. *Introduction to Contact Mechanics.* Springer US, 2 edition, 2007.

[29] C. A. Floudas and C. E. Gounaris. A review of recent advances in global optimization. *J Glob Optim*, 45:3–28, 2009.

[30] Somnath Ghosh, Masoud Anahid, and Pritam Chakraborty. Modeling Fatigue Crack Nucleation Using Crystal Plasticity Finite Element Simulations and Multi-time Scaling. *Computational Methods for Microstructure-Property Relationships*, 2011.

[31] Joseph E. Grady. CMC Technology Advancements CMC Technology Advancements for Gas Turbine Engine Applications, 2013.

[32] M Grujicic, JS Snipes, R Galgalikar, et al. Multi-length scale derivation of the room-temperature material constitutive model for SiC/SiC ceramic-matrix composites. *Proc IMechE Part L. J Materials: Design and Applications*, 2015.

[33] Mica Grujicic, R. Galgalikar, J.S. Snipes, and S. Ramaswami. Material Constitutive Models for Creep and Rupture of SiC/SiC Ceramic-Matrix Composites (CMCs) Under Multiaxial Loading. *Journal of Materials Engineering and Performance*, 25(5):1697–1708, 2016.

[34] Mica Grujicic, R. Galgalikar, J.S. Snipes, and S. Ramaswami. Multi-length-scale Material Model for SiC/SiC Ceramic-Matrix Composites (CMCs): Inclusion of In-Service Environmental Effects. *Journal of Materials Engineering and Performance*, 25(1):199–219, 2016.

[35] Michael C. Halbig, Martha H. Jaskowiak, James D. Kiser, and Dongming Zhu. Evaluation of Ceramic Matrix Composite Technology for Aircraft Turbine Engine Applications Read More: https://arc.aiaa.org/doi/abs/10.2514/6.2013-539. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Aerospace Sciences Meetings,*, 2013.

[36] Z. Hashin. Analysis of Composite Materials - A Survey. *Journal of Applied Mechanics*, 50(481), 1983.

[37] Z. Hashin and S. Shtrikman. A Variational Approach To The Theory of the Elastic Behavious of Multiphase Materials. *J. Mech. Phys. Solids*, 11:127–140, 1963.

[38] et al. J.Kwon, E. Yildizdag. Designing 3D Printer Functional Materials Using Block Copolymer Nanocomposites. Technical report, UC Berkeley, 2017.

[39] LM Kachanov. *Introduction to continuum damage mechanics.* Springer Science and Business Media, 1986.

[40] T. Kellner. Space Age Ceramic Are Aviation's New Cup of Tea, July 2013.

[41] M Kuntz and G Grathwohl. Crack Resistance of Ceramic Matrix Composites with Coulomb Friction Controlled Interfacial Processes. *Fracture Mechanics of Ceramics*, 12, 1996.

[42] Chai L. and L. R Dharani. Analysis of elastic crack bridging in ceramic matrix composites. *Theoretical and Applied Fracture Mechanics*, 15:105–112, 1991.

[43] Lawrence Livermore National Laboratory. Energy Flow Charts.

[44] Lee S. Langston and George Opdyke. Introduction to Gas Turbines for Non-Engineers. *Global Gas Turbine News*, 37(2), 1997.

[45] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations.* SIAM, 2007.

[46] Dawn Levy. Ceramic matrix composites take flight in LEAP jet engine, 2017.

[47] I. M. Low. *Advances in Ceramic Matrix Composites.* Woodhead Publishing, 2014.

[48] J.B. Min, D. Xue, and Y. Shi. Micromechanics modeling for fatigue damage analysis designed for fabric reinforced ceramic matrix composites. *Composite Structures*, 111:213–223, 2014.

[49] Sumio Murakami. *Continuum Damage Mechanics, Solid Mechanics and Its Applications.* Springer Science and Business Media, 2012.

[50] Roger R. Naslain. SiC-Matrix Composites: Nonbrittle Ceramics for Thermo-Structural Applications. *Int. J. Appl. Ceram. Technol.*, 2(2):755–84, 2005.

[51] NRC. Nuclear Power for Electrical Generation. Technical report, United States Nuclear Regulatory Commission, 2017.

[52] Hisaichi Ohnabe, Shoju Masaki, Masakazu Onozuka, Kaoru Miyahara, and Tadashi Sasa. Potential application of ceramic matrix composites to aero-engine components. *Composites Part A: Applied Science and Manufacturing*, 30(4):489–496, 1999.

[53] Nitin P. Padture, Maurice Gell, and Eric H. Jordan. Thermal Barrier Coatings for Gas-Turbine Engine Applications. *Science*, 296, 2002.

[54] Bavesh Patel and Tarek I. Zohdi. Numerical estimation of effective electromagnetic properties for design of particulate composites. *Materials and Design*, 94:546–553, 2016.

[55] Robert O. Ritchie. Mechanisms of fatigue-crack propagation in ductile and brittle solids. *International Journal of Fracture*, 100:55–83, 1998.

[56] David Royland. Introduction to Fracture Mechanics. Online, June 2001.

[57] Martin Siebart. Breakthrough with 3D printed Gas Turbine Blades, February 2017.

[58] William Smothers and R. W. Rice. *Ceramic Matrix Composite Toughening Mechanisms: An Update.* John Wiley and Sons, 2008.

[59] Bent F Sorensen, E. Kristofer Gamstedt, Rasmus C. Ostergaard, and Stergios Goutianos. Micromechanical model of cross-over fibre bridging - Prediction of mixed mode bridging laws. *Mechanics of Materials*, 40(220-234), 2008.

[60] I. Spitsberg and J. Steibel. Thermal and Environmental Barrier Coatings for SiC/SiC CMCs in Aircraft Engine Applications. *Int. J. Appl. Ceram. Technol*, 1(4):291–301, 2004.

[61] Peter Spittle. Gas turbine technology. *Physics Education*, 38(6), 2003.

[62] Marco Brunelli Stefan Lampenscherf. Siemens AG. Presentation, 2016.

[63] S. Suresh. *Fatigue of Materials.* Cambridge University Press, 1998.

[64] R Talreja. Continuum modelling of damage in ceramic matrix composites. *Mechanics of Materials*, 12:165–180, 1991.

[65] Likun Tan. Multiple Length and Time-scale Approaches in Materials Modeling. *Advances in Materials*, 6(1):1–9, 2017.

[66] Ian Wright. GE Aviation to Industrialize Ceramic Matrix Composites.

[67] Tarek I. Zohdi. Genetic design of solids possessing a random-particulate microstructure. *Phil. Trans. R. Soc. Lond. A*, pages 1021–1043, 2003.

[68] Tarek I. Zohdi. *A Finite Element Primer for Beginners.* Springer Verlag, 2014.

[69] Tarek I. Zohdi. Modeling and simulation of cooling-induced residual stresses in heated particulate mixture depositions in additive manufacturing. *Comput Mech*, 2015.

[70] Tarek I. Zohdi. An explicit macro-micro phase-averaged stress correaltion for particle-enhanced composite materials in loaded structures. *International Journal of Engineering Science*, 109:1–13, 2016.

[71] Tarek I. Zohdi. Modeling and Simulation of Laser Processing of Particulate-Functionalized Materials. *Arch Computat Methods Eng*, 2017.

[72] Tarek I. Zohdi and Peter Wriggers. *An Introduction to Computational Micromechanics*. Springer-Verlag Berlin Heidelberg, 2005.

[73] Frank W. Zok and Carlos G. Levi. Mechanical Properties of Porous-Matrix Ceramic Composites. *Advanced Engineering Materials*, 3(1-2), 2001.

Appendix A

Additional Derivations for Methods

A.1 Gradient of Cost Function

Since it does not occur often that the cost function minimizes to unquestionably low values, one can verify the potential minimum found by the genetic algorithm by checking the gradient of the cost function. If the gradient of the cost function reaches 0, then the genetic algorithm has indeed found a minimum of the cost function. The minimum may not be an absolute minimum, but it does represent an initial result of the optimization procedure.

