THERMO-MECHANICAL FATIGUE USING THE EXTENDED

SPACE-TIME FINITE ELEMENT METHOD

by

Ryan T. Schlinkman

APPROVED BY SUPERVISORY COMMITTEE:

____

Dong Qian, Chair

____

Hongbing Lu

____

Ill Ryu

This thesis is dedicated to my parents,

who gave me encouragement through some

of the toughest years of my life.

THERMO-MECHANICAL FATIGUE USING THE EXTENDED SPACE-TIME FINITE ELEMENT METHOD

by

RYAN T. SCHLINKMAN, BS

THESIS

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN

MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

August 2019

ACKNOWLEDGMENTS

I would like to thank Prof. Dong Qian, my thesis adviser, who has supported me in my academic pursuits as well as Rui Zhang, a fellow student, who helped me get through several classes and aided me in my research. I would also like to thank Dr. Samir Naboulsi of the DOD HPCMP and Dr. Thomas Eason of the AFRL for guiding me through this research. Lastly, I thank Dr. Hongbing Lu and Dr. Ill Ryu for serving on my supervising committee.

July 2019

THERMO-MECHANICAL FATIGUE USING THE EXTENDED

SPACE-TIME FINITE ELEMENT METHOD

Ryan T. Schlinkman, MS
The University of Texas at Dallas, 2019

Supervising Professor: Dong Qian, Chair

Thermo-mechanical high-cycle fatigue is a major failure mechanism for many engineering components in a diverse range of industries such as aerospace, automotive, and nuclear among others. Engineers trying to determine the fatigue life of a component typically rely on commercial fatigue analysis software which uses traditional fatigue criteria that are limited in their applicability. For instance, they are poor at handling multiaxial and variable amplitude loading. Furthermore, adding variable amplitude thermal loading into the mix makes using these traditional fatigue criteria even less appealing.

In recent years, there have been attempts to establish methods for simulating high cycle fatigue based on finite element calculations rather than using it as a post-processing step. These include cohesive zone and continuum damage mechanics models. However, all of these methods employ cycle jumping strategies to cut down on the enormous computational time required. However, cycle jumping is not applicable for a random loading history or with random or out-of-phase temperature variation. Motivated by these current developments, this thesis proposes the use of the extended space-time finite element method (XTFEM) in combination with a two scale progressive fatigue damage model for the direct numerical simulation of thermo-mechanical high cycle fatigue. Instead of using the conventional explicit or implicit finite difference time integration methods, temporal approximations are introduced

with FEM mesh and enriched based on the extended finite element method. After outlining the basic theory for XTFEM with thermo-mechanical coupling, the effectiveness of the computational framework is demonstrated in numerical examples including a coupled, thermo-mechanical fatigue simulation of a plate and hat stiffener model representative of a hypersonic aircraft's structure.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Fatigue: Definition, Mechanisms, and Importance

Fatigue is the dominant failure mechanism for many engineering components. Cyclic loading of these components causes the nucleation of microcracks in the material. As more and more cracks form and propagate, they coalesce to form a macrocrack which propagates until the remaining structure can no longer sustain the load resulting in the fracture of the component. The "total number of cycles or time to induce fatigue damage and to initiate a dominant fatigue flaw which is propagated to final failure" is called 'fatigue life' (Suresh (1998), p.221).

The failure category of fatigue may be further subdivided into different fatigue ranges with their own specific failure mechanisms. Table 1.1 is based off of the work of Dufailly and Lemaitre (Dufailly and Lemaitre (1995)). High and very high cycle fatigue occurs due to low-amplitude stresses typically below the material's yield strength causing elastic behavior globally (Suresh (1998), p.221). While there is some plastic deformation in these two ranges, the plasticity largely occurs around defects at the microscopic scale. Thus, most of the fatigue life is spent initiating the macroscopic fatigue crack (Bhamare (2012), p.3). For low and very low cycle fatigue, on the other hand, plastic strain occurs at the macroscopic scale with every cycle (Schijve (2009), p.161). This causes a relatively quick crack initiation, and thus, the time spent in crack growth is comparable to the time spent initiating the crack (Bhamare (2012), p.3).

In 1983, the U.S. government estimated that the total economic cost due to fracture was \$119 billion (in 1982 dollars; Reed et al. (1983), p.1). It also estimated that 80% of these costs involved cyclic loading or fatigue as a contributing factor which meant that the annual cost due to the fatigue of materials accounted for about 3% of the gross national product (Dowling (2013), p.399). Similar studies in Europe have also arrived at high economic costs

Table 1.1. Fatigue Ranges

Fatigue Ranges	Cycles to Failure	Typical Stress Level	Strain Ratio: $\frac{\Delta\varepsilon^p}{\Delta\varepsilon^e}$
Very High Cycle Fatigue	$> 10^7$	σ_f^∞	≈ 0
High Cycle Fatigue (HCF)	10^5 to 10^6	$< \sigma_Y$	≈ 0
Low Cycle Fatigue (LCF)	10^2 to 10^4	σ_Y to σ_U	1 to 10
Very Low Cycle Fatigue	1 to 20	$\approx \sigma_U$	10 to 100

for fatigue (Milne (1994), p.173). Additionally, the U.S. Air Force estimated that, between 1982 and 1996, 56% of Class A engine-related failures were due to high cycle fatigue (Bartsch (2003), p.xi), and in fiscal year 1994, approximately 850,000 maintenance man-hours were expended inspecting components as part of HCF risk management (Bartsch (2003), p.xi). In total, the cost of HCF to the Air Force and Navy is estimated at $400 million per annum (Bartsch (2003), p.xi).

1.2 Thermo-Mechanical Fatigue

1.2.1 Definition and the Effects of Cycling Temperatures

Thermal fatigue is defined as relating to structures which undergo thermal cycling leading to "the initiation of cracks, their propagation and failure," whereas thermo-mechanical fatigue (TMF) involves structures undergoing both mechanical as well as thermal cyclic loading leading to fatigue failure (Charkaluk and Rémy (2011), p.271). Reasons to include temperature and its affects in a fatigue life prediction are many. As a structure undergoes temperature changes, its mechanical properties change which affects how the structure responds to mechanical loading. For this reason alone, fatigue life prediction models based on isothermal data need to be modified before being applied to TMF (Bill (1986), p.1). Things become even more pronounced at higher temperatures. At temperatures in excess of one half the homologous temperature, strain-rate effects become significant and creep phenomenon must be taken into account (Suresh (1998), p.590). As as result, the amount of plastic deformation in the plastic zone of a fatigue crack is enhanced leading to higher fatigue damage accumulation (Schijve (2009), p.481). Indeed, it is well-known from experiments that crack-growth is faster at elevated temperatures than at room temperature (Bill (1986), p.2; Schijve (2009), p.488). Lastly and sometimes more importantly, temperature changes to an overly constrained structure can translate into mechanical stress, and when the temperature change is cyclic, these mechanical stresses will result in fatigue damage.

1.2.2 History, Examples and Importance

The study of TMF began with the work of Coffin who was researching the affects of TMF in the nuclear power industry (Coffin (1954a,b)). Around the same time, Manson studied TMF in application to turbine blades and disks in the aerospace industry (Manson (1953)). The nuclear and aerospace industries continue to be concerned with TMF phenomena in the present (Charkaluk and Rémy (2011), p.272), but the applications have proliferated to other areas. Examples illustrating the importance of TMF are many, but a few will suffice:

1. Aerospace engine parts: Aerospace engine turbine blades are exposed to enormous temperatures through the combustion of jet fuel in addition to immense centrifugal forces and vibration loads (Schijve (2009), p.482). The thermal and centrifugal loading correspond to low-cycle fatigue conditions where each start-up and shutdown constitutes a cycle whereas the blade vibrations yield high cycle fatigue conditions (Schijve (2009), p.483).

2. Nuclear pressure vessels: Nuclear fission generates an enormous amount of heat and pressure. This heat requires the periodic application of coolant to decrease the temperature of the vessels and piping thus creating cyclic temperature variations leading to low-cycle fatigue (Mohanty et al. (2012)).

3. The mixing of hot and cold fluids in nuclear plant pipes: In the piping systems of some nuclear power plants, there exist "mixing tees" where water flows of different temperatures turbulently mix and generate seemingly random and fluctuating temperature fields and corresponding stresses (Desmorat et al. (2007), p.923). These stresses do not typically exceed the yield strength, and so, this example falls into the high-cycle fatigue category.

4. A brake disk undergoing friction-heating from a brake pad: In this situation, the surface of the disk is heated to a high temperature over a short period of time creating a large temperature gradient across just a tiny fraction of the total thickness of the disk (Charkaluk and Rémy (2011), p.272). Because this affected thickness is negligible, we may assume that the deformation of the disk is negligible. This means that any temperature-induced strain will translate to mechanical strains, and when the gradients are high enough, the material will undergo plastic deformation. Over many heating and cooling cycles, this will lead to cyclic plasticity and thus low-cycle fatigue (Charkaluk and Rémy (2011), p.272).

5. Automotive engine parts: A number of parts in the internal combustion engine such as pistons, cylinder heads, exhaust manifolds, and turbo-compressors experience very high temperatures while in use but eventually cool down to ambient temperature when the vehicle is turned off. This thermal cycling induces repeated mechanical loads leading to fatigue (Charkaluk and Rémy (2011), pp.274-275).

6. Super- and hypersonic aircraft structures: Aircraft traveling at supersonic and hypersonic speeds generate an enormous amount of drag which causes the skin of the aircraft to heat up while the inner structure trails behind creating large temperature gradients. The situation is reversed upon landing (Schijve (2009), p.486). Additionally, these structures experience random acoustic loads during their trajectory. The combined thermal and acoustic loading yields a high-cycle fatigue condition.

1.3 Traditional Methods of Estimating Fatigue Life

Traditional methods of estimating the fatigue life of a component can be broken into two categories: the 'safe-life' concept and the 'damage-tolerant' concept. The safe-life concept is essentially theoretical in nature, and its goal is the prevention of crack initiation (Suresh (1998), pp.14-15). The safe-life concept generally adheres to the following procedure (Suresh (1998), pp.14-15):

1. The component's expected service loading is determined.

2. The component is analyzed using hand calculations, simulation software, and/or tested in a lab under service loading conditions. A fatigue life is estimated for the component.

3. The fatigue life is modified by a safety factor. This modified fatigue life becomes the component's 'safe-life.'

4. After the component in service has reached the end of its safe-life, it is "automatically retired from service" regardless of whether cracks have formed or not (Suresh (1998), p.14).

The analysis done by hand calculation and/or simulation software mentioned in step 2 above utilizes one of the 'total-life' approaches. These approaches, first begun in the mid-nineteenth by Wöhler (Wöhler (1860)), plot the service load stress against the number of cycles to failure, the classic S-N curve. A problem is presented when the loading is multiaxial since S-N curves are typically created using uniaxial loading. In this case, the multiaxial loading needs to be processed so that it can be compared with the uniaxial curve. These approaches are discussed further in chapter 3.

The damage-tolerant concept, on the other hand, is more empirical, and the goal is the tolerance of a fatigue crack without it leading to failure. The argument is that if there exists a structural redundancy, failure of one load path would not compromise the others, and the component or machine could still be operated safely until the crack is found during maintenance or inspection (Suresh (1998), p.15). The damage-tolerant concept makes use of the field of fracture mechanics which was first started by Irwin (Irwin (1957)) who built upon the work of Griffith (Griffith (1921)). Irwin showed that the magnitude of the crack-tip stress field can be represented by a scalar value called the stress-intensity factor, K (Hertzberg et al. (2013), p.318). The crack tip advances when the stress-intensity reaches a critical value, K_c (Sun and Jin (2012), p.3). The application of fracture mechanics to fatigue was initiated by Paris et al. (Paris et al. (1961)) who proposed that the rate of crack growth per cycle, da/dN, was related to the stress intensity factor range, ΔK (Suresh (1998), pp.5, 296):

$$\frac{da}{dN} = C\left(\Delta K\right)^m \tag{1.1}$$

where C and m are empirical constants which depend on the material properties, microstructure, loading conditions, temperature, etc. (Suresh (1998), p.297). These constants must be determined in a lab setting and are applied to the same component in service by the concept of similitude. That is, a crack in the lab component and a crack in the service component "subjected to the same range of stress intensity factor" at the same load ratio should grow at the same rate (Bhamare (2012), p.6). The fatigue life can then be defined as "the number of fatigue cycles or time to propagate the dominant crack from its initial size to some critical dimension" (Suresh (1998), p.13). The benefits of the damage-tolerant approach then are that: a.) the component or machine does not need to be retired from service after a set number of cycles and b.) even the presence of a crack does not necessarily require a retirement from service.

There are a number of problems with these traditional approaches. The problems with the various total-life approaches associated with the safe-life concept are discussed in more detail in chapter 3. The problems with the damage-tolerant approach of fracture mechanics are as follows:

1. Determination of a stress-intensity factor for a complex geometry under multiaxial loading is difficult (Bhamare (2012), p.6).

2. The extension of equation 1.1 from constant amplitude cyclic loading to random, variable amplitude loading is problematic since different loading conditions will change the empirical constants (Suresh (1998), p.333).

3. The damage-tolerant concept is not applicable to all structures. For some components, regular inspection is either impractical, too expensive relative to the cost of the component, or both. For other components, undetected cracks can easily grow to failure in between inspections resulting in catastrophic consequences.

Given the problems with the traditional methods for fatigue life estimation, we seek a different method of fatigue analysis. This method needs to be able to account for multiaxial, variable amplitude loading. It should also be able to handle the effects of non-zero mean stresses and random, thermal variation. Additionally, this method should avoid Miner's problem-fraught linear damage law as well as the somewhat arbitrary rainflow cycle counting methods (Desmorat et al. (2007), p.910).

Since the late 1950's, researchers have been developing the field of continuum damage mechanics wherein "the analysis of the damage development in mesoscopic and macroscopic fracture processes" is done "in the framework of continuum mechanics" (Murakami (2012), p.3). Specifically, the two-scale progressive fatigue damage model first proposed by Lemaitre (Lemaitre and Doghri (1994); Lemaitre et al. (1999); Lemaitre and Desmorat (2005)) and developed by Desmorat et al. to handle random, thermal variation (Desmorat et al. (2007)) is specifically designed for high cycle fatigue and can handle all the requirements listed in the previous paragraph. The two scale model, discussed more fully in chapter 4, separates the macroscale structure calculation from the microscale damage evaluation. The macroscale structure calculation is solved with the finite element method or other method using a purely elastic material model. The strains at this scale are then used to evaluate damage locally at the microscale where defects such as inclusions exist, plasticity occurs, and microcracks form.

The two scale model has been used to predict service life for both low and high cycle fatigue and applied to both uniaxial and multiaxial analyses (Lemaitre and Doghri (1994)). Lemaitre et al. (Lemaitre et al. (1999)) then showed the power of the two scale model for a non-zero mean stress and non-proportional loading with results comparing well with the Dang Van criterion. More recently, Latrou et al. (Lautrou et al. (2009)) used the two scale model to accurately predict the fatigue life of steel welded joints commonly used in naval structures. Additionally, dos Santos et al. (dos Santos et al. (2012)) modified Lemaitre et al.'s model (Lemaitre et al. (1999)) by means of the Soderberg fatigue relation to account for high mean stress effects in cardiovascular stents and showed that the modified model has good agreement with experimental results. Finally, Desmorat et al. (Desmorat et al. (2007)) added the ability for the two scale model to take into account random temperature changes and proved its predictive power even in random thermo-mechanical loading by showing good agreement with a pressure-vessel testing experiment. It is this latter, thermo-mechanical model which we will use to predict thermo-mechanical fatigue life.

1.4 Simulating Fatigue

1.4.1 Finite Difference Methods: Explicit and Implicit Time Integration

The standard computer-aided means for determining a component's fatigue life is to simulate a representative set of loading amplitudes in finite element software and run the results through a fatigue analysis post-processor (see chapter 3). However, there have been efforts in recent years to simulate the fatigue process within the finite element analysis itself. Roe and Siegmund (Roe and Siegmund (2003)) and Siegmund (Siegmund (2004)) employed a cohesive zone model to simulate the fatigue of a double cantilever beam bonded by an adhesive. Instead of making crack growth a function of cycle count, their model relies upon an "incremental solution process to the cyclic load problem" (Siegmund (2004), p.932). Cedergren et al. (Cedergren et al. (2004)) utilized an extended Gurson void growth model, a viscoplastic formulation, with plasticity to simulate low cycle fatigue in powder manufactured steel bars. The time integration scheme they used was an explicit Newmark method (Cedergren et al. (2004), p.902). Oller et al. (Oller et al. (2005)) used a continuum damage mechanics model combined with elasto-plasticity to simulate low cycle thermo-mechanical fatigue damage evolution. They utilized an implicit time-integration method to begin tracking the load history in small steps but then employed a time advancing strategy for load cycles of the same amplitude as the one previously tracked (Oller et al. (2005), p.185). Pirondi et al. (Pirondi et al. (2006)) performed finite element simulations of low cycle fatigue using both continuum damage mechanics and porous metal plasticity models for comparison. Takagaki and Nakamura (Takagaki and Nakamura (2007)) also utilized a continuum damage mechanics model with an anisotropic damage tensor and elasto-plasticity to simulate fatigue crack propagation for low cycle fatigue. They were able to simulate the simultaneous growth of two cracks in a plate which showed good agreement with experiments. Lu et al. (Lu et al. (2015)) simulated the low cycle fatigue behavior of 9Cr power plant steel at high temperature using a continuum damage mechanics approach coupled with the Chaboche elasto-visco-plastic model and employed an implicit scheme for time integration (Lu et al. (2015), p.151). All of the previously mentioned studies have looked at the simulation of low cycle fatigue which, given the relatively short time scale, is not computationally prohibitive using standard, finite difference time integration methods.

However, some researchers have attempted to simulate high cycle fatigue using the traditional, finite difference time integration schemes. Jiang et al. (Jiang et al. (2009)) used a cohesive zone model with cyclic plasticity and a damage variable to simulate high cycle fatigue for a compact-tension-shear specimen. Acknowledging that an implicit time-integration scheme for the entire life of the component is computationally prohibitive, they utilized an extrapolation scheme where the damage done over a small number of cycles is extrapolated over relatively small time spans (Jiang et al. (2009), p.680). Raje et al. (Raje et al. (2009)) examined the microstructure of polycrystaline materials under rolling contact fatigue loading using continuum damage mechanics with lattice springs connecting the grains. They utilized the explicit central difference scheme for time integration with a 'jump-in-cycles' procedure to speed up the simulation (Raje et al. (2009), 350). Lestriez et al. (Lestriez et al. (2007)) presented a continuum damage mechanics model combined with a flow surface of fatigue using the Sines criterion and element deletion to simulate the fatigue damage of roller bearings. They employed an explicit time integration scheme in Abaqus/Explicit's VUMAT, but in order to reduce CPU time, they used a cycle jump technique which extrapolates the damage over a given number of cycles (Lestriez et al. (2007), p.391). Kim (Kim (2013)) and Kim and Yoon (Kim and Yoon (2014)) used a bilinear, cycle-dependent cohesive zone law to study the high cycle fretting fatigue of 7050-T7451 aluminum alloy. They solved the equations of motion with an implicit time integration but used a cycle jump strategy to speed-up the computation (Kim (2013), p.684; Kim and Yoon (2014), p.31). Martin and Sun (Martin and Sun (2015)) developed a means for assessing fatigue damage in transcatheter aortic valves using a soft tissue damage model. To hasten the fatigue simulation, they multiplied the damage per simulated cycle by a factor (Martin and Sun (2015), p.3029). Barbu et al. (Barbu et al. (2015)), building off the work of Oller et al. (Oller et al. (2005)), proposed a "stepwise load-advancing strategy" for high cycle and very high cycle fatigue combined with a continuum damage mechanics model to simulate fatigue specimens and showed good agreement with experiments. Their load-advancing strategy consisted of two phases. The first, load-tracking phase, used small time increments following the loading path in order to "save the characteristics of the cyclical load" (Barbu et al. (2015), p.120). The second, large-increments phase simply multiplied the results of the first phase by however many cycles were at that load level (Barbu et al. (2015), p.122). Bak et al. (Bak et al. (2016))

simulated the formation of mixed-mode delamination cracks in laminated structures due to high cycle fatigue using a cohesive zone model combined with a "Paris law-like model" to determine crack growth rate. Instead of solving the equations of motion multiple times to form a loading cycle, they used an "envelope load approach" where Paris's law and fracture mechanics were used to determine the crack growth rate at the maximum load during a cycle, and they simply integrated that rate over time (Bak et al. (2016), p.164).

The takeaway from the previous high cycle fatigue studies is that they all had to use a form of cycle jumping extrapolation or a damage multiplier given that the time scale of high cycle fatigue makes a complete simulation impractical even with high-performance computers (Bhamare (2012), p.9). However, such methods cannot be used in cases with a random mechanical loading history (Bhamare (2012), p.10). Additionally, if a cyclic or random temperature variation that is not in phase with the mechanical loading is present, then extrapolation simply is not possible. Thus, we seek a time integration method that can perform a complete thermo-mechanical high cycle fatigue simulation without the high computational cost.

1.4.2 Extended Space-Time Finite Element Method (XTFEM)

Instead of the methods listed above, we propose the use of an extended space-time finite element method (XTFEM) formulation to solve thermo-mechanical high cycle fatigue problems. In contrast to the semi-discrete finite difference methods such as explicit or implicit time integration, space-time finite element methods discretize not only the spatial object being analyzed but also the temporal domain as well using finite elements. XTFEM builds off of the standard space-time finite element method (TFEM) by enriching the temporal shape functions with harmonic functions such as sine or cosine in order to match the nature of the loading conditions. By doing so, one can make the time step size equal to the harmonic loading's time period, a vast improvement over the finite difference methods or the standard

TFEM. A history and research developments with TFEM and XTFEM will be discussed in chapter 2.

1.5 Objective and Outline of the Thesis

The objective of this work is to develop a direct numerical simulation approach to calculating a component's life under thermo-mechanical high cycle fatigue loading. Since the two scale progressive fatigue damage model does not account for creep or other viscosity effects, our study will be limited to less than one-third of a material's melting temperature (Desmorat et al. (2007), p.911). We will demonstrate through several examples the power of our thermo-mechanical XTFEM code.

Chapter 2 will begin by explaining and deriving the regular space-time finite element method for both thermal analysis (section 2.2) as well as mechanical analysis (section 2.3). This will prepare us for the derivation of the thermo-mechanical extended space-time finite element method in section 2.4. Chapter 3 will give a brief overview of the traditional approaches to calculating multiaxial high-cycle fatigue damage. Approaches discussed here include equivalent stress methods (section 3.1.1), critical plane methods (section 3.1.2), and Dang Van's multiscale method (section 3.1.3). The last half of the chapter will examine the Palmgren-Miner rule (section 3.2), the cycle counting methods for random loading (section 3.3), and finish by summarizing the problems with the traditional approaches (section 3.4). Next, chapter 4 will explain the two scale progressive fatigue damage model from continuum damage mechanics. The chapter will start in section 4.1 by discussing the research history of continuum damage mechanics in general and the two scale model in particular and explain how the two scale model works. It is necessary in section 4.2 to describe the material parameters that are utilized in the two scale model since not all of them are familiar to many readers. The detailed derivation of the two scale formulation is given in section 4.3 followed by an outline of the implementation in section 4.4. Chapter 5 will describe three numerical examples used to demonstrate the power of the XTFEM code and discuss the results. Lastly, chapter 6 will then conclude this thesis and give a consideration of future work.

CHAPTER 2

EXTENDED SPACE-TIME FINITE ELEMENT METHOD (XTFEM)

2.1 Introduction

2.1.1 Chapter Outline

To solve the problem of the time scale associated with high cycle fatigue, this work utilizes a coupled thermo-mechanical extended space-time finite element method (XTFEM) formulation. In this chapter, the XTFEM method based on the time-discontinuous Galerkin (TDG) formulation is derived. First, regular space-time finite element method (TFEM) formulations for thermal and mechanical analyses are derived in sections 2.2 and 2.3, respectively. Section 2.4 then details the XTFEM formulation for mechanical analysis which is derived by building off of the previous TFEM mechanical formulation in section 2.3.

2.1.2 XTFEM Background

Generally, dynamic FEM problems have been integrated in the time domain using finite difference methods such as the central difference or Newmark-β methods. That is, while the partial differential equations in the spatial domain are solved using finite elements, the temporal domain uses finite differences making such methods "semi-discrete." While these methods work well in many cases including low-cycle fatigue, the time scale associated with high-cycle fatigue makes these methods computationally prohibitive (see section 1.4.1). Furthermore, these methods suffer from either time step size limitations or convergence issues "due to the oscillatory nature of the fatigue loading condition" (Bhamare et al. (2014), p.388). In contrast to the semi-discrete methods previously mentioned, space-time finite element methods discretize not only the spatial object being analyzed but also the temporal domain as well using finite elements. The concept of extending the use of finite element shape functions to the time domain was first proposed by Argyris and Scharpf (Argyris and Scharpf (1969)), Fried (Fried (1969)), Oden (Oden (1969)), and others. These early pioneers in this area utilized a formulation with a continuous temporal mesh called the time-continuous Galerkin (TCG). An alternative to the TCG formulation is to break-up the temporal domain into time segments and allow jumps in the solution variable between segments. This approach is known as the time-discontinuous Galerkin (TDG) formulation and was first used by Reed and Hill (Reed and Hill (1973)) and Lesaint and Raviart (Lasaint and Raviart (1974)). Hulbert and Hughes (Hulbert and Hughes (1990)), Hughes and Stewart (Hughes and Stewart (1996)), and Li and Wiberg (Li and Wiberg (1996, 1998)) extended the TDG formulation to solve second-order hyperbolic systems such as structural dynamics. Lesaint and Raviart (Lasaint and Raviart (1974)) and others (Delfour et al. (1981); Johnson (1984)) have proven that the TDG formulation results in systems that are A-stable and higher-order accurate.

2.2 Space-Time Finite Element Method (TFEM) for Thermal Analysis

2.2.1 Governing Equation to Weak Form

We start with the governing equations of initial-boundary-value thermal analysis over a 3D spatial domain, Ω, and temporal domain, $I =]0, T[$:

$$G = \rho\, c_p\, \dot{\phi} - \nabla\cdot (k\, \nabla \phi) \qquad \text{on} \quad Q \equiv \Omega \times I \tag{2.1}$$

$$\phi = \bar{\phi}\, (\Gamma, t) \qquad \text{on} \quad \Upsilon_D \equiv \Gamma_D \times I \tag{2.2}$$

$$q = k\, \nabla \phi\, \boldsymbol{n} \qquad \text{on} \quad \Upsilon_N \equiv \Gamma_N \times I \tag{2.3}$$

$$h\, (\phi_a - \phi) = k\, \nabla \phi\, \boldsymbol{n} \qquad \text{on} \quad \Upsilon_3 \equiv \Gamma_3 \times I \tag{2.4}$$

$$\phi\, (\boldsymbol{x}, 0) = \phi_0\, (\boldsymbol{x}) \qquad \text{on} \quad \boldsymbol{x} \in \Omega \tag{2.5}$$

$$\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_3 \tag{2.6}$$

$$\Gamma_D \cap \Gamma_N \cap \Gamma_3 = 0 \tag{2.7}$$

where equation 2.1 is the general heat equation, Γ is the boundary of Ω and is the combination of the non-overlapping essential (Dirichlet) temperature boundary Γ_D, natural (Neumann) surface heat flux boundary Γ_N, and convection boundary Γ_3, G is the heat source generation, ϕ is the temperature, k is the conductivity, ρ is the volumetric mass density, c_p is the specific heat, $\boldsymbol{n}$ is the outward normal to Γ, $\bar{\phi}$ is the prescribed temperature on Γ_D, q is the prescribed surface heat flux on Γ_N, h is the prescribed convection coefficient on Γ_3, and ϕ_0 is the initial temperature. Lastly, the superposed dot in $\dot{\phi}$ denotes a partial derivative with respect to time.

Since our formulation is the time-discontinuous Galerkin, we discretize the temporal domain into segments which, combined with the spatial mesh, comprise space-time slabs. Figure 2.1 shows a 1D spatial mesh combined with the discretization in time to form the space-time slabs. The temporal domain, $I =]0, T[$, consists of N segments, $I_n =]t_{n-1}, t_n[$, such that $0 = t_0 < t_1 < \ldots < t_n < ... < t_N = T$. The n^{th} space-time slab is given as $Q_n \equiv \Omega \times I_n$. Space-time boundary conditions are defined as $(\Upsilon_D)_n \equiv \Gamma_D \times I_n$, $(\Upsilon_N)_n \equiv \Gamma_N \times I_n$, and $(\Upsilon_3)_n \equiv \Gamma_3 \times I_n$ for the temperature, surface heat flux, and convection boundary conditions, respectively, such that $(\Upsilon)_n = \overline{(\Upsilon_D)_n \bigcup (\Upsilon_N)_n \bigcup (\Upsilon_3)_n}$. Each space-time slab is divided into n_{el} space-time elements where Q_n^e is the e^{th} space-time element and $(\Upsilon^e)_n$ is the e^{th} element's boundary. Thus, the interior of the slab is given as:

$$Q_n^{\Sigma} = \bigcup_{e=1}^{n_{el}} Q_n^e \tag{2.8}$$

and the inter-element boundary of the slab is given as:

$$\left(\Upsilon^{\Sigma}\right)_n = \bigcup_{e=1}^{n_{el}} \left(\Upsilon^e\right)_n - \left(\Upsilon\right)_n \tag{2.9}$$

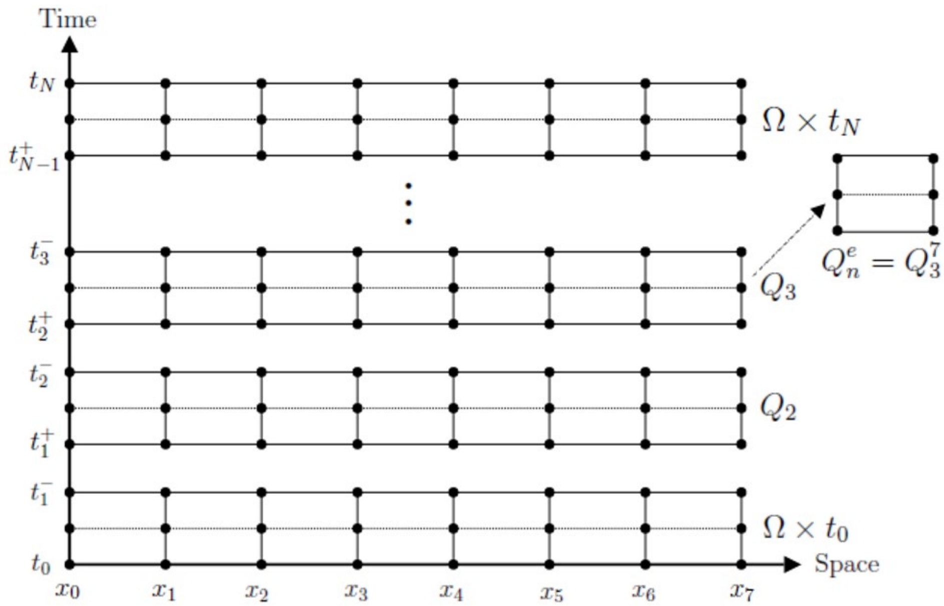

Figure 2.1. Space-Time Discretization (Linear Spatial; Quadratic Temporal)

Adding the natural and thermal convection boundary conditions to equation 2.1, multiplying by the variation in temperature, $\delta\phi$, and integrating over both time and space:

$$\begin{aligned} 0 = & \int_{I_n}\int_{\Omega} \delta\phi \left(\rho\, c_p\, \dot{\phi} - \nabla\cdot (k\,\nabla\phi) - G\right) d\Omega\, dt \\ & + \int_{I_n}\int_{\Gamma_N} \delta\phi\,(k\,\nabla\phi\cdot \boldsymbol{n} - q)\, d\Gamma\, dt \\ & + \int_{I_n}\int_{\Gamma_3} \delta\phi\,(k\,\nabla\phi\cdot \boldsymbol{n} - h\,(\phi_a - \phi))\, d\Gamma\, dt \end{aligned} \tag{2.10}$$

The essential boundary condition is not added since we make its test function vanish on that boundary (Krysl (2005), p.40). The middle term on the top line can be expanded by the product rule (Krysl (2005), p.39):

$$\begin{aligned} -\int_{I_n}\int_{\Omega} \delta\phi\,(\nabla\cdot (k\,\nabla\phi))\, d\Omega\, dt = & -\int_{I_n}\int_{\Omega} \nabla\cdot (\delta\phi\, k\,\nabla\phi)\, d\Omega\, dt \\ & + \int_{I_n}\int_{\Omega} \delta\nabla\phi\cdot (k\,\nabla\phi)\, d\Omega\, dt \end{aligned} \tag{2.11}$$

Using the divergence theorem, the first term on the RHS of equation 2.11 can be converted to (Krysl (2005), p.39):

$$\int_{I_n}\int_{\Omega} \nabla\cdot(\delta\phi\, k\, \nabla\phi)\, d\Omega\, dt = \int_{I_n}\int_{\Gamma_{int}} \delta\phi\, [\![k\, \nabla\phi\cdot \boldsymbol{n}]\!]\, d\Gamma\, dt + \int_{I_n}\int_{\Gamma} \delta\phi\, (k\, \nabla\phi\cdot \boldsymbol{n})\, d\Gamma\, dt \quad (2.12)$$

where the $[\![k\, \nabla\phi\cdot \boldsymbol{n}]\!]$ is the spatial, inter-element jump term which vanishes due to the inter-element continuity. Both terms on the RHS of equation 2.12 have $d\Gamma$ rather than $d\Omega$ because "the test function vanishes on the complement of the traction boundaries" (Belytschko et al. (2014), p.153). The RHS term of equation 2.12 cancels with both $k\, \nabla\phi\cdot \boldsymbol{n}$ terms in equation 2.10 since Γ in equation 2.12 is the union of all surfaces and would thus include both the surfaces where surface heat flux (Γ_N) and convection (Γ_3) boundary conditions are applied (Krysl (2005), p.40). This leaves us with:

$$\begin{aligned} &\int_{I_n}\int_{\Omega} \delta\phi\, \rho\, c_p\, \dot{\phi}\, d\Omega\, dt \\ &+\int_{I_n}\int_{\Omega} \delta\nabla\phi\cdot (k\, \nabla\phi)\, d\Omega\, dt \\ &\qquad = \int_{I_n}\int_{\Omega} \delta\phi\, G\, d\Omega\, dt \\ &\qquad + \int_{I_n}\int_{\Gamma_N} \delta\phi\, q\, d\Gamma\, dt \\ &\qquad + \int_{I_n}\int_{\Gamma_3} \delta\phi\, h\, (\phi_a - \phi)\, d\Gamma\, dt \end{aligned} \quad (2.13)$$

We now add a new term to allow for discontinuities across time element boundaries.

$$\begin{aligned} \int_{\Omega} \delta\phi\big(t_{n-1}^{+}\big)\, \rho\, c_p\, [\![\phi(t_{n-1})]\!]\, d\Omega = &\int_{\Omega} \delta\phi\big(t_{n-1}^{+}\big)\, \rho\, c_p\, \phi\big(t_{n-1}^{+}\big)\, d\Omega \\ &- \int_{\Omega} \delta\phi\big(t_{n-1}^{+}\big)\, \rho\, c_p\, \phi\big(t_{n-1}^{-}\big)\, d\Omega \end{aligned} \quad (2.14)$$

Note that this term is not integrated over the time step as it only occurs at the boundaries. Adding the RHS of equation 2.14 to the LHS of equation 2.13 and rearranging, we have:

$$\begin{aligned}
&\int_{I_n}\int_{\Omega} \delta\phi\, \rho\, c_p\, \dot{\phi}\, d\Omega\, dt \\
&+ \int_{I_n}\int_{\Omega} \delta\nabla\phi\cdot (k\,\nabla\phi)\, d\Omega\, dt \\
&+ \int_{\Omega} \delta\phi\big(t_{n-1}^{+}\big)\, \rho\, c_p\, \phi\big(t_{n-1}^{+}\big)\, d\Omega \\
&\qquad = \int_{I_n}\int_{\Omega} \delta\phi\, G\, d\Omega\, dt \\
&\qquad + \int_{I_n}\int_{\Gamma_N} \delta\phi\, q\, d\Gamma\, dt \\
&\qquad + \int_{I_n}\int_{\Gamma_3} \delta\phi\, h\,(\phi_a - \phi)\, d\Gamma\, dt \\
&\qquad + \int_{\Omega} \delta\phi\big(t_{n-1}^{+}\big)\, \rho\, c_p\, \phi\big(t_{n-1}^{-}\big)\, d\Omega
\end{aligned} \tag{2.15}$$

2.2.2 Discretization

We use the following definitions for the discretization of the weak form:

$$\nabla \boldsymbol{N}_x = \boldsymbol{B}_x$$

$$\boldsymbol{N}(x,t) = \boldsymbol{N}_t \otimes \boldsymbol{N}_x$$

$$\boldsymbol{B}(x,t) = \boldsymbol{N}_t \otimes \boldsymbol{B}_x$$

$$G(x,t) = G(t)\; G(x)$$

$$q(x,t) = q(t)\; q(x)$$

$$h(x,t) = h(t)\; h(x)$$

$$\phi(x,t) = \sum_{I=1}^{n} N_I(x,t)\,\phi_I$$

$$\nabla\phi(x,t) = \sum_{I=1}^{n} \nabla N_I(x,t)\;\phi_I = \sum_{I=1}^{n} B_I(x,t)\,\phi_I$$

$$\dot{\phi}(x,t) = \sum_{I=1}^{n} \dot{N}_I(x,t)\,\phi_I$$

$$\delta\phi(x,t) = \sum_{I=1}^{n} N_I(x,t)\,\delta\phi_I$$

$$\delta\nabla\phi(x,t) = \sum_{I=1}^{n} \nabla N_I(x,t)\,\delta\phi_I = \sum_{I=1}^{n} B_I(x,t)\,\delta\phi_I$$

where I is the node number. Substituting the discretized equations into the weak form of the general heat equation (2.15), moving the variation in the temperature outside the integrals, breaking up the $h(\phi_a - \phi)$ term, and moving the negative portion of the convection term to the LHS:

$$\begin{aligned}
&\delta\phi \int_{I_n} \int_{\Omega} \boldsymbol{N}^T \rho\, c_p\, \dot{\boldsymbol{N}}\, d\Omega\, dt\, \boldsymbol{\phi}_n \\
&+ \delta\phi \int_{I_n} \int_{\Omega} \boldsymbol{B}^T k\, \boldsymbol{B}\, d\Omega\, dt\, \boldsymbol{\phi}_n \\
&+ \delta\phi \int_{\Omega} \boldsymbol{N}^T\left(t_{n-1}^{+}\right) \rho\, c_p\, \boldsymbol{N}\left(t_{n-1}^{+}\right)\, d\Omega\, \boldsymbol{\phi}_n \\
&+ \delta\phi \int_{I_n} \int_{\Gamma_3} \boldsymbol{N}^T h\, \boldsymbol{N}\, d\Gamma\, dt\, \boldsymbol{\phi}_n \\
&= \delta\phi \int_{I_n} \int_{\Omega} \boldsymbol{N}^T G\, d\Omega\, dt \\
&+ \delta\phi \int_{I_n} \int_{\Gamma_N} \boldsymbol{N}^T q\, d\Gamma\, dt \\
&+ \delta\phi \int_{I_n} \int_{\Gamma_3} \boldsymbol{N}^T h\, d\Gamma\, dt\, \phi_a \\
&+ \delta\phi \int_{\Omega} \boldsymbol{N}^T\left(t_{n-1}^{+}\right) \rho\, c_p\, \boldsymbol{N}\left(t_{n-1}^{-}\right) d\Omega\, \boldsymbol{\phi}_{n-1}
\end{aligned} \tag{2.16}$$

Since $\delta\phi$ is arbitrary, we cancel it out. Next, we separate spatial and temporal integrals and use $h\left(x,t\right) = h\left(x\right)\ h\left(t\right)$, $G\left(x,t\right) = G\left(x\right)\ G\left(t\right)$, and $q\left(x,t\right) = q\left(x\right)\ q\left(t\right)$:

$$
\begin{aligned}
&\Bigg[\int_{I_n} \boldsymbol{N}_t^T\, \dot{\boldsymbol{N}}_t\, dt \otimes \int_{\Omega} \boldsymbol{N}_x^T\, \rho\, c_p\, \boldsymbol{N}_x\, d\Omega \\
&+ \int_{I_n} \boldsymbol{N}_t^T\, \boldsymbol{N}_t\, dt \otimes \int_{\Omega} \boldsymbol{B}_x^T\, k\, \boldsymbol{B}_x\, d\Omega \\
&+ \boldsymbol{N}_t^T(t_{n-1}^+)\, \boldsymbol{N}_t(t_{n-1}^+) \otimes \int_{\Omega} \boldsymbol{N}_x^T\, \rho\, c_p\, \boldsymbol{N}_x\, d\Omega \\
&+ \int_{I_n} \boldsymbol{N}_t^T\, h(t)\, \boldsymbol{N}_t\, dt \otimes \int_{\Gamma_3} \boldsymbol{N}_x^T\, h(x)\, \boldsymbol{N}_x\, d\Gamma\Bigg] \boldsymbol{\phi}_n \\
&= \int_{I_n} \boldsymbol{N}_t^T\, G(t)\, dt \otimes \int_{\Omega} \boldsymbol{N}_x^T\, G(x)\, d\Omega \\
&+ \int_{I_n} \boldsymbol{N}_t^T\, q(t)\, dt \otimes \int_{\Gamma_N} \boldsymbol{N}_x^T\, q(x)\, d\Gamma \\
&+ \int_{I_n} \boldsymbol{N}_t^T\, h(t)\, dt \otimes \int_{\Gamma_3} \boldsymbol{N}_x^T\, h(x)\, d\Gamma\, \phi_a \\
&+ \left[\boldsymbol{N}_t^T(t_{n-1}^+)\, \boldsymbol{N}_t(t_{n-1}^-) \otimes \int_{\Omega} \boldsymbol{N}_x^T\, \rho\, c_p\, \boldsymbol{N}_x\, d\Omega\right] \boldsymbol{\phi}_{n-1}
\end{aligned}
\tag{2.17}
$$

where $\boldsymbol{\phi}_n$ and $\boldsymbol{\phi}_{n-1}$ are space-time temperature vectors for the current and previous time steps, respectively. We define the following:

$$\boldsymbol{C} = \int_{\Omega} \boldsymbol{N}_x^T\, \rho\, c_p\, \boldsymbol{N}_x\, d\Omega$$

$$\boldsymbol{K} = \int_{\Omega} \boldsymbol{B}_x^T\, k\, \boldsymbol{B}_x\, d\Omega$$

$$\boldsymbol{H} = \int_{\Gamma_3} \boldsymbol{N}_x^T\, h\left(x\right)\ \boldsymbol{N}_x\, d\Gamma$$

$$\boldsymbol{PG} = \int_{\Omega} \boldsymbol{N}_x^T\, G\left(x\right)\ d\Omega$$

$$\boldsymbol{Pq} = \int_{\Gamma_N} \boldsymbol{N}_x^T\, q\left(x\right)\ d\Gamma$$

$$\boldsymbol{Ph} = \int_{\Gamma_3} \boldsymbol{N}_x^T\, h\left(x\right)\ d\Gamma\, \phi_a$$

$$\boldsymbol{INND} = \int_{I_n} \boldsymbol{N}_t^T\, \dot{\boldsymbol{N}}_t\, dt$$

$$\boldsymbol{INN} = \int_{I_n} \boldsymbol{N}_t^T\, \boldsymbol{N}_t\, dt$$

$$\boldsymbol{INNh} = \int_{I_n} \boldsymbol{N}_t^T \, h\left(t\right) \, \boldsymbol{N}_t \, dt$$

$$\boldsymbol{INh} = \int_{I_n} \boldsymbol{N}_t^T \, h\left(t\right) \, dt$$

$$\boldsymbol{ING} = \int_{I_n} \boldsymbol{N}_t^T \, G\left(t\right) \, dt$$

$$\boldsymbol{INq} = \int_{I_n} \boldsymbol{N}_t^T \, q\left(t\right) \, dt$$

$$\boldsymbol{NNP} = \boldsymbol{N}_t^T\left(t_{n-1}^{+}\right) \boldsymbol{N}_t\left(t_{n-1}^{+}\right)$$

$$\boldsymbol{NNM} = \boldsymbol{N}_t^T\left(t_{n-1}^{+}\right) \boldsymbol{N}_t\left(t_{n-1}^{-}\right)$$

Using the above definitions, equation 2.17 becomes:

$$\begin{gathered} \left[\boldsymbol{INND} \otimes \boldsymbol{C} + \boldsymbol{INN} \otimes \boldsymbol{K} + \boldsymbol{NNP} \otimes \boldsymbol{C} + \boldsymbol{INNh} \otimes \boldsymbol{H}\right] \boldsymbol{\phi}_n = \\ \boldsymbol{ING} \otimes \boldsymbol{PG} + \boldsymbol{INq} \otimes \boldsymbol{Pq} + \boldsymbol{INh} \otimes \boldsymbol{Ph} + \left[\boldsymbol{NNM} \otimes \boldsymbol{C}\right] \boldsymbol{\phi}_{n-1} \end{gathered} \tag{2.18}$$

Gathering like terms:

$$\begin{gathered} \left[\left(\boldsymbol{INND} + \boldsymbol{NNP}\right) \otimes \boldsymbol{C} + \boldsymbol{INN} \otimes \boldsymbol{K} + \boldsymbol{INNh} \otimes \boldsymbol{H}\right] \boldsymbol{\phi}_n = \\ \boldsymbol{ING} \otimes \boldsymbol{PG} + \boldsymbol{INq} \otimes \boldsymbol{Pq} + \boldsymbol{INh} \otimes \boldsymbol{Ph} + \left[\boldsymbol{NNM} \otimes \boldsymbol{C}\right] \boldsymbol{\phi}_{n-1} \end{gathered} \tag{2.19}$$

If $h\left(t\right) = G\left(t\right) = q\left(t\right) = 1$, then using $\boldsymbol{IN} = \int_{I_n} \boldsymbol{N}_t^T \, dt$ equation 2.19 becomes:

$$\begin{gathered} \left[\left(\boldsymbol{INND} + \boldsymbol{NNP}\right) \otimes \boldsymbol{C} + \boldsymbol{INN} \otimes \left(\boldsymbol{K} + \boldsymbol{H}\right)\right] \boldsymbol{\phi}_n = \\ \boldsymbol{IN} \otimes \left(\boldsymbol{PG} + \boldsymbol{Pq} + \boldsymbol{Ph}\right) + \left[\boldsymbol{NNM} \otimes \boldsymbol{C}\right] \boldsymbol{\phi}_{n-1} \end{gathered} \tag{2.20}$$

Equation 2.19 can be simplified into a space-time linear system:

$$\boldsymbol{K}_{st_n}^{temp} \, \boldsymbol{\phi}_n = \boldsymbol{F}_{st_n}^{temp} \tag{2.21}$$

where:

$$\boldsymbol{K}_{st_n}^{temp} = \left[\left(\boldsymbol{INND} + \boldsymbol{NNP}\right) \otimes \boldsymbol{C} + \boldsymbol{INN} \otimes \boldsymbol{K} + \boldsymbol{INNh} \otimes \boldsymbol{H}\right] \tag{2.22}$$

$$\boldsymbol{F}_{st_n}^{temp} = \boldsymbol{ING} \otimes \boldsymbol{PG} + \boldsymbol{INq} \otimes \boldsymbol{Pq} + \boldsymbol{INh} \otimes \boldsymbol{Ph} + \left[\boldsymbol{NNM} \otimes \boldsymbol{C}\right] \boldsymbol{\phi}_{n-1} \tag{2.23}$$

The presence of $\boldsymbol{\phi}_{n-1}$naturally raises the question of what to do for the case when n=1. For this case, the last term of equation 2.23 becomes:

$$\left[\int_{\Omega_0} \boldsymbol{N}^{+T}\,\rho\,c_p\,\boldsymbol{N}_x\,d\Omega\right]\boldsymbol{\phi}_0=\left[\boldsymbol{N}_t^T\left(0^+\right)\otimes\int_{\Omega_0}\boldsymbol{N}_x^T\,\rho\,c_p\,\boldsymbol{N}_x\,d\Omega\right]\boldsymbol{\phi}_0=\left[\boldsymbol{NP_0}\otimes\boldsymbol{C}\right]\boldsymbol{\phi}_0 \tag{2.24}$$

where $\boldsymbol{NP_0}=\boldsymbol{N}_t^T\left(0^+\right)$ and $\boldsymbol{\phi}_0$ is the initial temperature vector whose size is the number of spatial nodes only. That is, $\boldsymbol{\phi}_0$ is a spatial temperature vector and not a space-time temperature vector.

2.2.3 Space-Time Matrix Formulation

The time Lagrangian shape function has the following form (Cook et al. (2002), p.86):

$$N_{t_i}=\frac{(t_1-t)\,(t_2-t)\,...\,[t_i-t]\,...\,(t_k-t)}{(t_1-t_i)\,(t_2-t_i)\,...\,[t_i-t_i]\,...\,(t_k-t_i)} \tag{2.25}$$

where the bracketed terms are omitted to obtain the i^{th} shape function. If we want non-zero accelerations for the mechanical analysis and desire to make the thermal analysis consistent, then we should choose quadratic (or higher) polynomial order time shape functions. For these quadratic time shape functions, there are three, equally-spaced nodes at t_{n-1}, $t_{n-1/2}=t_{n-1}+\frac{1}{2}\Delta t$, and $t_n=t_{n-1}+\Delta t$ where n is the time-step number:

$$\boldsymbol{N}_t=\begin{bmatrix}\frac{\left(t_{n-1/2}-t\right)(t_n-t)}{\left(t_{n-1/2}-t_{n-1}\right)(t_n-t_{n-1})}\\ \frac{(t_{n-1}-t)(t_n-t)}{\left(t_{n-1}-t_{n-1/2}\right)\left(t_n-t_{n-1/2}\right)}\\ \frac{(t_{n-1}-t)\left(t_{n-1/2}-t\right)}{(t_{n-1}-t_n)\left(t_{n-1/2}-t_n\right)}\end{bmatrix}^T \tag{2.26}$$

Or:

$$\boldsymbol{N}_t=\begin{bmatrix}\frac{1}{\Delta t^2}\left(\Delta t-t+t_{n-1}\right)\left(\Delta t-2t+2t_{n-1}\right)\\ \frac{1}{\Delta t^2}\left(4\left(t-t_{n-1}\right)\left(\Delta t-t+t_{n-1}\right)\right)\\ \frac{1}{\Delta t^2}\left(t_{n-1}-t\right)\left(\Delta t-2t+2t_{n-1}\right)\end{bmatrix}^T \tag{2.27}$$

The first time derivatives of these shape functions are:

$$\dot{\boldsymbol{N}}_t = \begin{bmatrix} \frac{1}{\Delta t^2}\left(-3\Delta t + 4t - 4t_{n-1}\right) \\ \frac{1}{\Delta t^2}\left(4\Delta t - 8t + 8t_{n-1}\right) \\ \frac{1}{\Delta t^2}\left(-\Delta t + 4t - 4t_{n-1}\right) \end{bmatrix}^T \tag{2.28}$$

Using equations 2.27 and 2.28, the definitions used in the previous section become:

$$\boldsymbol{NP_0} = \boldsymbol{N}_t^T\left(0^+\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{INND} = \begin{bmatrix} -\frac{1}{2} & \frac{2}{3} & -\frac{1}{6} \\ -\frac{2}{3} & 0 & \frac{2}{3} \\ \frac{1}{6} & -\frac{2}{3} & \frac{1}{2} \end{bmatrix}$$

$$\boldsymbol{INN} = \Delta t \begin{bmatrix} \frac{2}{15} & \frac{1}{15} & -\frac{1}{30} \\ \frac{1}{15} & \frac{8}{15} & \frac{1}{15} \\ -\frac{1}{30} & \frac{1}{15} & \frac{2}{15} \end{bmatrix}$$

$$\boldsymbol{NNP} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{NNM} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

If $h\left(t\right) = G\left(t\right) = q\left(t\right) = 1$, then:

$$\boldsymbol{INh}_{const} = \boldsymbol{ING}_{const} = \boldsymbol{INq}_{const} = \boldsymbol{IN} = \int_{I_n} \boldsymbol{N}_t^T \, dt = \Delta t \begin{bmatrix} \frac{1}{6} \\ \frac{2}{3} \\ \frac{1}{6} \end{bmatrix}$$

$$
\boldsymbol{INNh}_{const} = \boldsymbol{INN} = \Delta t \begin{bmatrix} \frac{2}{15} & \frac{1}{15} & -\frac{1}{30} \\ \frac{1}{15} & \frac{8}{15} & \frac{1}{15} \\ -\frac{1}{30} & \frac{1}{15} & \frac{2}{15} \end{bmatrix}
$$

$$
\boldsymbol{K}_{st_n}^{temp} = \begin{bmatrix} \frac{1}{2}\boldsymbol{C} + \frac{2}{15}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t & \frac{2}{3}\boldsymbol{C} + \frac{1}{15}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t & -\frac{1}{6}\boldsymbol{C} - \frac{1}{30}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t \\ -\frac{2}{3}\boldsymbol{C} + \frac{1}{15}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t & \frac{8}{15}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t & \frac{2}{3}\boldsymbol{C} + \frac{1}{15}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t \\ \frac{1}{6}\boldsymbol{C} - \frac{1}{30}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t & -\frac{2}{3}\boldsymbol{C} + \frac{1}{15}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t & \frac{1}{2}\boldsymbol{C} + \frac{2}{15}\left(\boldsymbol{K}+\boldsymbol{H}\right)\Delta t \end{bmatrix} \tag{2.29}
$$

$$
\boldsymbol{F}_{st_n}^{temp} = \begin{bmatrix} 0 & 0 & \boldsymbol{C} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{\phi}_{n-1} + \frac{\Delta t}{6} \begin{bmatrix} \boldsymbol{PG}+\boldsymbol{Pq}+\boldsymbol{Ph} \\ 4\left(\boldsymbol{PG}+\boldsymbol{Pq}+\boldsymbol{Ph}\right) \\ \boldsymbol{PG}+\boldsymbol{Pq}+\boldsymbol{Ph} \end{bmatrix} \tag{2.30}
$$

For n=1:

$$
\boldsymbol{F}_{st_1}^{temp} = \begin{bmatrix} \boldsymbol{C} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{\phi}_{0} + \frac{\Delta t}{6} \begin{bmatrix} \boldsymbol{PG}+\boldsymbol{Pq}+\boldsymbol{Ph} \\ 4\left(\boldsymbol{PG}+\boldsymbol{Pq}+\boldsymbol{Ph}\right) \\ \boldsymbol{PG}+\boldsymbol{Pq}+\boldsymbol{Ph} \end{bmatrix} \tag{2.31}
$$

If, on the other hand, $q\left(t\right) = \sin\left(\omega\, t\right)$ in the case of a cyclic heat flux, then $\boldsymbol{INq}$ becomes:

$$
\boldsymbol{INq}_{sin} = \begin{bmatrix} \frac{1}{\omega^3 \Delta t^2}\left[4\cos(\omega\,[t_{n-1}+\Delta t]) - 4\cos(\omega\, t_{n-1}) + \omega^2 \Delta t^2 \cos(\omega\, t_{n-1}) + \omega\,\Delta t \sin(\omega\,[t_{n-1}+\Delta t]) + 3\,\omega\,\Delta t \sin(\omega\, t_{n-1})\right] \\ \frac{1}{\omega^3 \Delta t^2}\left[8\cos(\omega\, t_{n-1}) - 8\cos(\omega\,[t_{n-1}+\Delta t]) - 4\,\omega\,\Delta t \sin(\omega\,[t_{n-1}+\Delta t]) - 4\,\omega\,\Delta t \sin(\omega\, t_{n-1})\right] \\ \frac{1}{\omega^3 \Delta t^2}\left[4\cos(\omega\,[t_{n-1}+\Delta t]) - 4\cos(\omega\, t_{n-1}) - \omega^2 \Delta t^2 \cos(\omega\,[t_{n-1}+\Delta t]) + 3\,\omega\,\Delta t \sin(\omega\,[t_{n-1}+\Delta t]) + \omega\,\Delta t \sin(\omega\, t_{n-1})\right] \end{bmatrix} \tag{2.32}
$$

A cyclic heat flux with a non-zero mean flux will be a combination of the previous two $\boldsymbol{INq}$ matrices above:

$$
\left(\boldsymbol{INq} \otimes \boldsymbol{Pq}\right)_{total} = \boldsymbol{INq}_{const} \otimes \boldsymbol{Pq}_{const} + \boldsymbol{INq}_{sin} \otimes \boldsymbol{Pq}_{sin} \tag{2.33}
$$

where $\boldsymbol{Pq}_{const}$ is the mean heat flux load and $\boldsymbol{Pq}_{sin}$ is the sinusoidal heat flux amplitude.

2.2.4 Thermal TFEM Implemention

The implementation of the thermal TFEM formulation is as follows:

1. Generate a mesh of the spatial domain.

2. Assemble the global, spatial capacitance, conductance, and convection-stiffness matrices, $\boldsymbol{C}$, $\boldsymbol{K}$, and $\boldsymbol{H}$, respectively.

3. Assemble the global, spatial heat source, heat flux, and convection-load vectors, $\boldsymbol{PG}$, $\boldsymbol{Pq}$, and $\boldsymbol{Ph}$, respectively.

4. Initiate the time loop:

   (a) Calculate the various time matrices (e.g., $\boldsymbol{INND}$, $\boldsymbol{INNH}$, $\boldsymbol{NNP}$, etc.) for the given time step's Δt.

   (b) Assemble the thermal space-time stiffness matrix, $\boldsymbol{K}_{st_n}^{temp}$, using equation 2.29.

   (c) Calculate the space-time thermal load vector, $\boldsymbol{F}_{st_1}^{temp}$ or $\boldsymbol{F}_{st_n}^{temp}$, using equations 2.31 or 2.30 for the first time step or subsequent time steps, respectively.

   (d) Apply boundary conditions.

   (e) Solve the linear system, equation 2.21, for the space-time temperature vector, $\boldsymbol{\phi}_n$.

   (f) Store the space-time temperature vector for the next space-time time-step, $\boldsymbol{\phi}_{n-1}$.

   (g) Repeat (a) through (f) for all time intervals.

5. End time loop.

2.3 Space-Time Finite Element Method (TFEM) for Mechanical Analysis

2.3.1 Governing Equation to Weak Form

We start with the governing equations of initial-boundary-value linear elastodynamics over a 3D spatial domain, $\varOmega$, and temporal domain, $I =]0, T[$:

$$\rho\, \ddot{\boldsymbol{u}} = \nabla\cdot \boldsymbol{\sigma} + \rho\, \boldsymbol{b} \qquad \text{on} \quad Q \equiv \varOmega \times I \tag{2.34}$$

$$\boldsymbol{\sigma} = \boldsymbol{D} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta) \tag{2.35}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on} \quad \Upsilon_u \equiv \Gamma_u \times I \tag{2.36}$$

$$\boldsymbol{n} \cdot \boldsymbol{\sigma} \left(\nabla \boldsymbol{u} \right) = \boldsymbol{h} \qquad \text{on} \quad \Upsilon_h \equiv \Gamma_h \times I \tag{2.37}$$

$$\boldsymbol{u} \left(\boldsymbol{x}, 0 \right) = \boldsymbol{u}_0 \left(\boldsymbol{x} \right) \qquad \text{on} \quad \boldsymbol{x} \in \Omega \tag{2.38}$$

$$\dot{\boldsymbol{u}} \left(\boldsymbol{x}, 0 \right) = \boldsymbol{v}_0 \left(\boldsymbol{x} \right) \qquad \text{on} \quad \boldsymbol{x} \in \Omega \tag{2.39}$$

$$\Gamma = \Gamma_u \cup \Gamma_h \tag{2.40}$$

$$\Gamma_u \cap \Gamma_h = 0 \tag{2.41}$$

where equation 2.34 is the conservation of linear momentum, Γ is the boundary of Ω and is the combination of the non-overlapping essential (Dirichlet) displacement boundary Γ_u and natural (Neumann) traction boundary Γ_h, ρ is the volumetric mass density, $\boldsymbol{b}$ is the body acceleration, $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{n}$ is the outward normal to Γ, $\bar{\boldsymbol{u}}$ and $\boldsymbol{h}$ are the prescribed boundary displacement and traction, respectively, and $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ are the initial displacement and velocity, respectively. Lastly, the superposed dots in $\dot{\boldsymbol{u}}$ and $\ddot{\boldsymbol{u}}$ denote partial derivatives with respect to time. The temporal and spatial domains and corresponding boundary terms are similar to the thermal TFEM and will not be repeated here (see section 2.2.1).

To get equation 2.34 in its weak form, we multiply by the virtual velocity, $\delta \boldsymbol{v}$, and integrate over the space-time domain:

$$\int_{Q_n} \delta \boldsymbol{v} \, \rho \, \ddot{\boldsymbol{u}} \, dQ = \int_{Q_n} \delta \boldsymbol{v} \left(\nabla \cdot \boldsymbol{\sigma} \right) dQ + \int_{Q_n} \delta \boldsymbol{v} \, \rho \, \boldsymbol{b} \, dQ \tag{2.42}$$

The first term on the RHS can be expanded by the product rule (Belytschko et al. (2014), p.153):

$$\int_{Q_n} \delta \boldsymbol{v} \left(\nabla \cdot \boldsymbol{\sigma} \right) dQ = \int_{Q_n} \left[\nabla \cdot \left(\delta \boldsymbol{v} \cdot \boldsymbol{\sigma} \right) - \nabla \delta \boldsymbol{v} \cdot \boldsymbol{\sigma} \right] dQ \tag{2.43}$$

The first term on the RHS of equation 2.43 can itself be expanded by Gauss's Theorem:

$$\int_{Q_n} \nabla \cdot \left(\delta \boldsymbol{v} \cdot \boldsymbol{\sigma} \right) dQ = \int_{I_n} \int_{\Gamma_{int}} \delta \boldsymbol{v} \left[\!\left[\boldsymbol{n} \cdot \boldsymbol{\sigma} \right]\!\right] d\Gamma \, dt + \int_{I_n} \int_{\Gamma_h} \delta \boldsymbol{v} \cdot \boldsymbol{h} \, d\Gamma \, dt \tag{2.44}$$

where $\boldsymbol{h}$ is the traction and $[\![\boldsymbol{n}\cdot\boldsymbol{\sigma}]\!]$ is the inter-element traction jump term. Both terms on the RHS of equation 2.44 have $d\Gamma$ rather than $d\Omega$ because "the test function vanishes on the complement of the traction boundaries" (Belytschko et al. (2014), p.153). The $[\![\boldsymbol{n}\cdot\boldsymbol{\sigma}]\!]$ becomes zero due to the traction continuity. Substituting these results into equation 2.42, we have:

$$\int_{Q_n} \delta\boldsymbol{v}\cdot \rho\,\ddot{\mathbf{u}}\,dQ = \int_{I_n}\int_{\Gamma_h} \delta\boldsymbol{v}\cdot \boldsymbol{h}\,d\Gamma\,dt - \int_{Q_n} \nabla\delta\boldsymbol{v}\cdot \boldsymbol{\sigma}\,dQ + \int_{Q_n} \delta\boldsymbol{v}\cdot \rho\,\boldsymbol{b}\,dQ \tag{2.45}$$

Substituting equation 2.35 into equation 2.45 and rearranging, we get:

$$\begin{aligned} &\int_{Q_n} \delta\boldsymbol{v}\cdot \rho\,\ddot{\mathbf{u}}\,dQ \\ &+ \int_{Q_n} \nabla\delta\boldsymbol{v}\cdot \left(\boldsymbol{D}:\boldsymbol{\varepsilon}\right)\,dQ = \int_{I_n}\int_{\Gamma_h} \delta\boldsymbol{v}\cdot \boldsymbol{h}\,d\Gamma\,dt \\ &\qquad + \int_{Q_n} \delta\boldsymbol{v}\cdot \rho\,\boldsymbol{b}\,dQ \\ &\qquad + \int_{Q_n} \nabla\delta\boldsymbol{v}\cdot \left(\boldsymbol{D}:\boldsymbol{\varepsilon}_\theta\right)\,dQ \end{aligned} \tag{2.46}$$

Given that $\int_{I_n}\int_{\Gamma_h} d\Gamma\,dt = \int_{(\Upsilon_h)_n} d\Upsilon$:

$$\begin{aligned} &\int_{Q_n} \delta\boldsymbol{v}\cdot \rho\,\ddot{\mathbf{u}}\,dQ \\ &+ \int_{Q_n} \nabla\delta\boldsymbol{v}\cdot \left(\boldsymbol{D}:\boldsymbol{\varepsilon}\right)\,dQ = \int_{(\Upsilon_h)_n} \delta\boldsymbol{v}\cdot \boldsymbol{h}\,d\Upsilon \\ &\qquad + \int_{Q_n} \delta\boldsymbol{v}\cdot \rho\,\boldsymbol{b}\,dQ \\ &\qquad + \int_{Q_n} \nabla\delta\boldsymbol{v}\cdot \left(\boldsymbol{D}:\boldsymbol{\varepsilon}_\theta\right)\,dQ \end{aligned} \tag{2.47}$$

We now introduce two terms to allow for discontinuities across time element boundaries:

$$\begin{aligned} \int_{\Omega} \delta\boldsymbol{v}\left(t_{n-1}^{+}\right)\cdot \rho\,[\![\dot{\boldsymbol{u}}\left(t_{n-1}\right)]\!]\,d\Omega &= \int_{\Omega} \delta\boldsymbol{v}\left(t_{n-1}^{+}\right)\cdot \rho\,\dot{\boldsymbol{u}}\left(t_{n-1}^{+}\right)\,d\Omega \\ &\quad - \int_{\Omega} \delta\boldsymbol{v}\left(t_{n-1}^{+}\right)\cdot \rho\,\dot{\boldsymbol{u}}\left(t_{n-1}^{-}\right)\,d\Omega \end{aligned} \tag{2.48}$$

$$
\begin{aligned}
\int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot[\![\boldsymbol{\sigma}\left(t_{n-1}\right)]\!]\;d\Omega &= \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot[\![\boldsymbol{D}:\left(\boldsymbol{\varepsilon}\left(t_{n-1}\right)-\boldsymbol{\varepsilon}_{\theta}\left(t_{n-1}\right)\right)]\!]\;d\Omega \\
&= \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}\left(t_{n-1}^{+}\right)\;d\Omega \\
&- \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}_{\theta}\left(t_{n-1}^{+}\right)\;d\Omega \\
&- \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}\left(t_{n-1}^{-}\right)\;d\Omega \\
&+ \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}_{\theta}\left(t_{n-1}^{-}\right)\;d\Omega
\end{aligned}
\tag{2.49}
$$

Adding equations 2.48 and 2.49 to the LHS of equation 2.45 and moving everything to one side:

$$
\begin{aligned}
0 &= \int_{Q_n} \delta\boldsymbol{v}\cdot\rho\,\ddot{\boldsymbol{u}}\,dQ \\
&+ \int_{Q_n} \nabla\delta\boldsymbol{v}\cdot\left(\boldsymbol{D}:\boldsymbol{\varepsilon}\right)\;dQ \\
&- \int_{Q_n} \nabla\delta\boldsymbol{v}\cdot\left(\boldsymbol{D}:\boldsymbol{\varepsilon}_{\theta}\right)\;dQ \\
&+ \int_{\Omega} \delta\boldsymbol{v}\left(t_{n-1}^{+}\right)\cdot\rho\,\dot{\boldsymbol{u}}\left(t_{n-1}^{+}\right)\;d\Omega \\
&+ \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}\left(t_{n-1}^{+}\right)\;d\Omega \\
&- \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}_{\theta}\left(t_{n-1}^{+}\right)\;d\Omega \\
&- \int_{(\Upsilon_h)_n} \delta\boldsymbol{v}\cdot\boldsymbol{h}\,d\Upsilon \\
&- \int_{Q_n} \delta\boldsymbol{v}\cdot\rho\,\boldsymbol{b}\,dQ \\
&- \int_{\Omega} \delta\boldsymbol{v}\left(t_{n-1}^{+}\right)\cdot\rho\,\dot{\boldsymbol{u}}\left(t_{n-1}^{-}\right)\;d\Omega \\
&- \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}\left(t_{n-1}^{-}\right)\;d\Omega \\
&+ \int_{\Omega} \nabla\delta\boldsymbol{u}\left(t_{n-1}^{+}\right)\cdot\boldsymbol{D}:\boldsymbol{\varepsilon}_{\theta}\left(t_{n-1}^{-}\right)\;d\Omega
\end{aligned}
\tag{2.50}
$$

The presence of $\dot{\boldsymbol{u}}\left(t_{n-1}^{-}\right)$, $\boldsymbol{\varepsilon}\left(t_{n-1}^{-}\right)$, and $\boldsymbol{\varepsilon}_{\theta}\left(t_{n-1}^{-}\right)$ naturally raises the question of what to do for the case when n=1. For this case, the last three terms evaluated at $\left(t_{n-1}^{-}\right)$ in equation

2.50 become:

$$-\int_{\Omega} \delta \boldsymbol{v}\left(0^{+}\right) \cdot \rho\, \boldsymbol{v}_0 \, d\Omega - \int_{\Omega} \nabla \delta \boldsymbol{u}\left(0^{+}\right) \cdot \boldsymbol{\varepsilon}_0 \, d\Omega + \int_{\Omega} \nabla \delta \boldsymbol{u}\left(0^{+}\right) \cdot \boldsymbol{D} : \boldsymbol{\varepsilon}_{\theta}^{0} \, d\Omega \tag{2.51}$$

where $\boldsymbol{v}_0$ is the initial velocity vector, $\boldsymbol{\varepsilon}_0$ is the initial total strain tensor, and $\boldsymbol{\varepsilon}_{\theta}^{0}$ is the initial thermal strain tensor. Note that $\boldsymbol{v}_0$ is a *spatial* velocity vector and not a space-time velocity vector.

2.3.2 Discretization

We use the following definitions for the discretization of the weak form:

$$\nabla \boldsymbol{N}_x = \boldsymbol{B}_x$$

$$\boldsymbol{N}(x,t) = \boldsymbol{N}_t \otimes \boldsymbol{N}_x$$

$$\boldsymbol{B}(x,t) = \boldsymbol{N}_t \otimes \boldsymbol{B}_x$$

$$\boldsymbol{h} = \boldsymbol{h}(x,t) = h(t)\ \boldsymbol{h}(x)$$

$$\boldsymbol{b} = \boldsymbol{b}(x,t) = b(t)\ \boldsymbol{b}(x)$$

$$\boldsymbol{\varepsilon}_{\theta} = \boldsymbol{\varepsilon}_{\theta}(x,t) = \varepsilon_{\theta}(t)\ \boldsymbol{\varepsilon}_{\theta}(x)$$

$$\boldsymbol{u}(x,t) = \sum_{I=1}^{n} N_I(x,t)\, \boldsymbol{d}_I$$

$$\nabla \boldsymbol{u}(x,t) = \sum_{I=1}^{n} \nabla N_I(x,t)\, \boldsymbol{d}_I = \sum_{I=1}^{n} B_I(x,t)\, \boldsymbol{d}_I$$

$$\dot{\boldsymbol{u}}(x,t) = \sum_{I=1}^{n} \dot{N}_I(x,t)\, \boldsymbol{d}_I$$

$$\nabla \dot{\boldsymbol{u}}(x,t) = \sum_{I=1}^{n} \nabla \dot{N}_I(x,t)\, \boldsymbol{d}_I = \sum_{I=1}^{n} \dot{B}_I(x,t)\, \boldsymbol{d}_I$$

$$\ddot{\boldsymbol{u}}(x,t) = \sum_{I=1}^{n} \ddot{N}_I(x,t)\, \boldsymbol{d}_I$$

$$\delta \boldsymbol{u}(x,t) = \sum_{I=1}^{n} N_I(x,t)\,\delta \boldsymbol{d}_I$$

$$\delta \boldsymbol{v}(x,t) = \delta \dot{\boldsymbol{u}}(x,t) = \sum_{I=1}^{n} \dot{N}_I(x,t)\,\delta \boldsymbol{d}_I$$

$$\nabla \delta \boldsymbol{u}(x,t) = \sum_{I=1}^{n} \nabla N_I(x,t)\,\delta \boldsymbol{d}_I = \sum_{I=1}^{n} B_I(x,t)\,\delta \boldsymbol{d}_I$$

$$\nabla \delta \boldsymbol{v}(x,t) = \nabla \dot{\delta \boldsymbol{u}}(x,t) = \sum_{I=1}^{n} \nabla \dot{N}_I(x,t)\,\delta \boldsymbol{d}_I = \sum_{I=1}^{n} \dot{B}_I(x,t)\,\delta \boldsymbol{d}_I$$

where $\boldsymbol{d}_I$ are the nodal displacements. Substituting the above discretized equations into equation 2.50 and moving the variations in displacement outside the integrals:

$$\begin{aligned}
0 = {} & \delta \boldsymbol{d}_n \int_{(Q_n)^\Sigma} \dot{\boldsymbol{N}}^T \rho\, \ddot{\boldsymbol{N}}\, \boldsymbol{d}_n\, dQ \\
& + \delta \boldsymbol{d}_n \int_{Q_n} \dot{\boldsymbol{B}}^T \boldsymbol{D}\, \boldsymbol{\varepsilon}\, dQ \\
& - \delta \boldsymbol{d}_n \int_{Q_n} \dot{\boldsymbol{B}}^T \boldsymbol{D}\, \boldsymbol{\varepsilon}_\theta\, dQ \\
& + \delta \boldsymbol{d}_n \int_{\Omega} \dot{\boldsymbol{N}}^T \left(t_{n-1}^+\right) \rho\, \dot{\boldsymbol{N}} \left(t_{n-1}^+\right) \boldsymbol{d}_n\, d\Omega \\
& + \delta \boldsymbol{d}_n \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \boldsymbol{\varepsilon} \left(t_{n-1}^+\right)\, d\Omega \\
& - \delta \boldsymbol{d}_n \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \boldsymbol{\varepsilon}_\theta \left(t_{n-1}^+\right)\, d\Omega \\
& - \delta \boldsymbol{d}_n \int_{(\Upsilon_h)_n} \dot{\boldsymbol{N}}^T \boldsymbol{h}\, d\Upsilon \\
& - \delta \boldsymbol{d}_n \int_{Q_n} \dot{\boldsymbol{N}}^T \rho\, \boldsymbol{b}\, dQ \\
& - \delta \boldsymbol{d}_n \int_{\Omega} \dot{\boldsymbol{N}}^T \left(t_{n-1}^+\right) \rho\, \dot{\boldsymbol{N}} \left(t_{n-1}^-\right) \boldsymbol{d}_{n-1}\, d\Omega \\
& - \delta \boldsymbol{d}_n \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \boldsymbol{\varepsilon} \left(t_{n-1}^-\right)\, d\Omega \\
& + \delta \boldsymbol{d}_n \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \boldsymbol{\varepsilon}_\theta \left(t_{n-1}^-\right)\, d\Omega
\end{aligned} \tag{2.52}$$

For the case where n=1, the last three terms of equation 2.52 become:

$$-\delta \boldsymbol{d}_n \int_{\Omega} \dot{\boldsymbol{N}}^T \left(0^+\right) \rho\, \boldsymbol{v}_0\, d\Omega - \delta \boldsymbol{d}_n \int_{\Omega} \boldsymbol{B}^T \left(0^+\right) \boldsymbol{D}\, \boldsymbol{\varepsilon}_0\, d\Omega + \delta \boldsymbol{d}_n \int_{\Omega} \boldsymbol{B}^T \left(0^+\right) \boldsymbol{D}\, \boldsymbol{\varepsilon}_\theta^0\, d\Omega \tag{2.53}$$