The cost function is dependent on the different material parameters that are included in the optimization

$$\Pi(\kappa_1, \kappa_2\mu_1, \mu_2, \lambda_1, \lambda_2, \rho_1, \rho_2)$$

therefore the gradient of the cost function is

$$\nabla\Pi(\kappa_1, \kappa_2\mu_1, \mu_2, \lambda_1, \lambda_2, \rho_1, \rho_2) = (\frac{\partial\Pi}{\kappa_1}, \frac{\partial\Pi}{\kappa_2}, \frac{\partial\Pi}{\mu_1}, \frac{\partial\Pi}{\mu_2}, \frac{\partial\Pi}{\lambda_1}, \frac{\partial\Pi}{\lambda_2}, \frac{\partial\Pi}{\rho_1}, \frac{\partial\Pi}{\rho_2}, \frac{\partial\Pi}{\upsilon_2})$$

The partial derivative of of a multi-variable function can be numerically approximated using the following formula

$$\frac{\partial u}{\partial x_1} = \frac{u(x_1 + \Delta x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) - u(x_1 - \Delta x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)}{2\Delta x_1}$$

allowing one to split up the gradient into its components

$$\frac{\partial\Pi}{\partial\kappa_1} = \frac{\partial\Pi}{\partial\kappa^*}\frac{\partial\kappa^*}{\partial\kappa_1} + \frac{\partial\Pi}{\partial\mu^*}\frac{\partial\mu^*}{\partial\kappa_1} + \frac{\partial\Pi}{\partial\bar{C}_\kappa}\frac{\partial\bar{C}_\kappa}{\partial\kappa_1} + \frac{\partial\Pi}{\partial\bar{C}_\mu}\frac{\partial\bar{C}_\mu}{\partial\kappa_1} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\kappa}\frac{\partial\bar{\bar{\bar{C}}}_\kappa}{\partial\kappa_1} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\mu}\frac{\partial\bar{\bar{\bar{C}}}_\mu}{\partial\kappa_1}$$

$$\frac{\partial\Pi}{\partial\kappa_2} = \frac{\partial\Pi}{\partial\kappa^*}\frac{\partial\kappa^*}{\partial\kappa_2} + \frac{\partial\Pi}{\partial\mu^*}\frac{\partial\mu^*}{\partial\kappa_2} + \frac{\partial\Pi}{\partial\bar{C}_\kappa}\frac{\partial\bar{C}_\kappa}{\partial\kappa_2} + \frac{\partial\Pi}{\partial\bar{C}_\mu}\frac{\partial\bar{C}_\mu}{\partial\kappa_2} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\kappa}\frac{\partial\bar{\bar{\bar{C}}}_\kappa}{\partial\kappa_2} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\mu}\frac{\partial\bar{\bar{\bar{C}}}_\mu}{\partial\kappa_2}$$

$$\frac{\partial\Pi}{\partial\mu_1} = \frac{\partial\Pi}{\partial\kappa^*}\frac{\partial\kappa^*}{\partial\mu_1} + \frac{\partial\Pi}{\partial\mu^*}\frac{\partial\mu^*}{\partial\mu_1} + \frac{\partial\Pi}{\partial\bar{C}_\kappa}\frac{\partial\bar{C}_\kappa}{\partial\mu_1} + \frac{\partial\Pi}{\partial\bar{C}_\mu}\frac{\partial\bar{C}_\mu}{\partial\mu_1} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\kappa}\frac{\partial\bar{\bar{\bar{C}}}_\kappa}{\partial\mu_1} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\mu}\frac{\partial\bar{\bar{\bar{C}}}_\mu}{\partial\mu_1}$$

$$\frac{\partial\Pi}{\partial\mu_2} = \frac{\partial\Pi}{\partial\kappa^*}\frac{\partial\kappa^*}{\partial\mu_2} + \frac{\partial\Pi}{\partial\mu^*}\frac{\partial\mu^*}{\partial\mu_2} + \frac{\partial\Pi}{\partial\bar{C}_\kappa}\frac{\partial\bar{C}_\kappa}{\partial\mu_2} + \frac{\partial\Pi}{\partial\bar{C}_\mu}\frac{\partial\bar{C}_\mu}{\partial\mu_2} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\kappa}\frac{\partial\bar{\bar{\bar{C}}}_\kappa}{\partial\mu_2} + \frac{\partial\Pi}{\partial\bar{\bar{\bar{C}}}_\mu}\frac{\partial\bar{\bar{\bar{C}}}_\mu}{\partial\mu_2}$$

$$\frac{\partial\Pi}{\partial\lambda_1} = \frac{\partial\Pi}{\partial\lambda^*}\frac{\partial\lambda^*}{\partial\lambda_1}$$

$$\frac{\partial\Pi}{\partial\lambda_2} = \frac{\partial\Pi}{\partial\lambda^*}\frac{\partial\lambda^*}{\partial\lambda_2}$$

$$\frac{\partial\Pi}{\partial\rho_1} = \frac{\partial\Pi}{\partial\rho^*}\frac{\partial\rho^*}{\partial\rho_1}$$

$$\frac{\partial \Pi}{\partial \rho_2} = \frac{\partial \Pi}{\partial \rho^*}\frac{\partial \rho^*}{\partial \rho_2}$$

$$\frac{\partial \Pi}{\partial v_2} = \frac{\partial \Pi}{\partial \kappa^*}\frac{\partial \kappa^*}{\partial v_2} + \frac{\partial \Pi}{\partial \mu^*}\frac{\partial \mu^*}{\partial v_2} + \frac{\partial \Pi}{\partial \lambda^*}\frac{\partial \lambda^*}{\partial v_2} + \frac{\partial \Pi}{\partial \rho^*}\frac{\partial \rho^*}{\partial v_2} + \frac{\partial \Pi}{\partial \bar{C}_\kappa}\frac{\partial \bar{C}_\kappa}{\partial v_2} + \frac{\partial \Pi}{\partial \bar{C}_\mu}\frac{\partial \bar{C}_\mu}{\partial v_2} + \frac{\partial \Pi}{\partial \bar{\bar{C}}_\kappa}\frac{\partial \bar{\bar{C}}_\kappa}{\partial v_2} + \frac{\partial \Pi}{\partial \bar{\bar{C}}_\mu}\frac{\partial \bar{\bar{C}}_\mu}{\partial v_2}$$

where the effective properties are computed using the previously defined bounds

$$\kappa^* = \phi_\kappa \kappa^{(-)} + (1-\phi_\kappa)\kappa^{(+)}$$

$$\mu^* = \phi_\mu \mu^{(-)} + (1-\phi_\mu)\mu^{(+)}$$
$$\lambda^* = \phi_\lambda \lambda^{(-)} + (1-\phi_\lambda)\lambda^{(+)}$$
$$\rho^* = \phi_\rho \rho^{(-)} + (1-\phi_\rho)\rho^{(+)}$$

with the numerical discretization being plugged into the bounds themselves. Therefore

$$\frac{\partial \Pi}{\partial \kappa_1} = w_1 \frac{\left|\frac{\phi_\kappa \kappa^{(-)}(\kappa_1+\Delta\kappa_1)+(1-\phi_\kappa)\kappa^{(+)}(\kappa_1+\Delta\kappa_1)}{\kappa^{*,D}} - 1\right| - \left|\frac{\phi_\kappa \kappa^{(-)}(\kappa_1-\Delta\kappa_1)+(1-\phi_\kappa)\kappa^{(+)}(\kappa_1-\Delta\kappa_1)}{\kappa^{*,D}} - 1\right|}{2\Delta\kappa_1}$$

$$+w_2 \frac{\left|\frac{\phi_\kappa \mu^{(-)}(\kappa_1+\Delta\kappa_1)+(1-\phi_\kappa)\mu^{(+)}(\kappa_1+\Delta\kappa_1)}{\mu^{*,D}} - 1\right| - \left|\frac{\phi_\kappa \mu^{(-)}(\kappa_1-\Delta\kappa_1)+(1-\phi_\kappa)\mu^{(+)}(\kappa_1-\Delta\kappa_1)}{\mu^{*,D}} - 1\right|}{2\Delta\kappa_1}$$

$$+w_5 \frac{(\left|\frac{((\bar{C}_\kappa(\kappa_1+\Delta\kappa_1)-1)/\bar{C}_\kappa(\kappa_1+\Delta\kappa_1))}{tol_\kappa} - 1\right|) - (\left|\frac{((\bar{C}_\kappa(\kappa_1-\Delta\kappa_1)-1)/\bar{C}_\kappa(\kappa_1-\Delta\kappa_1))}{tol_\kappa} - 1\right|)}{2\Delta\kappa_1}$$

$$+w_6 \frac{(\left|\frac{((\bar{C}_\mu(\kappa_1+\Delta\kappa_1)-1)/\bar{C}_\mu(\kappa_1+\Delta\kappa_1))}{tol_\mu} - 1\right|) - (\left|\frac{((\bar{C}_\mu(\kappa_1-\Delta\kappa_1)-1)/\bar{C}_\mu(\kappa_1-\Delta\kappa_1))}{tol_\mu} - 1\right|)}{2\Delta\kappa_1}$$

$$+w_7 \frac{(\left|\frac{((\bar{\bar{C}}_\kappa(\kappa_1+\Delta\kappa_1)-1)/\bar{\bar{C}}_\kappa(\kappa_1+\Delta\kappa_1))}{tol_\kappa} - 1\right|) - (\left|\frac{((\bar{\bar{C}}_\kappa(\kappa_1-\Delta\kappa_1)-1)/\bar{\bar{C}}_\kappa(\kappa_1-\Delta\kappa_1))}{tol_\kappa} - 1\right|)}{2\Delta\kappa_1}$$

$$+w_8 \frac{(\left|\frac{((\bar{\bar{C}}_\mu(\kappa_1+\Delta\kappa_1)-1)/\bar{\bar{C}}_\mu(\kappa_1+\Delta\kappa_1))}{tol_\mu} - 1\right|) - (\left|\frac{((\bar{\bar{C}}_\mu(\kappa_1-\Delta\kappa_1)-1)/\bar{\bar{C}}_\mu(\kappa_1-\Delta\kappa_1))}{tol_\mu} - 1\right|)}{2\Delta\kappa_1}$$

for the first component

$$\frac{\partial \Pi}{\partial \kappa_2} = w_1 \frac{\left|\frac{\phi_\kappa \kappa^{(-)}(\kappa_2+\Delta\kappa_2)+(1-\phi_\kappa)\kappa^{(+)}(\kappa_2+\Delta\kappa_2)}{\kappa^{*,D}} - 1\right| - \left|\frac{\phi_\kappa \kappa^{(-)}(\kappa_2-\Delta\kappa_2)+(1-\phi_\kappa)\kappa^{(+)}(\kappa_2-\Delta\kappa_2)}{\kappa^{*,D}} - 1\right|}{2\Delta\kappa_2}$$

$$+w_2 \frac{\left|\frac{\phi_\kappa \mu^{(-)}(\kappa_2+\Delta\kappa_2)+(1-\phi_\kappa)\mu^{(+)}(\kappa_2+\Delta\kappa_2)}{\mu^{*,D}} - 1\right| - \left|\frac{\phi_\kappa \mu^{(-)}(\kappa_2-\Delta\kappa_2)+(1-\phi_\kappa)\mu^{(+)}(\kappa_2-\Delta\kappa_2)}{\mu^{*,D}} - 1\right|}{2\Delta\kappa_2}$$

$$+w_5 \frac{(\left|\frac{((\bar{C}_\kappa(\kappa_2+\Delta\kappa_2)-1)/\bar{C}_\kappa(\kappa_2+\Delta\kappa_2))}{tol_\kappa} - 1\right|) - (\left|\frac{((\bar{C}_\kappa(\kappa_2-\Delta\kappa_2)-1)/\bar{C}_\kappa(\kappa_2-\Delta\kappa_2))}{tol_\kappa} - 1\right|)}{2\Delta\kappa_2}$$

$$+w_6\,\frac{(\left|\frac{((\bar{C}_\mu(\kappa_2+\Delta\kappa_2)-1)/\bar{C}_\mu(\kappa_2+\Delta\kappa_2))}{tol_\mu}-1\right|)-(\left|\frac{((\bar{C}_\mu(\kappa_2-\Delta\kappa_2)-1)/\bar{C}_\mu(\kappa_2-\Delta\kappa_2))}{tol_\mu}-1\right|)}{2\Delta\kappa_2}$$

$$+w_7\,\frac{(\left|\frac{((\bar{\bar{C}}_\kappa(\kappa_2+\Delta\kappa_2)-1)/\bar{\bar{C}}_\kappa(\kappa_2+\Delta\kappa_2))}{tol_\kappa}-1\right|)-(\left|\frac{((\bar{\bar{C}}_\kappa(\kappa_2-\Delta\kappa_2)-1)/\bar{\bar{C}}_\kappa(\kappa_2-\Delta\kappa_2))}{tol_\kappa}-1\right|)}{2\Delta\kappa_2}$$

$$+w_8\,\frac{(\left|\frac{((\bar{\bar{C}}_\mu(\kappa_2+\Delta\kappa_2)-1)/\bar{\bar{C}}_\mu(\kappa_2+\Delta\kappa_2))}{tol_\mu}-1\right|)-(\left|\frac{((\bar{\bar{C}}_\mu(\kappa_2-\Delta\kappa_2)-1)/\bar{\bar{C}}_\mu(\kappa_2-\Delta\kappa_2))}{tol_\mu}-1\right|)}{2\Delta\kappa_2}$$

for the second component

$$\frac{\partial\Pi}{\partial\mu_1}=w_1\,\frac{\left|\frac{\phi_\mu\kappa^{(-)}(\mu_1+\Delta\mu_1)+(1-\phi_\mu)\kappa^{(+)}(\mu_1+\Delta\mu_1)}{\kappa^{*,D}}-1\right|-\left|\frac{\phi_\mu\kappa^{(-)}(\mu_1-\Delta\mu_1)+(1-\phi_\mu)\kappa^{(+)}(\mu_1-\Delta\mu_1)}{\kappa^{*,D}}-1\right|}{2\Delta\mu_1}$$

$$+w_2\,\frac{\left|\frac{\phi_\mu\mu^{(-)}(\mu_1+\Delta\mu_1)+(1-\phi_\mu)\mu^{(+)}(\mu_1+\Delta\mu_1)}{\mu^{*,D}}-1\right|-\left|\frac{\phi_\mu\mu^{(-)}(\mu_1-\Delta\mu_1)+(1-\phi_\mu)\mu^{(+)}(\mu_1-\Delta\mu_1)}{\mu^{*,D}}-1\right|}{2\Delta\mu_1}$$

$$+w_5\,\frac{(\left|\frac{((\bar{C}_\kappa(\mu_1+\Delta\mu_1)-1)/\bar{C}_\kappa(\mu_1+\Delta\mu_1))}{tol_\kappa}-1\right|)-(\left|\frac{((\bar{C}_\kappa(\mu_1-\Delta\mu_1)-1)/\bar{C}_\kappa(\mu_1-\Delta\mu_1))}{tol_\kappa}-1\right|)}{2\Delta\mu_1}$$

$$+w_6\,\frac{(\left|\frac{((\bar{C}_\mu(\mu_1+\Delta\mu_1)-1)/\bar{C}_\mu(\mu_1+\Delta\mu_1))}{tol_\mu}-1\right|)-(\left|\frac{((\bar{C}_\mu(\mu_1-\Delta\mu_1)-1)/\bar{C}_\mu(\mu_1-\Delta\mu_1))}{tol_\mu}-1\right|)}{2\Delta\mu_1}$$

$$+w_7\,\frac{(\left|\frac{((\bar{\bar{C}}_\kappa(\mu_1+\Delta\mu_1)-1)/\bar{\bar{C}}_\kappa(\mu_1+\Delta\mu_1))}{tol_\kappa}-1\right|)-(\left|\frac{((\bar{\bar{C}}_\kappa(\mu_1-\Delta\mu_1)-1)/\bar{\bar{C}}_\kappa(\mu_1-\Delta\mu_1))}{tol_\kappa}-1\right|)}{2\Delta\mu_1}$$

$$+w_8\,\frac{(\left|\frac{((\bar{\bar{C}}_\mu(\mu_1+\Delta\mu_1)-1)/\bar{\bar{C}}_\mu(\mu_1+\Delta\mu_1))}{tol_\mu}-1\right|)-(\left|\frac{((\bar{\bar{C}}_\mu(\mu_1-\Delta\mu_1)-1)/\bar{\bar{C}}_\mu(\mu_1-\Delta\mu_1))}{tol_\mu}-1\right|)}{2\Delta\mu_1}$$

for the third component

$$\frac{\partial\Pi}{\partial\mu_2}=w_1\,\frac{\left|\frac{\phi_\mu\kappa^{(-)}(\mu_2+\Delta\mu_2)+(1-\phi_\mu)\kappa^{(+)}(\mu_2+\Delta\mu_2)}{\kappa^{*,D}}-1\right|-\left|\frac{\phi_\mu\kappa^{(-)}(\mu_2-\Delta\mu_2)+(1-\phi_\mu)\kappa^{(+)}(\mu_2-\Delta\mu_2)}{\kappa^{*,D}}-1\right|}{2\Delta\mu_2}$$

$$+w_2\,\frac{\left|\frac{\phi_\mu\mu^{(-)}(\mu_2+\Delta\mu_2)+(1-\phi_\mu)\mu^{(+)}(\mu_2+\Delta\mu_2)}{\mu^{*,D}}-1\right|-\left|\frac{\phi_\mu\mu^{(-)}(\mu_2-\Delta\mu_2)+(1-\phi_\mu)\mu^{(+)}(\mu_2-\Delta\mu_2)}{\mu^{*,D}}-1\right|}{2\Delta\mu_2}$$

$$+w_5\,\frac{(\left|\frac{((\bar{C}_\kappa(\mu_2+\Delta\mu_2)-1)/\bar{C}_\kappa(\mu_2+\Delta\mu_2))}{tol_\kappa}-1\right|)-(\left|\frac{((\bar{C}_\kappa(\mu_2-\Delta\mu_2)-1)/\bar{C}_\kappa(\mu_2-\Delta\mu_2))}{tol_\kappa}-1\right|)}{2\Delta\mu_2}$$

$$+w_6\,\frac{(\left|\frac{((\bar{C}_\mu(\mu_2+\Delta\mu_2)-1)/\bar{C}_\mu(\mu_2+\Delta\mu_2))}{tol_\mu}-1\right|)-(\left|\frac{((\bar{C}_\mu(\mu_2-\Delta\mu_2)-1)/\bar{C}_\mu(\mu_2-\Delta\mu_2))}{tol_\mu}-1\right|)}{2\Delta\mu_2}$$

$$+w_7 \frac{(\left|\frac{((\bar{\bar{C}}_\kappa(\mu_2+\Delta\mu_2)-1)/\bar{\bar{C}}_\kappa(\mu_2+\Delta\mu_2))}{tol_\kappa}-1\right|)-(\left|\frac{((\bar{\bar{C}}_\kappa(\mu_2-\Delta\mu_2)-1)/\bar{\bar{C}}_\kappa(\mu_2-\Delta\mu_2))}{tol_\kappa}-1\right|)}{2\Delta\mu_2}$$

$$+w_8 \frac{(\left|\frac{((\bar{\bar{C}}_\mu(\mu_2+\Delta\mu_2)-1)/\bar{\bar{C}}_\mu(\mu_2+\Delta\mu_2))}{tol_\mu}-1\right|)-(\left|\frac{((\bar{\bar{C}}_\mu(\mu_2-\Delta\mu_2)-1)/\bar{\bar{C}}_\mu(\mu_2-\Delta\mu_2))}{tol_\mu}-1\right|)}{2\Delta\mu_2}$$