Since the virtual displacements are arbitrary, they may be dropped. Furthermore, the nodal displacements are independent of the integrals and may be moved outside:

$$
\begin{aligned}
0 = & \int_{(Q_n)^\Sigma} \dot{\boldsymbol{N}}^T \rho \, \ddot{\boldsymbol{N}} \, dQ \, \boldsymbol{d}_n \\
& + \int_{Q_n} \dot{\boldsymbol{B}}^T \boldsymbol{D} \, \boldsymbol{\varepsilon} \, dQ \\
& - \int_{Q_n} \dot{\boldsymbol{B}}^T \boldsymbol{D} \, \boldsymbol{\varepsilon}_\theta \, dQ \\
& + \int_{\Omega} \dot{\boldsymbol{N}}^T \left(t_{n-1}^+\right) \rho \, \dot{\boldsymbol{N}} \left(t_{n-1}^+\right) \, d\Omega \, \boldsymbol{d}_n \\
& + \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D} \, \boldsymbol{\varepsilon} \left(t_{n-1}^+\right) \, d\Omega \\
& - \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D} \, \boldsymbol{\varepsilon}_\theta \left(t_{n-1}^+\right) \, d\Omega \\
& - \int_{(\Upsilon_h)_n} \dot{\boldsymbol{N}}^T \boldsymbol{h} \, d\Upsilon \\
& - \int_{Q_n} \dot{\boldsymbol{N}}^T \rho \, \boldsymbol{b} \, dQ \\
& - \int_{\Omega} \dot{\boldsymbol{N}}^T \left(t_{n-1}^+\right) \rho \, \dot{\boldsymbol{N}} \left(t_{n-1}^-\right) d\Omega \, \boldsymbol{d}_{n-1} \\
& - \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D} \, \boldsymbol{\varepsilon} \left(t_{n-1}^-\right) d\Omega \\
& + \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D} \, \boldsymbol{\varepsilon}_\theta \left(t_{n-1}^-\right) \, d\Omega
\end{aligned}
\tag{2.54}
$$

Likewise for the case where n=1, the last three terms of equation 2.54 become:

$$
-\int_{\Omega} \dot{\boldsymbol{N}}^T \left(0^+\right) \, \rho \, d\Omega \, \boldsymbol{v}_0 - \int_{\Omega} \boldsymbol{B}^T \left(0^+\right) \cdot \boldsymbol{D} \, \boldsymbol{\varepsilon}_0 \, d\Omega + \int_{\Omega} \boldsymbol{B}^T \left(0^+\right) \boldsymbol{D} \, \boldsymbol{\varepsilon}_\theta^0 \, d\Omega \tag{2.55}
$$

Substituting $\boldsymbol{\varepsilon} = \boldsymbol{B} \, \boldsymbol{d}$ and $\boldsymbol{\varepsilon}_\theta = \alpha \, \Delta T \, \boldsymbol{I}$ where $\boldsymbol{d}$ is the displacement vector, α is the coefficient of thermal expansion, $\Delta T = \Delta T\,(t)\ \Delta T\,(x) = T\,(x, t) - T_{ref}$, T_{ref} is the reference temperature, and $\boldsymbol{I}$ is the second-order identity tensor into equation 2.54, we have:

$$
\begin{aligned}
0 = &\int_{(Q_n)^\Sigma} \dot{\boldsymbol{N}}^T \rho\, \ddot{\boldsymbol{N}}\, dQ\, \boldsymbol{d}_n \\
&+ \int_{Q_n} \dot{\boldsymbol{B}}^T \boldsymbol{D}\, \boldsymbol{B}\, dQ\, \boldsymbol{d}_n \\
&- \int_{Q_n} \dot{\boldsymbol{B}}^T \boldsymbol{D}\, \alpha\, \Delta T\, \boldsymbol{I}\, dQ \\
&+ \int_{\Omega} \dot{\boldsymbol{N}}^T \left(t_{n-1}^+\right) \rho\, \dot{\boldsymbol{N}} \left(t_{n-1}^+\right) d\Omega\, \boldsymbol{d}_n \\
&+ \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \boldsymbol{B} \left(t_{n-1}^+\right) d\Omega\, \boldsymbol{d}_n \\
&- \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \alpha \left[T\left(t_{n-1}^+\right) - T_{ref}\right] \boldsymbol{I}\, d\Omega \\
&- \int_{(\Upsilon_h)_n} \dot{\boldsymbol{N}}^T \boldsymbol{h}\, d\Upsilon \\
&- \int_{Q_n} \dot{\boldsymbol{N}}^T \rho\, \boldsymbol{b}\, dQ \\
&- \int_{\Omega} \dot{\boldsymbol{N}}^T \left(t_{n-1}^+\right) \rho\, \dot{\boldsymbol{N}} \left(t_{n-1}^-\right) d\Omega\, \boldsymbol{d}_{n-1} \\
&- \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \boldsymbol{B} \left(t_{n-1}^-\right) d\Omega\, \boldsymbol{d}_{n-1} \\
&+ \int_{\Omega} \boldsymbol{B}^T \left(t_{n-1}^+\right) \boldsymbol{D}\, \alpha \left[T\left(t_{n-1}^-\right) - T_{ref}\right] \boldsymbol{I}\, d\Omega
\end{aligned}
\tag{2.56}
$$

Likewise for the case where n=1, the last three terms of equation 2.56 become:

$$
\begin{aligned}
&- \int_{\Omega} \dot{\boldsymbol{N}}^T \left(0^+\right) \rho\, d\Omega\, \boldsymbol{v}_0 \\
&- \int_{\Omega} \boldsymbol{B}^T \left(0^+\right) \boldsymbol{D}\, \boldsymbol{B}_x\, d\Omega\, \boldsymbol{u}_0 \\
&+ \int_{\Omega} \boldsymbol{B}^T \left(0^+\right) \boldsymbol{D}\, \alpha \left(T_0 - T_{ref}\right) \boldsymbol{I}\, d\Omega
\end{aligned}
\tag{2.57}
$$

where $\boldsymbol{u}_0$ is the initial displacement vector and T_0 is the initial temperature. Note that $\boldsymbol{u}_0$ is a *spatial* displacement vector and not a space-time displacement vector. Separating the spatial and temporal integrals and combining the thermal jump terms:

$$
\begin{aligned}
0 = \Bigg[& \int_{I_n} \dot{\boldsymbol{N}}_t^T \ddot{\boldsymbol{N}}_t \, dt \otimes \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \\
& + \int_{I_n} \dot{\boldsymbol{N}}_t^T \boldsymbol{N}_t \, dt \otimes \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \\
& + \dot{\boldsymbol{N}}_t^T \left(t_{n-1}^+\right) \dot{\boldsymbol{N}}_t \left(t_{n-1}^+\right) \otimes \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \\
& + \boldsymbol{N}_t^T \left(t_{n-1}^+\right) \boldsymbol{N}_t \left(t_{n-1}^+\right) \otimes \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \Bigg] \, \boldsymbol{d}_n \\
& - \int_{I_n} \dot{\boldsymbol{N}}_t^T \, h\left(t\right) \, dt \otimes \int_{\Gamma_n} \boldsymbol{N}_x^T \, \boldsymbol{h}\left(x\right) \, d\Gamma \\
& - \int_{I_n} \dot{\boldsymbol{N}}_t^T \, b\left(t\right) \, dt \otimes \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{b}\left(x\right) \, d\Omega \\
& - \int_{I_n} \dot{\boldsymbol{N}}_t^T \, \Delta T\left(t\right) \, dt \otimes \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \Delta T\left(x\right) \, \boldsymbol{I} \, d\Omega \\
& - \Bigg[\dot{\boldsymbol{N}}_t^T \left(t_{n-1}^+\right) \dot{\boldsymbol{N}}_t \left(t_{n-1}^-\right) \otimes \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \\
& + \boldsymbol{N}_t^T \left(t_{n-1}^+\right) \boldsymbol{N}_t \left(t_{n-1}^-\right) \otimes \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \Bigg] \, \boldsymbol{d}_{n-1} \\
& - \boldsymbol{N}_t^T \left(t_{n-1}^+\right) \otimes \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \left[T\left(t_{n-1}^+\right) - T\left(t_{n-1}^-\right)\right] \boldsymbol{I} \, d\Omega
\end{aligned}
\tag{2.58}
$$

When n=1, the last three terms become:

$$
\begin{aligned}
& - \dot{\boldsymbol{N}}_t^T \left(0^+\right) \otimes \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \, \boldsymbol{v}_0 \\
& - \boldsymbol{N}_t^T \left(0^+\right) \otimes \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \, \boldsymbol{u}_0 \\
& - \boldsymbol{N}_t^T \left(0^+\right) \otimes \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \left[T\left(0^+\right) - T_0\right] \boldsymbol{I} \, d\Omega
\end{aligned}
\tag{2.59}
$$

We define the following:

$$
\begin{aligned}
\boldsymbol{M} &= \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \\
\boldsymbol{K} &= \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \\
\boldsymbol{H} &= \int_{\Gamma_n} \boldsymbol{N}_x^T \, \boldsymbol{h}\left(x\right) \, d\Gamma \\
\boldsymbol{B}_f &= \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{b}\left(x\right) \, d\Omega
\end{aligned}
$$

$$\boldsymbol{\Theta} = \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \Delta T\left(x\right) \, \boldsymbol{I} \, d\Omega$$

$$\boldsymbol{\Theta J} = \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \left[T\left(t_{n-1}^{+}\right) - T\left(t_{n-1}^{-}\right)\right] \boldsymbol{I} \, d\Omega$$

$$\boldsymbol{\Theta J_0} = \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \left[T\left(0^{+}\right) - T_0\right] \boldsymbol{I} \, d\Omega$$

$$\boldsymbol{INN3} = \int_{I_n} \dot{\boldsymbol{N}}_t^T \, \ddot{\boldsymbol{N}}_t \, dt$$

$$\boldsymbol{INN1} = \int_{I_n} \dot{\boldsymbol{N}}_t^T \, \boldsymbol{N}_t \, dt$$

$$\boldsymbol{IN1H} = \int_{I_n} \dot{\boldsymbol{N}}_t^T \, h\left(t\right) \, dt$$

$$\boldsymbol{IN1B} = \int_{I_n} \dot{\boldsymbol{N}}_t^T \, b\left(t\right) \, dt$$

$$\boldsymbol{IN1\Theta} = \int_{I_n} \dot{\boldsymbol{N}}_t^T \, \Delta T\left(t\right) \, dt$$

$$\boldsymbol{NN2P} = \dot{\boldsymbol{N}}_t^T \left(t_{n-1}^{+}\right) \dot{\boldsymbol{N}}_t \left(t_{n-1}^{+}\right)$$

$$\boldsymbol{NN2M} = \dot{\boldsymbol{N}}_t^T \left(t_{n-1}^{+}\right) \dot{\boldsymbol{N}}_t \left(t_{n-1}^{-}\right)$$

$$\boldsymbol{NNP} = \boldsymbol{N}_t^T \left(t_{n-1}^{+}\right) \boldsymbol{N}_t \left(t_{n-1}^{+}\right)$$

$$\boldsymbol{NNM} = \boldsymbol{N}_t^T \left(t_{n-1}^{+}\right) \boldsymbol{N}_t \left(t_{n-1}^{-}\right)$$

$$\boldsymbol{NP} = \boldsymbol{N}_t^T \left(t_{n-1}^{+}\right)$$

$$\boldsymbol{N1P_0} = \dot{\boldsymbol{N}}_t^T \left(0^{+}\right)$$

$$\boldsymbol{NP_0} = \boldsymbol{N}_t^T \left(0^{+}\right)$$

Using the above definitions and moving the negative terms to the other side, equation 2.58 becomes:

$$\begin{aligned}
&[(\boldsymbol{INN3} + \boldsymbol{NN2P}) \otimes \boldsymbol{M} + \\
&(\boldsymbol{INN1} + \boldsymbol{NNP}) \otimes \boldsymbol{K}] \, \boldsymbol{d}_n = \boldsymbol{IN1H} \otimes \boldsymbol{H} + \boldsymbol{IN1B} \otimes \boldsymbol{B}_f \\
&\qquad\qquad + \boldsymbol{IN1\Theta} \otimes \boldsymbol{\Theta} + \boldsymbol{NP} \otimes \boldsymbol{\Theta J} \\
&\qquad\qquad + [\boldsymbol{NN2M} \otimes \boldsymbol{M} + \boldsymbol{NNM} \otimes \boldsymbol{K}] \, \boldsymbol{d}_{n-1}
\end{aligned} \tag{2.60}$$

For n=1, the last three terms on the RHS of equation 2.60 become:

$$\left[\boldsymbol{N1P_0} \otimes \boldsymbol{M}\right] \boldsymbol{v}_0 + \left[\boldsymbol{NP_0} \otimes \boldsymbol{K}\right] \boldsymbol{u}_0 + \boldsymbol{NP_0} \otimes \boldsymbol{\Theta J_0} \tag{2.61}$$

Equation 2.60 can be simplified into a space-time linear system:

$$\boldsymbol{K}_{st_n}^{mech} \boldsymbol{d}_n = \boldsymbol{F}_{st_n}^{mech} \tag{2.62}$$

where:

$$\boldsymbol{K}_{st_n}^{mech} = \left[\left(\boldsymbol{INN3} + \boldsymbol{NN2P}\right) \otimes \boldsymbol{M} + \left(\boldsymbol{INN1} + \boldsymbol{NNP}\right) \otimes \boldsymbol{K}\right] \tag{2.63}$$

$$\begin{aligned} \boldsymbol{F}_{st_n}^{mech} &= \boldsymbol{IN1H} \otimes \boldsymbol{H} + \boldsymbol{IN1B} \otimes \boldsymbol{B}_f \\ &+ \boldsymbol{IN1\Theta} \otimes \boldsymbol{\Theta} + \boldsymbol{NP} \otimes \boldsymbol{\Theta J} \\ &+ \left[\boldsymbol{NN2M} \otimes \boldsymbol{M} + \boldsymbol{NNM} \otimes \boldsymbol{K}\right] \boldsymbol{d}_{n-1} \end{aligned} \tag{2.64}$$

$$\begin{aligned} \boldsymbol{F}_{st_1}^{mech} &= \boldsymbol{IN1H} \otimes \boldsymbol{H} + \boldsymbol{IN1B} \otimes \boldsymbol{B}_f \\ &+ \boldsymbol{IN1\Theta} \otimes \boldsymbol{\Theta} + \boldsymbol{NP_0} \otimes \boldsymbol{\Theta J_0} \\ &+ \left[\boldsymbol{N1P_0} \otimes \boldsymbol{M}\right] \boldsymbol{v}_0 + \left[\boldsymbol{NP_0} \otimes \boldsymbol{K}\right] \boldsymbol{u}_0 \end{aligned} \tag{2.65}$$

2.3.3 Space-Time Matrix Formulation

Assuming a quadratic interpolation (see section 2.2.3), the shape functions and their first and second derivatives become:

$$\boldsymbol{N}_t = \begin{bmatrix} \frac{1}{\Delta t^2}\left(\Delta t - t + t_{n-1}\right)\left(\Delta t - 2t + 2t_{n-1}\right) \\ \frac{1}{\Delta t^2}\left(4\left(t - t_{n-1}\right)\left(\Delta t - t + t_{n-1}\right)\right) \\ \frac{1}{\Delta t^2}\left(t_{n-1} - t\right)\left(\Delta t - 2t + 2t_{n-1}\right) \end{bmatrix}^T$$

$$\dot{\boldsymbol{N}}_t = \begin{bmatrix} \frac{1}{\Delta t^2}\left(-3\Delta t + 4t - 4t_{n-1}\right) \\ \frac{1}{\Delta t^2}\left(4\Delta t - 8t + 8t_{n-1}\right) \\ \frac{1}{\Delta t^2}\left(-\Delta t + 4t - 4t_{n-1}\right) \end{bmatrix}^T$$

$$\ddot{\boldsymbol{N}}_t = \begin{bmatrix} \frac{4}{\Delta t^2} \\ \frac{-8}{\Delta t^2} \\ \frac{4}{\Delta t^2} \end{bmatrix}^T$$

The time matrices then become:

$$\boldsymbol{NP} = \boldsymbol{NP_0} = \boldsymbol{N}_t^T\left(0^+\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{N1P_0} = \dot{\boldsymbol{N}}_t^T\left(0^+\right) = \frac{1}{\Delta t}\begin{bmatrix} -3 \\ 4 \\ -1 \end{bmatrix}$$

$$\boldsymbol{INN3} = \frac{1}{\Delta t^2}\begin{bmatrix} -4 & 8 & -4 \\ 0 & 0 & 0 \\ 4 & -8 & 4 \end{bmatrix}$$

$$\boldsymbol{INN1} = \begin{bmatrix} -\frac{1}{2} & -\frac{2}{3} & \frac{1}{6} \\ \frac{2}{3} & 0 & -\frac{2}{3} \\ -\frac{1}{6} & \frac{2}{3} & \frac{1}{2} \end{bmatrix}$$

$$\boldsymbol{NN2P} = \frac{1}{\Delta t^2}\begin{bmatrix} 9 & -12 & 3 \\ -12 & 16 & -4 \\ 3 & -4 & 1 \end{bmatrix}$$

$$\boldsymbol{NN2M} = \frac{1}{\Delta t^2}\begin{bmatrix} -3 & 12 & -9 \\ 4 & -16 & 12 \\ -1 & 4 & -3 \end{bmatrix}$$

$$\boldsymbol{NNP} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{NNM} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Assuming $h\,(t) = b\,(t) = \Delta T\,(t) = 1$, $\boldsymbol{IN1H}$, $\boldsymbol{IN1B}$, and $\boldsymbol{IN1\Theta}$ become:

$$\boldsymbol{IN1H}_{const} = \boldsymbol{IN1B}_{const} = \boldsymbol{IN1\Theta}_{const} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

Using the above, the parts of the linear system become:

$$\boldsymbol{K}^{mech}_{st_n} = \begin{bmatrix} \frac{5}{\Delta t^2}\boldsymbol{M} + \frac{1}{2}\boldsymbol{K} & -\frac{4}{\Delta t^2}\boldsymbol{M} - \frac{2}{3}\boldsymbol{K} & -\frac{1}{\Delta t^2}\boldsymbol{M} + \frac{1}{6}\boldsymbol{K} \\ -\frac{12}{\Delta t^2}\boldsymbol{M} + \frac{2}{3}\boldsymbol{K} & \frac{16}{\Delta t^2}\boldsymbol{M} & -\frac{4}{\Delta t^2}\boldsymbol{M} - \frac{2}{3}\boldsymbol{K} \\ \frac{7}{\Delta t^2}\boldsymbol{M} - \frac{1}{6}\boldsymbol{K} & -\frac{12}{\Delta t^2}\boldsymbol{M} + \frac{2}{3}\boldsymbol{K} & \frac{5}{\Delta t^2}\boldsymbol{M} + \frac{1}{2}\boldsymbol{K} \end{bmatrix} \tag{2.66}$$

$$\boldsymbol{F}^{mech}_{st_n} = \begin{bmatrix} -\frac{3}{\Delta t^2}\boldsymbol{M} & \frac{12}{\Delta t^2}\boldsymbol{M} & -\frac{9}{\Delta t^2}\boldsymbol{M} + \frac{1}{6}\boldsymbol{K} \\ \frac{4}{\Delta t^2}\boldsymbol{M} & -\frac{16}{\Delta t^2}\boldsymbol{M} & \frac{12}{\Delta t^2}\boldsymbol{M} \\ -\frac{1}{\Delta t^2}\boldsymbol{M} & \frac{4}{\Delta t^2}\boldsymbol{M} & -\frac{3}{\Delta t^2}\boldsymbol{M} \end{bmatrix} \boldsymbol{d}_{n-1} + \begin{bmatrix} -\left(\boldsymbol{B}_f + \boldsymbol{H} + \boldsymbol{\Theta}\right) + \boldsymbol{\Theta}\boldsymbol{J} \\ 0 \\ \boldsymbol{B}_f + \boldsymbol{H} + \boldsymbol{\Theta} \end{bmatrix} \tag{2.67}$$

$$\boldsymbol{F}^{mech}_{st_1} = \begin{bmatrix} -\frac{3}{\Delta t}\boldsymbol{M}\,\boldsymbol{v}_o + \boldsymbol{K}\,\boldsymbol{u}_0 - \left(\boldsymbol{B}_f + \boldsymbol{H} + \boldsymbol{\Theta}\right) + \boldsymbol{\Theta}\boldsymbol{J}_0 \\ \frac{4}{\Delta t}\boldsymbol{M}\,\boldsymbol{v}_o \\ -\frac{1}{\Delta t}\boldsymbol{M}\,\boldsymbol{v}_o + \boldsymbol{B}_f + \boldsymbol{H} + \boldsymbol{\Theta} \end{bmatrix} \tag{2.68}$$

If, on the other hand, $h\,(t) = \sin\left(\omega\,t\right)$ in the case of cyclic mechanical fatigue loading, then $\boldsymbol{IN1H}$ becomes:

$$\boldsymbol{IN1H}_{sin} =$$

$$\begin{bmatrix} \frac{1}{\omega^2\Delta t^2}\left[4\sin(\omega\,[t_{n-1}+\Delta t]) - 4\sin(\omega\,t_{n-1}) - \omega\,\Delta t\cos(\omega\,[t_{n-1}+\Delta t]) - 3\,\omega\,\Delta t\cos(\omega\,t_{n-1})\right] \\ \frac{1}{\omega^2\Delta t^2}\left[8\sin(\omega\,t_{n-1}) - 8\sin(\omega\,[t_{n-1}+\Delta t]) + 4\,\omega\,\Delta t\cos(\omega\,[t_{n-1}+\Delta t]) + 4\,\omega\,\Delta t\cos(\omega\,t_{n-1})\right] \\ \frac{1}{\omega^2\Delta t^2}\left[4\sin(\omega\,[t_{n-1}+\Delta t]) - 4\sin(\omega\,t_{n-1}) - 3\,\omega\,\Delta t\cos(\omega\,[t_{n-1}+\Delta t]) - \omega\,\Delta t\cos(\omega\,t_{n-1})\right] \end{bmatrix} \tag{2.69}$$

A cyclic fatigue loading with a non-zero mean stress will be a combination of the previous two $\boldsymbol{INH}1$ matrices above:

$$(\boldsymbol{IN}1\boldsymbol{H} \otimes \boldsymbol{H})_{total} = \boldsymbol{IN}1\boldsymbol{H}_{const} \otimes \boldsymbol{H}_{const} + \boldsymbol{IN}1\boldsymbol{H}_{sin} \otimes \boldsymbol{H}_{sin} = \left[\begin{array}{c} -\boldsymbol{H}_{const} + \frac{1}{\omega^2 \Delta t^2}\left[4\sin(\omega\,[t_{n-1}+\Delta t]) - 4\sin(\omega\, t_{n-1}) - \omega\,\Delta t\cos(\omega\,[t_{n-1}+\Delta t]) - 3\,\omega\,\Delta t\cos(\omega\, t_{n-1})\right]\boldsymbol{H}_{sin} \\ \frac{1}{\omega^2 \Delta t^2}\left[8\sin(\omega\, t_{n-1}) - 8\sin(\omega\,[t_{n-1}+\Delta t]) + 4\,\omega\,\Delta t\cos(\omega\,[t_{n-1}+\Delta t]) + 4\,\omega\,\Delta t\cos(\omega\, t_{n-1})\right]\boldsymbol{H}_{sin} \\ \boldsymbol{H}_{const} + \frac{1}{\omega^2 \Delta t^2}\left[4\sin(\omega\,[t_{n-1}+\Delta t]) - 4\sin(\omega\, t_{n-1}) - 3\,\omega\,\Delta t\cos(\omega\,[t_{n-1}+\Delta t]) - \omega\,\Delta t\cos(\omega\, t_{n-1})\right]\boldsymbol{H}_{sin} \end{array}\right] \quad (2.70)$$

where $\boldsymbol{H}_{const}$ is the mean traction load and $\boldsymbol{H}_{sin}$ is the sinusoidal traction amplitude.

2.4 Extended Space-Time Finite Element Method for Mechanical Analysis

2.4.1 Introduction

Given the cyclic character of fatigue loading, it is difficult to accurately capture the structural response of a component using the standard polynomial shape functions without using small time steps, somewhere around 100 or more per period. This would nullify much of the advantage over the finite difference methods mentioned earlier. To be able to use longer time steps, one method employed has been to enrich the temporal shape functions with harmonic functions such as sine or cosine in order to match the nature of the loading conditions (see Figures 2.2 and 2.3). By doing so, one can make the time step size equal to (or multiples of) the harmonic loading's time period, a vast improvement over the finite difference methods and the standard TFEM. This translates to much smaller computation times. The process of including functions matching the problem physics is called *enrichment*, and the resulting formulation is commonly known as an *extended* formulation. In the case of TFEM enrichment, the result is the extended space-time finite element method (XTFEM).

The idea of enriching TFEM is an outgrowth of the extended finite element method (XFEM) (Moës et al. (1999)) which was itself based on the generalized finite element method (GFEM) and partition of unity method (PUM) (Belytschko et al. (2014), p.646; Melenk and Babuska (1996)). The first researchers to use a space-time enrichment were Chessa and

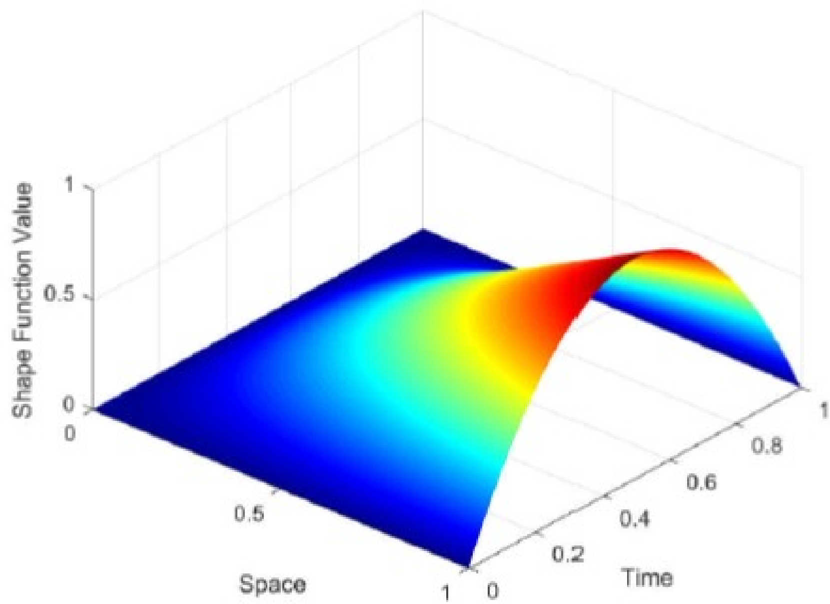

Figure 2.2. Standard Polynomial (Quadratic) Space-Time Shape Function

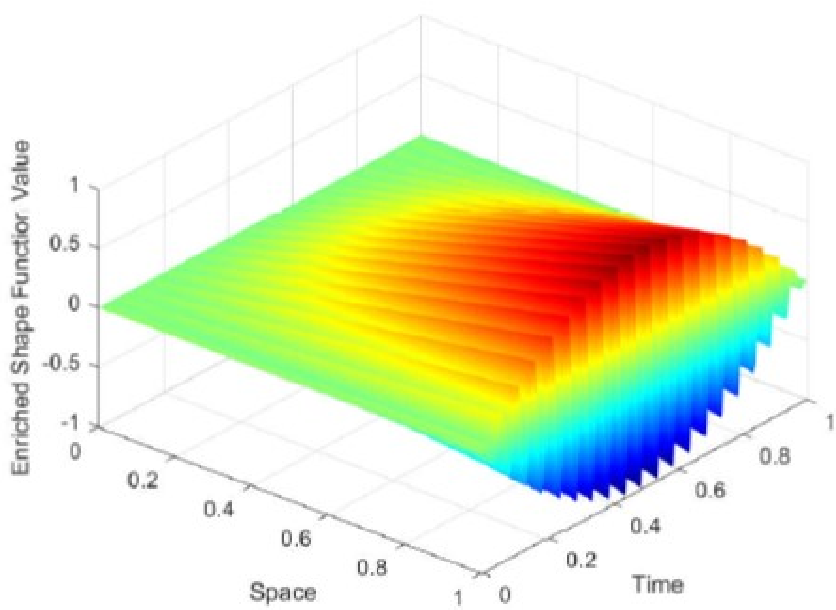

Figure 2.3. Enriched (Harmonic) Space-Time Shape Function

Belytschko (Chessa and Belytschko (2004)) who used it to solve linear, first-order wave and non-linear Burgers' equations. Qian and Chirputkar (Qian and Chirputkar (2014)) used an enriched space-time formulation to capture high frequency waves in a coupled molecular-FEM simulation of dynamic fracture. Using "examples involving wave propagations and dynamic fracture in harmonic lattice," Yang et al. (Yang et al. (2012)) proved the robustness of XTFEM and its superior convergence properties over the traditional TFEM for problems involving multiple time scales. The first researchers to use XTFEM to predict component life for high cycle fatigue problems were Bhamare et al. (Bhamare et al. (2014)) who performed a direct numerical simulation of a single-edge notched plate specimen for over 1 million fatigue cycles with a high degree of accuracy. Zhang et al. (Zhang et al. (2016)) demonstrated XTFEM's potential for parallel computing resulting in a drastic reduction in computation time. This section details the formulation and implementation of XTFEM for mechanical analysis.

2.4.2 Formulation

The extended finite element method (XFEM) is based on the partition of unity framework (Belytschko et al. (2014), p.647, Melenk and Babuska (1996)):

$$\tilde{N}_J\left(X,t\right) = N_J\left(X,t\right)\ \Phi_J\left(X,t\right) \tag{2.71}$$

where $\tilde{N}_J$ is the enriched nodal shape function, J is the node number, and Φ_J is the shifted enrichment function given by:

$$\Phi_J\left(X,t\right) = \Phi\left(X,t\right) - \Phi\left(X_J,t_J\right) \tag{2.72}$$

Thus, the total displacement approximation is:

$$u^h\left(X,t\right) = \sum_{I=1}^{n_s} N_I\left(X,t\right)\ d_I\ +\ \sum_{J=1}^{n_e} \tilde{N}_J\left(X,t\right)\ a_J$$

where n_s is the total number of regular nodes, n_e is the total number of enriched nodes, and a_J is the additional degrees of freedom due to enrichment. For fatigue applications, we introduce the enrichment function:

$$\Phi_J(t) = \Phi(t) - \Phi(t_J) = \sin(\omega\, t) - \sin(\omega\, t_J) \tag{2.73}$$

which is only a function of time and so will only be applied to the time shape functions. Assuming quadratic time shape functions (see section 2.2.3), the enrichment approximation is:

$$\begin{aligned} u^h = \tilde{\boldsymbol{N}}\, \boldsymbol{d}^X &= \left[\tilde{\boldsymbol{N}}_t \otimes \boldsymbol{N}_x\right] \begin{bmatrix} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{bmatrix} \\ &= \left[N_{t1}\boldsymbol{N}_x \;\; N_{t2}\boldsymbol{N}_x \;\; N_{t3}\boldsymbol{N}_x \;\; \tilde{N}_{t1}\boldsymbol{N}_x \;\; \tilde{N}_{t2}\boldsymbol{N}_x \;\; \tilde{N}_{t3}\boldsymbol{N}_x\right] \begin{bmatrix} d_{1_x 1_t} \\ \vdots \\ d_{n_s 3_t} \\ a_{1_x 1_t} \\ \vdots \\ a_{n_e 3_t} \end{bmatrix} \end{aligned} \tag{2.74}$$

where $\tilde{\boldsymbol{N}} = \left[\tilde{\boldsymbol{N}}_t \otimes \boldsymbol{N}_x\right]$, $\boldsymbol{d}^X = \begin{bmatrix} \boldsymbol{d}_n & \boldsymbol{a}_n \end{bmatrix}^T$, and:

$$\begin{aligned} \tilde{N}_{t1} &= \frac{\left(t_{n-1/2} - t\right)\left(t_n - t\right)}{\left(t_{n-1/2} - t_{n-1}\right)\left(t_n - t_{n-1}\right)} \left(\sin(\omega\, t) - \sin(\omega\, t_{n-1})\right) \\ \tilde{N}_{t2} &= \frac{\left(t_{n-1} - t\right)\left(t_n - t\right)}{\left(t_{n-1} - t_{n-1/2}\right)\left(t_n - t_{n-1/2}\right)} \left(\sin(\omega\, t) - \sin\left(\omega\, t_{n-1/2}\right)\right) \\ \tilde{N}_{t3} &= \frac{\left(t_{n-1} - t\right)\left(t_{n-1/2} - t\right)}{\left(t_{n-1} - t_n\right)\left(t_{n-1/2} - t_n\right)} \left(\sin(\omega\, t) - \sin(\omega\, t_n)\right) \end{aligned} \tag{2.75}$$

Moving back to equation 2.56, we now substitute $\begin{bmatrix} \boldsymbol{N} & \tilde{\boldsymbol{N}} \end{bmatrix}$ in place of $\boldsymbol{N}$, $\begin{bmatrix} \dot{\boldsymbol{N}} & \dot{\tilde{\boldsymbol{N}}} \end{bmatrix}$ in place of $\dot{\boldsymbol{N}}$, $\begin{bmatrix} \ddot{\boldsymbol{N}} & \ddot{\tilde{\boldsymbol{N}}} \end{bmatrix}$ in place of $\ddot{\boldsymbol{N}}$, $\begin{bmatrix} \boldsymbol{B} & \tilde{\boldsymbol{B}} \end{bmatrix}$ in place of $\boldsymbol{B}$, $\begin{bmatrix} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{bmatrix}$ in place of $\boldsymbol{d}_n$, etc.:

$$
\begin{aligned}
0 = & \int_{(Q_n)^\Sigma} \left[\begin{array}{cc} \dot{\boldsymbol{N}} & \dot{\tilde{\boldsymbol{N}}} \end{array} \right]^T \rho \left[\begin{array}{cc} \ddot{\boldsymbol{N}} & \ddot{\tilde{\boldsymbol{N}}} \end{array} \right] dQ \left[\begin{array}{c} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{array} \right] \\
& + \int_{Q_n} \left[\begin{array}{cc} \dot{\boldsymbol{B}} & \dot{\tilde{\boldsymbol{B}}} \end{array} \right]^T \boldsymbol{D} \left[\begin{array}{cc} \boldsymbol{B} & \tilde{\boldsymbol{B}} \end{array} \right] dQ \left[\begin{array}{c} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{array} \right] \\
& - \int_{Q_n} \left[\begin{array}{cc} \dot{\boldsymbol{B}} & \dot{\tilde{\boldsymbol{B}}} \end{array} \right]^T \boldsymbol{D}\, \alpha\, \Delta T\, \boldsymbol{I}\, dQ \\
& + \int_{\Omega} \left[\begin{array}{cc} \dot{\boldsymbol{N}}\left(t_{n-1}^+\right) & \dot{\tilde{\boldsymbol{N}}}\left(t_{n-1}^+\right) \end{array} \right]^T \rho \left[\begin{array}{cc} \dot{\boldsymbol{N}}\left(t_{n-1}^+\right) & \dot{\tilde{\boldsymbol{N}}}\left(t_{n-1}^+\right) \end{array} \right] d\Omega \left[\begin{array}{c} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{array} \right] \\
& + \int_{\Omega} \left[\begin{array}{cc} \boldsymbol{B}\left(t_{n-1}^+\right) & \tilde{\boldsymbol{B}}\left(t_{n-1}^+\right) \end{array} \right]^T \boldsymbol{D} \left[\begin{array}{cc} \boldsymbol{B}\left(t_{n-1}^+\right) & \tilde{\boldsymbol{B}}\left(t_{n-1}^+\right) \end{array} \right] d\Omega \left[\begin{array}{c} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{array} \right] \\
& - \int_{\Omega} \left[\begin{array}{cc} \boldsymbol{B}\left(t_{n-1}^+\right) & \tilde{\boldsymbol{B}}\left(t_{n-1}^+\right) \end{array} \right]^T \boldsymbol{D}\, \alpha\, \left[T\left(t_{n-1}^+\right) - T_{ref} \right] \boldsymbol{I}\, d\Omega \\
& - \int_{(\Upsilon_h)_n} \left[\begin{array}{cc} \dot{\boldsymbol{N}} & \dot{\tilde{\boldsymbol{N}}} \end{array} \right]^T \boldsymbol{h}\, d\Upsilon \\
& - \int_{Q_n} \left[\begin{array}{cc} \dot{\boldsymbol{N}} & \dot{\tilde{\boldsymbol{N}}} \end{array} \right]^T \rho\, \boldsymbol{b}\, dQ \\
& - \int_{\Omega} \left[\begin{array}{cc} \dot{\boldsymbol{N}}\left(t_{n-1}^+\right) & \dot{\tilde{\boldsymbol{N}}}\left(t_{n-1}^+\right) \end{array} \right]^T \rho \left[\begin{array}{cc} \dot{\boldsymbol{N}}\left(t_{n-1}^-\right) & \dot{\tilde{\boldsymbol{N}}}\left(t_{n-1}^-\right) \end{array} \right] d\Omega \left[\begin{array}{c} \boldsymbol{d}_{n-1} \\ \boldsymbol{a}_{n-1} \end{array} \right] \\
& - \int_{\Omega} \left[\begin{array}{cc} \boldsymbol{B}\left(t_{n-1}^+\right) & \tilde{\boldsymbol{B}}\left(t_{n-1}^+\right) \end{array} \right]^T \boldsymbol{D} \left[\begin{array}{cc} \boldsymbol{B}\left(t_{n-1}^-\right) & \tilde{\boldsymbol{B}}\left(t_{n-1}^-\right) \end{array} \right] d\Omega \left[\begin{array}{c} \boldsymbol{d}_{n-1} \\ \boldsymbol{a}_{n-1} \end{array} \right] \\
& + \int_{\Omega} \left[\begin{array}{cc} \boldsymbol{B}\left(t_{n-1}^+\right) & \tilde{\boldsymbol{B}}\left(t_{n-1}^+\right) \end{array} \right]^T \boldsymbol{D}\, \alpha \left[T\left(t_{n-1}^-\right) - T_{ref} \right] \boldsymbol{I}\, d\Omega
\end{aligned}
\tag{2.76}
$$

Likewise for n=1, the last three terms are:

$$
\begin{aligned}
& - \int_{\Omega} \left[\begin{array}{cc} \dot{\boldsymbol{N}}\left(0^+\right) & \dot{\tilde{\boldsymbol{N}}}\left(0^+\right) \end{array} \right]^T \rho\, \boldsymbol{N}_x\, d\Omega \left[\begin{array}{c} \boldsymbol{v}_0 \\ 0 \end{array} \right] \\
& - \int_{\Omega} \left[\begin{array}{cc} \boldsymbol{B}\left(0^+\right) & \tilde{\boldsymbol{B}}\left(0^+\right) \end{array} \right]^T \boldsymbol{D}\, \boldsymbol{B}_x\, d\Omega \left[\begin{array}{c} \boldsymbol{u}_0 \\ 0 \end{array} \right] \\
& + \int_{\Omega} \left[\begin{array}{cc} \boldsymbol{B}\left(0^+\right) & \tilde{\boldsymbol{B}}\left(0^+\right) \end{array} \right]^T \boldsymbol{D}\, \alpha \left(T_0 - T_{ref} \right) \boldsymbol{I}\, d\Omega
\end{aligned}
\tag{2.77}
$$

where $\boldsymbol{u}_0$, $\boldsymbol{v}_0$, and T_0 are the initial, spatial (not space-time) nodal displacements, nodal velocities, and nodal temperatures, respectively. Separating the spatial and temporal integrals:

$$
\begin{aligned}
0 = & \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T \begin{bmatrix} \ddot{\boldsymbol{N}}_t & \ddot{\tilde{\boldsymbol{N}}}_t \end{bmatrix} dt \otimes \int_\Omega \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \begin{bmatrix} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{bmatrix} \\
& + \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T \begin{bmatrix} \boldsymbol{N}_t & \tilde{\boldsymbol{N}}_t \end{bmatrix} dt \otimes \int_\Omega \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \begin{bmatrix} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{bmatrix} \\
& + \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^+\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^+\right) \end{bmatrix}^T \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^+\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^+\right) \end{bmatrix} \otimes \int_\Omega \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \begin{bmatrix} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{bmatrix} \\
& + \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^+\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^+\right) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^+\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^+\right) \end{bmatrix} \otimes \int_\Omega \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \begin{bmatrix} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{bmatrix} \\
& - \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T h\left(t\right) \, dt \otimes \int_{\Gamma_N} \boldsymbol{N}_x^T \, \boldsymbol{h}\left(x\right) \, d\Gamma \\
& - \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T b\left(t\right) \, dt \otimes \int_\Omega \boldsymbol{N}_x^T \, \rho \, \boldsymbol{b}\left(x\right) \, d\Omega \\
& - \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T \Delta T\left(t\right) \, dt \otimes \int_\Omega \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \Delta T\left(x\right) \, \boldsymbol{I} \, d\Omega \\
& - \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^+\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^+\right) \end{bmatrix}^T \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^-\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^-\right) \end{bmatrix} \otimes \int_\Omega \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \begin{bmatrix} \boldsymbol{d}_{n-1} \\ \boldsymbol{a}_{n-1} \end{bmatrix} \\
& - \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^+\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^+\right) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^-\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^-\right) \end{bmatrix} \otimes \int_\Omega \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \begin{bmatrix} \boldsymbol{d}_{n-1} \\ \boldsymbol{a}_{n-1} \end{bmatrix} \\
& - \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^+\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^+\right) \end{bmatrix}^T \otimes \int_\Omega \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \left[T\left(t_{n-1}^+\right) - T\left(t_{n-1}^-\right)\right] \boldsymbol{I} \, d\Omega
\end{aligned}
\tag{2.78}
$$

For the case when n=1, the last three terms become:

$$
\begin{aligned}
& - \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(0^+\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(0^+\right) \end{bmatrix}^T \otimes \int_\Omega \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega \begin{bmatrix} \boldsymbol{v}_0 \\ 0 \end{bmatrix} \\
& - \begin{bmatrix} \boldsymbol{N}_t\left(0^+\right) & \tilde{\boldsymbol{N}}_t\left(0^+\right) \end{bmatrix}^T \otimes \int_\Omega \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega \begin{bmatrix} \boldsymbol{u}_0 \\ 0 \end{bmatrix} \\
& - \begin{bmatrix} \boldsymbol{N}_t\left(0^+\right) & \tilde{\boldsymbol{N}}_t\left(0^+\right) \end{bmatrix}^T \otimes \int_\Omega \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \left[T\left(0^+\right) - T_0\right] \boldsymbol{I} \, d\Omega
\end{aligned}
\tag{2.79}
$$

We define the following:

$$\boldsymbol{M} = \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{N}_x \, d\Omega$$

$$\boldsymbol{K} = \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \boldsymbol{B}_x \, d\Omega$$

$$\boldsymbol{H} = \int_{\Gamma_n} \boldsymbol{N}_x^T \, \boldsymbol{h}\left(x\right) \, d\Gamma$$

$$\boldsymbol{B}_f = \int_{\Omega} \boldsymbol{N}_x^T \, \rho \, \boldsymbol{b}\left(x\right) \, d\Omega$$

$$\boldsymbol{\Theta} = \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \Delta T\left(x\right) \, \boldsymbol{I} \, d\Omega$$

$$\boldsymbol{\Theta J} = \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \, \left[T\left(t_{n-1}^{+}\right) - T\left(t_{n-1}^{-}\right)\right] \boldsymbol{I} \, d\Omega$$

$$\boldsymbol{\Theta J_0} = \int_{\Omega} \boldsymbol{B}_x^T \, \boldsymbol{D} \, \alpha \left[T\left(0^{+}\right) - T_0\right] \boldsymbol{I} \, d\Omega$$

$$\boldsymbol{EINN3} = \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T \begin{bmatrix} \ddot{\boldsymbol{N}}_t & \ddot{\tilde{\boldsymbol{N}}}_t \end{bmatrix} dt$$

$$\boldsymbol{EINN1} = \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T \begin{bmatrix} \boldsymbol{N}_t & \tilde{\boldsymbol{N}}_t \end{bmatrix} dt$$

$$\boldsymbol{EIN1H} = \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T h\left(t\right) \, dt$$

$$\boldsymbol{EIN1B} = \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T b\left(t\right) \, dt$$

$$\boldsymbol{EIN1\Theta} = \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{bmatrix}^T \Delta T\left(t\right) \, dt$$

$$\boldsymbol{ENN2P} = \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^{+}\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^{+}\right) \end{bmatrix}^T \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^{+}\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^{+}\right) \end{bmatrix}$$

$$\boldsymbol{ENN2M} = \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^{+}\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^{+}\right) \end{bmatrix}^T \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(t_{n-1}^{-}\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(t_{n-1}^{-}\right) \end{bmatrix}$$

$$\boldsymbol{ENNP} = \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^{+}\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^{+}\right) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^{+}\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^{+}\right) \end{bmatrix}$$

$$\boldsymbol{ENNM} = \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^{+}\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^{+}\right) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^{-}\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^{-}\right) \end{bmatrix}$$

$$\boldsymbol{ENP} = \begin{bmatrix} \boldsymbol{N}_t\left(t_{n-1}^{+}\right) & \tilde{\boldsymbol{N}}_t\left(t_{n-1}^{+}\right) \end{bmatrix}^T$$

$$\boldsymbol{EN1P_0} = \begin{bmatrix} \dot{\boldsymbol{N}}_t\left(0^{+}\right) & \dot{\tilde{\boldsymbol{N}}}_t\left(0^{+}\right) \end{bmatrix}^T$$

$$\boldsymbol{ENP_0} = \begin{bmatrix} \boldsymbol{N}_t\left(0^{+}\right) & \tilde{\boldsymbol{N}}_t\left(0^{+}\right) \end{bmatrix}^T$$

$$\boldsymbol{d}_n^X = \begin{bmatrix} \boldsymbol{d}_n \\ \boldsymbol{a}_n \end{bmatrix}$$

$$\boldsymbol{d}_{n-1}^X = \begin{bmatrix} \boldsymbol{d}_{n-1} \\ \boldsymbol{a}_{n-1} \end{bmatrix}$$

$$\boldsymbol{u}_0^X = \begin{bmatrix} \boldsymbol{u}_0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{v}_0^X = \begin{bmatrix} \boldsymbol{v}_0 \\ 0 \end{bmatrix}$$

Using the above definitions and moving the negative terms to the other side, equation 2.78 becomes:

$$\begin{aligned} &[(\boldsymbol{EINN3} + \boldsymbol{ENN2P}) \otimes \boldsymbol{M} + \\ &(\boldsymbol{EINN1} + \boldsymbol{ENNP}) \otimes \boldsymbol{K}]\ \boldsymbol{d}_n^X = \boldsymbol{EIN1H} \otimes \boldsymbol{H} + \boldsymbol{EIN1B} \otimes \boldsymbol{B}_f \\ &\qquad + \boldsymbol{EIN1\Theta} \otimes \boldsymbol{\Theta} + \boldsymbol{ENP} \otimes \boldsymbol{\Theta J} \\ &\qquad + [\boldsymbol{ENN2M} \otimes \boldsymbol{M} + \boldsymbol{ENNM} \otimes \boldsymbol{K}]\ \boldsymbol{d}_{n-1}^X \end{aligned} \tag{2.80}$$

For n=1, the RHS of equation 2.80 becomes:

$$\begin{aligned} &+ \boldsymbol{EIN1H} \otimes \boldsymbol{H} + \boldsymbol{EIN1B} \otimes \boldsymbol{B}_f \\ &+ \boldsymbol{EIN1\Theta} \otimes \boldsymbol{\Theta} + \boldsymbol{ENP_0} \otimes \boldsymbol{\Theta J_0} \\ &+ [\boldsymbol{EN1P} \otimes \boldsymbol{M}]\ \boldsymbol{v}_0^X + [\boldsymbol{ENP} \otimes \boldsymbol{K}]\ \boldsymbol{u}_0^X \end{aligned} \tag{2.81}$$

Equation 2.60 can be simplified into a space-time linear system:

$$\boldsymbol{K}_{Xst_n}^{mech}\,\boldsymbol{d}_n^X = \boldsymbol{F}_{Xst_n}^{mech} \tag{2.82}$$

where:

$$\boldsymbol{K}_{Xst_n}^{mech} = \left[(\boldsymbol{EINN3} + \boldsymbol{ENN2P}) \otimes \boldsymbol{M} + (\boldsymbol{EINN1} + \boldsymbol{ENNP}) \otimes \boldsymbol{K}\right] \tag{2.83}$$

$$\begin{aligned} \boldsymbol{F}_{Xst_n}^{mech} &= \boldsymbol{EIN1H} \otimes \boldsymbol{H} + \boldsymbol{EIN1B} \otimes \boldsymbol{B}_f \\ &\quad + \boldsymbol{EIN1\Theta} \otimes \boldsymbol{\Theta} + \boldsymbol{ENP} \otimes \boldsymbol{\Theta J} \\ &\quad + \left[\boldsymbol{ENN2M} \otimes \boldsymbol{M} + \boldsymbol{ENNM} \otimes \boldsymbol{K}\right] \boldsymbol{d}_{n-1}^X \end{aligned} \tag{2.84}$$

$$\begin{aligned} \boldsymbol{F}_{Xst_1}^{mech} &= \boldsymbol{EIN1H} \otimes \boldsymbol{H} + \boldsymbol{EIN1B} \otimes \boldsymbol{B}_f \\ &\quad + \boldsymbol{EIN1\Theta} \otimes \boldsymbol{\Theta} + \boldsymbol{ENP_0} \otimes \boldsymbol{\Theta J_0} \\ &\quad + \left[\boldsymbol{EN1P} \otimes \boldsymbol{M}\right] \boldsymbol{v}_0^X + \left[\boldsymbol{ENP} \otimes \boldsymbol{K}\right] \boldsymbol{u}_0^X \end{aligned} \tag{2.85}$$

$\boldsymbol{K}_{Xst_n}^{mech}$ can be broken down into its normal, enriched, and coupled parts:

$$\boldsymbol{K}_{Xst_n}^{mech} = \begin{bmatrix} \boldsymbol{K}_{st_n}^{mech} & \boldsymbol{K}_{ea_n}^{mech} \\ \boldsymbol{K}_{eb_n}^{mech} & \boldsymbol{K}_{ee_n}^{mech} \end{bmatrix} \tag{2.86}$$

where $\boldsymbol{K}_{st_n}^{mech}$ is the normal space-time stiffness matrix, $\boldsymbol{K}_{ea_n}^{mech}$ and $\boldsymbol{K}_{eb_n}^{mech}$ represent the coupling between regular and enriched degrees of freedom, and $\boldsymbol{K}_{ee_n}^{mech}$ represents the coupling between enriched degrees of freedom (Bhamare (2012), p.93). Note that $\boldsymbol{K}_{ea_n}^{mech} \neq \left(\boldsymbol{K}_{eb_n}^{mech}\right)^T$.

2.4.3 Mechanical XTFEM Implementation

The implementation of the mechanical XTFEM formulation is as follows:

1. Generate a mesh of the spatial domain.

2. Assemble the global, spatial mass and stiffness matrices, $\boldsymbol{M}$ and $\boldsymbol{K}$, respectively.

3. Assemble the global, spatial external traction and body force vectors, $\boldsymbol{H}$ and $\boldsymbol{B}_f$, respectively.

4. Initiate the time loop:

   (a) Solve the thermal space-time equations for the temperatures (see section 2.2.4).

   (b) Calculate the various time matrices (e.g., $\boldsymbol{EINN3}$, $\boldsymbol{EINN1}$, $\boldsymbol{ENN2P}$, etc.) for the given time step's Δt.

   (c) Assemble the space-time stiffness matrix, $\boldsymbol{K}^{mech}_{Xst_n}$, using equation 2.83.

   (d) Calculate the space-time thermal "force" and thermal jump terms, $\boldsymbol{EIN1\Theta} \otimes \boldsymbol{\Theta}$ and $\boldsymbol{ENP} \otimes \boldsymbol{\Theta J}$, respectively.

   (e) Calculate the space-time force vector, $\boldsymbol{F}^{mech}_{Xst_1}$ or $\boldsymbol{F}^{mech}_{Xst_n}$, using equations 2.85 or 2.84 for the first time step or subsequent time steps, respectively.

   (f) Apply boundary conditions.

   (g) Solve the linear system, equation 2.82, for the extended space-time displacement vector, $\boldsymbol{d}^X_n$.

   (h) Store the extended space-time displacement vector for the next time-step, $\boldsymbol{d}^X_{n-1}$.

   (i) Repeat (a) through (f) for all time intervals.

5. End time loop.

CHAPTER 3

TRADITIONAL APPROACHES TO CALCULATING MULTIAXIAL HCF DAMAGE

There are several traditional models employed by modern virtual fatigue analysis software packages for evaluating HCF for multiaxial problems. This chapter will give a brief overview of some of these models. Since the goal of this study is HCF life estimation, we will neither discuss the various low-cycle fatigue (LCF) methods nor the damage-tolerant approach of fracture mechanics.

3.1 Multiaxial Methods for Complex Loading

3.1.1 Equivalent Stress Methods

Equivalent stress methods calculate an equivalent stress from the stress tensor values. These methods are easy to implement and computationally very fast.

3.1.1.1 Principal Stress Criterion

Fatigue initiation will occur if:

$$\sigma_{PS,a} = \sigma_{1,a} \geq \sigma_{E,R=-1} \tag{3.1}$$

where $\sigma_{PS,a}$ is the principal stress amplitude, $\sigma_{1,a}$ is the maximum principal stress amplitude, and $\sigma_{E,R=-1}$ is the fully reversed fatigue limit for normal stress.

One problem with the principal stress criterion is that it assumes that fatigue life is dependent only on the largest principal stress, σ_1, and that the other principal stresses play no role (Safe Technology Ltd. (2002), p.7-15). To use a circular shaft in pure torsion as a counter-example, the principal stress criterion would predict that "the fatigue strength in torsion is the same as the fatigue strength under axial loading," an assumption that is not supported by experimental results (Safe Technology Ltd. (2002), p.7-15). Furthermore, the principal stress criterion gives very non-conservative fatigue lives for ductile metals and should only be used (if at all) for brittle metals such as cast irons and very high strength steels (Safe Technology Ltd. (2002), p.7-16). Lastly, this criterion shows poor correlation for non-proportional loading (Papuga et al. (2012), p.99).

3.1.1.2 Maximum Shear Stress Criterion

Fatigue initiation will occur if:

$$\tau_{MS,a} = \sigma_{1,a} - \sigma_{3,a} \geq \tau_{E,R=-1} \tag{3.2}$$

where $\tau_{MS,a}$ is the maximum shear stress amplitude, $\sigma_{1,a}$ is the maximum principal stress amplitude, $\sigma_{3,a}$ is the minimum principal stress amplitude, and $\tau_{E,R=-1}$ is the fully reversed fatigue limit for shear stress. In general, the maximum shear stress criterion gives conservative fatigue lives for ductile metals but non-conservative (i.e., unsafe) fatigue lives for brittle metals (Safe Technology Ltd. (2002), p.7-18). Furthermore, it shows poor correlation for non-proportional loading (Papuga et al. (2012), p.99).

3.1.1.3 von Mises Stress Criterion

Fatigue initiation will occur if:

$$\sigma_{vM,a} + \alpha_{vM}\, \sigma_{vM,m} \geq \sigma_{E,R=-1} \tag{3.3}$$

where $\sigma_{vM,a}$ is the von Mises stress amplitude, $\sigma_{vM,m}$ is the mean stress, and α_{vM} is the mean stress sensitivity factor. $\sigma_{vM,a}$ and $\sigma_{vM,m}$ are defined as follows:

$$\sigma_{vM,a} = \sqrt{\frac{3}{2}\, \sigma^D_{ij} : \sigma^D_{ij}} \tag{3.4}$$

$$\sigma_{vM,m} = \sigma_{1,m} + \sigma_{2,m} + \sigma_{3,m} \tag{3.5}$$

where $\sigma_{1,m}$, $\sigma_{2,m}$, and $\sigma_{3,m}$ are the three principal stresses, and σ_{ij}^D is the deviatoric stress tensor.

One problem with the von Mises stress criterion is that the von Mises stress is always positive, and thus, one cannot use cycle counting methods directly (see section 3.3). Though there are methods for assigning a sign to the von Mises stress such as using the sign of the largest stress tensor component or using the sign of the hydrostatic stress, these different methods give vastly disparate life estimates (Safe Technology Ltd. (2002), p.7-23). A second, more important problem is that the von Mises criterion does not correlate well with experimental results, especially for non-proportional loading (Safe Technology Ltd. (2002), p.7-23; Papuga et al. (2012), p.99).

3.1.1.4 Sines Criterion

The Sines criterion calculates fatigue damage on the basis of the octahedral shear stress amplitude and the mean hydrostatic stress (Sines (1959)). Fatigue initiation will occur if:

$$\tau_{Sines,a} = \tau_{oct,a} + \alpha_{oct}\left(3\,\sigma_m^H\right) \geq \tau_E \tag{3.6}$$

where α_{oct} is the hydrostatic stress sensitivity factor, σ_m^H is the hydrostatic part of the mean stress, $\tau_{oct,a}$ is the octahedral stress amplitude, and τ_E is the fatigue limit for shear stress. σ_m^H and $\tau_{oct,a}$ are defined as:

$$\sigma_m^H = \frac{1}{3}\left(\sigma_{1,m} + \sigma_{2,m} + \sigma_{3,m}\right) \tag{3.7}$$

$$\tau_{oct,a} = \sqrt{\frac{1}{3}\,\sigma_{ij}^D : \sigma_{ij}^D} \tag{3.8}$$

where $\sigma_{1,m}$, $\sigma_{2,m}$, and $\sigma_{3,m}$ are the mean principal stresses. According to Lee and Barkey (Lee and Barkey (2011a), p.172), the Sines criterion “shows satisfactory correlation with experimental investigations.” It also correctly reproduces the independence of the fatigue life in torsion from a superimposed mean shear stress in the very high cycle fatigue range

($> 10^7$cycles) as well as a dependence of fatigue life in bending upon a superimposed normal mean stress (Papadopoulos et al. (1997), p.227).

However, the Sines criterion requires a torsion test and a repeated bending test to calibrate parameters (Papadopoulos et al. (1997), p.227). Furthermore, the criterion predicts that "the fatigue limits in torsion and fully reversed bending are in a constant ratio for all metals," which does not line up with experimental results (Papadopoulos et al. (1997), p.227). Lastly, the Sines criterion can only be applied to components under proportional loading conditions since it has poor predictive power for components under non-proportional loading (Santos et al. (2003), p.470; Tchoupou and Fotsing (2015), p.2507).

3.1.2 Critical Plane Methods

Critical plane methods are typically used for fatigue analyses involving variable amplitude, nonproportional, multiaxial loading (Lee and Barkey (2011a), p.186). These approaches are premised on the fact that, for most ductile materials, fatigue cracks tend to "initiate on the plane with the maximum shear stress amplitude" (Karpanan (2016), p.2). Critical plane methods involve searching for a particular plane in which a fatigue damage parameter for the entire loading history is maximized. All possible planes must be evaluated at each material point of interest (Karpanan (2016), p.2).

3.1.2.1 Critical Plane Search Algorithm

The search is done by transforming the stress tensor through coordinate system rotations:

$$\boldsymbol{\sigma}^* = \boldsymbol{Q}^T \, \boldsymbol{\sigma} \, \boldsymbol{Q} \tag{3.9}$$

or in indicial notation:

$$\sigma^*_{ij} = Q_{mi} \, \sigma_{mn} \, Q_{nj} \tag{3.10}$$

where $Q_{ij} = \cos\left(e_i, e_j^*\right)$ is the transformation matrix (Lai et al. (2010), pp.24-25), e_i are the original coordinate system base vectors, and e_j^* are the new coordinate system base vectors. The steps are:

1. First, the original coordinate system is rotated about a single axis by a certain angle, θ.
2. Then, the new coordinate system is rotated again but about a different axis by a certain angle, ϕ.
3. The transformed stress tensor's history is evaluated at this point and the number of cycles to failure is calculated. The popular damage parameter criteria are discussed in the subsections below.
4. Repeat steps 2 and 3 for all angles $0 \leq \phi \leq 180°$.
5. Repeat steps 1 through 4 for all angles $0 \leq \theta \leq 180°$.
6. The plane that has the fewest cycles to failure determines the component's life.

A common practice is to vary θ and ϕ by 10° (Safe Technology Ltd. (2002), p.7-34; Karpanan (2016), p.4) or 5° intervals though even this small of an increment can still lead to discretization errors (Lee and Barkey (2011a), p.188).

3.1.2.2 Findley Damage Parameter

Findley's parameter is premised on the theory that friction between crack faces can provide resistance to shear forces and thus reduce the force seen at the crack tips which, in turn, reduces crack growth (see Figure 3.1). When the crack faces separate as in the case of a tensile load, the shear force is allowed to concentrate at the crack tips (see Figure 3.2). The criterion is thus a combination of the shear stress amplitude and the maximum normal stress on the plane being investigated. A crack will form on the plane being investigated if:

$$\left(\tau_{Findey,a}\right)_{max} = \tau_E \tag{3.11}$$

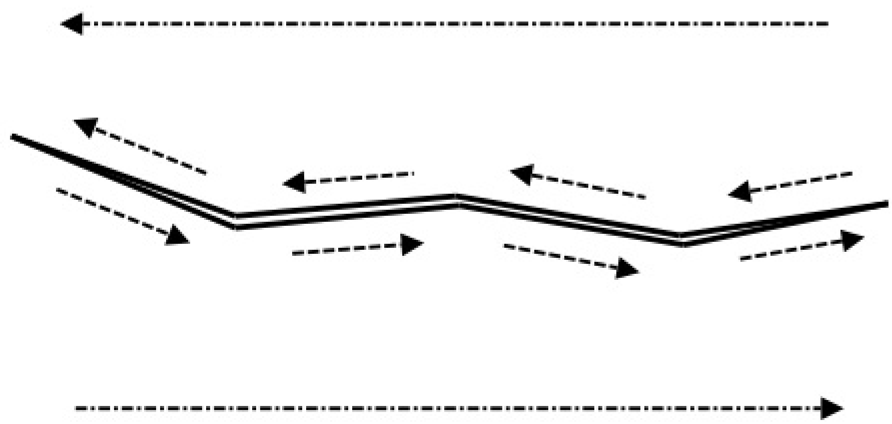

Figure 3.1. Effect of Crack Face Friction on Crack Propagation

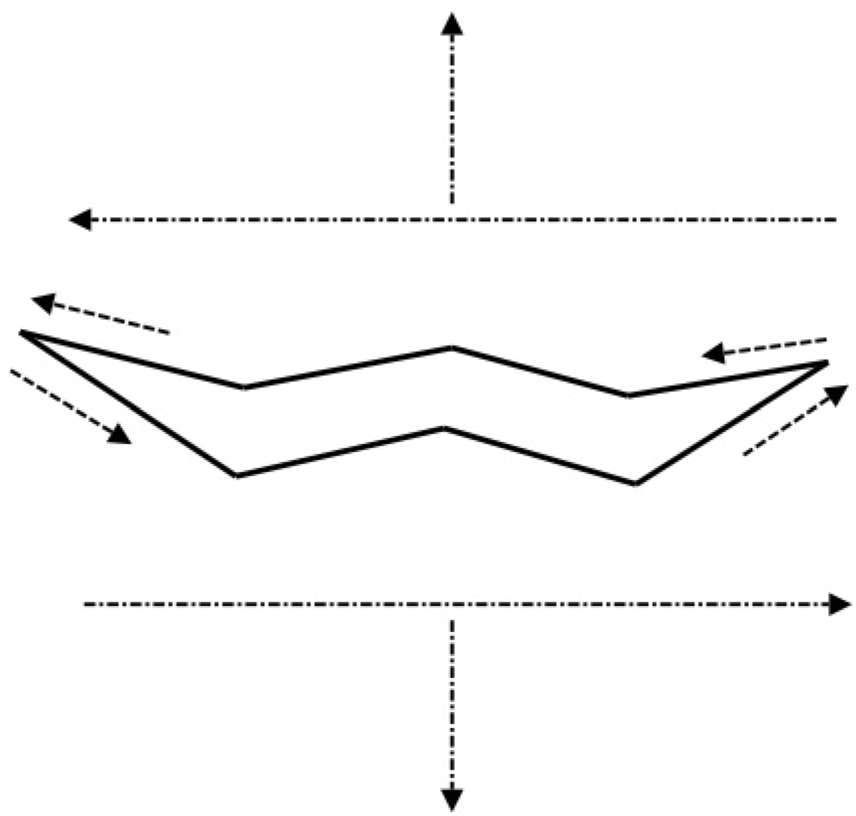

Figure 3.2. Effect of Crack Face Opening on Crack Propagation

$$\tau_{Findey,a} = \tau_a + k_F \, \sigma_{n,max} \tag{3.12}$$

where τ_E is the shear fatigue limit, τ_a is the shear stress amplitude, $\sigma_{n,max}$ is the maximum normal stress, and k_F is the normal stress sensitivity factor which represents "the influence of the maximum normal stress on the maximum shear stress amplitude" (Lee and Barkey (2011a), p.192). One criticism is that the Findley criterion fails to predict that, in the very high cycle fatigue range (i.e., $> 10^7$), fatigue life for cyclic torsion is independent of the applied mean shear stress (Papadopoulos et al. (1997), p.225).

3.1.2.3 McDiarmid Damage Parameter

McDiarmid's fatigue criterion is based on Findley's method, but his critical plane definition is based on the maximum shear stress amplitude only (Papadopoulos et al. (1997), p.226; Karpanan (2016), p.3):

$$\frac{\tau_a}{\tau_E} + \frac{\sigma_{n,max}}{2\,\sigma_{u,t}} = 1 \tag{3.13}$$

where τ_a is the shear stress amplitude, τ_E is the shear fatigue limit, $\sigma_{u,t}$ is the ultimate tensile strength, and $\sigma_{n,max}$ is the maximum normal stress on the critical plane with the maximum shear stress amplitude. McDiarmid's criterion correctly captures the fact that the fatigue limit in bending is highly dependent upon a superimposed mean normal stress and that, in the very high cycle fatigue range (i.e., $> 10^7$), fatigue life for cyclic torsion is independent of the applied mean shear stress (Papadopoulos et al. (1997), p.226). As a result, McDiarmid's criterion has shown good correlation in the very high cycle range (Safe Technology Ltd. (2002), p.7-19).

However, there are two main drawbacks to McDiarmid's parameter. In order to calibrate the shear fatigue limit, τ_E, one must have torsion test data (Safe Technology Ltd. (2002), p.7-19). Furthermore, it is not as accurate in fatigue life predictions below the very high cycle range (Safe Technology Ltd. (2002), p.7-50).

3.1.2.4 Evaluation of Critical Plane Methods

Critical plane methods can be very useful in the case of multiaxial, non-proportional loading. In addition to the criticisms of the specific criterions, however, they are computationally expensive especially for long and complex loading histories (Karpanan (2016), p.4).

3.1.3 Dang Van's Multiscale Method

Dang Van's method (Dang Van (1973); Dang Van et al. (1989); Dang Van (1993)) is a variation of the Crossland (Crossland (1956)) and Sines (Sines (1959)) criterions, but instead of calculating fatigue life, the Dang Van criterion asseses whether or not "a component will have infinite life" (Safe Technology Ltd. (2002), p.7-48). It is a multiscale approach that calculates mesoscopic stress values based on the macroscopic (i.e., finite element) stress tensor. Dang Van assumed that for an infinite component life, "crack nucleation in slip bands may occur in the most unfavorably oriented grains, which are subjected to plastic deformation even if the macroscopic stress is elastic" (Lee and Barkey (2011a), p.199). These plastic stresses result in residual stresses "due to the restraining effect of the adjacent grains" (Lee and Barkey (2011a), p.199). A second assumption is that an elastic shakedown occurs in the macroscale and that plastic strains at the mesoscale reach constant values after a number of iterations (Lee and Barkey (2011a), p.200). The last assumption in this criterion is that the mesoscopic shear stress, τ_{meso}, (which acts on the plane of maximum shear stress) "on a grain is responsible for crack nucleation in slip bands within a grain and the mesoscopic hydrostatic stress," σ^H_{meso}, "will influence the opening of these cracks" (Lee and Barkey (2011a), p.202).

Dang Van assumed that for an infinite component life, the macroscopic elastic strain tensor is equal to the sum of the mesoscopic elastic and plastic strain tensors:

$$\boldsymbol{\varepsilon}^e_{macro_n} = \boldsymbol{\varepsilon}^e_{meso_n} + \boldsymbol{\varepsilon}^p_{meso_n} \tag{3.14}$$

where the subscript, n, is the nth stress value from the loading history. Dang Van defines the following:

$$\boldsymbol{\varepsilon}^e_{meso_n} = \frac{1}{E_{meso}}\boldsymbol{\sigma}_{meso_n} \tag{3.15}$$

$$\boldsymbol{\varepsilon}^e_{macro_n} = \frac{1}{E_{macro}}\boldsymbol{\sigma}_{macro_n} \tag{3.16}$$

where E_{meso} and E_{macro} are the elastic moduli at mesoscale and macroscale, respectively. Substituting equations 3.15 and 3.16 into equation 3.14, we have:

$$\frac{1}{E_{macro}}\boldsymbol{\sigma}_{macro_n} = \frac{1}{E_{meso}}\boldsymbol{\sigma}_{meso_n} + \boldsymbol{\varepsilon}^p_{meso_n} \tag{3.17}$$

Rearranging equation 3.17:

$$\boldsymbol{\sigma}_{meso_n} = \frac{E_{meso}}{E_{macro}}\boldsymbol{\sigma}_{macro_n} - E_{meso}\boldsymbol{\varepsilon}^p_{meso_n} \tag{3.18}$$

Assuming that $\frac{E_{meso}}{E_{macro}} = 1$, we have:

$$\boldsymbol{\sigma}_{meso_n} = \boldsymbol{\sigma}_{macro_n} - E_{meso}\boldsymbol{\varepsilon}^p_{meso_n} \tag{3.19}$$

Defining the backstress tensor as $\boldsymbol{\alpha}^*_n = E_{meso}\boldsymbol{\varepsilon}^p_{meso_n}$ and substituting this definition into equation 3.19:

$$\boldsymbol{\sigma}_{meso_n} = \boldsymbol{\sigma}_{macro_n} - \boldsymbol{\alpha}^*_n \tag{3.20}$$

Because of the elastic shakedown assumption, $\boldsymbol{\alpha}^*_n$ and $\boldsymbol{\varepsilon}^p_{meso_n}$ become constant after a number of loading cycles:

$$\boldsymbol{\sigma}_{meso_n} = \boldsymbol{\sigma}_{macro_n} - \boldsymbol{\alpha}^* \tag{3.21}$$

Because $\boldsymbol{\alpha}^*$ is deviatoric, the hydrostatic mesoscopic and macroscopic stresses are equal:

$$\begin{gathered} \sigma^H_{meso_n} = \sigma^H_{macro_n} \\ \frac{1}{3}tr\left(\boldsymbol{\sigma}_{meso_n}\right) = \frac{1}{3}tr\left(\boldsymbol{\sigma}_{macro_n}\right) \end{gathered} \tag{3.22}$$

The deviatoric part of equation 3.21 is:

$$\boldsymbol{\sigma}^D_{meso_n} = \boldsymbol{\sigma}^D_{macro_n} - \boldsymbol{\alpha}^* \tag{3.23}$$

Under the assumption that $\boldsymbol{\alpha}^*$ is "the center of the smallest von Mises yield surface, σ_{vM}, that completely encloses the path described by the macroscopic deviatoric stress tensor" (Lee and Barkey (2011a), p.201), $\boldsymbol{\alpha}^*$ is found by minimizing "the maximum von Mises stresses calculated from the macro deviatoric stress tensor with respect to the updated yield surface center" (Lee and Barkey (2011a), p.201):

$$\boldsymbol{\alpha}^* = \min_{\boldsymbol{\alpha}} \left\{ \max_{n} \sigma_{vM_n} \right\} \tag{3.24}$$

$$\sigma_{vM_n} = \sqrt{\frac{3}{2} \left(\boldsymbol{\sigma}^D_{macro_n} - \boldsymbol{\alpha}^* \right) : \left(\boldsymbol{\sigma}^D_{macro_n} - \boldsymbol{\alpha}^* \right)} \tag{3.25}$$

As an initial guess, one might start with the average of the macro deviatoric stress points as the yield surface center (Lee and Barkey (2011a), p.201). Once $\boldsymbol{\alpha}^*$ is found, equation 3.23 is calculated. The mesoscopic shear is then given by:

$$\tau_{meso_n} = \frac{1}{2} \left(\sigma^D_{meso,1_n} - \sigma^D_{meso,3_n} \right) \tag{3.26}$$

where $\sigma^D_{meso,1_n}$ and $\sigma^D_{meso,3_n}$ are the largest and smallest principal stresses of the mesoscopic deviatoric stress tensor. The Dang Van criterion calculates an equivalent stress from τ_{meso_n} and $\sigma^H_{meso_n}$ to compare with a shear fatigue limit, τ_E. No fatigue damage will occur if (Lee and Barkey (2011a), p.202):

$$\max \left\{ \max_{n} \left| \tau_{meso_n} + b\, \sigma^H_{meso_n} \right| \right\} \leq \tau_E \tag{3.27}$$

where b is the hydrostatic stress sensitivity. A graphic depiction of the Dang Van criterion is shown in Figure 3.3. As long as the $\left(\sigma^H_{meso_n}, \tau_{meso_n} \right)$ coordinate at each point in the loading history remains inside the "safe zone" depicted in Figure 3.3, the component will not accrue fatigue damage.

A safety factor at any point in the loading history can be calculated as:

$$SF_n = \frac{\tau_E}{\tau_{meso_n} + b\, \sigma^H_{meso_n}} \tag{3.28}$$

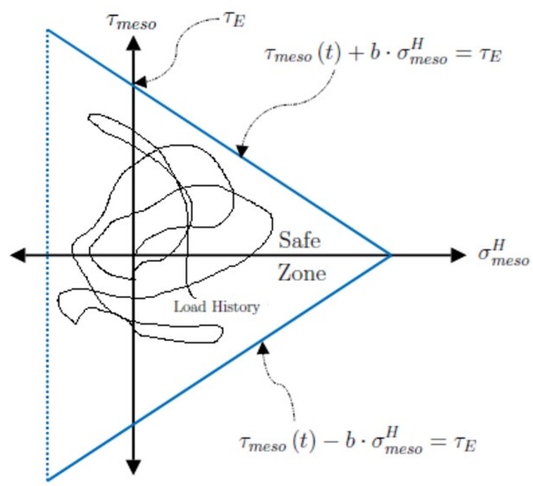

Figure 3.3. Dang Van Criterion

As a design tool, the Dang Van multiscale method is very effective for components intended for infinite life (Safe Technology Ltd. (2002), p.7-51) and is the standard solution technique implemented in commercial fatigue analysis software (e.g., fe-safe, eFatigue, nCode DesignLife, etc.). However, there are some drawbacks. Since it is a design tool for predicting infinite life, it is unreliable for predicting finite life when the stress sum exceeds the shear fatigue limit, τ_E (Safe Technology Ltd. (2002), p.7-49) though there have been proposed modifications to allow for finite life (e.g., Charkaluk et al. (2009)). Second, it requires more materials data than other criterions such as the shear fatigue limit, τ_E, and the hydrostatic stress sensitivity, b.

3.2 Palmgren-Miner Rule

Real world engineering components are rarely subjected to a single load amplitude. This brings up the question of how to predict failure for an engineering component that is subject to several different load amplitudes. According to Palmgren (Palmgren (1924)) and Miner (Miner (1945)), the total damage is the linear sum of individual damages caused by each set of cycles at their respective amplitudes. Thus, the Palmgren-Miner rule (or, simply, “Miner's rule") states that failure occurs when Miner (1945)):

$$1 = \sum_{i=1}^{kn} \frac{n_i}{N_{f,i}} \tag{3.29}$$

where kn are the total number of stress amplitude bins, n_i is the number of cycles of the ith amplitude bin, and $N_{f,i}$ is the fatigue life at the ith amplitude. Palmgren and Miner assumed that the sum of the damages from all amplitudes, the critical damage value, D, would be equal to one at failure. However, researchers have since determined that D depends on the material, varying from 0.15 to 1.06 (Lee and Barkey (2011b), p.150):

$$D = \sum_{i=1}^{kn} \frac{n_i}{N_{f,i}} \tag{3.30}$$

Based on experiments, the recommendation for steels, steel casings, and aluminum alloys is $D = 0.6$ and $D = 1.0$ for ductile irons, grey cast irons, and malleable cast irons (Lee and Barkey (2011b), p.150).