for the fourth component

$$\frac{\partial\Pi}{\partial\lambda_1} = w_3 \frac{\left|\frac{\phi_\lambda\lambda^{(-)}(\lambda_1+\Delta\lambda_1)+(1-\phi_\lambda)\lambda^{(+)}(\lambda_1+\Delta\lambda_1)}{\lambda^{*,D}}-1\right|-\left|\frac{\phi_\lambda\lambda^{(-)}(\lambda_1-\Delta\lambda_1)+(1-\phi_\lambda)\lambda^{(+)}(\lambda_1-\Delta\lambda_1)}{\lambda^{*,D}}-1\right|}{2\Delta\lambda_1}$$

for the fifth component

$$\frac{\partial\Pi}{\partial\lambda_2} = w_3 \frac{\left|\frac{\phi_\lambda\lambda^{(-)}(\lambda_2+\Delta\lambda_2)+(1-\phi_\lambda)\lambda^{(+)}(\lambda_2+\Delta\lambda_2)}{\lambda^{*,D}}-1\right|-\left|\frac{\phi_\lambda\lambda^{(-)}(\lambda_2-\Delta\lambda_2)+(1-\phi_\lambda)\lambda^{(+)}(\lambda_2-\Delta\lambda_2)}{\lambda^{*,D}}-1\right|}{2\Delta\lambda_2}$$

for the sixth component

$$\frac{\partial\Pi}{\partial\rho_1} = w_4 \frac{\left|\frac{\phi_\rho\rho^{(-)}(\rho_1+\Delta\rho_1)+(1-\phi_\rho)\rho^{(+)}(\rho_1+\Delta\rho_1)}{\rho^{*,D}}-1\right|-\left|\frac{\phi_\rho\rho^{(-)}(\rho_1-\Delta\rho_1)+(1-\phi_\rho)\rho^{(+)}(\rho_1-\Delta\rho_1)}{\rho^{*,D}}-1\right|}{2\Delta\rho_1}$$

for the seventh component

$$\frac{\partial\Pi}{\partial\rho_2} = w_4 \frac{\left|\frac{\phi_\rho\rho^{(-)}(\rho_2+\Delta\rho_2)+(1-\phi_\rho)\rho^{(+)}(\rho_2+\Delta\rho_2)}{\rho^{*,D}}-1\right|-\left|\frac{\phi_\rho\rho^{(-)}(\rho_2-\Delta\rho_2)+(1-\phi_\rho)\rho^{(+)}(\rho_2-\Delta\rho_2)}{\rho^{*,D}}-1\right|}{2\Delta\rho_2}$$

for the eighth component

$$\frac{\partial\Pi}{\partial v_2} = \frac{\Pi(v_2+\Delta v_2)-\Pi(v_2-\Delta v_2)}{2\Delta v_2}$$

A.2 Effective Elasticity for a Composite Material

The full derivation for computing the elasticity of a composite materials is shown below.

To obtain the elasticity using the above method, six different load states have to be applied

$$\mathcal{E} = \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta \end{bmatrix}, \begin{bmatrix} 0 & \beta & 0 \\ \beta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \beta \\ 0 & \beta & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \beta \\ 0 & 0 & 0 \\ \beta & 0 & 0 \end{bmatrix}$$

where the load state is applied as a strain to the boundary of the RVE using displacement boundary conditions for better stability

$$\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{bmatrix} \beta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

to yield Dirichlet Boundary Conditions on a specified boundary surface. Following post-processing of the FEM solution, each of the load states will set up the following system of equations

$$\begin{Bmatrix} \langle \sigma_{11} \rangle_\Omega \\ \langle \sigma_{22} \rangle_\Omega \\ \langle \sigma_{33} \rangle_\Omega \\ \langle \sigma_{12} \rangle_\Omega \\ \langle \sigma_{23} \rangle_\Omega \\ \langle \sigma_{13} \rangle_\Omega \end{Bmatrix} = \begin{bmatrix} E^*_{1111} & E^*_{1122} & E^*_{1133} & E^*_{1112} & E^*_{1123} & E^*_{1113} \\ E^*_{2211} & E^*_{2222} & E^*_{2233} & E^*_{2212} & E^*_{2223} & E^*_{2213} \\ E^*_{3311} & E^*_{3322} & E^*_{3333} & E^*_{3312} & E^*_{3323} & E^*_{3313} \\ E^*_{1211} & E^*_{1222} & E^*_{1233} & E^*_{1212} & E^*_{1223} & E^*_{1213} \\ E^*_{2311} & E^*_{2322} & E^*_{2333} & E^*_{2312} & E^*_{2323} & E^*_{2313} \\ E^*_{1311} & E^*_{1322} & E^*_{1333} & E^*_{1312} & E^*_{1323} & E^*_{1313} \end{bmatrix} \begin{Bmatrix} \langle \epsilon_{11} \rangle_\Omega \\ \langle \epsilon_{22} \rangle_\Omega \\ \langle \epsilon_{33} \rangle_\Omega \\ 2\langle \epsilon_{12} \rangle_\Omega \\ 2\langle \epsilon_{23} \rangle_\Omega \\ 2\langle \epsilon_{13} \rangle_\Omega \end{Bmatrix}$$

where the left-hand side of the equation is the imposed load state and the right-hand side of the equation can be obtained using the average strain theorem $\langle \epsilon \rangle = \frac{1}{2|\Omega|} \int_\Omega (\nabla u + (\nabla u)^T) d\Omega$.

This will set up a system of 6 equations for each load state, leading to 36 total equations for the 36 unknowns in the elasticity tensor. The system of equations will looks as follows:

Load State 1 (1,1)

$$\begin{aligned} \langle \sigma_{11} \rangle_\Omega^{11} &= E^*_{1111} \langle \epsilon_{11} \rangle_\Omega^{11} + E^*_{1122} \langle \epsilon_{22} \rangle_\Omega^{11} + E^*_{1133} \langle \epsilon_{33} \rangle_\Omega^{11} \\ &+ E^*_{1112}(2\langle \epsilon_{12} \rangle_\Omega^{11}) + E^*_{1123}(2\langle \epsilon_{23} \rangle_\Omega^{11}) + E^*_{1113}(2\langle \epsilon_{13} \rangle_\Omega^{11}) \\ \langle \sigma_{22} \rangle_\Omega^{11} &= E^*_{2211} \langle \epsilon_{11} \rangle_\Omega^{11} + E^*_{2222} \langle \epsilon_{22} \rangle_\Omega^{11} + E^*_{2233} \langle \epsilon_{33} \rangle_\Omega^{11} \\ &+ E^*_{2212}(2\langle \epsilon_{12} \rangle_\Omega^{11}) + E^*_{2223}(2\langle \epsilon_{23} \rangle_\Omega^{11}) + E^*_{2213}(2\langle \epsilon_{13} \rangle_\Omega^{11}) \\ \langle \sigma_{33} \rangle_\Omega^{11} &= E^*_{3311} \langle \epsilon_{11} \rangle_\Omega^{11} + E^*_{3322} \langle \epsilon_{22} \rangle_\Omega^{11} + E^*_{3333} \langle \epsilon_{33} \rangle_\Omega^{11} \\ &+ E^*_{3312}(2\langle \epsilon_{12} \rangle_\Omega^{11}) + E^*_{3323}(2\langle \epsilon_{23} \rangle_\Omega^{11}) + E^*_{3313}(2\langle \epsilon_{13} \rangle_\Omega^{11}) \\ \langle \sigma_{12} \rangle_\Omega^{11} &= E^*_{1211} \langle \epsilon_{11} \rangle_\Omega^{11} + E^*_{1222} \langle \epsilon_{22} \rangle_\Omega^{11} + E^*_{1233} \langle \epsilon_{33} \rangle_\Omega^{11} \\ &+ E^*_{1212}(2\langle \epsilon_{12} \rangle_\Omega^{11}) + E^*_{1223}(2\langle \epsilon_{23} \rangle_\Omega^{11}) + E^*_{1213}(2\langle \epsilon_{13} \rangle_\Omega^{11}) \\ \langle \sigma_{23} \rangle_\Omega^{11} &= E^*_{2311} \langle \epsilon_{11} \rangle_\Omega^{11} + E^*_{2322} \langle \epsilon_{22} \rangle_\Omega^{11} + E^*_{2333} \langle \epsilon_{33} \rangle_\Omega^{11} \\ &+ E^*_{2312}(2\langle \epsilon_{12} \rangle_\Omega^{11}) + E^*_{2323}(2\langle \epsilon_{23} \rangle_\Omega^{11}) + E^*_{2313}(2\langle \epsilon_{13} \rangle_\Omega^{11}) \\ \langle \sigma_{13} \rangle_\Omega^{11} &= E^*_{1311} \langle \epsilon_{11} \rangle_\Omega^{11} + E^*_{1322} \langle \epsilon_{22} \rangle_\Omega^{11} + E^*_{1333} \langle \epsilon_{33} \rangle_\Omega^{11} \\ &+ E^*_{1312}(2\langle \epsilon_{12} \rangle_\Omega^{11}) + E^*_{1323}(2\langle \epsilon_{23} \rangle_\Omega^{11}) + E^*_{1313}(2\langle \epsilon_{13} \rangle_\Omega^{11}) \end{aligned}$$

Load State 2 (2,2)

$$\langle\sigma_{11}\rangle_{\Omega}^{22} = E^{*}_{1111}\langle\epsilon_{11}\rangle_{\Omega}^{22} + E^{*}_{1122}\langle\epsilon_{22}\rangle_{\Omega}^{22} + E^{*}_{1133}\langle\epsilon_{33}\rangle_{\Omega}^{22}$$

$$+E^{*}_{1112}(2\langle\epsilon_{12}\rangle_{\Omega}^{22}) + E^{*}_{1123}(2\langle\epsilon_{23}\rangle_{\Omega}^{22}) + E^{*}_{1113}(2\langle\epsilon_{13}\rangle_{\Omega}^{22})$$

$$\langle\sigma_{22}\rangle_{\Omega}^{22} = E^{*}_{2211}\langle\epsilon_{11}\rangle_{\Omega}^{22} + E^{*}_{2222}\langle\epsilon_{22}\rangle_{\Omega}^{22} + E^{*}_{2233}\langle\epsilon_{33}\rangle_{\Omega}^{22}$$

$$+E^{*}_{2212}(2\langle\epsilon_{12}\rangle_{\Omega}^{22}) + E^{*}_{2223}(2\langle\epsilon_{23}\rangle_{\Omega}^{22}) + E^{*}_{2213}(2\langle\epsilon_{13}\rangle_{\Omega}^{22})$$

$$\langle\sigma_{33}\rangle_{\Omega}^{22} = E^{*}_{3311}\langle\epsilon_{11}\rangle_{\Omega}^{22} + E^{*}_{3322}\langle\epsilon_{22}\rangle_{\Omega}^{22} + E^{*}_{3333}\langle\epsilon_{33}\rangle_{\Omega}^{22}$$

$$+E^{*}_{3312}(2\langle\epsilon_{12}\rangle_{\Omega}^{22}) + E^{*}_{3323}(2\langle\epsilon_{23}\rangle_{\Omega}^{22}) + E^{*}_{3313}(2\langle\epsilon_{13}\rangle_{\Omega}^{22})$$

$$\langle\sigma_{12}\rangle_{\Omega}^{22} = E^{*}_{1211}\langle\epsilon_{11}\rangle_{\Omega}^{22} + E^{*}_{1222}\langle\epsilon_{22}\rangle_{\Omega}^{22} + E^{*}_{1233}\langle\epsilon_{33}\rangle_{\Omega}^{22}$$

$$+E^{*}_{1212}(2\langle\epsilon_{12}\rangle_{\Omega}^{22}) + E^{*}_{1223}(2\langle\epsilon_{23}\rangle_{\Omega}^{22}) + E^{*}_{1213}(2\langle\epsilon_{13}\rangle_{\Omega}^{22})$$

$$\langle\sigma_{23}\rangle_{\Omega}^{22} = E^{*}_{2311}\langle\epsilon_{11}\rangle_{\Omega}^{22} + E^{*}_{2322}\langle\epsilon_{22}\rangle_{\Omega}^{22} + E^{*}_{2333}\langle\epsilon_{33}\rangle_{\Omega}^{22}$$

$$+E^{*}_{2312}(2\langle\epsilon_{12}\rangle_{\Omega}^{22}) + E^{*}_{2323}(2\langle\epsilon_{23}\rangle_{\Omega}^{22}) + E^{*}_{2313}(2\langle\epsilon_{13}\rangle_{\Omega}^{22})$$

$$\langle\sigma_{13}\rangle_{\Omega}^{22} = E^{*}_{1311}\langle\epsilon_{11}\rangle_{\Omega}^{22} + E^{*}_{1322}\langle\epsilon_{22}\rangle_{\Omega}^{22} + E^{*}_{1333}\langle\epsilon_{33}\rangle_{\Omega}^{22}$$

$$+E^{*}_{1312}(2\langle\epsilon_{12}\rangle_{\Omega}^{22}) + E^{*}_{1323}(2\langle\epsilon_{23}\rangle_{\Omega}^{22}) + E^{*}_{1313}(2\langle\epsilon_{13}\rangle_{\Omega}^{22})$$

Load State 3 (3,3)

$$\langle\sigma_{11}\rangle_{\Omega}^{33} = E^{*}_{1111}\langle\epsilon_{11}\rangle_{\Omega}^{33} + E^{*}_{1122}\langle\epsilon_{22}\rangle_{\Omega}^{33} + E^{*}_{1133}\langle\epsilon_{33}\rangle_{\Omega}^{33}$$

$$+E^{*}_{1112}(2\langle\epsilon_{12}\rangle_{\Omega}^{33}) + E^{*}_{1123}(2\langle\epsilon_{23}\rangle_{\Omega}^{33}) + E^{*}_{1113}(2\langle\epsilon_{13}\rangle_{\Omega}^{33})$$

$$\langle\sigma_{22}\rangle_{\Omega}^{33} = E^{*}_{2211}\langle\epsilon_{11}\rangle_{\Omega}^{33} + E^{*}_{2222}\langle\epsilon_{22}\rangle_{\Omega}^{33} + E^{*}_{2233}\langle\epsilon_{33}\rangle_{\Omega}^{33}$$

$$+E^{*}_{2212}(2\langle\epsilon_{12}\rangle_{\Omega}^{33}) + E^{*}_{2223}(2\langle\epsilon_{23}\rangle_{\Omega}^{33}) + E^{*}_{2213}(2\langle\epsilon_{13}\rangle_{\Omega}^{33})$$

$$\langle\sigma_{33}\rangle_{\Omega}^{33} = E^{*}_{3311}\langle\epsilon_{11}\rangle_{\Omega}^{33} + E^{*}_{3322}\langle\epsilon_{22}\rangle_{\Omega}^{33} + E^{*}_{3333}\langle\epsilon_{33}\rangle_{\Omega}^{33}$$

$$+E^{*}_{3312}(2\langle\epsilon_{12}\rangle_{\Omega}^{33}) + E^{*}_{3323}(2\langle\epsilon_{23}\rangle_{\Omega}^{33}) + E^{*}_{3313}(2\langle\epsilon_{13}\rangle_{\Omega}^{33})$$

$$\langle\sigma_{12}\rangle_{\Omega}^{33} = E^{*}_{1211}\langle\epsilon_{11}\rangle_{\Omega}^{33} + E^{*}_{1222}\langle\epsilon_{22}\rangle_{\Omega}^{33} + E^{*}_{1233}\langle\epsilon_{33}\rangle_{\Omega}^{33}$$

$$+E^{*}_{1212}(2\langle\epsilon_{12}\rangle_{\Omega}^{33}) + E^{*}_{1223}(2\langle\epsilon_{23}\rangle_{\Omega}^{33}) + E^{*}_{1213}(2\langle\epsilon_{13}\rangle_{\Omega}^{33})$$

$$\langle\sigma_{23}\rangle_{\Omega}^{33} = E^{*}_{2311}\langle\epsilon_{11}\rangle_{\Omega}^{33} + E^{*}_{2322}\langle\epsilon_{22}\rangle_{\Omega}^{33} + E^{*}_{2333}\langle\epsilon_{33}\rangle_{\Omega}^{33}$$

$$+E^{*}_{2312}(2\langle\epsilon_{12}\rangle_{\Omega}^{33}) + E^{*}_{2323}(2\langle\epsilon_{23}\rangle_{\Omega}^{33}) + E^{*}_{2313}(2\langle\epsilon_{13}\rangle_{\Omega}^{33})$$

$$\langle\sigma_{13}\rangle_{\Omega}^{33} = E^{*}_{1311}\langle\epsilon_{11}\rangle_{\Omega}^{33} + E^{*}_{1322}\langle\epsilon_{22}\rangle_{\Omega}^{33} + E^{*}_{1333}\langle\epsilon_{33}\rangle_{\Omega}^{33}$$

$$+E^{*}_{1312}(2\langle\epsilon_{12}\rangle_{\Omega}^{33}) + E^{*}_{1323}(2\langle\epsilon_{23}\rangle_{\Omega}^{33}) + E^{*}_{1313}(2\langle\epsilon_{13}\rangle_{\Omega}^{33})$$

Load State 4 (1,2)

$$\langle\sigma_{11}\rangle_{\Omega}^{12} = E^{*}_{1111}\langle\epsilon_{11}\rangle_{\Omega}^{12} + E^{*}_{1122}\langle\epsilon_{22}\rangle_{\Omega}^{12} + E^{*}_{1133}\langle\epsilon_{33}\rangle_{\Omega}^{12}$$

$$+E^*_{1112}(2\left\langle \epsilon_{12}\right\rangle^{12}_{\Omega}) + E^*_{1123}(2\left\langle \epsilon_{23}\right\rangle^{12}_{\Omega}) + E^*_{1113}(2\left\langle \epsilon_{13}\right\rangle^{12}_{\Omega})$$

$$\left\langle \sigma_{22}\right\rangle^{12}_{\Omega} = E^*_{2211}\left\langle \epsilon_{11}\right\rangle^{12}_{\Omega} + E^*_{2222}\left\langle \epsilon_{22}\right\rangle^{12}_{\Omega} + E^*_{2233}\left\langle \epsilon_{33}\right\rangle^{12}_{\Omega}$$

$$+E^*_{2212}(2\left\langle \epsilon_{12}\right\rangle^{12}_{\Omega}) + E^*_{2223}(2\left\langle \epsilon_{23}\right\rangle^{12}_{\Omega}) + E^*_{2213}(2\left\langle \epsilon_{13}\right\rangle^{12}_{\Omega})$$

$$\left\langle \sigma_{33}\right\rangle^{12}_{\Omega} = E^*_{3311}\left\langle \epsilon_{11}\right\rangle^{12}_{\Omega} + E^*_{3322}\left\langle \epsilon_{22}\right\rangle^{12}_{\Omega} + E^*_{3333}\left\langle \epsilon_{33}\right\rangle^{12}_{\Omega}$$