Over the years, many researchers have pointed out two serious problems with the Palmgren-Miner rule. First, the Miner rule assumes that stress amplitudes below the fatigue limit will not cause damage (Schijve (2009), p.299). This is false since "stress cycles with amplitudes below the fatigue limit could become damaging if some of the subsequent stress amplitudes exceed the original fatigue limit" because it "is believed that the increase in crack driving force due to periodic overloads will overcome the original grain barrier strength and help the crack to propagate until failure" (Lee and Barkey (2011b), p.119). Another criticism is that the Miner rule assumes that the sequence of loading amplitudes is irrelevant (Suresh (1998), p.227). However, if a larger amplitude which results in a compressive residual stress precedes a smaller amplitude stress, this would help lower the stress during the tensile portion of the low amplitude cycles resulting in a longer fatigue life than Miner's rule would predict (Schijve (2009), p.301). Conversely, if the larger amplitude ended with a tensile residual stress, this would amplify the tensile peaks of the low amplitude cycles resulting in a shorter fatigue life than Miner's rule would predict (Schijve (2009), p.302). Nevertheless, for many sequences of loading these criticisms are less problematic, and Miner's rule can give a rough approximation of fatigue life (Lemaitre and Desmorat (2005), p.280).

3.3 Cycle Counting Methods for Random Loading

For variable amplitude loading, it is difficult to assess when a cycle has occurred and at what amplitude. The various means for determining cycle count, amplitude, and mean stress are briefly discussed in this section. These algorithms are then paired with the Palmgren-Miner rule discussed in the previous section to calculate accumulated damage.

3.3.1 Counting Methods for Uniaxial Loading

It is necessary to discuss cycle counting methods for uniaxial loading because they can be used in multiaxial cases with some modifications.

3.3.1.1 Rainflow Cycle Counting (Matsuishi-Endo)

Prior to the late 1960's, there were several counting methods such as the level crossing analysis, peak and valley counting, and range counting that sought to provide a solution to the variable amplitude problem, but each of these methods gave less than satisfactory results (Safe Technology Ltd. (2002), pp.4-10 to 4-13). Then in the late 1960's, Japanese engineers Matsuishi and Endo (Matsuishi and Endo (1968)) developed the rainflow algorithm, the first widely accepted algorithm for extracting cycles from variable amplitude loading (Lee and Tjhung (2011), p.90). The term, 'rainflow,' comes from the fact that when one turns the loading history 90° clockwise and starts counting cycles, one is to envision rain flowing down the roof levels of a Japanese pagoda (Safe Technology Ltd. (2002), p.4-4). The rainflow cycle counting algorithm is as follows (Matsuishi and Endo (1968); Lee and Tjhung (2011), pp.91-92; Suresh (1998), pp.266-267):

1. Rotate the loading history 90° clockwise so that the time axis points downward.

2. For a periodic loading history, rearrange the loading history so that the peak with the largest magnitude of stress (positive or negative) comes first. This way, the algorithm will be able to identify full cycles. For a non-periodic loading history, the algorithm will have to identify half-cycles in addition to full cycles.

3. The "rain" flows on both "sides" of the plot and starts flowing at the upper part of each "roof" level.

4. A loading reversal (i.e., a half-cycle) happens by allowing the rain flow to continue until:

   (a) It "drips" down past an upper roof level of greater stress magnitude than the upper "roof" level from which it came.

   (b) It merges with "rain" flowing from a "roof" above.

   (c) It falls past all subsequent peaks.

5. Identify full cycles by pairing half-cycles of identical magnitude but opposite sense.

3.3.1.2 Three-Point Cycle Counting (Socie-Downing)

Although Matsuishi and Endo's cycle counting method was a breakthrough at the time, it has since been replaced by simpler algorithms which are computationally more efficient (Safe Technology Ltd. (2002), p.4-4; Lee and Tjhung (2011), p.101). The three-point cycle counting algorithm of Downing and Socie (Downing and Socie (1982)), for instance, is recommended by the American Society for Testing and Materials (ASTM (2005)) and is frequently implemented in commercial fatigue analysis codes (Safe Technology Ltd. (2002), p.4-3). As its name suggests, the three-point algorithm simplifies things by considering only three points at a time. Once a cycle is found, it will eliminate the first two points of the three under consideration from the history and re-start the analysis from the beginning. This way, it can automatically pair-up reversals in the earlier loading history with opposite reversals that happen only later in the history. The three-point algorithm is as follows (Downing and Socie (1982)):

1. Initialize load sequence counter: $n = 2$.
2. $n = n + 1$
3. Let $\sigma_1 = \sigma(n-2)$, $\sigma_2 = \sigma(n-1)$, and $\sigma_3 = \sigma(n)$.
4. Let $X = |\sigma_3 - \sigma_2|$ where the two vertical bars denote the absolute value operation.
5. Let $Y = |\sigma_1 - \sigma_2|$.
6. Compare X and Y:
   (a) If $X \geq Y$ and σ_1 <u>is</u> the first point in the loading history sequence (i.e., $\sigma_1 = \sigma(1)$), then:
      i. The range is equal to Y.
      ii. The mean is $\frac{1}{2}(\sigma_1 + \sigma_2)$.
      iii. A half-cycle is counted, and the above range and mean are recorded.
      iv. Remove only σ_1 from the loading history and re-label the data points starting at 1.
      v. Repeat analysis starting with step 1.
   (b) If $X \geq Y$ and σ_1 is <u>not</u> the first point in the loading history sequence (i.e., $\sigma_1 \neq \sigma(1)$), then:
      i. The range is equal to Y.

ii. The mean is $\frac{1}{2}(\sigma_1 + \sigma_2)$.

iii. A full cycle is counted, and the above range and mean are recorded.

iv. Remove σ_1 and σ_2 from the loading history and re-label the data points starting at 1.

v. Repeat analysis starting with step 1.

(c) If $X < Y$, then:

i. No cycle is formed.

ii. Go to step 2.

3.3.2 Counting Methods for Multiaxial Loading

Since most engineered parts experience loading multiaxially, it is necessary to formulate counting methods that can handle the whole stress tensor rather than a single stress value. In the case of proportional loading, any uniaxial cycle counting method may be used along with a signed equivalent stress (Lee and Tjhung (2011), p.106). For non-proportional loading, one could use a critical plane approach along with any of the uniaxial cycle counting methods for each potential failure plane (Lee and Tjhung (2011), p.113).

Alternatively, Wang and Brown (Wang and Brown (1996)) developed a fully multiaxial cycle counting technique based on Matsuishi and Endo's rainflow method (see section 3.3.1.1) to extract reversals from a complex loading history. It uses the maximum von Mises equivalent stress range, $\Delta\sigma_{eq}$, as the basis for reversal counting. The following description of the algorithm is based on that found in Lee and Tjhung (Lee and Tjhung (2011), pp.108-109):

1. Calculate the von Mises equivalent stress history:

$$\sigma_{eq}(t) = \sqrt{\frac{1}{2}\left[(\sigma_x(t) - \sigma_y(t))^2 + (\sigma_y(t) - \sigma_z(t))^2 + (\sigma_z(t) - \sigma_x(t))^2\right] + 3\left[\tau_{xy}^2(t) + \tau_{yz}^2(t) + \tau_{zx}^2(t)\right]} \tag{3.31}$$

2. Reorder the von Mises equivalent stress history to begin with the maximum von Mises stress. This point now starts at time t_0, and the stress at t_0 becomes the reference stress. Any stress points that came before this maximum are moved to the end of the loading history.

3. Calculate the relative von Mises stress history with respect to the reference point:

$$\Delta\sigma_{eq}(t) = \sqrt{\frac{1}{2}\left[(\Delta\sigma_x(t) - \Delta\sigma_y(t))^2 + (\Delta\sigma_y(t) - \Delta\sigma_z(t))^2 + (\Delta\sigma_z(t) - \Delta\sigma_x(t))^2\right] + 3\left[\Delta\tau_{xy}^2(t) + \Delta\tau_{yz}^2(t) + \Delta\tau_{zx}^2(t)\right]} \tag{3.32}$$

where:

$$\Delta\sigma_x\left(t\right) = \sigma_x\left(t\right) - \sigma_x\left(t_0\right) \tag{3.33}$$

$$\Delta\sigma_y\left(t\right) = \sigma_y\left(t\right) - \sigma_y\left(t_0\right) \tag{3.34}$$

$$\Delta\sigma_z\left(t\right) = \sigma_z\left(t\right) - \sigma_z\left(t_0\right) \tag{3.35}$$

$$\Delta\tau_{xy}\left(t\right) = \tau_{xy}\left(t\right) - \tau_{xy}\left(t_0\right) \tag{3.36}$$

$$\Delta\tau_{yz}\left(t\right) = \tau_{yz}\left(t\right) - \tau_{yz}\left(t_0\right) \tag{3.37}$$

$$\Delta\tau_{zx}\left(t\right) = \tau_{zx}\left(t\right) - \tau_{zx}\left(t_0\right) \tag{3.38}$$

4. Identify the point at which $\Delta\sigma_{eq}$ is maximized, $\max\left(\Delta\sigma_{eq}\right)$. This point is at time t_E: $\max\left(\Delta\sigma_{eq}\right) = \Delta\sigma_{eq}\left(t_E\right)$.

5. All the points that cause $\Delta\sigma_{eq}\left(t\right)$ to monotonically increase from $(t_0, 0)$ to (t_E, $\max(\Delta\sigma_{eq})$) form a reversal. If there are valleys and possibly small peaks in between (i.e., values of $\Delta\sigma_{eq}$ below where it stopped monotonically increasing), then all these points are excluded from the reversal. They will be examined in later passes.

6. The remaining points are grouped into segments of continuous points. Each segment is evaluated individually.

7. For the new segment being evaluated, the first stress point becomes the new reference stress, and the time at which it occurs becomes the new t_0.

8. Repeat steps 3 through 7 until all data segments have been evaluated.

3.4 Conclusion to Traditional Approaches

Over the course of this chapter, we've seen several problems with the traditional approaches for calculating fatigue damage.

Equivalent Stress Criterions

1. Some of the criterions are too simple. The principal stress criterion only uses the first principal stress, σ_1, and ignores the other two. The Sines criterion assumes that the fatigue limits in torsion and fully-reversed bending are in a constant ratio for all metals.

2. Some of the criterions are limited to the type of material. Some are limited to brittle solids only while others are limited to ductile solids.

3. None of these approaches works well for non-proportional loading.

4. The von Mises criterion has an issue with assigning a sign to the von Mises stress.

5. The Sines criterion requires extra test data.

Critical Plane Methods

1. These approaches are limited to ductile solids.

2. These approaches are computationally expensive.

3. Even small angular increments during the plane search can lead to discretization errors.

4. The Findley approach does not predict that, in the very high cycle fatigue range (i.e., $> 10^7$), fatigue life in cyclic torsion is independent of a superimposed mean shear stress.

5. The McDiarmid approach requires extra test data and is limited to the very high cycle fatigue range.

The Dang Van Criterion

1. This approach is limited to infinite life design and cannot be used to calculate a finite life.

2. This approach also requires extra material data.

The Palmgren-Miner Rule

1. The Miner rule assumes that amplitudes below the fatigue limit cause no fatigue damage (which is false).

2. It also assumes that the sequence of loading amplitudes doesn't matter (which is also false).

Given that fatigue is a complex interaction of grains at the microscopic level, it is difficult to create an algorithm that will be able to avoid all of the above pitfalls. Creating a single algorithm that can handle both ductile and brittle solids, for instance, would be very challenging. Nonetheless, we can avoid some of the above drawbacks by utilizing a continuum damage mechanics approach which will be detailed in the next chapter.

CHAPTER 4

THE TWO SCALE PROGRESSIVE FATIGUE DAMAGE MODEL

Having detailed the extended space-time finite element method, we now need a means for calculating the cumulative fatigue damage from the finite element results. This chapter briefly lays out the idea behind continuum damage mechanics (CDM) and then details the formulation of the two scale progressive fatigue damage model and its implementation.

4.1 Introduction

The idea of using a scalar variable, D, to quantify damage started with L.M. Kachanov (Kachanov (1958)) who used it to model material deterioration, crack initiation, and fracture due to brittle creep in metals. Rabotnov (Rabotnov (1969)) extended Kachanov's idea "by allowing for the increase in creep rate $\dot{\varepsilon}^c$ due to creep damage" (Murakami (2012), p.219) and expressed an equation for damage evolution. Jan Hult (Hult (1972)) was the first to use the term, "continuum damage mechanics," again, in reference to creep damage. Lemaitre (Lemaitre (1971)) extended the Kachanov-Rabotnov damage law to low-cycle fatigue by using the law to derive a damage evolution equation that was a function of energy. This damage evolution equation was derived from a thermodynamics framework, and this framework became the standard manner of modeling fatigue damage evolution up to the present since it "provided the necessary scientific basis to justify continuous damage mechanics as a theory" (Lemaitre (1985), p.83). Several high- and low-cycle fatigue models were then developed by Chaboche (Chaboche (1974, 1981, 1987)) and Lemaitre (Lemaitre (1985)). Like Lemaitre's earlier work (Lemaitre (1971)), these models used a damage evolution equation that was a function of "macroscale" (Lemaitre (1985), p.83) variable amplitudes of the form: $\frac{dD}{dN} = f\left(\Delta\sigma, \Delta\varepsilon\right)$ (Bhamare (2012), p.98).

With a fully coupled analysis like those used in the previous works cited above, strain and damage affect each other globally (Lemaitre and Doghri (1994), p.199). This sort of analysis is necessary for certain types of failures such as creep because "the damage is not localized but diffused in a large region" (Lemaitre and Doghri (1994), p.199). Given that the damage in HCF situations is highly localized, however, an opportunity was seen to simplify the CDM fatigue analysis by separating the macroscale structure calculation from the microscale damage evaluation. As with the fully coupled models mentioned above, the microscale evaluation is derived from the thermodynamics framework. In this locally coupled analysis, the macroscale structure calculation can be solved with the finite element method or other method using a purely elastic material model. The strains at this scale can then be used to evaluate damage locally at microscale where defects such as inclusions exist and where plasticity occurs and microcracks form. This locally coupled method became known as the two scale progressive fatigue damage model (Lemaitre et al. (1999)).

The two scale model has been used to predict service life for both low and high cycle fatigue and applied to both uniaxial and multiaxial analyses (Lemaitre and Doghri (1994)). Lemaitre et al. (Lemaitre et al. (1999)) then showed the power of the two scale model for a non-zero mean stress and non-proportional loading comparing well with the Dang Van criterion. More recently, Latrou et al. (Lautrou et al. (2009)) used the two scale model to accurately predict the fatigue life of steel welded joints commonly used in naval structures. Additionally, dos Santos et al. (dos Santos et al. (2012)) modified Lemaitre et al.'s model (Lemaitre et al. (1999)) by means of the Soderberg fatigue relation to account for high mean stress effects in cardiovascular stents and showed that the modified model has good agreement with experimental results. Others (e.g., Flaceliere et al. (2007a,b)) have created two scale models of their own. Finally, Desmorat et al. (Desmorat et al. (2007)) added the ability for the two scale model to take into account random temperature changes and proved its predictive power even in random thermo-mechanical loading by showing good agreement with a pressure-vessel testing experiment. It is this model which we will use to predict thermo-mechanical fatigue life from our XTFEM simulation.

A few definitions are in order:

- Atomic Scale - "The mechanical properties of materials are determined by the constituent atoms or molecules, their array, and the kind of interatomic or intermolecular forces between them," and, the "damage of materials in the atomic scale is induced by the separation of these interatomic or intermolecular bonds" (Murakami (2012), p.4).
- Microscopic Scale - Visually, the material at this scale has a discontinuous structure but may have some continuous regions. The damage at this scale is found in "microcavities, microcracks, or in decohesion in microstructures of materials" (Murakami (2012), p.4).
- Mesoscopic Scale - "If the value representing a mechanical property or a mechanical state of the material averaged over the small region can be expressed as a continuous function of the position x of the material point P, then the material can be idealized as a continuum" (Murakami (2012), p.5)
- Macroscopic Scale - At this scale, "every point in a material can be viewed as a material point in a continuum" (Murakami (2012), p.5).

The thermo-mechanical two scale model works as follows (see Figure 4.1):

1. The macroscale stucture is broken down into representative volume elements (RVE). An RVE is the smallest mesoscale region in a macroscopic body in which "the material with discontinuous structures...can be statistically homogeneous and the mechanical state of the material...can be represented by the statistical average of the mechanical variables" (Murakami (2012), p.11). When the two scale model is combined with finite element analysis as in this work, the RVE is equivalent to a finite element.
2. Stress and strain calculations at the RVE level are done using a thermo-elastic material model in finite element analysis software.

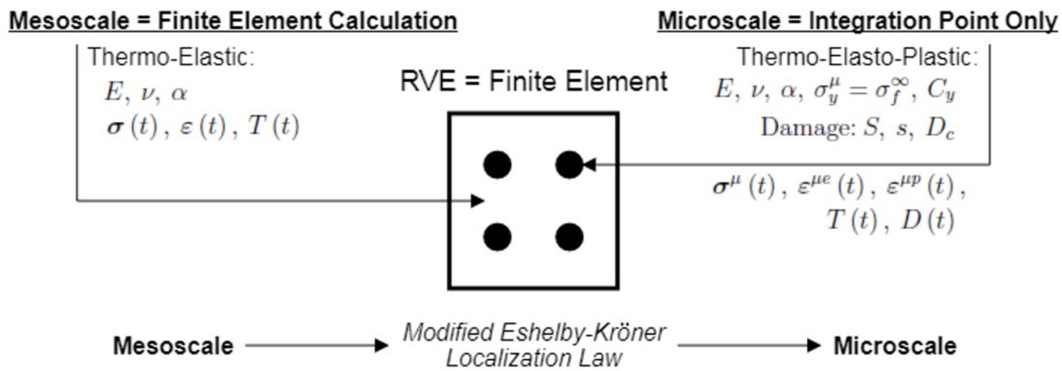

Figure 4.1. Two Scale Progressive Fatigue Damage Model

3. As a post-processing stage after each time step, these values at mesoscale are applied to each microscale material point (i.e., integration points when using FEM) through the modified Eshelby-Kröner localization law (Eshelby (1957); Kröner (1961); Desmorat et al. (2015)). At this scale, plasticity with kinematic hardening and damage can occur. The asymptotic fatigue limit, σ_f^∞, is used as the yield strength, and damage, D, is a function of the accumulated plastic strain, p^μ. Finally, the material parameters such as the damage strength, S, and the damage exponent, s, are obtained from an isothermal, uniaxial S-N curve.

4. Once the damage at a microscale material point reaches a critical value, D_c, a crack initiates, and crack propagation is modeled using fracture mechanics tools (Desmorat et al. (2007), p.913). In the XTFEM simulation, we have chosen to model this process using element deletion, but one could utilize more sophisticated FEM fracture techniques such as XFEM.

4.2 Material Parameters

There are a number of material parameters utilized in the two scale damage model, and not all of them are familiar to many readers.

4.2.1 E, ν, α, σ_u, C_y, ε_{pD}

Young's Modulus, E, Poisson's Ratio, ν, the coefficient of thermal expansion, α, and the ultimate tensile strength, σ_u, can all be determined from uniaxial tension tests. The plastic modulus, C_y, can be estimated by drawing a straight line on a uniaxial tension curve from the yield point to the point where the curve reaches the ultimate strength. The slope of this line is the plastic modulus. The monotonic plastic strain damage threshold, ε_{pD}, is the amount of plastic strain a material undergoes by the time it reaches its ultimate strength. It can be derived from the uniaxial tension curve by drawing a line of slope E from the point of ultimate strength to the strain axis. The value of the strain-intercept is ε_{pD}. For all of these parameters, no scale transition needs to be made (see subsection 4.3.2).

4.2.2 h, D_c

From experimental data, the micro-defects closure parameter, h, is typically $h \approx 0.2$ for metals (Lemaitre and Desmorat (2005), p.16; Desmorat et al. (2007), p.912). The critical damage, D_c, is also derived from experimental data and is between 0.2 and 0.5 for many materials, especially ductile ones (Lemaitre and Desmorat (2005), p.65; Murakami (2012), p.153). The default value for metals is $D_c = 0.3$ (Desmorat et al. (2007), p.920). For ductile fracture, the critical damage can be estimated using (Murakami (2012), p.153):

$$D_c = 1 - \frac{\sigma_R}{\sigma_u} \tag{4.1}$$

where σ_R is the stress at final fracture and σ_u is the ultimate tensile strength.

4.2.3 σ_f^∞, S, s

The asymptotic fatigue limit, σ_f^∞, also known as the endurance limit, is the fatigue strength at which an isothermal, fully-reversed ($R = -1$) S-N curve plateaus, usually around 10^6 cycles (Suresh (1998), p.222). If the stress amplitude is below this strength, fatigue failure will, theoretically, never occur. For non-ferrous materials which do not exhibit a clear asymptotic fatigue limit, σ_f^∞ may be taken as the fatigue strength at 10^7 cycles or more depending on the material (Suresh (1998), p.223). The damage strength, S, and damage exponent, s, are also derived from a fully-reversed S-N curve. Using a standalone script, the two parameters are optimized using a non-linear least-squares fit algorithm so as to fit the following equation to the experimental S-N curve data (Desmorat et al. (2007), p.914):

$$N_R = N_D + \frac{(2\,E\,S)^s\,\mathcal{G}\,D_c}{\left(\sigma_f^\infty\right)^{2s}\left[\Delta\sigma - 2\,\sigma_f^\infty\right]\left[R_{\nu_{min}}^{s} + R_{\nu_{max}}^{s}\right]} \tag{4.2}$$

where:

$$N_D = \frac{1}{4}\,\varepsilon_{pD}\,\frac{\mathcal{G}^2}{C_y}\,\frac{\sigma_u - \sigma_f^\infty}{\left(\frac{\Delta\sigma}{2} - \sigma_f^\infty\right)^2} \tag{4.3}$$

$$\mathcal{G} = \frac{3\,E}{2\,(1+\nu)}\,(1-b) + C_y\,(1-D) \approx \frac{3\,E}{2\,(1+\nu)}\,(1-b) \tag{4.4}$$

$$b = \frac{2}{15}\left(\frac{4-5\nu}{1-\nu}\right) \tag{4.5}$$

$$R_{\nu_{min}} = \frac{2}{3}\,(1+\nu) + \frac{1}{3}\,(1-2\,\nu)\left[\frac{\sigma_{min}}{\sigma_f^\infty}\right]^2 \tag{4.6}$$

$$R_{\nu_{max}} = \frac{2}{3}\,(1+\nu) + \frac{1}{3}\,(1-2\,\nu)\left[\frac{\sigma_{max}}{\sigma_f^\infty}\right]^2 \tag{4.7}$$

N_R is the number of cycles to rupture, N_D is the number of cycles to crack initiation, $R_{\nu_{min}}$ and $R_{\nu_{max}}$ are the stress triaxiality ratio during loading at $\sigma = \sigma_{min}$ and $\sigma = \sigma_{max}$, respectively, and $\Delta\sigma > 2\,\sigma_f^\infty$. Equations 4.6 and 4.7 assume that the micro-defects closure parameter has been set equal to $h = 1$ which is fine if the experimental S-N curve was created using a zero mean stress (i.e., $R = -1$). When there is a non-zero mean stress, however, the fact that the damage evolution in compression is less than that in tension necessitates the use of the micro-defects closure parameter, h (Lemaitre and Desmorat (2005), pp.12-13;

Desmorat et al. (2007), p.915). For a non-zero mean stress, $R_{\nu_{min}}$ and $R_{\nu_{max}}$ in equation 4.2 are replaced with $R_{\nu h_{min}}$ and $R_{\nu h_{max}}$, respectively:

$$\begin{aligned} R_{\nu h_{min}} = \frac{1+\nu}{9} & \left[\left\langle 2 + \frac{\sigma_{min}}{\sigma_f^\infty} \right\rangle^2 + 2 \left\langle -1 + \frac{\sigma_{min}}{\sigma_f^\infty} \right\rangle^2 + h \left\langle -2 - \frac{\sigma_{min}}{\sigma_f^\infty} \right\rangle^2 + 2\, h \left\langle 1 - \frac{\sigma_{min}}{\sigma_f^\infty} \right\rangle^2 \right] \\ & - \nu \left\langle \frac{\sigma_{min}}{\sigma_f^\infty} \right\rangle^2 - \nu\, h \left\langle \frac{\sigma_{min}}{\sigma_f^\infty} \right\rangle^2 \end{aligned} \tag{4.8}$$

$$\begin{aligned} R_{\nu h_{max}} = \frac{1+\nu}{9} & \left[\left\langle 2 + \frac{\sigma_{max}}{\sigma_f^\infty} \right\rangle^2 + 2 \left\langle -1 + \frac{\sigma_{max}}{\sigma_f^\infty} \right\rangle^2 + h \left\langle -2 - \frac{\sigma_{max}}{\sigma_f^\infty} \right\rangle^2 + 2\, h \left\langle 1 - \frac{\sigma_{max}}{\sigma_f^\infty} \right\rangle^2 \right] \\ & - \nu \left\langle \frac{\sigma_{max}}{\sigma_f^\infty} \right\rangle^2 - \nu\, h \left\langle \frac{\sigma_{max}}{\sigma_f^\infty} \right\rangle^2 \end{aligned} \tag{4.9}$$

4.2.4 Stored Energy Damage Threshold, w_D

While the microscale accumulated plastic strain at damage initiation, p_D^μ, is typically only 0.1 to 0.3 in monotonic tension (for metals), it can be several hundreds of percent in fatigue depending on the loading condition (Desmorat et al. (2007), p.913). Because it is load dependent and thus cannot be a material parameter, p_D^μ cannot be used to determine when the material begins to accumulate damage (Desmorat et al. (2007), p.913). Instead, we must find a different parameter not dependent upon the load magnitude. The stored energy density meets this requirement as it is loading independent and thus can serve as a material property (Desmorat et al. (2007), p.913). For both isotropic and kinematic hardening, we have (Desmorat et al. (2007), p.913):

$$w_s = \int_0^t \left(R\, \dot{p} + X_{ij}\, \dot{\alpha}_{ij} \right) dt \tag{4.10}$$

In fatigue, kinematic hardening is periodic (and will be lost) while isotropic hardening is monotonic (Desmorat et al. (2007), p.913). So, only using the isotropic terms and applying equation 4.10 to the microscale, we have:

$$w_s^\mu = \int_0^t \left(\sigma_{eq}^\mu - \sigma_y^\mu \right) \dot{p}^\mu\, dt \tag{4.11}$$

This is considered the "envelope of the maxima in stored energy reached during complex loading" (Desmorat et al. (2007), p.913). For monotonic loading with linear hardening and assuming no scale transition has to be considered, the stored energy damage threshold is:

$$w_D = w_s^{\mu}\,(p = \varepsilon_{pD}) = \left(\sigma_y + \frac{1}{2}\,C_y\,\varepsilon_{pD} - \sigma_y^{\mu}\right)\varepsilon_{pD} \tag{4.12}$$

Using the approximation $\sigma_u \approx \sigma_y + \frac{1}{2}\,C_y\,\varepsilon_{pD}$ and the assumption that the microscale yield stress is equal to the asymptotic fatigue limit, σ_f^{∞} (see subsection 4.3.2 below), we have:

$$w_D \approx \left(\sigma_u - \sigma_f^{\infty}\right)\varepsilon_{pD} \tag{4.13}$$

4.3 Formulation

This section derives the equations used in the two-scale progressive fatigue damage model subroutine.

4.3.1 Mesoscale

We begin with the laws of linear elasticity at mesoscale. The constitutive law for an isotropic elastic material is:

$$\sigma_{ij} = \lambda\left[\varepsilon_{kk} - 3\,\alpha\,(T - T_{ref})\right]\delta_{ij} + \left[2\,\mu\,\varepsilon_{ij} - \alpha\,(T - T_{ref})\,\delta_{ij}\right] \tag{4.14}$$

Solving for the strains, we arrive at:

$$\varepsilon_{ij} = \varepsilon_{ij}^{e} = \frac{1+\nu}{E}\,\sigma_{ij} - \frac{\nu}{E}\,\sigma_{kk}\,\delta_{ij} + \alpha\,(T - T_{ref})\,\delta_{ij} \tag{4.15}$$

For cases involving plasticity, the strains can be additively decomposed into elastic and plastic strains:

$$\varepsilon_{ij} = \varepsilon_{ij}^{e} + \varepsilon_{ij}^{p} \tag{4.16}$$

For high-cycle fatigue (HCF) applications, microcracks only form in a small number of crystal grains (Murakami (2012), p.202). Consequently, we are justified in assuming that, at mesoscale, $\varepsilon_{ij}^{p} \approx 0$.

4.3.2 Modified Eshelby-Kröner Localization Law

Applying the Eshelby-Kröner localization law (Eshelby (1957); Kröner (1961)) to equation 4.15:

$$\varepsilon_{ij}^{\mu e} = \frac{1+\nu}{E\,(1-D)}\,\sigma_{ij}^{\mu} - \frac{\nu}{E\,(1-D)}\,\sigma_{kk}^{\mu}\,\delta_{ij} + \alpha^{\mu}\,(T - T_{ref})\,\delta_{ij} \tag{4.17}$$

where $\varepsilon_{ij}^{\mu e}$ is the microscale elastic strain tensor, σ_{ij}^{μ} is the microscale stress tensor, α^{μ} is the microscale coefficient of thermal expansion, and D is the damage variable. More generally, the damage variable is a tensor, but reduces to a scalar for isotropic materials (Lemaitre and Doghri (1994), p.198). It represents "the maximum equivalent area density of microcracks or microcavities which lies in any cross plane that can be defined in a representative volume element" (Lemaitre and Doghri (1994), p.198). Applying the localization law to equation 4.16:

$$\varepsilon_{ij}^{\mu} = \varepsilon_{ij}^{\mu e} + \varepsilon_{ij}^{\mu p} \tag{4.18}$$

where ε_{ij}^{μ} is the microscale total strain tensor and $\varepsilon_{ij}^{\mu p}$ is the microscale plastic strain tensor. Assuming that $\alpha^{\mu} = \alpha$ and that, for HCF applications, $\varepsilon_{ij}^{p} \approx 0$, the modified Eshelby-Kröner localization law is (Eshelby (1957); Kröner (1961); Desmorat et al. (2015)):

$$\varepsilon^{\mu H} = \frac{1}{1 - a\,D}\left[\varepsilon^{H} - a\,D\,\alpha\,(T - T_{ref})\right] \tag{4.19}$$

$$\varepsilon_{ij}^{\mu D} = \frac{1}{1 - b\,D}\left[\varepsilon_{ij}^{D} - b\,(1-D)\,\varepsilon_{ij}^{\mu p}\right] \tag{4.20}$$

where $\varepsilon^{\mu H}$ is the hydrostatic microstrain, $\varepsilon_{ij}^{\mu D}$ is the deviatoric microstrain tensor, and a and b are the Eshelby parameters for a spherical inclusion (Eshelby (1957)):

$$a = \frac{1+\nu}{3\,(1-\nu)} \tag{4.21}$$

$$b = \frac{2}{15}\left(\frac{4-5\nu}{1-\nu}\right) \tag{4.22}$$

Given equations 4.19, 4.20, and using $\varepsilon_{ij}^{\mu} = \varepsilon_{ij}^{\mu D} + \varepsilon^{\mu H}\delta_{ij}$, the total microscale strains are given as:

$$\varepsilon_{ij}^{\mu} = \frac{1}{1 - b\,D}\left[\varepsilon_{ij} + \frac{(a-b)\,D}{3\,(1-a\,D)}\,\varepsilon_{kk}\,\delta_{ij} + b\,(1-D)\,\varepsilon_{ij}^{\mu p}\right] - \frac{a\,D\,\alpha}{1-a\,D}\,(T - T_{ref})\,\delta_{ij} \tag{4.23}$$

Because we assume that plasticity with kinematic hardening can happen at microscale, $\varepsilon_{ij}^{\mu p} \geq 0$.

According to Lemaitre's damage law, we distinguish between the actual stress at microscale and the effective stress at microscale. The relationship between the two is given as:

$$\tilde{\sigma}_{ij}^{\mu} = \frac{\sigma_{ij}^{\mu}}{(1-D)} \tag{4.24}$$

where $\tilde{\sigma}_{ij}^{\mu}$ is the effective stress at microscale. For HCF applications, we assume that the microscale yield strength is the asymptotic fatigue limit, σ_f^{∞}. Hence, the microscale yield criterion, f^{μ}, is given by:

$$f^{\mu} = \left(\tilde{\sigma}_{ij}^{\mu} - X_{ij}^{\mu}\right)_{eq} - \sigma_f^{\infty} \tag{4.25}$$

where $(\cdot)_{eq}$ denotes the von Mises norm. The plastic strain rate definition applied to microscale is:

$$\dot{\varepsilon}_{ij}^{\mu p} = \frac{3}{2}\left(\frac{\tilde{\sigma}_{ij}^{\mu D} - X_{ij}^{\mu}}{\left(\tilde{\sigma}_{ij}^{\mu} - X_{ij}^{\mu}\right)_{eq}}\right)\dot{p}^{\mu} \tag{4.26}$$

The anisothermal variation of the Prager linear kinematic hardening law is:

$$\frac{d}{dt}\left(\frac{X_{ij}^{\mu}}{C_y}\right) = \frac{2}{3}\,\dot{\varepsilon}_{ij}^{\mu p}\,(1-D) \tag{4.27}$$

Lemaitre's damage evolution law applied to microscale is:

$$\dot{D} = \left(\frac{Y^{\mu}}{S}\right)^{s}\dot{p}^{\mu} \tag{4.28}$$

Using the consistency condition ($f^{\mu} = 0$; $\dot{f}^{\mu} = 0$), the microscale plastic multiplier is:

$$\dot{\lambda} = \dot{p}^{\mu}\,(1-D) \tag{4.29}$$

4.3.3 Derivation of the Damage Energy Release Rate

The damage energy release rate, Y^μ, in Lemaitre's damage evolution law (equation 4.28) is the release rate of the elastic strain energy caused by the development of the damage variable, D (Murakami (2012), p.97). The damage energy release rate thus plays a similar role to the strain energy release rate in fracture mechanics (Chaboche (1977)). To derive the expression for Y^μ used in the two scale model, we first start with its definition (Murakami (2012), p.97):

$$Y^\mu = \frac{w_e^\mu}{1-D} \tag{4.30}$$

where w_e^μ is the microscale elastic strain energy density given as:

$$w_e^\mu = \frac{1}{2}\,\sigma_{ij}^\mu\,\varepsilon_{ij}^\mu \tag{4.31}$$

Note that the strain tensor, ε_{ij}^μ, in equation 4.31 does not include any thermal strains. Combining equation 4.31 with 4.17 gives:

$$\begin{aligned} w_e^\mu &= \frac{1}{2}\,\sigma_{ij}^\mu \left(\frac{1+\nu}{E\,(1-D)}\,\sigma_{ij}^\mu - \frac{\nu}{E\,(1-D)}\,\sigma_{kk}^\mu\,\delta_{ij} \right) \\ &= \frac{1+\nu}{2\,E\,(1-D)}\,\sigma_{ij}^\mu\,\sigma_{ij}^\mu - \frac{\nu}{2\,E\,(1-D)}\,\sigma_{kk}^\mu\,\sigma_{ij}^\mu\,\delta_{ij} \\ &= \frac{1+\nu}{2\,E\,(1-D)}\,\sigma_{ij}^\mu\,\sigma_{ij}^\mu - \frac{\nu}{2\,E\,(1-D)}\,(\sigma_{kk}^\mu)^2 \end{aligned} \tag{4.32}$$

Splitting the stress tensor, σ_{ij}^μ, into hydrostatic and deviatoric parts:

$$w_e^\mu = \frac{1+\nu}{2\,E\,(1-D)} \left(\sigma_{ij}^{\mu D} + \sigma^{\mu H}\delta_{ij}\right) \left(\sigma_{ij}^{\mu D} + \sigma^{\mu H}\delta_{ij}\right) - \frac{\nu}{2\,E\,(1-D)} \left(3\,\sigma^{\mu H}\right)^2 \tag{4.33}$$

Given that $\sigma_{ij}^{\mu D}\sigma^{\mu H}\delta_{ij} = 0$, we have:

$$w_e^\mu = \frac{1+\nu}{2\,E\,(1-D)} \left[\sigma_{ij}^{\mu D}\sigma_{ij}^{\mu D} + \left(\sigma^{\mu H}\right)^2 \delta_{ij}\,\delta_{ij}\right] - \frac{\nu}{2\,E\,(1-D)} \left(3\,\sigma^{\mu H}\right)^2 \tag{4.34}$$

Using the definition of the von Mises norm, $\sigma_{eq}^\mu = \sqrt{\frac{3}{2}\,\sigma_{ij}^{\mu D}\sigma_{ij}^{\mu D}}$, then $\sigma_{ij}^{\mu D}\sigma_{ij}^{\mu D} = \frac{2}{3}\left(\sigma_{eq}^\mu\right)^2$, and given that $\delta_{ij}\,\delta_{ij} = 3$, we have:

$$w_e^\mu = \frac{1+\nu}{2\,E\,(1-D)} \left[\frac{2}{3}\left(\sigma_{eq}^\mu\right)^2 + 3\left(\sigma^{\mu H}\right)^2\right] - \frac{\nu}{2\,E\,(1-D)} \left(3\,\sigma^{\mu H}\right)^2 \tag{4.35}$$

Simplifying equation 4.35, we get:

$$w_e^\mu = \frac{\left(\sigma_{eq}^\mu\right)^2}{2\,E\,(1-D)}\left[\frac{2}{3}\,(1+\nu) + 3\,(1-2\nu)\left(\frac{\sigma^{\mu H}}{\sigma_{eq}^\mu}\right)^2\right] \tag{4.36}$$

where $\frac{\sigma^{\mu H}}{\sigma_{eq}^\mu}$ is the triaxiality ratio. Combining equation 4.36 with equation 4.30, we have:

$$Y^\mu = \frac{\left(\sigma_{eq}^\mu\right)^2}{2\,E\,(1-D)^2}\left[\frac{2}{3}\,(1+\nu) + 3\,(1-2\nu)\left(\frac{\sigma^{\mu H}}{\sigma_{eq}^\mu}\right)^2\right] \tag{4.37}$$

Backing up a bit, we next combine equation 4.30 with equation 4.32 to get:

$$Y^\mu = \frac{1+\nu}{2\,E\,(1-D)^2}\,\sigma_{ij}^\mu\,\sigma_{ij}^\mu - \frac{\nu}{2\,E\,(1-D)^2}\,(\sigma_{kk}^\mu)^2 \tag{4.38}$$