$$+E^*_{3312}(2\left\langle \epsilon_{12}\right\rangle^{12}_{\Omega}) + E^*_{3323}(2\left\langle \epsilon_{23}\right\rangle^{12}_{\Omega}) + E^*_{3313}(2\left\langle \epsilon_{13}\right\rangle^{12}_{\Omega})$$

$$\left\langle \sigma_{12}\right\rangle^{12}_{\Omega} = E^*_{1211}\left\langle \epsilon_{11}\right\rangle^{12}_{\Omega} + E^*_{1222}\left\langle \epsilon_{22}\right\rangle^{12}_{\Omega} + E^*_{1233}\left\langle \epsilon_{33}\right\rangle^{12}_{\Omega}$$

$$+E^*_{1212}(2\left\langle \epsilon_{12}\right\rangle^{12}_{\Omega}) + E^*_{1223}(2\left\langle \epsilon_{23}\right\rangle^{12}_{\Omega}) + E^*_{1213}(2\left\langle \epsilon_{13}\right\rangle^{12}_{\Omega})$$

$$\left\langle \sigma_{23}\right\rangle^{12}_{\Omega} = E^*_{2311}\left\langle \epsilon_{11}\right\rangle^{12}_{\Omega} + E^*_{2322}\left\langle \epsilon_{22}\right\rangle^{12}_{\Omega} + E^*_{2333}\left\langle \epsilon_{33}\right\rangle^{12}_{\Omega}$$

$$+E^*_{2312}(2\left\langle \epsilon_{12}\right\rangle^{12}_{\Omega}) + E^*_{2323}(2\left\langle \epsilon_{23}\right\rangle^{12}_{\Omega}) + E^*_{2313}(2\left\langle \epsilon_{13}\right\rangle^{12}_{\Omega})$$

$$\left\langle \sigma_{13}\right\rangle^{12}_{\Omega} = E^*_{1311}\left\langle \epsilon_{11}\right\rangle^{12}_{\Omega} + E^*_{1322}\left\langle \epsilon_{22}\right\rangle^{12}_{\Omega} + E^*_{1333}\left\langle \epsilon_{33}\right\rangle^{12}_{\Omega}$$

$$+E^*_{1312}(2\left\langle \epsilon_{12}\right\rangle^{12}_{\Omega}) + E^*_{1323}(2\left\langle \epsilon_{23}\right\rangle^{12}_{\Omega}) + E^*_{1313}(2\left\langle \epsilon_{13}\right\rangle^{12}_{\Omega})$$

Load State 5 (2,3)

$$\left\langle \sigma_{11}\right\rangle^{23}_{\Omega} = E^*_{1111}\left\langle \epsilon_{11}\right\rangle^{23}_{\Omega} + E^*_{1122}\left\langle \epsilon_{22}\right\rangle^{23}_{\Omega} + E^*_{1133}\left\langle \epsilon_{33}\right\rangle^{23}_{\Omega}$$

$$+E^*_{1112}(2\left\langle \epsilon_{12}\right\rangle^{23}_{\Omega}) + E^*_{1123}(2\left\langle \epsilon_{23}\right\rangle^{23}_{\Omega}) + E^*_{1113}(2\left\langle \epsilon_{13}\right\rangle^{23}_{\Omega})$$

$$\left\langle \sigma_{22}\right\rangle^{23}_{\Omega} = E^*_{2211}\left\langle \epsilon_{11}\right\rangle^{23}_{\Omega} + E^*_{2222}\left\langle \epsilon_{22}\right\rangle^{23}_{\Omega} + E^*_{2233}\left\langle \epsilon_{33}\right\rangle^{23}_{\Omega}$$

$$+E^*_{2212}(2\left\langle \epsilon_{12}\right\rangle^{23}_{\Omega}) + E^*_{2223}(2\left\langle \epsilon_{23}\right\rangle^{23}_{\Omega}) + E^*_{2213}(2\left\langle \epsilon_{13}\right\rangle^{23}_{\Omega})$$

$$\left\langle \sigma_{33}\right\rangle^{23}_{\Omega} = E^*_{3311}\left\langle \epsilon_{11}\right\rangle^{23}_{\Omega} + E^*_{3322}\left\langle \epsilon_{22}\right\rangle^{23}_{\Omega} + E^*_{3333}\left\langle \epsilon_{33}\right\rangle^{23}_{\Omega}$$

$$+E^*_{3312}(2\left\langle \epsilon_{12}\right\rangle^{23}_{\Omega}) + E^*_{3323}(2\left\langle \epsilon_{23}\right\rangle^{23}_{\Omega}) + E^*_{3313}(2\left\langle \epsilon_{13}\right\rangle^{23}_{\Omega})$$

$$\left\langle \sigma_{12}\right\rangle^{23}_{\Omega} = E^*_{1211}\left\langle \epsilon_{11}\right\rangle^{23}_{\Omega} + E^*_{1222}\left\langle \epsilon_{22}\right\rangle^{23}_{\Omega} + E^*_{1233}\left\langle \epsilon_{33}\right\rangle^{23}_{\Omega}$$

$$+E^*_{1212}(2\left\langle \epsilon_{12}\right\rangle^{23}_{\Omega}) + E^*_{1223}(2\left\langle \epsilon_{23}\right\rangle^{23}_{\Omega}) + E^*_{1213}(2\left\langle \epsilon_{13}\right\rangle^{23}_{\Omega})$$

$$\left\langle \sigma_{23}\right\rangle^{23}_{\Omega} = E^*_{2311}\left\langle \epsilon_{11}\right\rangle^{23}_{\Omega} + E^*_{2322}\left\langle \epsilon_{22}\right\rangle^{23}_{\Omega} + E^*_{2333}\left\langle \epsilon_{33}\right\rangle^{23}_{\Omega}$$

$$+E^*_{2312}(2\left\langle \epsilon_{12}\right\rangle^{23}_{\Omega}) + E^*_{2323}(2\left\langle \epsilon_{23}\right\rangle^{23}_{\Omega}) + E^*_{2313}(2\left\langle \epsilon_{13}\right\rangle^{23}_{\Omega})$$

$$\left\langle \sigma_{13}\right\rangle^{23}_{\Omega} = E^*_{1311}\left\langle \epsilon_{11}\right\rangle^{23}_{\Omega} + E^*_{1322}\left\langle \epsilon_{22}\right\rangle^{23}_{\Omega} + E^*_{1333}\left\langle \epsilon_{33}\right\rangle^{23}_{\Omega}$$

$$+E^*_{1312}(2\left\langle \epsilon_{12}\right\rangle^{23}_{\Omega}) + E^*_{1323}(2\left\langle \epsilon_{23}\right\rangle^{23}_{\Omega}) + E^*_{1313}(2\left\langle \epsilon_{13}\right\rangle^{23}_{\Omega})$$

Load State 6 (1,3)

$$\left\langle \sigma_{11}\right\rangle^{13}_{\Omega} = E^*_{1111}\left\langle \epsilon_{11}\right\rangle^{13}_{\Omega} + E^*_{1122}\left\langle \epsilon_{22}\right\rangle^{13}_{\Omega} + E^*_{1133}\left\langle \epsilon_{33}\right\rangle^{13}_{\Omega}$$

$$+E^*_{1112}(2\left\langle \epsilon_{12}\right\rangle^{13}_{\Omega}) + E^*_{1123}(2\left\langle \epsilon_{23}\right\rangle^{13}_{\Omega}) + E^*_{1113}(2\left\langle \epsilon_{13}\right\rangle^{13}_{\Omega})$$

$$\left\langle \sigma_{22}\right\rangle^{13}_{\Omega} = E^*_{2211}\left\langle \epsilon_{11}\right\rangle^{13}_{\Omega} + E^*_{2222}\left\langle \epsilon_{22}\right\rangle^{13}_{\Omega} + E^*_{2233}\left\langle \epsilon_{33}\right\rangle^{13}_{\Omega}$$

$$+E^*_{2212}(2\left\langle \epsilon_{12}\right\rangle^{13}_{\Omega}) + E^*_{2223}(2\left\langle \epsilon_{23}\right\rangle^{13}_{\Omega}) + E^*_{2213}(2\left\langle \epsilon_{13}\right\rangle^{13}_{\Omega})$$

$$\langle\sigma_{33}\rangle_{\Omega}^{13} = E^*_{3311}\langle\epsilon_{11}\rangle_{\Omega}^{13} + E^*_{3322}\langle\epsilon_{22}\rangle_{\Omega}^{13} + E^*_{3333}\langle\epsilon_{33}\rangle_{\Omega}^{13}$$
$$+E^*_{3312}(2\langle\epsilon_{12}\rangle_{\Omega}^{13}) + E^*_{3323}(2\langle\epsilon_{23}\rangle_{\Omega}^{13}) + E^*_{3313}(2\langle\epsilon_{13}\rangle_{\Omega}^{13})$$
$$\langle\sigma_{12}\rangle_{\Omega}^{13} = E^*_{1211}\langle\epsilon_{11}\rangle_{\Omega}^{13} + E^*_{1222}\langle\epsilon_{22}\rangle_{\Omega}^{13} + E^*_{1233}\langle\epsilon_{33}\rangle_{\Omega}^{13}$$
$$+E^*_{1212}(2\langle\epsilon_{12}\rangle_{\Omega}^{13}) + E^*_{1223}(2\langle\epsilon_{23}\rangle_{\Omega}^{13}) + E^*_{1213}(2\langle\epsilon_{13}\rangle_{\Omega}^{13})$$
$$\langle\sigma_{23}\rangle_{\Omega}^{13} = E^*_{2311}\langle\epsilon_{11}\rangle_{\Omega}^{13} + E^*_{2322}\langle\epsilon_{22}\rangle_{\Omega}^{13} + E^*_{2333}\langle\epsilon_{33}\rangle_{\Omega}^{13}$$
$$+E^*_{2312}(2\langle\epsilon_{12}\rangle_{\Omega}^{13}) + E^*_{2323}(2\langle\epsilon_{23}\rangle_{\Omega}^{13}) + E^*_{2313}(2\langle\epsilon_{13}\rangle_{\Omega}^{13})$$
$$\langle\sigma_{13}\rangle_{\Omega}^{13} = E^*_{1311}\langle\epsilon_{11}\rangle_{\Omega}^{13} + E^*_{1322}\langle\epsilon_{22}\rangle_{\Omega}^{13} + E^*_{1333}\langle\epsilon_{33}\rangle_{\Omega}^{13}$$

$$+E^*_{1312}(2\langle\epsilon_{12}\rangle_{\Omega}^{13}) + E^*_{1323}(2\langle\epsilon_{23}\rangle_{\Omega}^{13}) + E^*_{1313}(2\langle\epsilon_{13}\rangle_{\Omega}^{13})$$

which can be rearranged to the following matrix system in the form $\mathbf{A}x = b$.

$$\begin{bmatrix} \langle\epsilon_{11}\rangle_{\Omega}^{11} & \langle\epsilon_{22}\rangle_{\Omega}^{11} & \langle\epsilon_{33}\rangle_{\Omega}^{11} & \langle\epsilon_{12}\rangle_{\Omega}^{11} & \langle\epsilon_{23}\rangle_{\Omega}^{11} & \langle\epsilon_{13}\rangle_{\Omega}^{11} & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \langle\epsilon_{11}\rangle_{\Omega}^{11} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ \langle\epsilon_{11}\rangle_{\Omega}^{22} & \langle\epsilon_{22}\rangle_{\Omega}^{22} & \langle\epsilon_{33}\rangle_{\Omega}^{22} & \langle\epsilon_{12}\rangle_{\Omega}^{22} & \langle\epsilon_{23}\rangle_{\Omega}^{22} & \langle\epsilon_{13}\rangle_{\Omega}^{22} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \begin{Bmatrix} E^*_{1111} \\ E^*_{1122} \\ E^*_{1133} \\ E^*_{1112} \\ \vdots \\ E^*_{1313} \end{Bmatrix} = \begin{Bmatrix} \langle\sigma_{11}\rangle_{\Omega}^{11} \\ \langle\sigma_{22}\rangle_{\Omega}^{11} \\ \langle\sigma_{33}\rangle_{\Omega}^{11} \\ \langle\sigma_{12}\rangle_{\Omega}^{11} \\ \vdots \\ \langle\sigma_{13}\rangle_{\Omega}^{13} \end{Bmatrix}$$

A.3 Computation of Effective Thermal Conductivity Matrix

The thermal conductivity matrix can be obtained in a similar fashion. In this case, three distinct load states with the spatial temperature derivative are imposed:

$$\mathcal{Q} = \left\{ \begin{matrix} \nabla_x T & 0 & 0 \end{matrix} \right\}, \left\{ \begin{matrix} 0 & \nabla_x T & 0 \end{matrix} \right\}, \left\{ \begin{matrix} 0 & 0 & \nabla_x T \end{matrix} \right\}$$

which can be applied to the boundary using displacement boundary conditions for better stability as follows

$$T = \left\{ \begin{matrix} \nabla_x T & 0 & 0 \end{matrix} \right\} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

leading to the following system of equation

$$\begin{Bmatrix} \langle q_1\rangle_{\Omega} \\ \langle q_2\rangle_{\Omega} \\ \langle q_3\rangle_{\Omega} \end{Bmatrix} = \begin{Bmatrix} k_{11} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_3 \end{Bmatrix} \begin{Bmatrix} \frac{\langle\partial T_{\Omega}\rangle}{\partial x_1} \\ \frac{\langle\partial T_{\Omega}\rangle}{\partial x_2} \\ \frac{\langle\partial T_{\Omega}\rangle}{\partial x_3} \end{Bmatrix}$$

where the left hand side is the imposed load state and the right-hand side can be computed using the finite element solution for the derivate the temperature $\frac{\langle \partial T_{11\Omega} \rangle}{\partial x} = \left\langle \frac{\partial \phi}{\partial x} T \right\rangle_{\Omega}$. This yields a system of 9 equations for the conductivity matrix that looks as follows:

State 1

$$\langle q_1 \rangle_{\Omega}^{1} = k_{11} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_1} + k_{12} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_2} + k_{13} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_3}$$

$$\langle q_2 \rangle_{\Omega}^{1} = k_{21} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_1} + k_{22} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_2} + k_{23} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_3}$$

$$\langle q_3 \rangle_{\Omega}^{1} = k_{31} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_1} + k_{32} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_2} + k_{33} \frac{\partial \langle T \rangle_{\Omega}^{1}}{\partial x_3}$$

State 2

$$\langle q_1 \rangle_{\Omega}^{2} = k_{11} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_1} + k_{12} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_2} + k_{13} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_3}$$

$$\langle q_2 \rangle_{\Omega}^{2} = k_{21} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_1} + k_{22} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_2} + k_{23} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_3}$$

$$\langle q_3 \rangle_{\Omega}^{2} = k_{31} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_1} + k_{32} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_2} + k_{33} \frac{\partial \langle T \rangle_{\Omega}^{2}}{\partial x_3}$$

State 3

$$\langle q_1 \rangle_{\Omega}^{3} = k_{11} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_1} + k_{12} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_2} + k_{13} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_3}$$

$$\langle q_2 \rangle_{\Omega}^{3} = k_{21} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_1} + k_{22} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_2} + k_{23} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_3}$$

$$\langle q_3 \rangle_{\Omega}^{3} = k_{31} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_1} + k_{32} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_2} + k_{33} \frac{\partial \langle T \rangle_{\Omega}^{3}}{\partial x_3}$$

which can be rearranged to the following system of matrix equations in the form of $\mathbf{A}x = b$.

$$\begin{bmatrix}
\frac{\partial\langle T\rangle_\Omega^1}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^1}{\partial x_3} \\
\frac{\partial\langle T\rangle_\Omega^2}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^2}{\partial x_3} \\
\frac{\partial\langle T\rangle_\Omega^3}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_1} & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_2} & \frac{\partial\langle T\rangle_\Omega^3}{\partial x_3}
\end{bmatrix}
\begin{Bmatrix} k_{11} \\ k_{12} \\ k_{13} \\ k_{21} \\ k_{22} \\ k_{23} \\ k_{31} \\ k_{32} \\ k_{33} \end{Bmatrix}
=
\begin{Bmatrix} \langle q_1\rangle_\Omega^1 \\ \langle q_2\rangle_\Omega^1 \\ \langle q_3\rangle_\Omega^1 \\ \langle q_1\rangle_\Omega^2 \\ \langle q_2\rangle_\Omega^2 \\ \langle q_3\rangle_\Omega^2 \\ \langle q_1\rangle_\Omega^3 \\ \langle q_2\rangle_\Omega^3 \\ \langle q_3\rangle_\Omega^3 \end{Bmatrix}$$

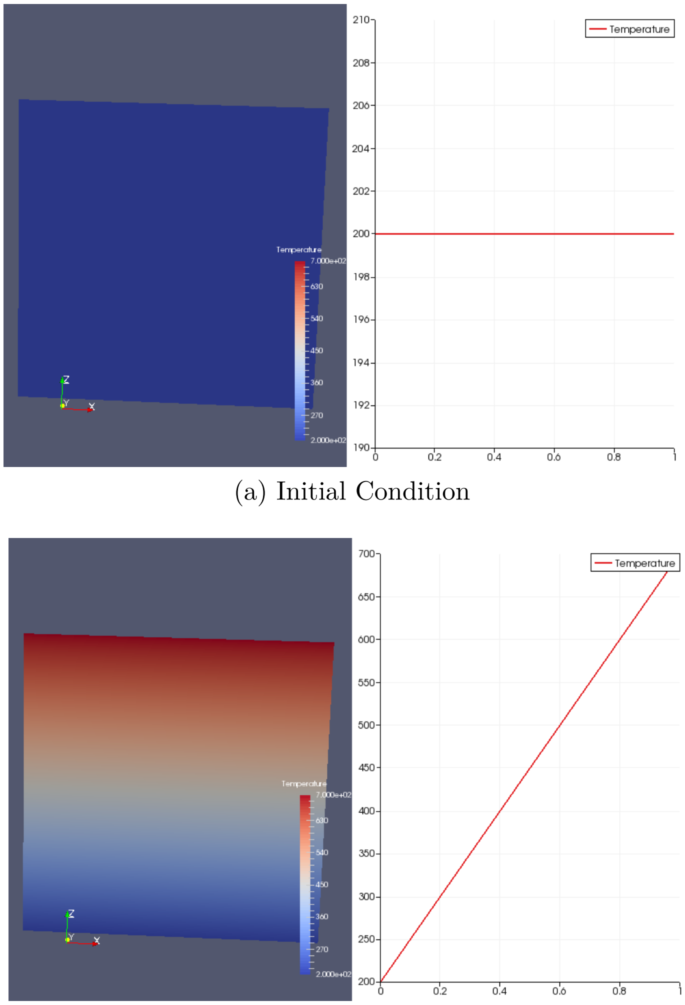
Appendix B

Finite Element Method Verification Studies

The thermomechanical finite element method was verified by performing test cases for the computer program. The verification was performed for both the mechanical equation, the balance of linear momentum, and the thermal equation, the thermodynamic energy balance. The code was verified with various unit tests throughout the development of a standalone module and various sections of larger code. Once a the finite element solver framework had been completed for each equation, a series of test was run to determine the robustness of the solver in easy test cases. The test cases illustrated here will focus on the case in which a linear solution between one surface of the cube and another surface of the cube is expected. The cases shown here are meant to be illustrative, as more exhaustive tests were performed throughout the development of the solver.