We need to account for the fact that damage is less in compression than in tension due to the closure of micro-defects in compression (Desmorat et al. (2007), p.912). We first split up the stress tensor into tensile and compressive parts. To do this, we introduce the Macauley bracket:

$$\langle x\rangle = x\,H(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{4.39}$$

where $H(x)$ is the Heaviside function. Thus, the principal stress tensor becomes:

$$\begin{aligned} \sigma_{ij}^\mu &= \begin{bmatrix} \sigma_{11}^\mu & 0 & 0 \\ 0 & \sigma_{22}^\mu & 0 \\ 0 & 0 & \sigma_{33}^\mu \end{bmatrix} = \begin{bmatrix} \langle\sigma_{11}^\mu\rangle & 0 & 0 \\ 0 & \langle\sigma_{22}^\mu\rangle & 0 \\ 0 & 0 & \langle\sigma_{33}^\mu\rangle \end{bmatrix} - \begin{bmatrix} \langle-\sigma_{11}^\mu\rangle & 0 & 0 \\ 0 & \langle-\sigma_{22}^\mu\rangle & 0 \\ 0 & 0 & \langle-\sigma_{33}^\mu\rangle \end{bmatrix} \\ &= \begin{bmatrix} \langle\sigma_{11}^\mu\rangle^+ & 0 & 0 \\ 0 & \langle\sigma_{22}^\mu\rangle^+ & 0 \\ 0 & 0 & \langle\sigma_{33}^\mu\rangle^+ \end{bmatrix} - \begin{bmatrix} \langle\sigma_{11}^\mu\rangle^- & 0 & 0 \\ 0 & \langle\sigma_{22}^\mu\rangle^- & 0 \\ 0 & 0 & \langle\sigma_{33}^\mu\rangle^- \end{bmatrix} \end{aligned} \tag{4.40}$$

Using $\sigma_{ij}^\mu\,\sigma_{ij}^\mu = \left\langle\sigma_{ij}^\mu\right\rangle^+\left\langle\sigma_{ij}^\mu\right\rangle^+ + \left\langle\sigma_{ij}^\mu\right\rangle^-\left\langle\sigma_{ij}^\mu\right\rangle^-$ and $(\sigma_{kk}^\mu)^2 = \left(\langle\sigma_{kk}^\mu\rangle^+\right)^2 + \left(\langle\sigma_{kk}^\mu\rangle^-\right)^2$ (see proofs A.1 and A.3 in the Appendix), we have:

$$Y^\mu = \frac{1+\nu}{2\,E\,(1-D)^2}\left[\left\langle\sigma_{ij}^\mu\right\rangle^+\left\langle\sigma_{ij}^\mu\right\rangle^+ + \left\langle\sigma_{ij}^\mu\right\rangle^-\left\langle\sigma_{ij}^\mu\right\rangle^-\right] - \frac{\nu}{2\,E\,(1-D)^2}\left[\left(\langle\sigma_{kk}^\mu\rangle^+\right)^2 + \left(\langle\sigma_{kk}^\mu\rangle^-\right)^2\right] \tag{4.41}$$

We now introduce the micro-defects closure parameter, h, to account for the lower damage in compression. Multiplying the compressive terms and their corresponding damage variable by the closure parameter gives us:

$$Y^{\mu} = \frac{1+\nu}{2\,E}\left[\frac{\left\langle\sigma^{\mu}_{ij}\right\rangle^{+}\left\langle\sigma^{\mu}_{ij}\right\rangle^{+}}{(1-D)^{2}} + h\,\frac{\left\langle\sigma^{\mu}_{ij}\right\rangle^{-}\left\langle\sigma^{\mu}_{ij}\right\rangle^{-}}{(1-h\,D)^{2}}\right] - \frac{\nu}{2\,E}\left[\frac{\left\langle\sigma^{\mu}_{kk}\right\rangle^{2}}{(1-D)^{2}} + h\,\frac{\left\langle-\sigma^{\mu}_{kk}\right\rangle^{2}}{(1-h\,D)^{2}}\right] \tag{4.42}$$

where $h \approx 0.2$ for metals (Lemaitre and Desmorat (2005), p.84). Substituting equation 4.24 into equation 4.42, we have:

$$Y^{\mu} = \frac{1+\nu}{2\,E}\left[\left\langle\tilde{\sigma}^{\mu}_{ij}\right\rangle^{+}\left\langle\tilde{\sigma}^{\mu}_{ij}\right\rangle^{+} + h\left(\frac{1-D}{1-h\,D}\right)^{2}\left\langle\tilde{\sigma}^{\mu}_{ij}\right\rangle^{-}\left\langle\tilde{\sigma}^{\mu}_{ij}\right\rangle^{-}\right] - \frac{\nu}{2\,E}\left[\left\langle\tilde{\sigma}^{\mu}_{kk}\right\rangle^{2} + h\left(\frac{1-D}{1-h\,D}\right)^{2}\left\langle-\tilde{\sigma}^{\mu}_{kk}\right\rangle^{2}\right] \tag{4.43}$$

4.3.4 Derivation of the Two-Scale Numerical Scheme

4.3.4.1 Elastic Prediction

For the elastic prediction step, we assume constant damage $D_{n+1} = D_n$, constant microscale plastic strain $\boldsymbol{\varepsilon}^{\mu p}_{n+1} = \boldsymbol{\varepsilon}^{\mu p}_{n}$, and constant microscale kinematic hardening backstress $\boldsymbol{X}_{n+1} = \boldsymbol{X}_{n}$. We start with the microscale total strains (equation 4.23):

$$\boldsymbol{\varepsilon}^{\mu}_{elpr} = \frac{1}{1-b\,D_n}\left[\boldsymbol{\varepsilon}_{n+1} + \frac{(a-b)\,D_n}{3\,(1-a\,D_n)} tr\left(\boldsymbol{\varepsilon}_{n+1}\right)\boldsymbol{I} + b\,(1-D_n)\,\boldsymbol{\varepsilon}^{\mu p}_{n}\right] - \frac{a\,D_n\,\alpha_{n+1}}{1-a\,D_n}\left(T_{n+1}-T_{ref}\right)\boldsymbol{I} \tag{4.44}$$

By rearranging equation 4.18 and removing the thermal strains, we get the elastic prediction's microscale elastic strains:

$$\boldsymbol{\varepsilon}^{\mu e}_{elpr} = \boldsymbol{\varepsilon}^{\mu}_{elpr} - \boldsymbol{\varepsilon}^{\mu p}_{n} - \alpha_{n+1}\left(T_{n+1}-T_{ref}\right)\boldsymbol{I} \tag{4.45}$$

We can now calculate the elastic prediction's microscale effective stress:

$$\tilde{\boldsymbol{\sigma}}^{\mu}_{elpr} = \boldsymbol{E}_{n+1} : \boldsymbol{\varepsilon}^{\mu e}_{elpr} \tag{4.46}$$

To determine yielding we discretize equation 4.25:

$$f^{\mu}_{n+1} = \left(\tilde{\boldsymbol{\sigma}}^{\mu}_{elpr} - \boldsymbol{X}^{\mu}_{n}\right)_{eq} - \sigma^{\infty}_{f_{n+1}} \tag{4.47}$$

If $f^{\mu}_{n+1} \leq 0$, then the material does not yield and the variables for the next time step can be updated using the elastic prediction variables:

$$\begin{aligned}
\boldsymbol{\varepsilon}^{\mu}_{n+1} &= \boldsymbol{\varepsilon}^{\mu}_{elpr} \\
\boldsymbol{\varepsilon}^{\mu e}_{n+1} &= \boldsymbol{\varepsilon}^{\mu e}_{elpr} \\
\boldsymbol{\varepsilon}^{\mu p}_{n+1} &= \boldsymbol{\varepsilon}^{\mu p}_{n} \\
\boldsymbol{X}^{\mu}_{n+1} &= \boldsymbol{X}^{\mu}_{n} \\
\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} &= \tilde{\boldsymbol{\sigma}}^{\mu}_{elpr} \\
\left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1}\right)_{eq} &= \left(\tilde{\boldsymbol{\sigma}}^{\mu}_{elpr}\right)_{eq} \\
D_{n+1} &= D_n \\
p^{\mu}_{n+1} &= p^{\mu}_{n} \\
\Delta p^{\mu}_{n+1} &= \Delta p^{\mu}_{n}
\end{aligned} \tag{4.48}$$

4.3.4.2 Plastic Correction

If $f^{\mu}_{n+1} > 0$, then the material yields and the elastic prediction is corrected by ensuring the consistency condition (Desmorat et al. (2007), p.918). This is performed using an implicit Euler backward scheme. We should also note that in HCF, "the maximum damage increment per cycle is of the order of magnitude of $D_c/N_R < 1/N_R$" where N_R is the total number of cycles (Desmorat et al. (2007), p.918). Thus, we are justified in making the damage variable, D, be constant over an increment. This helps us eliminate the need for a computationally intensive Newton iterative method (Bhamare et al. (2014), p.393).

The plasticity with damage equations discretized using the Euler backward scheme are (Desmorat et al. (2007), p.918):

$$\boldsymbol{\varepsilon}^{\mu}_{n+1} = \boldsymbol{\varepsilon}^{\mu e}_{n+1} + \boldsymbol{\varepsilon}^{\mu p}_{n+1} \tag{4.49}$$

$$\boldsymbol{\varepsilon}^{\mu e}_{n+1} = \frac{1+\nu}{E_{n+1}}\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} - \frac{\nu}{E_{n+1}} tr\left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1}\right)\boldsymbol{I} + \alpha^{\mu}_{n+1}\left(T_{n+1} - T_{ref}\right)\boldsymbol{I} \tag{4.50}$$

$$\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} = \frac{3}{2}\,\frac{\left(\tilde{\boldsymbol{\sigma}}^{\mu D}_{n+1} - \boldsymbol{X}^{\mu}_{n+1}\right)}{\left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} - \boldsymbol{X}^{\mu}_{n+1}\right)_{eq}}\,\Delta p^{\mu}_{n+1} \tag{4.51}$$

$$\frac{\boldsymbol{X}^{\mu}_{n+1}}{C_{y_{n+1}}} - \frac{\boldsymbol{X}^{\mu}_{n}}{C_{y_n}} = \frac{2}{3}\,\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1}\,(1 - D_n) \tag{4.52}$$

$$\Delta D = \left(\frac{Y^{\mu}_{n+1}}{S_{n+1}}\right)^{s_{n+1}} \Delta p^{\mu} \tag{4.53}$$

Writing the variables in equations 4.49 and 4.23 in incremental form (again, subtracting the thermal strains):

$$\begin{aligned}
\Delta\boldsymbol{\varepsilon}^{\mu}_{n+1} &= \Delta\boldsymbol{\varepsilon}^{\mu e}_{n+1} + \Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} \\
&= \frac{1}{1 - b\,D_n}\left[\Delta\boldsymbol{\varepsilon}_{n+1} + \frac{(a-b)\,D_n}{3\,(1 - a\,D_n)}\,tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{I} + b\,(1 - D_n)\,\Delta\boldsymbol{\varepsilon}^{\mu p}_{n}\right] \\
&- \frac{a\,D_n\,\alpha_{n+1}}{1 - a\,D_n}\,\Delta T_{n+1}\,\boldsymbol{I} - \alpha_{n+1}\Delta T_{n+1}\,\boldsymbol{I}
\end{aligned} \tag{4.54}$$

Multiplying $\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1}$ in the first RHS above by $\frac{1-b\,D_n}{1-b\,D_n}$, and using the two RHS's above, we move everything to one side:

$$\begin{aligned}
\mathbf{0} &= \Delta\boldsymbol{\varepsilon}^{\mu e}_{n+1} + \frac{1 - b\,D_n}{1 - b\,D_n}\,\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} - \frac{b - b\,D_n}{1 - b\,D_n}\,\Delta\boldsymbol{\varepsilon}^{\mu p}_{n} - \frac{1}{1 - b\,D_n}\,\Delta\boldsymbol{\varepsilon}_{n+1} \\
&- \frac{(a-b)\,D_n}{3\,(1 - a\,D_n)\,(1 - b\,D_n)}\,tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{I} + \frac{a\,D_n\,\alpha_{n+1}}{1 - a\,D_n}\,\Delta T_{n+1}\,\boldsymbol{I} + \alpha_{n+1}\Delta T_{n+1}\,\boldsymbol{I}
\end{aligned} \tag{4.55}$$

Combining the $\Delta\boldsymbol{\varepsilon}^{\mu p}$ terms and multiplying by the constitutive elasticity tensor, $\boldsymbol{E}$:

$$\begin{aligned}
\mathbf{0} &= \boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}^{\mu e}_{n+1} + \frac{1 - b}{1 - b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} - \frac{1}{1 - b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} \\
&- \frac{(a-b)\,D_n}{3\,(1 - a\,D_n)\,(1 - b\,D_n)}\,tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{E}_{n+1} : \boldsymbol{I} \\
&+ \frac{a\,D_n\,\alpha_{n+1}}{1 - a\,D_n}\,\Delta T_{n+1}\boldsymbol{E}_{n+1} : \boldsymbol{I} + \alpha_{n+1}\Delta T_{n+1}\boldsymbol{E}_{n+1} : \boldsymbol{I}
\end{aligned} \tag{4.56}$$

Noting that $\boldsymbol{E} : \boldsymbol{I} = 3\,K\,\boldsymbol{I}$:

$$\begin{aligned}
\mathbf{0} &= \boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}^{\mu e}_{n+1} + \frac{1 - b}{1 - b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} - \frac{1}{1 - b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} \\
&- K_{n+1}\frac{(a-b)\,D_n}{(1 - a\,D_n)\,(1 - b\,D_n)}\,tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{I} + \frac{3\,a\,D_n\,\alpha_{n+1}}{1 - a\,D_n}\,\Delta T_{n+1}K_{n+1}\boldsymbol{I} \\
&+ 3\,\alpha_{n+1}\Delta T_{n+1}K_{n+1}\boldsymbol{I}
\end{aligned} \tag{4.57}$$

Combining the temperature terms:

$$\begin{aligned}\mathbf{0} = \boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}^{\mu e}_{n+1} + \frac{1-b}{1-b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} - \frac{1}{1-b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} \\ - K_{n+1}\frac{(a-b)\,D_n}{(1-a\,D_n)\,(1-b\,D_n)}\,tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{I} + \frac{3\,K_{n+1}\alpha_{n+1}}{1-a\,D_n}\,\Delta T_{n+1}\boldsymbol{I}\end{aligned} \tag{4.58}$$

We use the following:

$$\boldsymbol{E}_{n+1} = 2\,\mu_{n+1}\boldsymbol{P}^d + 3\,K_{n+1}\boldsymbol{P}^s \tag{4.59}$$

$$\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}^{\mu e}_{n+1} = \Delta\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \tag{4.60}$$

where $\boldsymbol{P}^d$ is the deviatoric projection tensor and $\boldsymbol{P}^s$ is the spherical projection tensor. These tensors have the following properties:

$$\boldsymbol{I}^4 = \boldsymbol{P}^s + \boldsymbol{P}^d \tag{4.61}$$

$$\boldsymbol{P}^d : \boldsymbol{P}^d = \boldsymbol{P}^d \tag{4.62}$$

$$\boldsymbol{P}^s : \boldsymbol{P}^s = \boldsymbol{P}^s \tag{4.63}$$

where $\boldsymbol{I}^4$ is the fourth-order identity tensor ($I^4_{ijkl} = \frac{1}{2}\left[\delta_{ik}\,\delta_{jl} + \delta_{il}\,\delta_{jk}\right]$). Substituting equations 4.59 and 4.60 into equation 4.58:

$$\begin{aligned}\mathbf{0} = \Delta\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} + \frac{1-b}{1-b\,D_n}\left(2\,\mu_{n+1}\boldsymbol{P}^d : \Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} + 3\,K_{n+1}\boldsymbol{P}^s : \Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1}\right) \\ - \frac{1}{1-b\,D_n}\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} - K_{n+1}\frac{(a-b)\,D_n}{(1-a\,D_n)\,(1-b\,D_n)}tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{I} \\ + \frac{3\,K_{n+1}\alpha_{n+1}}{1-a\,D_n}\Delta T_{n+1}\boldsymbol{I}\end{aligned} \tag{4.64}$$

Since $\boldsymbol{P}^s : \Delta\boldsymbol{\varepsilon}^{\mu p} = 0$ and $\boldsymbol{P}^d : \Delta\boldsymbol{\varepsilon}^{\mu p} = \Delta\boldsymbol{\varepsilon}^{\mu p}$:

$$\begin{aligned}\mathbf{0} = \Delta\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} + \frac{1-b}{1-b\,D_n}\,2\,\mu_{n+1}\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} - \frac{1}{1-b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} \\ - K_{n+1}\frac{(a-b)\,D_n}{(1-a\,D_n)\,(1-b\,D_n)}\,tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{I} + \frac{3\,K_{n+1}\alpha_{n+1}}{1-a\,D_n}\,\Delta T_{n+1}\boldsymbol{I}\end{aligned} \tag{4.65}$$

Using $\Delta\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} = \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} - \tilde{\boldsymbol{\sigma}}^{\mu}_{n}$ and adding $\boldsymbol{X}^{\mu}_{n+1} - \boldsymbol{X}^{\mu}_{n+1}$, we have:

$$\begin{aligned}\mathbf{0} = \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} - \tilde{\boldsymbol{\sigma}}^{\mu}_{n} + \boldsymbol{X}^{\mu}_{n+1} - \boldsymbol{X}^{\mu}_{n+1} + \frac{1-b}{1-b\,D_n}\,2\mu_{n+1}\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} - \frac{1}{1-b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} \\ - K_{n+1}\frac{(a-b)\,D_n}{(1-a\,D_n)\,(1-b\,D_n)}\,tr\,(\Delta\boldsymbol{\varepsilon}_{n+1})\,\boldsymbol{I} + \frac{3K_{n+1}\alpha_{n+1}}{1-a\,D_n}\,\Delta T_{n+1}\boldsymbol{I}\end{aligned} \tag{4.66}$$

Rearranging equation 4.52:

$$\boldsymbol{X}^{\mu}_{n+1} = \frac{C_{y_{n+1}}}{C_{y_n}}\,\boldsymbol{X}^{\mu}_{n} + \frac{2}{3}\,C_{y_{n+1}}\,(1-D_n)\,\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} \tag{4.67}$$

Substituting $\boldsymbol{s}^{\mu}_{n+1} = \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} - \boldsymbol{X}^{\mu}_{n+1}$ and equation 4.67 into equation 4.66:

$$\begin{aligned}\boldsymbol{0} = {} & \boldsymbol{s}^{\mu}_{n+1} - \tilde{\boldsymbol{\sigma}}^{\mu}_{n} + \frac{C_{y_{n+1}}}{C_{y_n}}\,\boldsymbol{X}^{\mu}_{n} + \frac{2}{3}\,C_{y_{n+1}}\,(1-D_n)\,\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} \\ & + \frac{1-b}{1-b\,D_n}\,2\,\mu_{n+1}\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} - \frac{1}{1-b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} \\ & - K_{n+1}\frac{(a-b)\,D_n}{(1-a\,D_n)\,(1-b\,D_n)}\,tr\left(\Delta\boldsymbol{\varepsilon}_{n+1}\right)\boldsymbol{I} + \frac{3\,K_{n+1}\alpha_{n+1}}{1-a\,D_n}\,\Delta T_{n+1}\boldsymbol{I}\end{aligned} \tag{4.68}$$

Grouping the $\boldsymbol{\varepsilon}^{\mu p}_{n+1}$ terms:

$$\begin{aligned}\boldsymbol{0} = {} & \boldsymbol{s}^{\mu}_{n+1} - \tilde{\boldsymbol{\sigma}}^{\mu}_{n} + \frac{C_{y_{n+1}}}{C_{y_n}}\boldsymbol{X}^{\mu}_{n} \\ & + \left[\frac{2}{3}C_{y_{n+1}}\,(1-D_n) + \frac{1-b}{1-b\,D_n}\,2\,\mu_{n+1}\right]\Delta\boldsymbol{\varepsilon}^{\mu p}_{n+1} \\ & - \frac{1}{1-b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} - K_{n+1}\frac{(a-b)\,D_n}{(1-a\,D_n)\,(1-b\,D_n)}\,tr\left(\Delta\boldsymbol{\varepsilon}_{n+1}\right)\boldsymbol{I} \\ & + \frac{3\,K_{n+1}\alpha_{n+1}}{1-a\,D_n}\,\Delta T_{n+1}\boldsymbol{I}\end{aligned} \tag{4.69}$$

Using $\tilde{\boldsymbol{\sigma}}^{\mu D}_{n+1} - \boldsymbol{X}^{\mu}_{n+1} = \boldsymbol{s}^{\mu D}_{n+1}$ and $\left(\tilde{\boldsymbol{\sigma}}^{\mu D}_{n+1} - \boldsymbol{X}^{\mu}_{n+1}\right)_{eq} = \left(\boldsymbol{s}^{\mu}_{n+1}\right)_{eq}$ in conjunction with equation 4.51:

$$\begin{aligned}\boldsymbol{0} = {} & \boldsymbol{s}^{\mu}_{n+1} - \tilde{\boldsymbol{\sigma}}^{\mu}_{n} + \frac{C_{y_{n+1}}}{C_{y_n}}\boldsymbol{X}^{\mu}_{n} \\ & + \left[C_{y_{n+1}}\,(1-D_n) + \frac{1-b}{1-b\,D_n}\,3\,\mu_{n+1}\right]\frac{\boldsymbol{s}^{\mu D}_{n+1}}{\left(\boldsymbol{s}^{\mu}_{n+1}\right)_{eq}}\,\Delta p^{\mu}_{n+1} \\ & - \frac{1}{1-b\,D_n}\,\boldsymbol{E}_{n+1} : \Delta\boldsymbol{\varepsilon}_{n+1} - K_{n+1}\frac{(a-b)\,D_n}{(1-a\,D_n)\,(1-b\,D_n)}\,tr\left(\Delta\boldsymbol{\varepsilon}_{n+1}\right)\boldsymbol{I} \\ & + \frac{3\,K_{n+1}\alpha_{n+1}}{1-a\,D_n}\,\Delta T_{n+1}\boldsymbol{I}\end{aligned} \tag{4.70}$$

We define the following:

$$\Gamma = C_{y_{n+1}}\,(1-D_n) + \frac{1-b}{1-b\,D_n}\,3\,\mu_{n+1} \tag{4.71}$$

$$
\begin{aligned}
\boldsymbol{Q}_s = & \frac{C_{y_{n+1}}}{C_{y_n}} \boldsymbol{X}_n^\mu - \tilde{\boldsymbol{\sigma}}_n^\mu \\
& - \frac{1}{1 - b\, D_n} \left[\boldsymbol{E}_{n+1} : \Delta \boldsymbol{\varepsilon}_{n+1} - K_{n+1} \frac{(a-b)\, D_n}{(1 - a\, D_n)}\, tr\left(\Delta \boldsymbol{\varepsilon}_{n+1}\right) \boldsymbol{I} \right] \\
& + \frac{3\, K_{n+1} \alpha_{n+1}}{1 - a\, D_n}\, \Delta T_{n+1} \boldsymbol{I}
\end{aligned} \tag{4.72}
$$

Substituting equations 4.71 and 4.72 into equation 4.70, we get:

$$
\mathbf{0} = \boldsymbol{s}_{n+1}^\mu + \Gamma \frac{\boldsymbol{s}_{n+1}^{\mu D}}{\left(\boldsymbol{s}_{n+1}^\mu\right)_{eq}}\, \Delta p_{n+1}^\mu + \boldsymbol{Q}_s \tag{4.73}
$$

Using $\boldsymbol{s}_{n+1}^\mu = \tilde{\boldsymbol{\sigma}}_{n+1}^\mu - \boldsymbol{X}_{n+1}^\mu$ and substituting it into equation 4.47, the yield function becomes:

$$
f_{n+1}^\mu = \left(\boldsymbol{s}_{n+1}^\mu\right)_{eq} - \sigma_{f_{n+1}}^\infty \tag{4.74}
$$

In order to find the unknowns, $\boldsymbol{s}_{n+1}^\mu$ and Δp_{n+1}^μ, we must solve equations 4.73 and 4.74 simultaneously. First, we multiply equation 4.73 by the spherical projection tensor, $\boldsymbol{P}^s$, to get the hydrostatic part:

$$
\mathbf{0} = \boldsymbol{P}^s : \boldsymbol{s}_{n+1}^\mu + \Gamma \frac{\boldsymbol{P}^s : \boldsymbol{s}_{n+1}^{\mu D}}{\left(\boldsymbol{s}_{n+1}^\mu\right)_{eq}}\, \Delta p_{n+1}^\mu + \boldsymbol{P}^s : \boldsymbol{Q}_s \tag{4.75}
$$

Using $\boldsymbol{P}^s : \boldsymbol{s}_{n+1}^{\mu D} = \mathbf{0}$, $\boldsymbol{P}^s : \boldsymbol{s}_{n+1}^\mu = s_{n+1}^{\mu H} : \boldsymbol{I}$, and $\boldsymbol{P}^s : \boldsymbol{Q}_s = Q_s^H : \boldsymbol{I}$, we have:

$$
s_{n+1}^{\mu H} = -Q_s^H \tag{4.76}
$$

Multiplying equation 4.73 by the deviatoric projection tensor, $\boldsymbol{P}^d$, to get the deviatoric part:

$$
\mathbf{0} = \boldsymbol{P}^d : \boldsymbol{s}_{n+1}^\mu + \Gamma \frac{\boldsymbol{P}^d : \boldsymbol{s}_{n+1}^{\mu D}}{\left(\boldsymbol{s}_{n+1}^\mu\right)_{eq}}\, \Delta p_{n+1}^\mu + \boldsymbol{P}^d : \boldsymbol{Q}_s \tag{4.77}
$$

Using $\boldsymbol{P}^d : \boldsymbol{s}_{n+1}^{\mu D} = \boldsymbol{s}_{n+1}^{\mu D}$, $\boldsymbol{P}^d : \boldsymbol{s}_{n+1}^\mu = \boldsymbol{s}_{n+1}^{\mu D}$, and $\boldsymbol{P}^d : \boldsymbol{Q}_s = \boldsymbol{Q}_s^D$, we have:

$$
\mathbf{0} = \boldsymbol{s}_{n+1}^{\mu D} + \Gamma \frac{\boldsymbol{s}_{n+1}^{\mu D}}{\left(\boldsymbol{s}_{n+1}^\mu\right)_{eq}}\, \Delta p_{n+1}^\mu + \boldsymbol{Q}_s^D \tag{4.78}
$$

Moving $\boldsymbol{Q}_s^D$ to the LHS:

$$
-\boldsymbol{Q}_s^D = \boldsymbol{s}_{n+1}^{\mu D} \left(1 + \Gamma \frac{\Delta p_{n+1}^\mu}{\left(\boldsymbol{s}_{n+1}^\mu\right)_{eq}} \right) \tag{4.79}
$$

Taking the von Mises norm of both sides:

$$\begin{aligned}\sqrt{\frac{3}{2}\,\boldsymbol{s}^{\mu D}_{n+1} : \boldsymbol{s}^{\mu D}_{n+1}}\left(1+\Gamma\,\frac{\Delta p^{\mu}_{n+1}}{\left(\boldsymbol{s}^{\mu}_{n+1}\right)_{eq}}\right) &= \sqrt{\frac{3}{2}\,\boldsymbol{Q}^{D}_{s} : \boldsymbol{Q}^{D}_{s}} \\ = \left(\boldsymbol{s}^{\mu}_{n+1}\right)_{eq}\left(1+\Gamma\,\frac{\Delta p^{\mu}_{n+1}}{\left(\boldsymbol{s}^{\mu}_{n+1}\right)_{eq}}\right) &= \left(\boldsymbol{Q}_{s}\right)_{eq}\end{aligned} \tag{4.80}$$

Solving for Δp^{μ}_{n+1}, we get:

$$\Delta p^{\mu}_{n+1} = \frac{1}{\Gamma}\left[\left(\boldsymbol{Q}_{s}\right)_{eq} - \left(\boldsymbol{s}^{\mu}_{n+1}\right)_{eq}\right] \tag{4.81}$$

Substituting equation 4.74 into equation 4.81:

$$\Delta p^{\mu}_{n+1} = \frac{1}{\Gamma}\left[\left(\boldsymbol{Q}_{s}\right)_{eq} - \sigma^{\infty}_{f_{n+1}}\right] \tag{4.82}$$

Solving 4.79 for $\boldsymbol{s}^{\mu D}_{n+1}$ and combining with equation 4.74:

$$\boldsymbol{s}^{\mu D}_{n+1} = -\frac{\boldsymbol{Q}^{D}_{s}}{\left(1+\Gamma\,\frac{\Delta p^{\mu}_{n+1}}{\sigma^{\infty}_{f_{n+1}}}\right)} \tag{4.83}$$

Thus $\boldsymbol{s}^{\mu}_{n+1}$ is:

$$\boldsymbol{s}^{\mu}_{n+1} = s^{\mu H}_{n+1}\boldsymbol{I} + \boldsymbol{s}^{\mu D}_{n+1} \tag{4.84}$$

The microscale effective stress is:

$$\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} = \boldsymbol{s}^{\mu}_{n+1} + \boldsymbol{X}^{\mu}_{n+1} \tag{4.85}$$

We can calculate the vector normal to the yield surface as:

$$\boldsymbol{m}^{\mu} = \frac{3}{2}\,\frac{\boldsymbol{s}^{\mu D}_{n+1}}{\sigma^{\infty}_{f\,n+1}} \tag{4.86}$$

With the yield surface normal, the new microscale plastic strains can be updated:

$$\boldsymbol{\varepsilon}^{\mu p}_{n+1} = \boldsymbol{\varepsilon}^{\mu p}_{n} + \boldsymbol{m}^{\mu}\,\Delta p^{\mu}_{n+1} \tag{4.87}$$

Microscale elastic strains and total strains can be calculated using:

$$\boldsymbol{\varepsilon}^{\mu e}_{n+1} = \boldsymbol{E}^{-1} : \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \tag{4.88}$$

$$\boldsymbol{\varepsilon}^{\mu}_{n+1} = \boldsymbol{\varepsilon}^{\mu e}_{n+1} + \boldsymbol{\varepsilon}^{\mu p}_{n+1} + \alpha_{n+1}\left(T_{n+1} - T_{ref}\right)\boldsymbol{I} \tag{4.89}$$

4.4 Two-Scale Model Implementation

To summarize, the implementation of the two-scale progressive fatigue damage model is as follows:

1. Import the mesoscale strains, $\boldsymbol{\varepsilon}_{n+1}$, calculated from the finite element analysis into the damage subroutine.

2. Calculate the elastic prediction microscale total strains, $\boldsymbol{\varepsilon}^{\mu}_{elpr}$, using equation 4.44, the elastic prediction microscale elastic strains, $\boldsymbol{\varepsilon}^{\mu e}_{elpr}$, using equation 4.45, and then the elastic prediction microscale effective stress, $\tilde{\boldsymbol{\sigma}}^{\mu}_{elpr}$, using equation 4.46.

3. Determine if there is microscale plastic yielding or not by calculating f^{μ}_{n+1} using equation 4.47.

   (a) If $f^{\mu}_{n+1} \leq 0$, then the material does not yield and the variables are updated as follows (equation 4.48):

   $$\begin{aligned} \boldsymbol{\varepsilon}^{\mu}_{n+1} &= \boldsymbol{\varepsilon}^{\mu}_{elpr} \\ \boldsymbol{\varepsilon}^{\mu e}_{n+1} &= \boldsymbol{\varepsilon}^{\mu e}_{elpr} \\ \boldsymbol{\varepsilon}^{\mu p}_{n+1} &= \boldsymbol{\varepsilon}^{\mu p}_{n} \\ \boldsymbol{X}^{\mu}_{n+1} &= \boldsymbol{X}^{\mu}_{n} \\ \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} &= \tilde{\boldsymbol{\sigma}}^{\mu}_{elpr} \\ \left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1}\right)_{eq} &= \left(\tilde{\boldsymbol{\sigma}}^{\mu}_{elpr}\right)_{eq} \\ D_{n+1} &= D_{n} \\ p^{\mu}_{n+1} &= p^{\mu}_{n} \\ \Delta p^{\mu}_{n+1} &= \Delta p^{\mu}_{n} \end{aligned}$$

   (b) If $f^{\mu}_{n+1} > 0$, then the material yields at the microscale and the elastic prediction is corrected by ensuring the consistency condition using an implicit Euler backward scheme:

i. Calculate $\boldsymbol{Q}_s$ using equation 4.72, and from $\boldsymbol{Q}_s$, calculate $\boldsymbol{Q}_s^D$ and $\boldsymbol{Q}_s^H$.

ii. Calculate $(\boldsymbol{Q}_s)_{eq}$ using the von Mises norm.

iii. Calculate Γ using equation 4.71.

iv. Calculate the new change in accumulated plastic strain, Δp^{μ}_{n+1}, using equation 4.82.

v. Update the total accumulated plastic strain using:

$$p^{\mu}_{n+1} = p^{\mu}_n + \Delta p^{\mu}_{n+1} \tag{4.90}$$

vi. Calculate $s^{\mu H}_{n+1}$ using equation 4.76, $\boldsymbol{s}^{\mu D}_{n+1}$ using equation 4.83, and $\boldsymbol{s}^{\mu}_{n+1}$ using equation 4.84.

vii. Calculate the yield surface normal, $\boldsymbol{m}^{\mu}$, using equation 4.86 and update the microscale plastic strains, $\boldsymbol{\varepsilon}^{\mu p}_{n+1}$, using equation 4.87.

viii. Update the backstress, $\boldsymbol{X}^{\mu}_{n+1}$, using equation 4.67.

ix. Update $\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1}$ using equation 4.85 and $\left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1}\right)_{eq}$ using the von Mises norm.

x. Update the microscale elastic strains, $\boldsymbol{\varepsilon}^{\mu e}_{n+1}$, using equation 4.88 and the microscale total strains, $\boldsymbol{\varepsilon}^{\mu}_{n+1}$, using equation 4.89.

xi. Calculate the stored energy damage threshold, w_D, using 4.13, and the stored energy density, $w_{s_{n+1}}$, can be updated using an incremental version of equation 4.11 with a simple numerical integration scheme:

$$\sigma^{\mu}_{new} = \left|\left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1}\right)_{eq} - \sigma^{\infty}_{f_{n+1}}\right| \tag{4.91}$$

$$\sigma^{\mu}_{old} = \left|(\tilde{\boldsymbol{\sigma}}^{\mu}_{n})_{eq} - \sigma^{\infty}_{f_n}\right| \tag{4.92}$$

$$w_{s_{n+1}} = w_{s_n} + \frac{1}{2}\left(\sigma^{\mu}_{new}\,\Delta p^{\mu}_{n+1} + \sigma^{\mu}_{old}\,\Delta p^{\mu}_{n}\right) \tag{4.93}$$

where the vertical bars in equations 4.91 and 4.92 are absolute value operators.

A. If $w_{s_{n+1}} < w_D$, then damage does not accumulate: $D_{n+1} = D_n$.

B. If $w_{s_{n+1}} \geq w_D$, then damage accumulates:

- Calculate the damage energy release rate, Y^{μ}_{n+1}, using an incremental version of equation 4.43:

$$\begin{aligned} Y^{\mu}_{n+1} = \frac{1+\nu_{n+1}}{2\,E_{n+1}} & \left[\left\langle \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \right\rangle^{+} : \left\langle \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \right\rangle^{+} + h \left(\frac{1-D_n}{1-h\,D_n} \right)^2 \left\langle \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \right\rangle^{-} : \left\langle \tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \right\rangle^{-} \right] \\ - \frac{\nu_{n+1}}{2\,E_{n+1}} & \left[\left\langle tr \left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \right) \right\rangle^2 + h \left(\frac{1-D_n}{1-h\,D_n} \right)^2 \left\langle -tr \left(\tilde{\boldsymbol{\sigma}}^{\mu}_{n+1} \right) \right\rangle^2 \right] \end{aligned} \tag{4.94}$$

- Update the damage using an incremental version of equation 4.28:

$$D_{n+1} = D_n + \left(\frac{Y^{\mu}_{n+1}}{S_{n+1}} \right)^{s_{n+1}} \Delta p^{\mu}_{n+1} \tag{4.95}$$

xii. If $D_{n+1} \geq D_c$, then a crack initiates at the mesoscale. In the XTFEM code, this translates to element deletion should any integration point reach the critical damage value.

CHAPTER 5

NUMERICAL EXAMPLES

In order to demonstrate the effectiveness of XTFEM for the direct numerical simulation of thermo-mechanical high cycle fatigue, we will present three cases of increasing complexity. First, we will establish the need for XTFEM by contrasting it with a simple case of a single element under fatigue loading using a finite difference time integration scheme. Next, we will present an elementary, academic example of a prismatic beam fixed at both ends subjected to a uniform fluctuating temperature resulting in thermal fatigue. As a capstone illustration of the XTFEM code's abilities we will present a coupled, thermo-mechanical fatigue simulation of a plate and hat stiffener representative of an aircraft's structure.

5.1 Abaqus UMAT

In order to illustrate the need for XTFEM, it was deemed prudent to simulate the fatigue of structures using a traditional, finite difference time integration algorithm while using the two scale progressive fatigue damage model for fatigue damage calculation (Desmorat et al. (2007)). To do this, an Abaqus' user-material model subroutine or UMAT was utilized and coded with the two scale model.

A single, eight-noded, reduced integration brick element (C3D8R in Abaqus) of 1 mm length on all three sides was used as the model. The material used was AISI 304L stainless steel, and the material properties and parameters specific to the two scale model were taken from Zhang et al. (Zhang et al. (2016), p.343; see Table 5.1). Symmetry boundary conditions were applied to three sides so that the model would be fully constrained while the Poisson's effect would not translate to stress (see Figure 5.1). A mechanical force was applied to one of the other sides in the direction normal to its face with a user-defined amplitude or UAMP. The UAMP had a sinusoidal character except for the first 10 seconds which served as a

Table 5.1. Material Parameters for AISI 304L Steel at Room Temperature

ρ $(\mathrm{kg/m^3})$	E (GPa)	ν	C_y (MPa)	σ_u (MPa)	σ_f^∞ (MPa)	ε_{pD}	h	D_c	S (MPa)	s
7,860	197	0.3	1,740	577	180	0.08	0.2	0.3	0.5	0.5

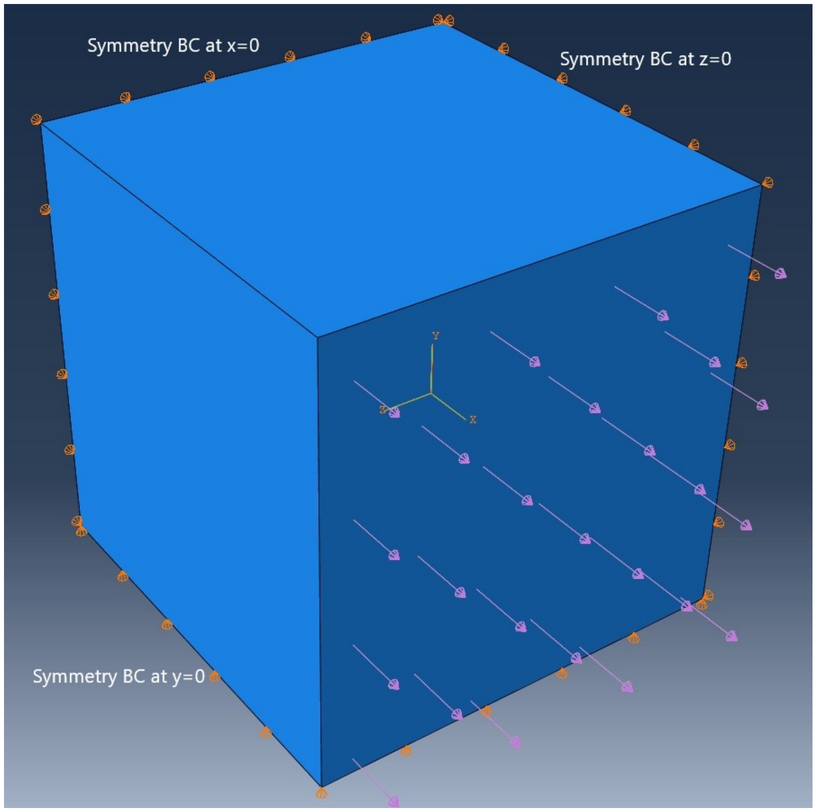

Figure 5.1. Single Element Model

load-ramping period to limit artificial oscillations:

$$A = \begin{cases} \frac{t}{10}\, M\, \sin(\omega\, t) & t < 10 \\ M\, \sin(\omega\, t) & t \geq 10 \end{cases} \tag{5.1}$$

where A is the amplitude, t is the time, M is the load magnitude (specified in the Edit Load dialogue within Abaqus' GUI), and ω is the circular frequency in radians per second. The frequency of the mechanical force was chosen to be 20 Hz making the circular frequency approximately 126 radians/sec. It was decided that these simulations would be isothermal so as to limit the complexity. This simulation was run at different magnitudes and the time to

Table 5.2. Abaqus UMAT and XTFEM Results

Load (MPa)	C++ Code 1,000 points/cycle (cycles)	XTFEM 200 points/cycle (cycles)	Abaqus 20 steps/cycle (cycles)	Abaqus 64 steps/cycle (cycles)	XTFEM 200 points/cycle (user time)	Abaqus 20 steps/cycle (user time)	Abaqus 64 steps/cycle (user time)
300	9,589	9,600	10,260	N/A	5 sec	1.2 hr	N/A
265	18,987	19,000	20,224	N/A	9 sec	7.6 hrs	N/A
235	45,080	45,100	47,482	N/A	21 sec	34.6 hrs	N/A
230	54,853	54,900	64,453	57,392 (projected)	27 sec	65.0 hrs	222 hrs (projected)
220	84,823	84,900	111,284	90,227 (projected)	39 sec	197.5 hrs	416 hrs (projected)

element deletion was recorded. In order to provide a contrast with XTFEM, the same model was run using the XTFEM code with time steps equal to 100 cycles, and the time to element deletion was recorded. In addition, a standalone C++ code which takes stress amplitudes as inputs (i.e., no finite element calculations) and calculates fatigue life using the two scale model with 1,000 interpolation points per cycle was also run as a control. Both Abaqus and the XTFEM code were run on a single processor. Table 5.2 provides the results of all codes.

The first observation from examining Table 5.2 is that the user time to run the XTFEM code was several orders of magnitude less than the standard, implicit time integration in Abaqus. The second observation is that the number of interpolation points per cycle is correlated with accuracy (i.e., similarity to the results of the C++ code). The Abaqus simulations with 64 implicit time steps per cycle were closer to the C++ code than the ones with 20 steps per cycle, and the XTFEM simulations with 200 interpolation points per cycle were even closer to the C++ code. This characteristic of the two scale model with sinusoidal loading has been studied by Zhang et al. who noted that, as loading amplitudes approach the asymptotic fatigue limit, more interpolation points are necessary to obtain accuracy (Zhang et al. (2016), p.344).

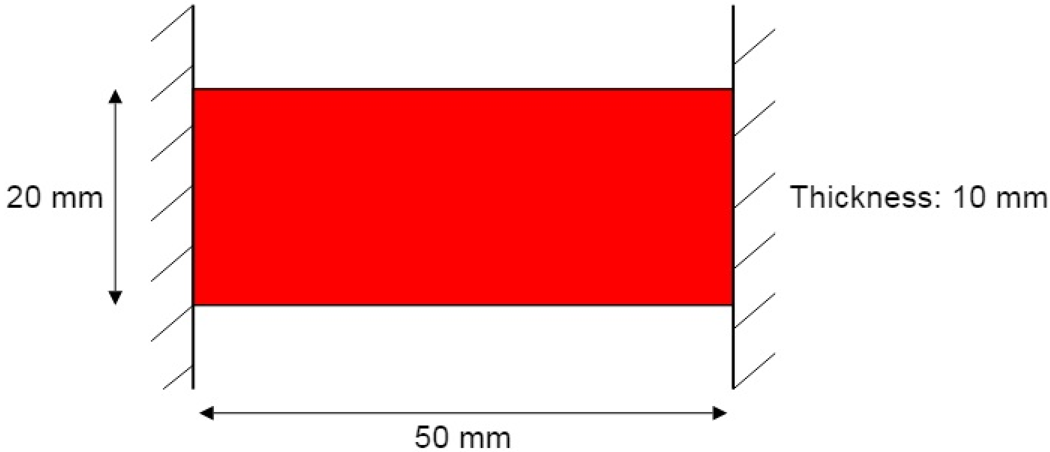

Figure 5.2. Constrained Prismatic Beam Geometry

5.2 Constrained Prismatic Beam

A simple, academic example was chosen to demonstrate the thermo-mechanical abilities of the XTFEM code. The example was a beam with a constant, rectangular cross-section prevented from expanding or contracting along its length and subjected to a spatially uniform but cyclically, time-varying temperature change (see Figure 5.2). The beam was assigned the same material properties and parameters as the previous example (see Table 5.1) with a coefficient of thermal expansion of $\alpha = 1.65\text{E-}5/^{\circ}\text{C}$. The beam was constrained from deforming along its length, and symmetry was utilized in the other two directions. The geometry was meshed with 20-node brick elements with reduced integration (C3D20R in Abaqus) with a total of 300 elements and 1,782 nodes. No mechanical force was used, but a sinusoidal, time-varying temperature was uniformly assigned to all nodes on the beam. In this case, we used an extended space-time temperature vector so that we could make the time steps multiples of

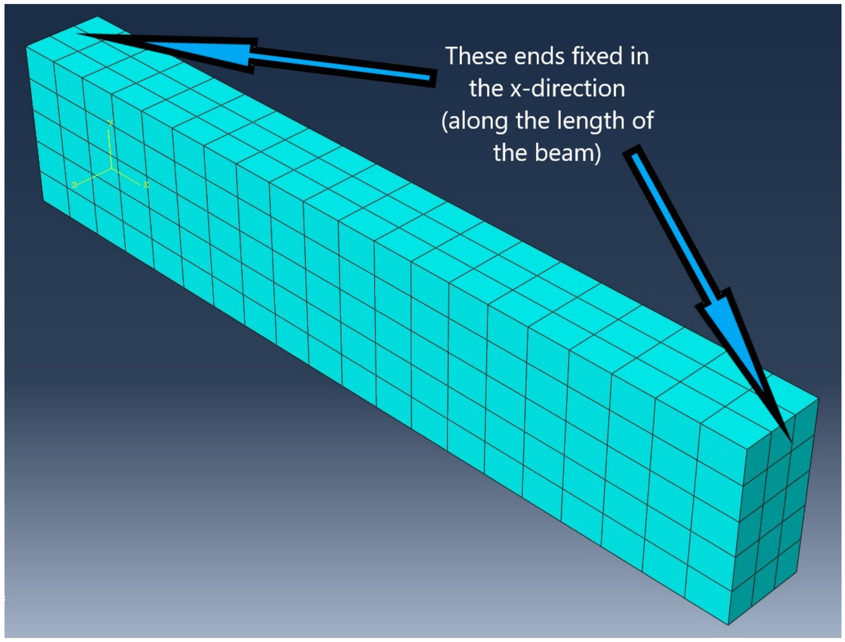

Figure 5.3. Beam Mesh

the temperature's time period:

$$\boldsymbol{\phi}_X = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \tilde{\phi}_1 \\ \tilde{\phi}_2 \\ \tilde{\phi}_3 \end{bmatrix} = \begin{bmatrix} T_{ref} + T_{amp} \sin(\omega\, t_1) \\ T_{ref} + T_{amp} \sin(\omega\, t_2) \\ T_{ref} + T_{amp} \sin(\omega\, t_3) \\ T_{amp} \\ T_{amp} \\ T_{amp} \end{bmatrix} \tag{5.2}$$

where ϕ_i is the spatial nodal temperature DOF at the ith time node, $\tilde{\phi}_i$ is the extended spatial nodal temperature DOF at the ith time node, T_{ref} is the initial or reference temperature, T_{amp} is the temperature amplitude, and $\omega = 2\,\pi\, f$ is the circular frequency. A frequency of $f = 1$ Hz was chosen for the temperature, and the model was simulated at various temperature amplitudes with a mean temperature and an initial temperature of zero. No thermal analysis was conducted as the temperatures were assigned directly.

Table 5.3. Constrained Prismatic Beam Results

Temperature Amplitude ($^{\circ}C$)	Stress (MPa)	C++ Cycles	XTFEM Cycles	Percent Difference
70.8	230	54,853	54,500	-0.6%
69.2	225	67,361	67,500	0.2%
67.7	220	84,823	84,500	-0.4%
66.1	215	111,041	111,800	0.7%
64.6	210	151,797	151,900	0.1%
63.1	205	218,987	217,500	-0.7%
61.5	200	338,378	341,100	0.8%
60.0	195	612,428	610,700	-0.3%
58.5	190	1,357,474	1,332,000	-1.9%
56.9	185	4,972,627	4,970,600 (Projected)	0.0%

As we recall from section 2.4, the thermal "force" term is given by:

$$\boldsymbol{EIN1\Theta} \otimes \boldsymbol{\Theta} = \int_{I_n} \left[\begin{array}{cc} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{array} \right]^T \Delta T\,(t)\; dt \otimes \int_{\Omega} \boldsymbol{B}_x^T\, \boldsymbol{D}\, \alpha\, \Delta T\,(x)\; \boldsymbol{I}\, d\Omega \tag{5.3}$$

Given that the temperature is uniformly distributed spatially and varies only with time, the above expression can be simplified as:

$$\boldsymbol{EIN1\Theta} \otimes \boldsymbol{\Theta} = \int_{I_n} \left[\begin{array}{cc} \dot{\boldsymbol{N}}_t & \dot{\tilde{\boldsymbol{N}}}_t \end{array} \right]^T (T_{amp} \sin{(\omega\, t)})\, dt \otimes \int_{\Omega} \boldsymbol{B}_x^T\, \boldsymbol{D}\, \alpha\, 1\, \boldsymbol{I}\, d\Omega \tag{5.4}$$

The simulations were run on the Texas Advanced Computing Center's (TACC) Lonestar5 high-performance computer using a single compute node with 24 processors. The results of the simulations were compared with those from the C++ code mentioned in the previous example using an equivalent uniaxial stress amplitude given by $\sigma_{amp} = E\, \alpha\, (T_{amp} - T_{ref})$ where T_{ref} in this case is zero. The results are displayed in Table 5.3. The last simulation at a temperature amplitude of 56.9°C had to be projected due to Lonestar5's 48 hour runtime limitation.

From the results displayed in Table 5.3, it is apparent that the XTFEM formulation can accurately translate temperature changes into equivalent "forces." The minor differences in cycles to failure between the C++ code and the XTFEM code were most likely due to the round-off in the calculated temperature values used to create the equivalent stress levels.

5.3 Plate and Hat Stiffener

At hypersonic speeds, aircraft structures experience extreme combined environments over their trajectory. By traveling at speeds in excess of Mach 5 (Sziroczak and Smith (2016), p.4), velocity is converted "into heat by viscous forces within the boundary layer that surrounds the vehicle" (Ho (2010), p.55). Temperatures on atmospheric, hypersonic flight vehicles can build up to several thousands of degrees Fahrenheit (Blevins et al. (1993), p.971). The intense heat creates large temperature gradients between the aircraft's skin and its inner structure causing large stresses. Complicating matters, the structural material is more flexible (i.e., lower elastic modulus) at higher temperatures which makes it more susceptible to oscillations (Ho (2010), p.54). Repeated stresses from combined (random) acoustic and thermal loads over the lifetime of the aircraft cause fatigue damage in the material (Ho (2010), p.55).

A physical experiment was conducted by the Aerospace Systems Directorate of the Air Force Research Laboratory (AFRL) in which a plate and hat stiffeners made of the titanium alloy Ti-6242S and representative of an aircraft's structure were subjected to a combined thermal and random mechanical loading eventually leading to crack formation (see Figure 5.4). This study, while highly simplified, sought to reproduce the results of the experiment.

5.3.1 The Computer Model

The Abaqus model of the experimental apparatus originally consisted of 4 hat stiffeners connected to a plate by means of spot welds 5×10^{-4} inch thick (see Figure 5.5). Given that the purpose of this study was a fatigue simulation, a decision was made to further simplify the model in order to decrease simulation time. So, the model was limited to one-quarter of a hat stiffener and corresponding plate section with symmetry boundary conditions applied to the

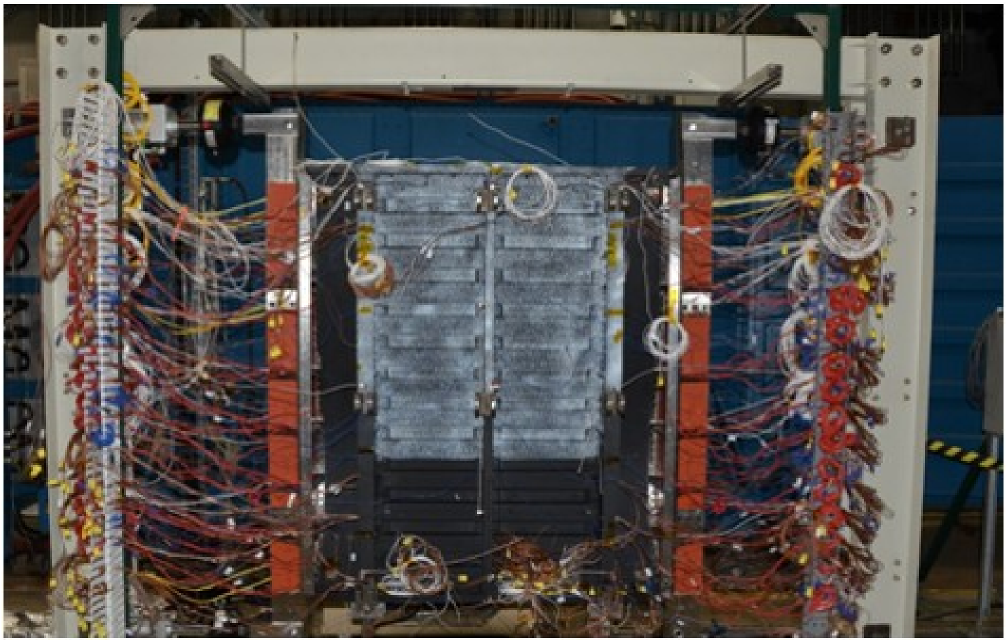

Figure 5.4. AFRL Experimental Apparatus (Picture from Case #: 88ABW-2018-1008)

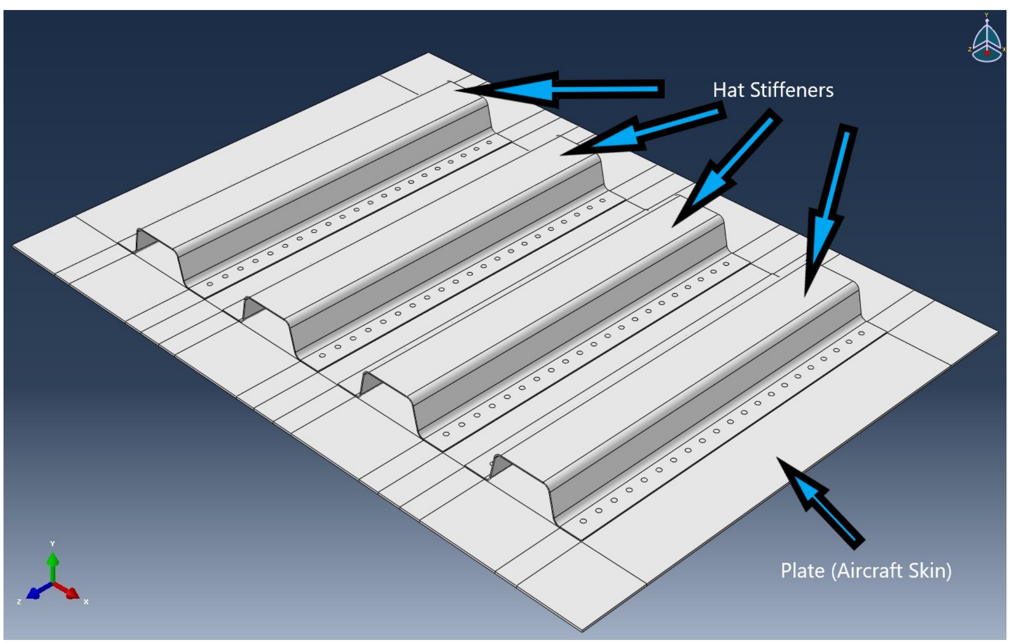

Figure 5.5. Original Abaqus Model Created by the AFRL

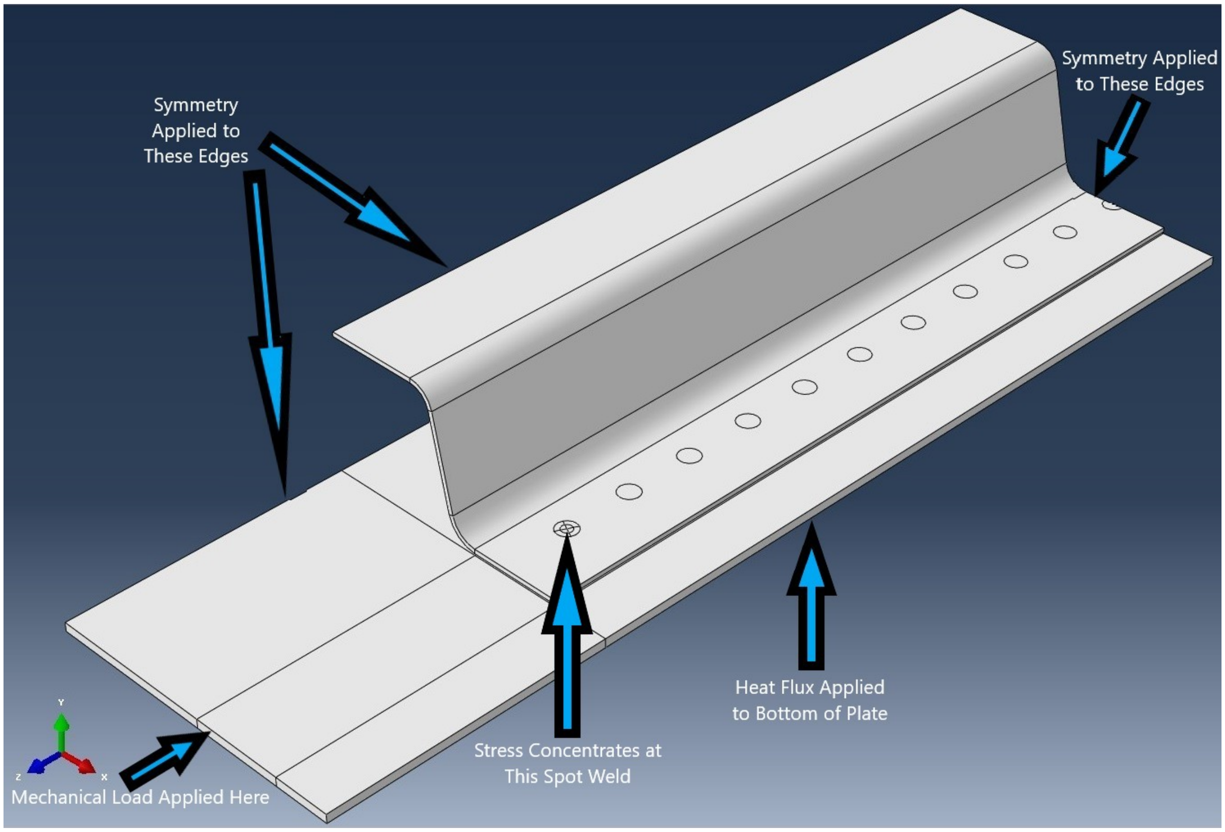

Figure 5.6. Quarter Plate and Hat Stiffener

sides where the hat was cut (see Figure 5.6). Furthermore, the spot welds were exaggerated in thickness and given fillets so as to make meshing and post-processing easier (see Figure 5.7). Lastly, it was determined from a preliminary, static analysis in Abaqus that the stress would concentrate in the first spot weld from the end of the hat (opposite the cut). So, that area was further partitioned for meshing (see Figure 5.8). Since the XTFEM code does not have the capacity to model components consisting of multiple materials at this time, the welds were modeled with the same material as the plate and hat stiffener, Ti-6242S.

5.3.2 The Mesh

Due to the highly curved geometry of the weld fillets (see Figure 5.7), the component was meshed with twenty node brick elements with reduced integration (C3D20R in Abaqus).

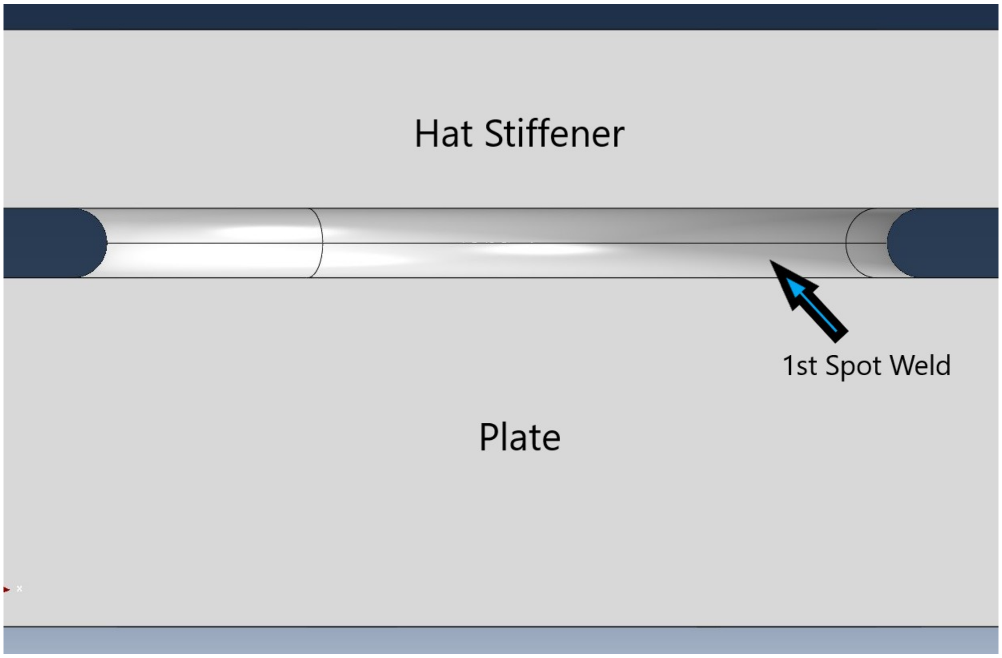

Figure 5.7. Thickened Spot Welds

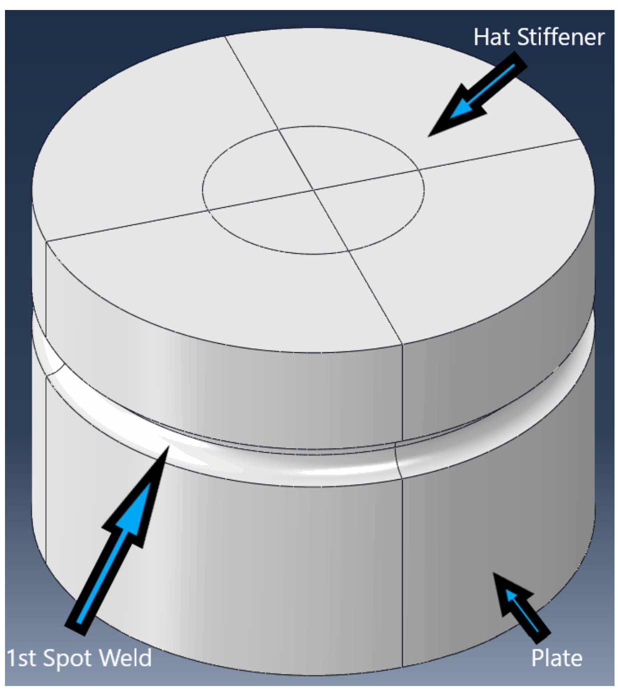

Figure 5.8. 1st Spot Weld Vicinity

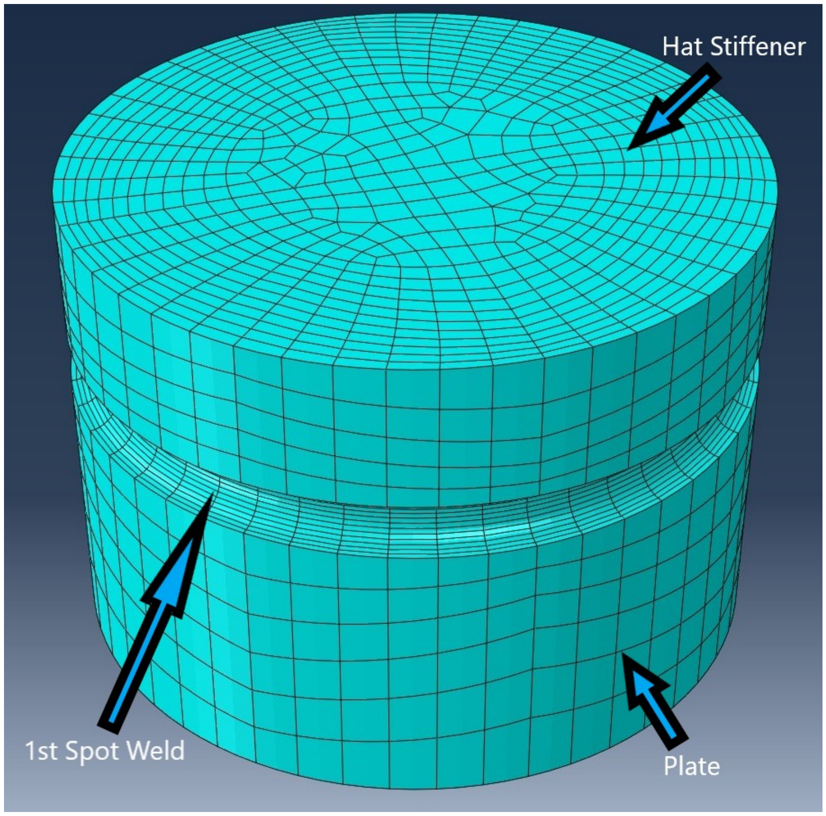

Figure 5.9. 1st Spot Weld Mesh

Since the XTFEM code can currently handle neither shell elements nor mixed element meshes of any kind, the entire model had to be meshed using the C3D20R element. This resulted in large increases in the number of elements every time the number of elements through the thickness of either the plate or the hat increased. Since the stress was going to concentrate in the first weld, that area was meshed first. The remaining spot welds were meshed with only a single element through their thicknesses. The rest of the model was then meshed based on the first spot weld using a general mesh size of 0.1 inch (see Figure 5.10).

A convergence study was performed to find the optimal mesh. The material properties used were those for Ti-6242S at room temperature since the material properties at higher temperatures had yet to be determined (see Table 5.4). Since a mesh convergence study is largely independent of the material properties for a linear-elastic material, this was deemed

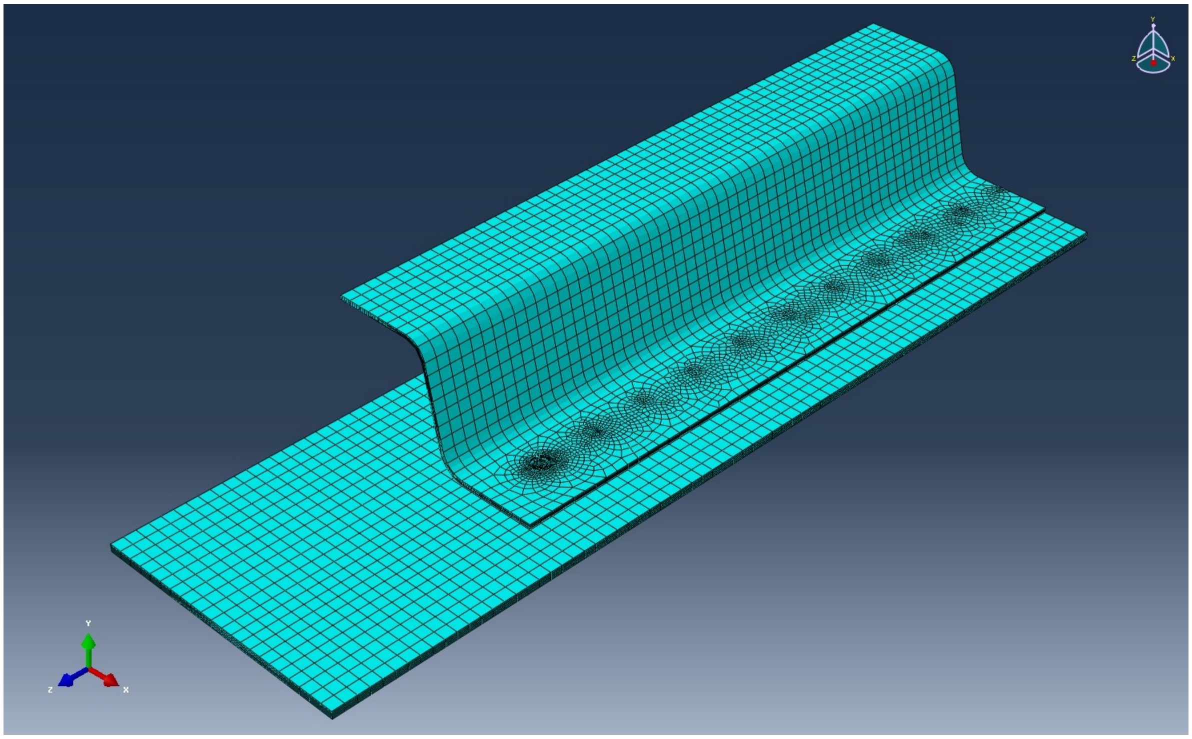

Figure 5.10. Quarter Plate and Hat Stiffener Mesh

Table 5.4. Material Properties for Ti-6242S at Room Temperature

$\rho \left(\frac{\text{lbf}\cdot \text{s}^2}{\text{in}^4}\right)$	E (ksi)	ν	$\alpha \left(\frac{\text{in/in}}{°\text{F}}\right)$	$k \left(\frac{\text{Btu}}{\text{in}\cdot\text{s}\cdot°\text{F}}\right)$	$c_p \left(\frac{\text{Btu}\cdot\text{in}}{\text{lbf}\cdot s^2\cdot°\text{F}}\right)$
4.244E-4	16,700	0.32	4.248E-6	9.259E-5	42.504

not to be an issue. The model was loaded with a constant heat flux to the bottom of the plate of 0.0018 $\frac{\text{Btu}}{\text{in}^2\cdot\text{s}}$, the equivalent flux amplitude used in the experiment, for 1,000 seconds to achieve a large temperature gradient between the plate and the top of the hat. The nodal temperature data was then used in a subsequent static mechanical simulation in which the side of the plate opposite the symmetry-cut was loaded with a pressure of 5,120 psi, the approximate mean stress used in the experiment. Symmetry boundary conditions were applied to the plate and hat along the sides where the hat was cut (x = 0 and z = 0) while the bottom of the plate (y = 0) was fixed in the y-direction to prevent mesh distortion from the plate bowing. As expected, the stress concentrated at the first spot weld (see Figures 5.11

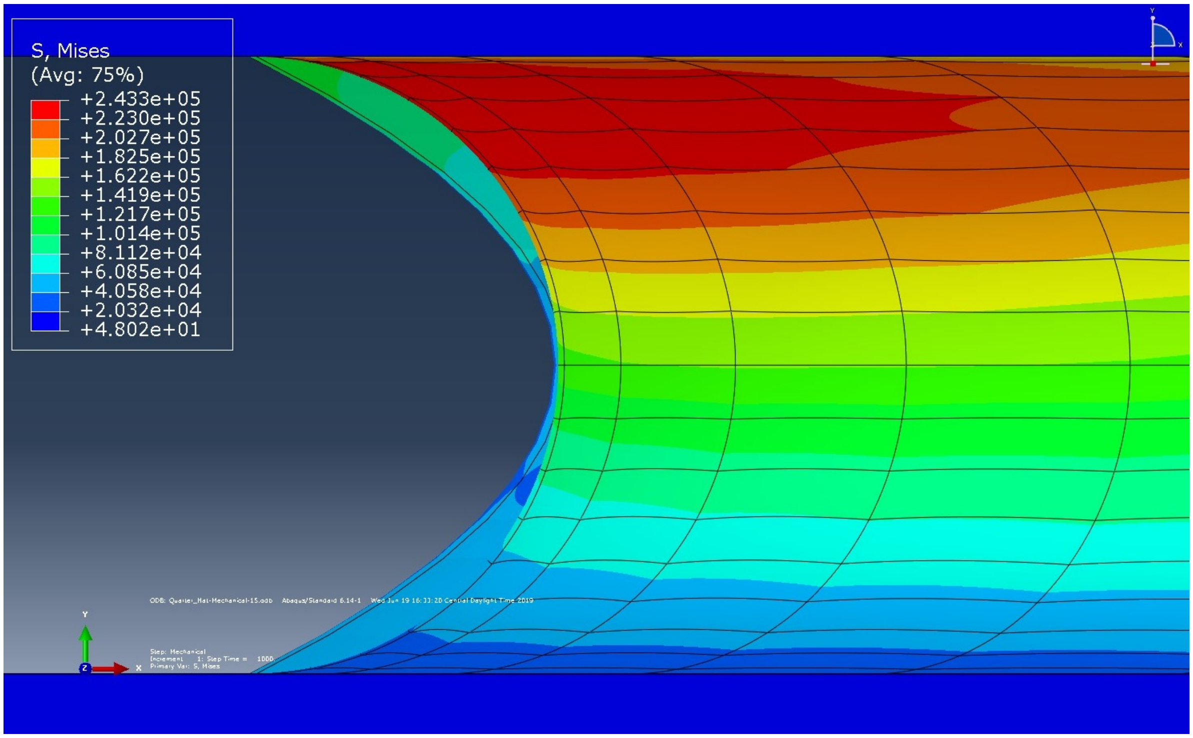

Figure 5.11. Mesh Test Results (1st Weld - Side View)

Table 5.5. Mesh Convergence Study

Mesh #	Elements	Nodes	Max Stress (ksi)
1	27,398	131,091	191.8
2	28,353	135,536	214.7
3	29,987	143,136	227.9
4	35,162	165,039	203.0
5	42,578	195,275	243.9
6	43,634	200,075	240.1
7	45,326	207,423	243.3
8	63,455	281,803	258.2
9	75,059	329,681	274.0
10	107,384	461,852	280.1
11	254,763	1,063,699	297.1

and 5.12). After many iterations of remeshing, heat transfer analysis, and static mechanical analysis, however, the mesh did not reach convergence (see Table 5.5). Refining the mesh even further would have been computationally prohibitive even for the XTFEM code. Since

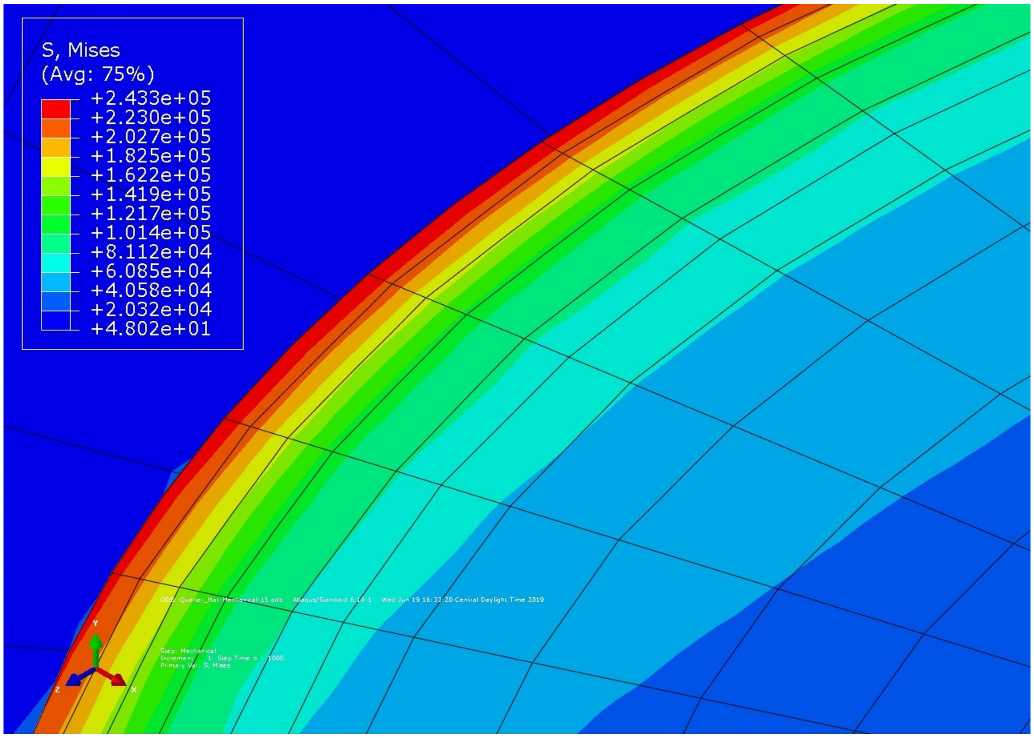

Figure 5.12. Mesh Test Results (1st Weld - Section Cut)

the objective of this study was to demonstrate the abilities of the XTFEM code and not to achieve the exact same answer as the AFRL experiment, mesh #7 was chosen for the study based on a balance of the mesh size and stress results.