B.1 Thermodynamic Energy Balance

For the finite element framework described in Chapter 2, several cases were tested to verify that the program was solving the intended problem. These tests included a changing of the meshing structure with irregular elements that were still mathematically viable, and an imposing of boundary conditions for which the solution of the equation was known. Even though many more test cases were performed, the case show below shows a linear temperature increase from the bottom surface at $z = 0$ to the top surface at $z = 1$. The temperature increases linearly from $200K$ to $700K$ as expected. Moreover, the problem arrives at equilibrium within a couple of timesteps, showcasing the stability of the time-stepping method for the test problem.



(a) Initial Condition

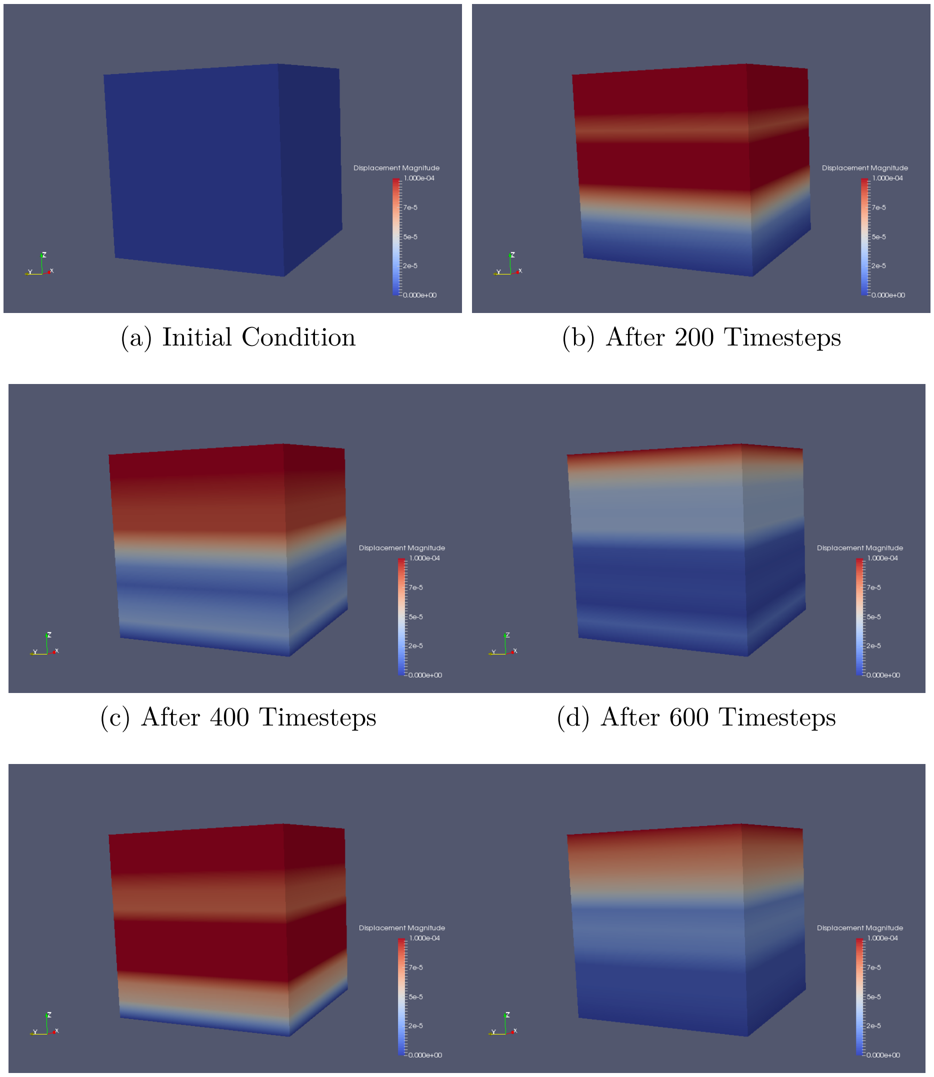
(b) After 50 Timesteps

Figure B.1: Thermodynamic Energy Balance Linear Test Case

B.2 Balance of Linear Momentum

The balance of linear momentum equation performed equally well in the static verification cases as the thermodynamic energy balance. In the dynamic case, however, the solution was prone to oscillate even within the implicit timestepping framework that converged to a stable solution. The oscillation of the solution showcases the difference between the equations, and how they need to treated individually for their mathematical behavior, especially when they are solved in staggered scheme.

The dynamic case for the balance of linear momentum initially exhibited severe oscillations for the linear test case.



(a) Initial Condition

(b) After 200 Timesteps

(c) After 400 Timesteps

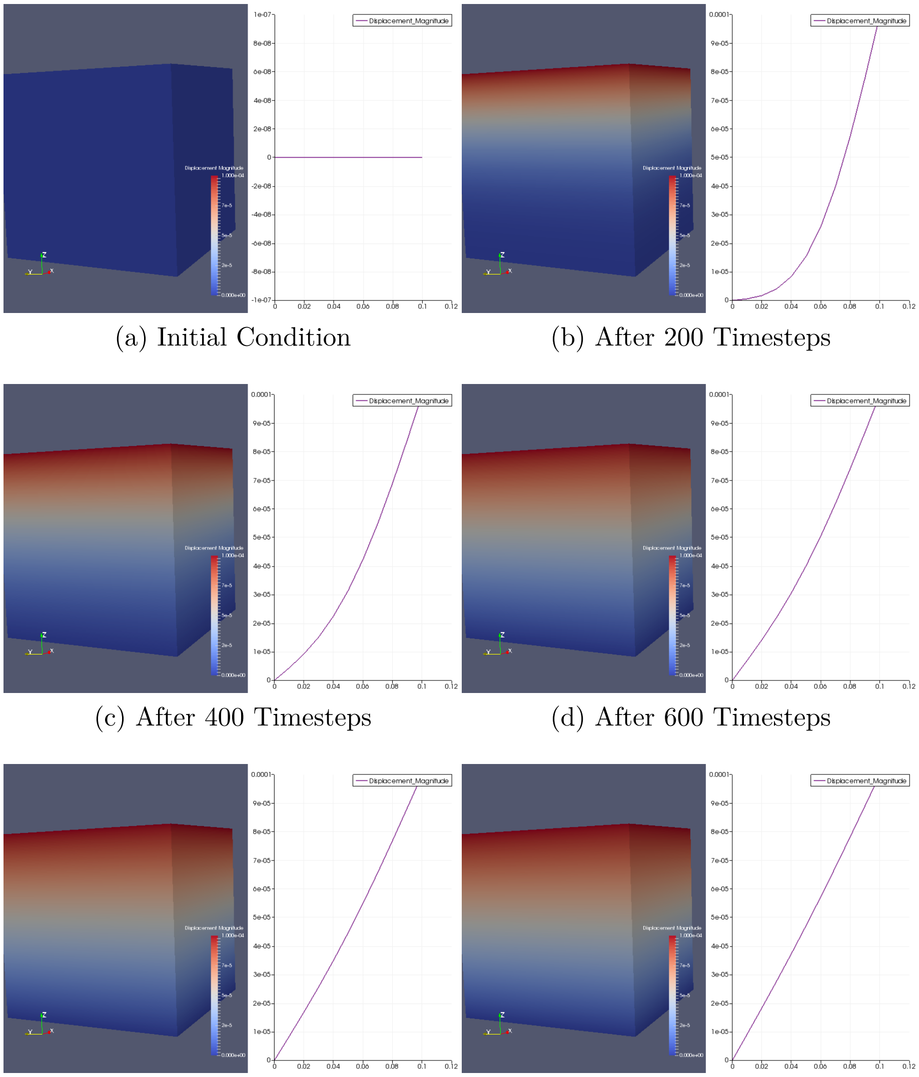
(d) After 600 Timesteps

(e) After 800 Timesteps

(f) After 1000 Timesteps

Figure B.2: Balance of Linear Momentum without Damping

This problem was addressed by adding a damping matrix that linearly damped the equation to converge to the expected solution. The damping matrix serves to reduce the velocity terms and was subtracted from the right-hand side of the equation shown in Chapter 2. As a result the equation arrived at the expected solution without major oscillations.



(a) Initial Condition

(b) After 200 Timesteps

(c) After 400 Timesteps

(d) After 600 Timesteps

(e) After 800 Timesteps

(f) After 1000 Timesteps

Figure B.3: Balance of Linear Momentum with Damping