5.3.3 Calibrating Material Parameters

The S-N curve chosen to be representative of the high temperature experiment was for Ti-6242S duplex annealed bar at 900°F (Welsch et al. (1994), p.362). Material properties at this temperature were then interpolated from ASM's Material Properties Handbook: Titanium Alloys (Welsch et al. (1994), pp.337-362). However, properties specific to the two scale damage model needed to be calibrated.

The paper by Desmorat et al. describing the two scale damage model also provides a means of calculating the damage parameters, S and s, from a uniaxial S-N curve (Desmorat et al. (2007), pp.914-917, 920). A standalone Fortran code was created to give preliminary

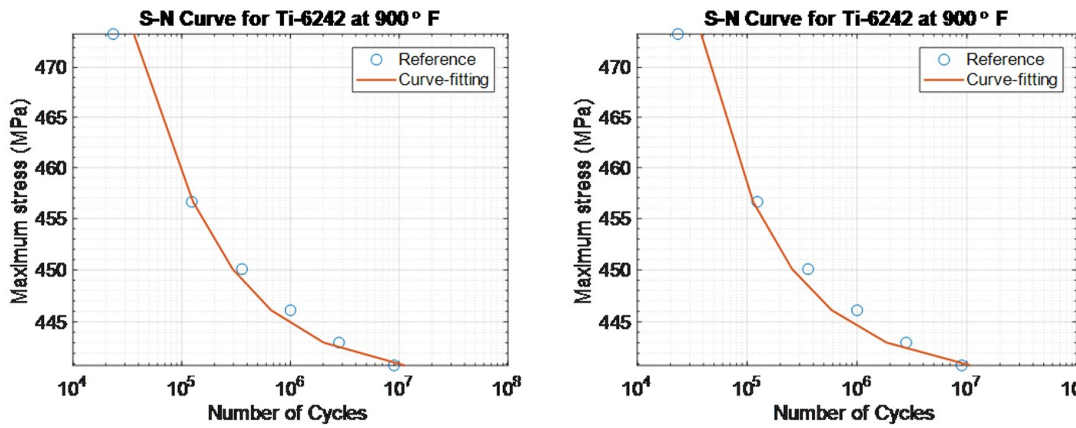

Figure 5.13. MATLAB Results for Two Scale Parameters (Metric Units): S=1.32 MPa, s=56.9 (left); S=2.88 MPa, s=6 (right)

values to the damage parameters, and a MATLAB code was used to further refine the values using a non-linear least squares fit. Using metric units at first, the MATLAB code yielded values of $S = 1.32$ MPa and $s = 56.9$. However, due to the oddity of such a high exponent value ($s = 56.9$), other values for the damage parameters were experimented with, and it was discovered that, since the S-N curve was so flat (dropping less than 10% in fatigue strength over the course of 10,000,000 cycles), just about any values of the damage parameters would give a good fit (see Figure 5.13). So, values of $S = 2.88$ MPa (0.42 ksi) and $s = 6$ were chosen.

However, the two calibration codes mentioned above assumed two things not applicable to our case. The first was that the micro-defects closure parameter, h (see section 4.2.2 and equation 4.94 above), was assumed to be equal to one (i.e., equal damage during compression and tension). This is not realistic since $h \approx 0.2$ for metals (Lemaitre and Desmorat (2005), p.16; Desmorat et al. (2007), p.912). The second assumption was that the S-N curve used for calibration was produced using triangular or "saw-tooth" loading. The curve taken from the Handbook, however, was created using smooth, rotating beam tests which generate sinusoidal loading (Welsch et al. (1994), p.362). Thus, when the parameters were run through the C++ code mentioned in section 5.1 above (which did use sinusoidal loading), the resulting

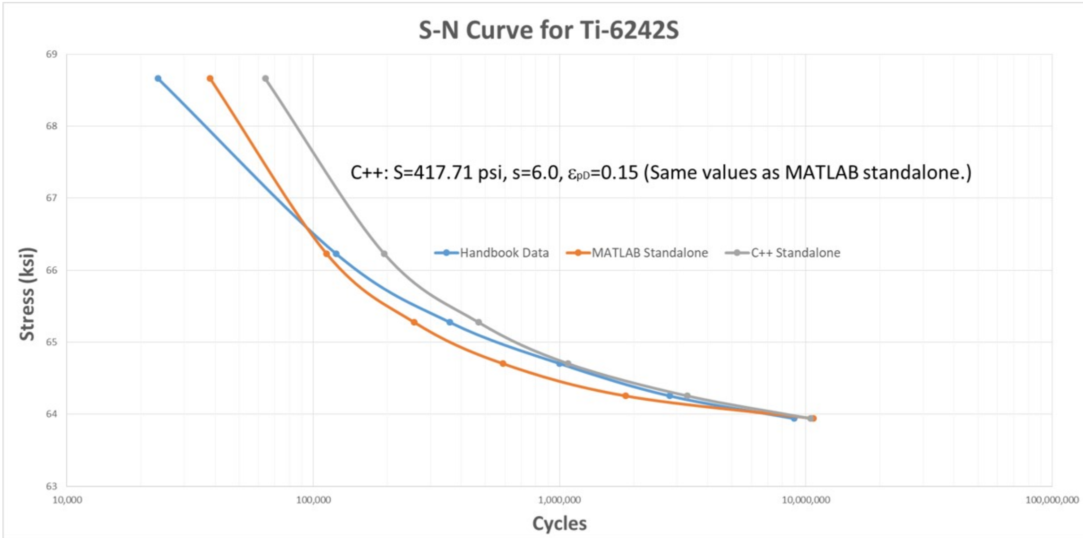

Figure 5.14. S-N Curve with C++ Results Using MATLAB Parameters

Table 5.6. Material Parameters for Ti-6242S at 900°F

ρ $\left(\frac{\text{lbf}\cdot\text{s}^2}{\text{in}^4}\right)$	E (ksi)	ν	C_y (ksi)	σ_u (ksi)	σ_f^∞ (ksi)	ε_{pD}
4.244E-4	11,548	0.353	188.1	108.8	63.7	0.125

h	D_c	S (ksi)	s	α $\left(\frac{\text{in/in}}{°\text{F}}\right)$	k $\left(\frac{\text{Btu}}{\text{in}\cdot\text{s}\cdot°\text{F}}\right)$	c_p $\left(\frac{\text{Btu}\cdot\text{in}}{\text{lbf}\cdot s^2\cdot°\text{F}}\right)$
0.2	0.3	0.0725	0.5	5.423E-6	1.725E-4	61.39

number of cycles did not match the MATLAB values (see the orange and grey curves in Figure 5.14). So, the C++ code was used to further correct the parameters. Again, given the flatness of the Handbook's S-N curve, changing the parameters proved fruitless until the value of the plastic strain damage threshold, ε_{pD} (i.e., the amount of plastic strain a material undergoes by the time it reaches its ultimate strength in monotonic tension), was changed. While the new value of $\varepsilon_{pD} = 0.125$ differs greatly from the value calculated from the Handbook ($\varepsilon_{pD} = 0.15$), it provides a very good fit to the S-N data (see the grey and blue curves in Figure 5.15). Since this variable is only used in the damage subroutine whose sole purpose is to give an estimation of fatigue failure, $\varepsilon_{pD} = 0.125$ was used. See Table 5.6 for the complete listing of the material parameters used in the simulation.

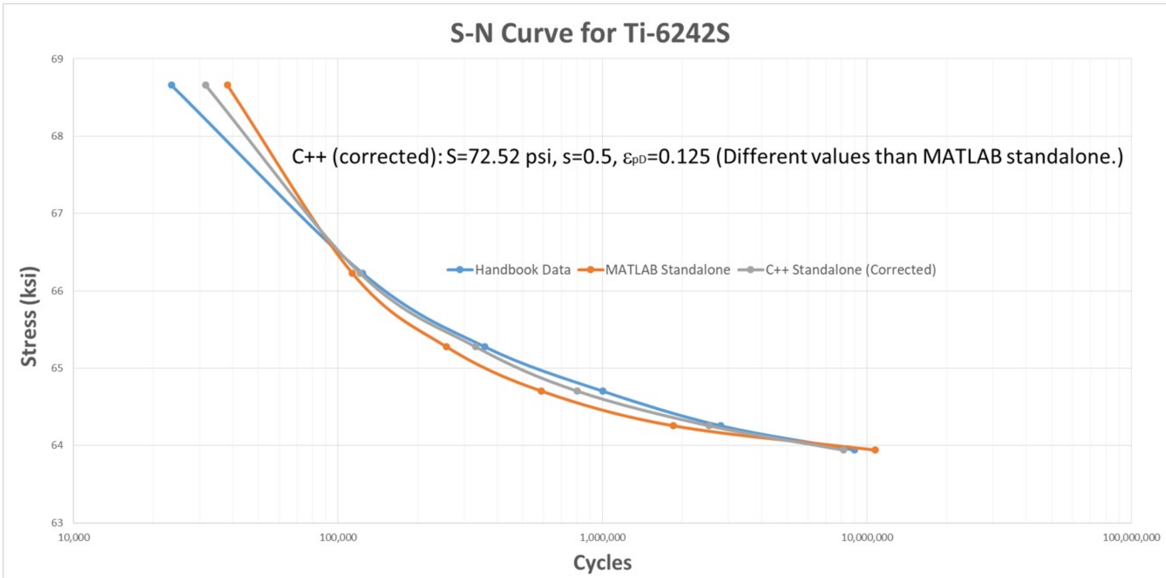

Figure 5.15. S-N Curve with C++ Results Using Corrected Parameters

5.3.4 Simulation and Results

Using the mesh and the material parameters mentioned above, the plate and hat stiffener model was simulated in the XTFEM code. The AFRL stated that the peak temperature during the thermal cycle should be about 1,510 Rankine (1,050°F) while using a heat flux amplitude of 0.0018 $\frac{\text{Btu}}{\text{in}^2 \cdot \text{s}}$. To achieve these parameters, a trial-and-error approach was taken using Abaqus, and it was determined that the frequency of the cyclic heat flux should be $f = 0.00025$ Hz or a period of 4,000 seconds with an initial temperature throughout the model of $T_{ref} = 765$ Rankine (305°F). The mechanical loading to the edge of the plate was 5 Hz as in the experiment. To simplify matters, the pressure amplitude was made to be 5,120 psi with a mean pressure of zero rather than 5,120 psi as the mean pressure and 512 psi as the amplitude as in the experiment. Likewise, the initial temperature for the whole model was set to zero with any rise or fall in temperature being understood as a change in temperature, ΔT, from the initial temperature of 765 Rankine.

A regular space-time temperature vector was used instead of the extended version in the previous example since we are solving for the nodal temperature values using a TFEM formulation instead of applying the temperatures directly. To calculate the extended space-time thermal "force" term (equation 5.3), the spatial thermal "force" as a function of time, $\varTheta\,(x,t)$, was approximated using a quadratic polynomial. To do this, the spatial thermal "force," $\varTheta$, was calculated at the three time-nodes, and a polynomial was fitted to the three data points by solving for the three coefficients, a_0, a_1, and a_2:

$$\varTheta_i\,(x,t_i) = \int_{\varOmega} \boldsymbol{B}_x^T\,\boldsymbol{D}\,\alpha\,\Delta T\,(x,t_i)\;\boldsymbol{I}\,d\varOmega \tag{5.5}$$

$$\varTheta\,(x,t) = a_0 + a_1\,t + a_2\,t^2 \tag{5.6}$$

$$\begin{bmatrix} \varTheta_1\,(x,t_1) \\ \varTheta_2\,(x,t_2) \\ \varTheta_3\,(x,t_3) \end{bmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \tag{5.7}$$

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{bmatrix}^{-1} \begin{bmatrix} \varTheta_1\,(x,t_1) \\ \varTheta_2\,(x,t_2) \\ \varTheta_3\,(x,t_3) \end{bmatrix} \tag{5.8}$$

$$\boldsymbol{EIN1\varTheta} \otimes \boldsymbol{\varTheta} = \int_{I_n} \begin{bmatrix} \dot{\boldsymbol{N}}_t & \dot{\dot{\boldsymbol{N}}}_t \end{bmatrix}^T \left(a_0 + a_1\,t + a_2\,t^2\right)\,dt \tag{5.9}$$

This procedure was done for each mechanical degree of freedom in the model.

The simulation was run on TACC's Lonestar5 high-performance computer using 4 compute nodes and 96 processors. Unfortunately, it became apparent after 24 hours that the simulation would not finish within Lonestar5's 48 hour maximum run time. So, the pressure amplitude was increased to 9,000 psi, and the simulation was re-run. It completed in about 28.8 hours CPU run time with the first element failing after about 36,600 cycles and completely finishing the simulation shortly thereafter at about 36,650 cycles.

The first point of interest in the simulation was the nonlinear accumulation of the stored energy density, w_s. As seen in Figure 5.16, the energy plateaus then rises repeatedly over the course of the simulation while the sharp rise in energy at the end was due to the accumulation

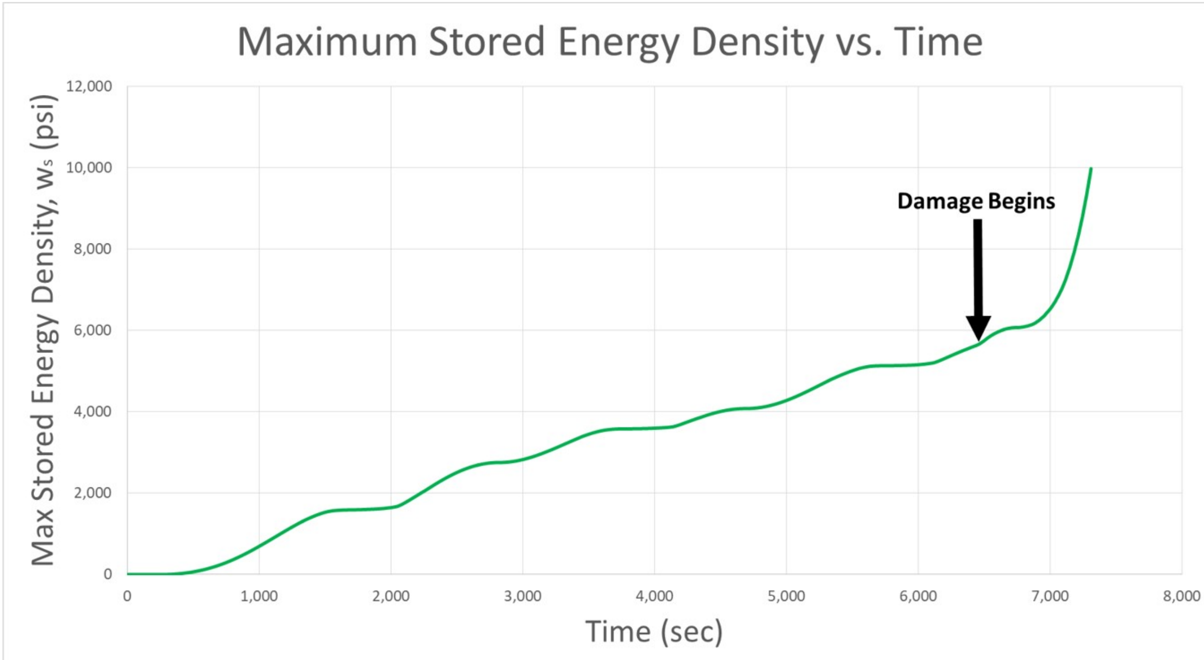

Figure 5.16. Maximum Stored Energy Density, w_s, vs. Time

of damage (see equation 4.24 above). In purely mechanical simulations, this value rises linearly up until damage accumulation begins. By plotting the temperature of a node in the stress concentration area of the first weld, we can see why the nonlinear accumulation occurs (see Figure 5.17). The energy seems to increase at a greater rate whenever the magnitude of the slope of the temperature curve is largest. We hypothesize that this is due to the relatively large temperature gradients present in the model during these times which generate extra mechanical stresses on top of those created by the pressure load (see Figure 5.18).

The second point of interest is the failure of the material at the end of the simulation. As seen in Figure 5.19, the stress concentrates at the first spot weld which was known from our static simulations. Once the damage reached the critical value in these elements, D_c, the elements were deleted. This created stress singularities due to the newly formed sharp, reentrant corners where the deleted elements used to be, a drawback of the element deletion method of crack modeling. Thus, the stress contour in Figure 5.20 was capped at 250,000 psi to show the stress distribution elsewhere. After a few more time steps, the 'crack' reached

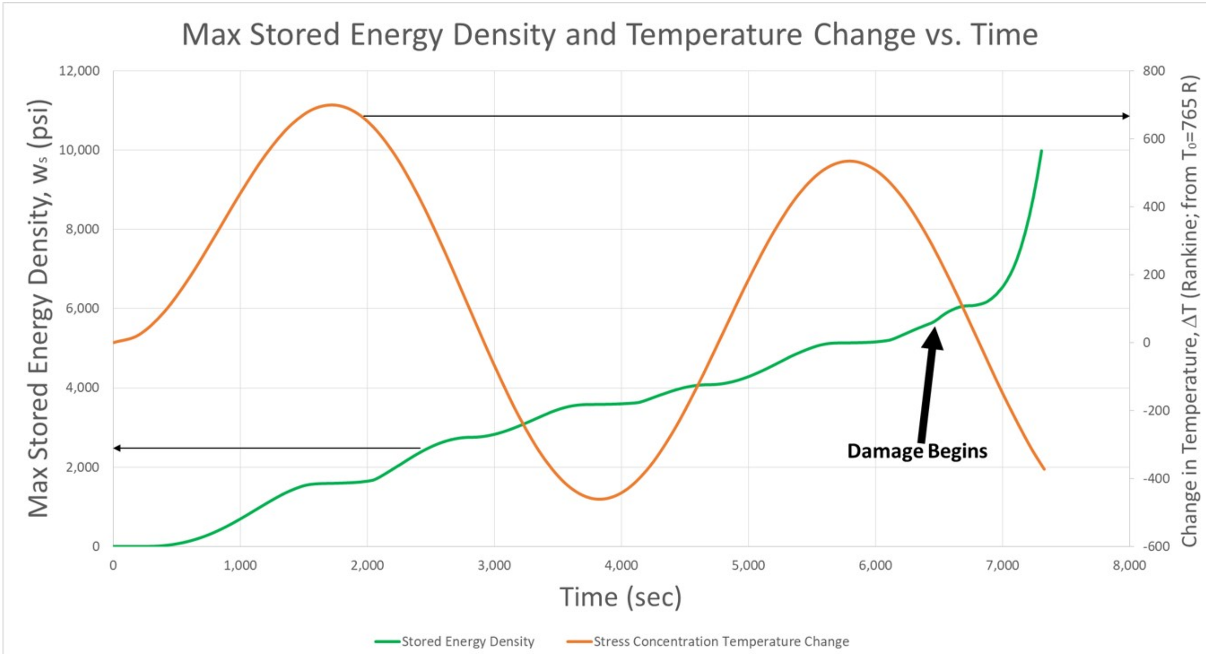

Figure 5.17. w_s and 1st Weld Nodal Temperature Change, ΔT, vs. Time

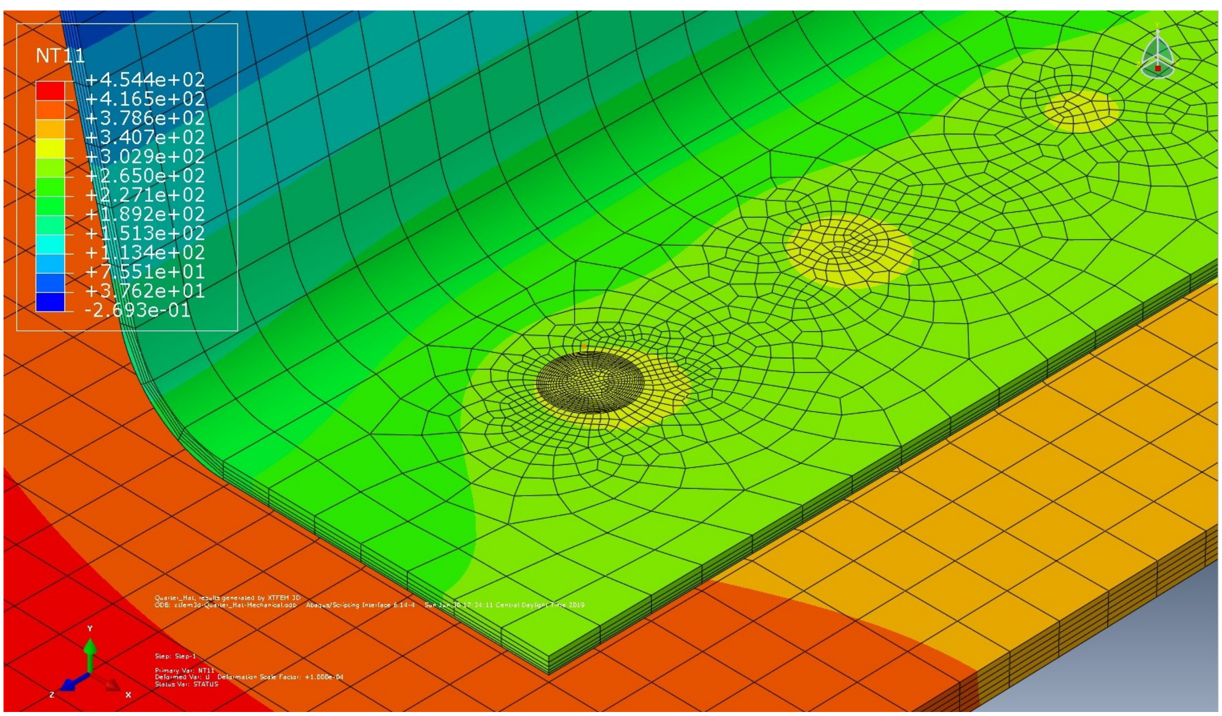

Figure 5.18. Temperature Distribution in the Model

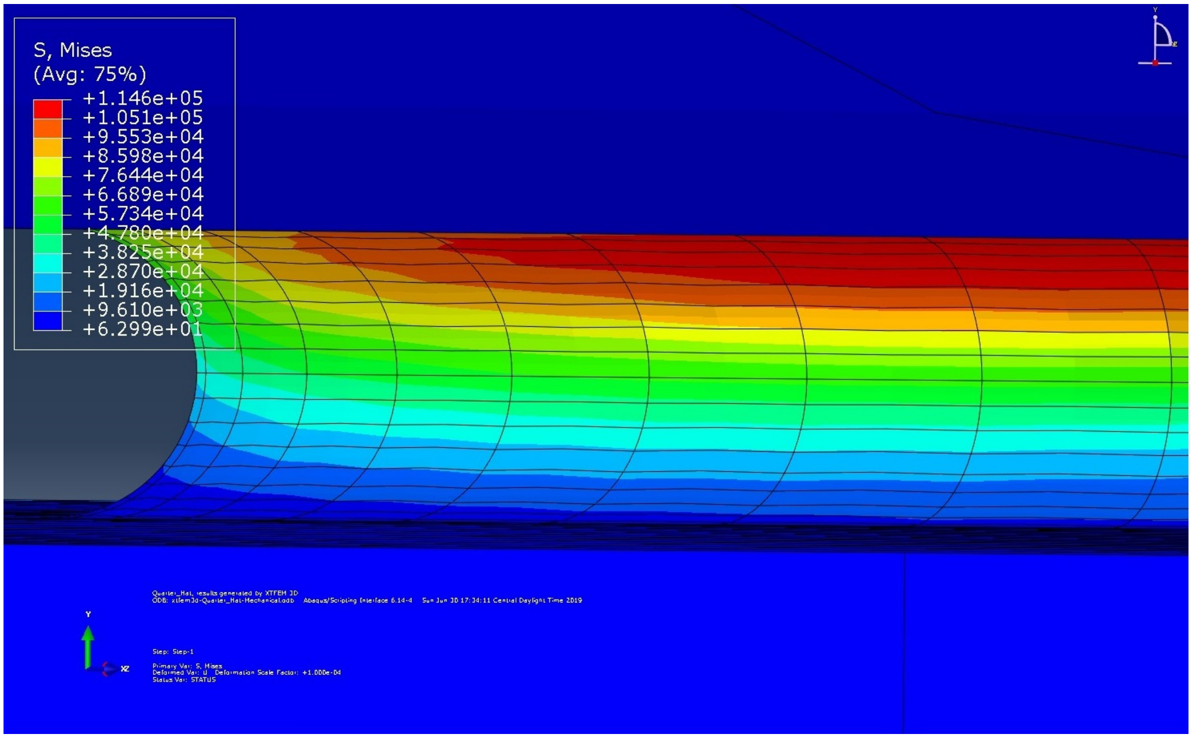

Figure 5.19. Stress Concentrated at 1st Spot Weld

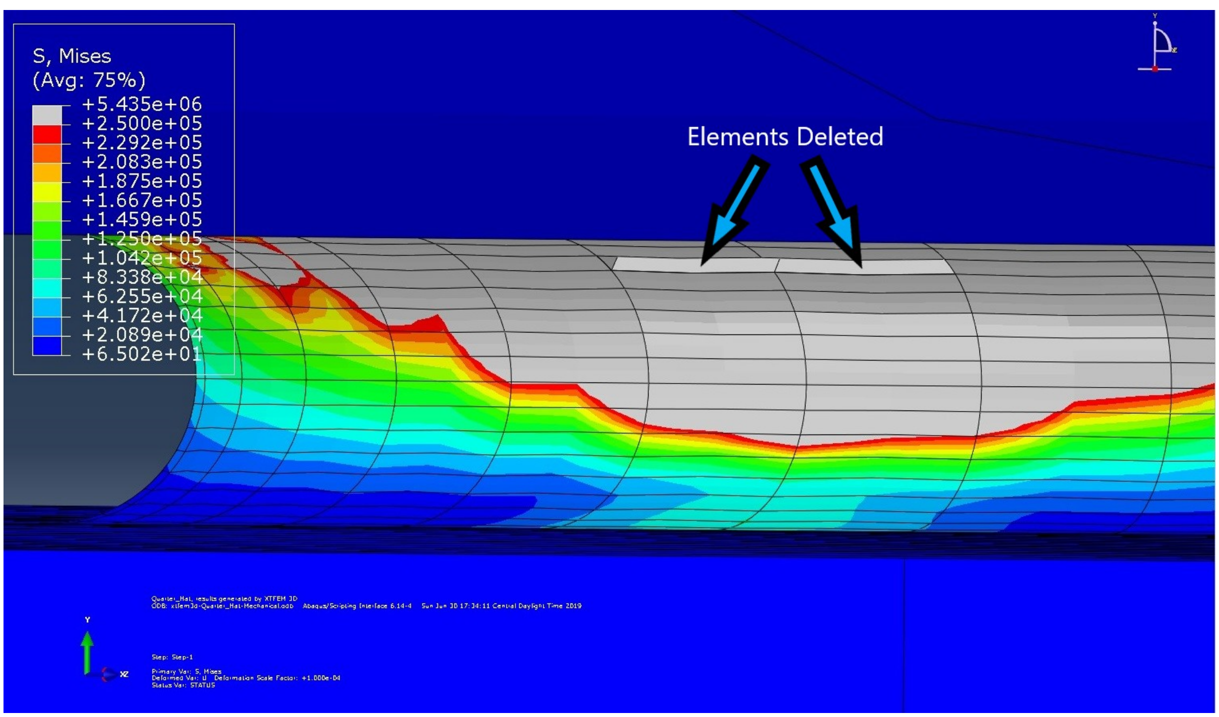

Figure 5.20. First Element Deletion at ∼36,600 Cycles Creating Stress Singularities

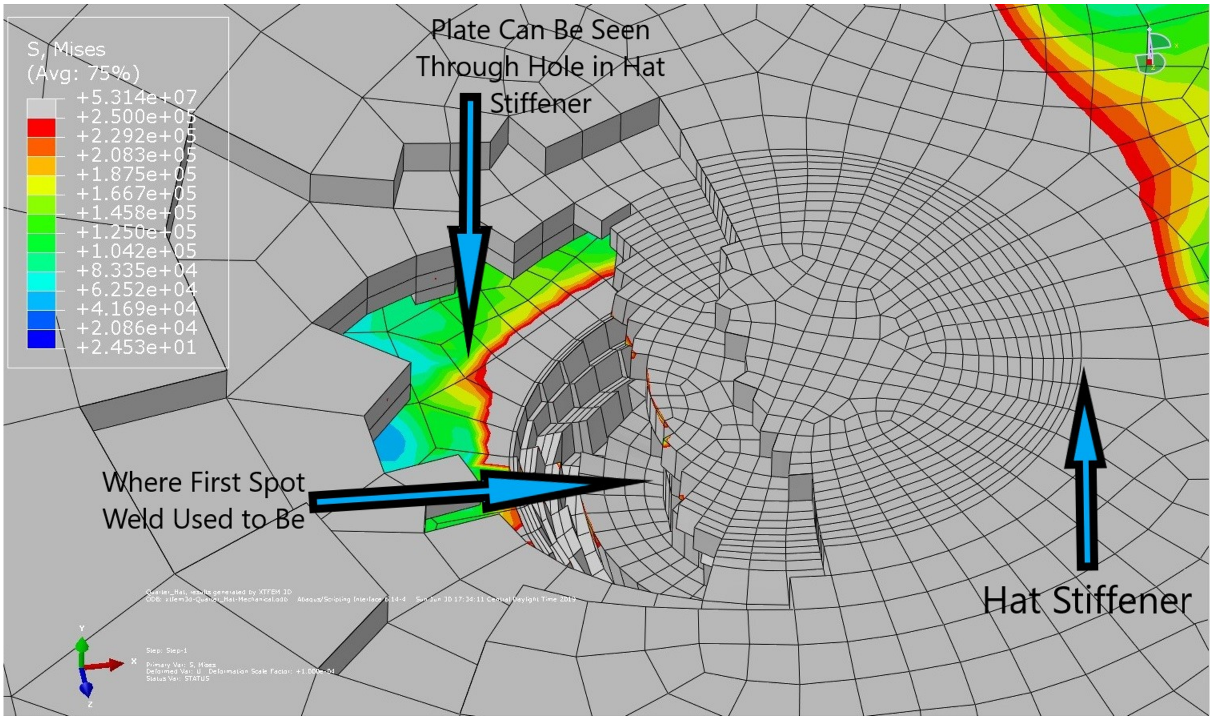

Figure 5.21. Fatigue 'Crack' Tearing Through the Surface of the Hat

the surface of the hat stiffener causing more stress singularities (see Figure 5.21). Figure 5.22 shows what remains of the first spot weld just prior to the end of the simulation. At the same time, Figure 5.23 shows the large stresses in the remaining spot welds. Within a few cycles of this frame, the remaining spot welds delete causing the hat stiffener to separate from the plate. This makes the stiffness matrix singular causing the simulation to stop.

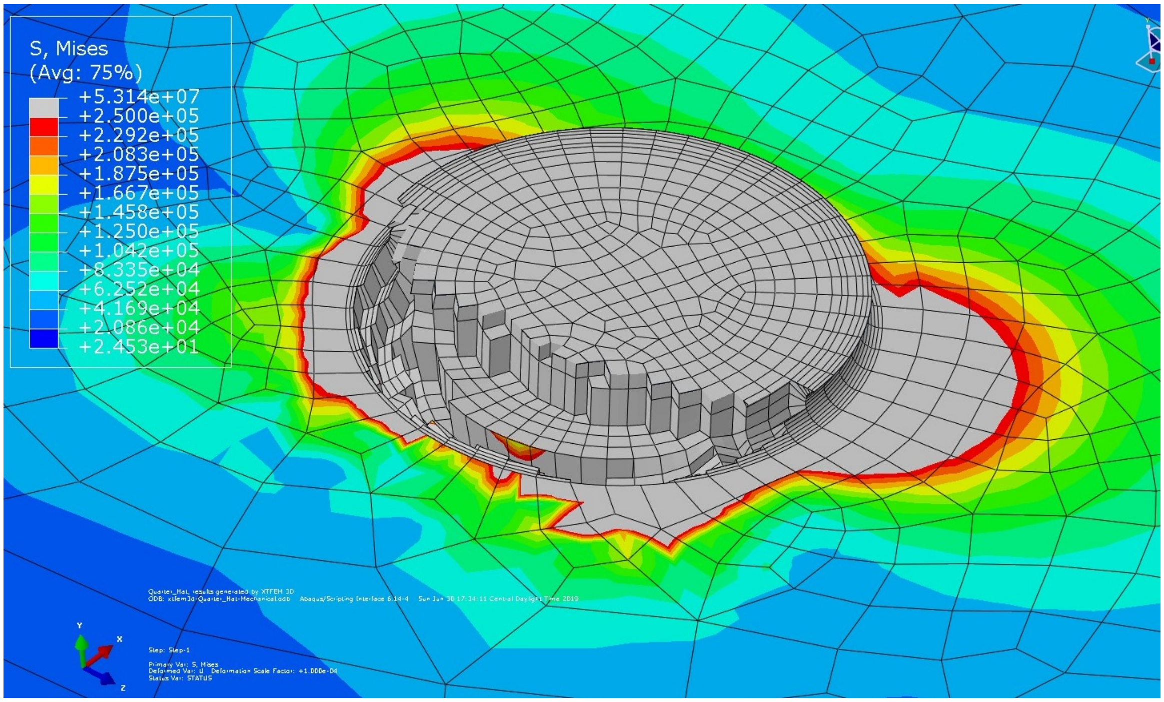

Figure 5.22. Section Cut Showing the Remains of the First Spot Weld

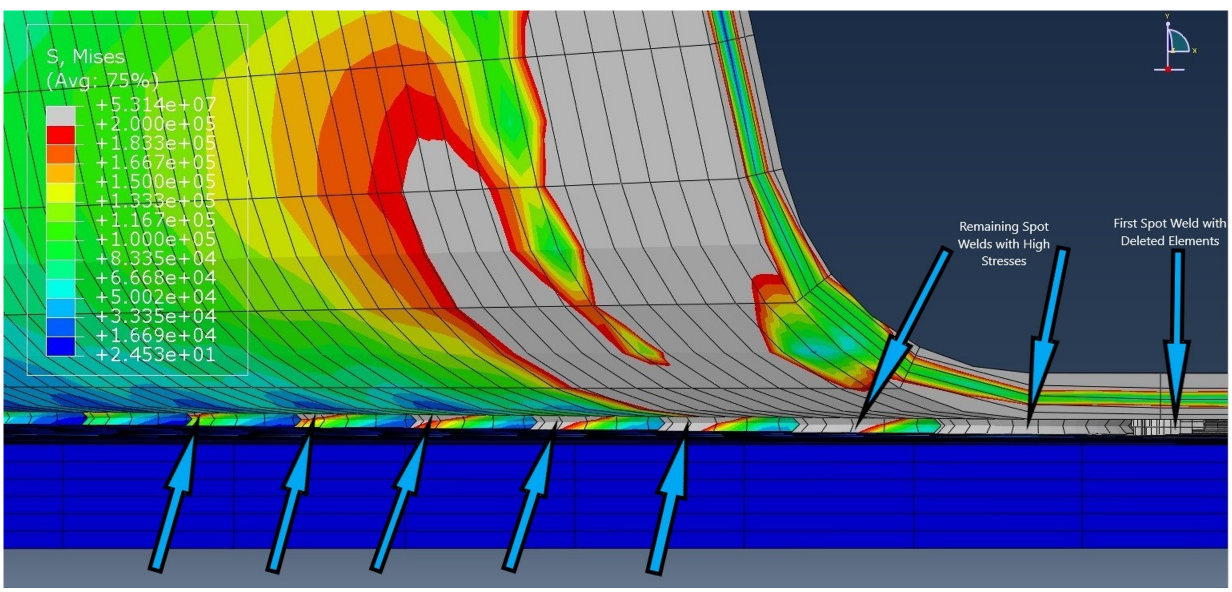

Figure 5.23. High Stresses on Other Spot Welds Before Failure

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

We have proven the effectiveness of the XTFEM code by simulating three different models of increasing complexity. In the single element model, we showed that the XTFEM formulation is superior to the standard, finite-difference time-integration methods by contrasting the user runtime of both formulations. The XTFEM formulation was several orders of magnitude faster even while using more interpolation points to give it greater accuracy. In the beam model, we simulated an academic problem of a beam constrained at both ends subjected to a cyclical temperature variation. We showed how the XTFEM code could accurately translate the temperature variation into an equivalent "force" which, due to the constraint, translated into stress. The minor differences in cycles to failure between the C++ code used as a control and the XTFEM code were most likely due to the round-off in calculated temperature values used to create the equivalent stresses. Lastly, we simulated a truly pertinent problem that is of interest to the Air Force: the thermo-mechanical high cycle fatigue of hypersonic aircraft structures. After performing a mesh convergence study and calibrating the material parameters, we simulated the fatigue of the plate and hat stiffener using similar conditions to the AFRL experiment as best we could. We saw that the relatively high temperature gradient decreased the fatigue life of the component by creating mechanical stresses on top of those made by the pressure load. We also saw the failure of the material initiating at the first spot weld and propagating up to the surface of the hat stiffener.

6.2 Future Work

While we have demonstrated the XTFEM formulation's capabilities, there is still more work to be done in order to simulate the full range of fatigue problems. These are just a few of the many possible developments that could be made to the code:

1. Plasticity: Currently, the only mesoscale (finite element level) material model the code is capable of running is for a linear-elastic, isotropic material. However, there are many high cycle fatigue cases which have a small amount of plasticity at the mesoscale, and this can make a difference in the fatigue life. In fact, the two scale progressive fatigue damage model (Desmorat et al. (2007)) is designed to allow for small amounts of mesoscale plastic strains.

2. Low Cycle Fatigue: Related to the previous proposal, plasticity would open up the possibility of simulating low cycle fatigue conditions. This would, of course, necessitate having another fatigue damage model either alongside the HCF two scale model or as a separate option in which the two scale model is turned off. This would allow the XTFEM code to simulate the full range of fatigue loading. This is especially pertinent for thermo-mechanical fatigue (TMF) since most TMF situations are either LCF or mixed LCF and HCF (see section 1.2.2 above).

3. Mixed Element Meshes: At this time, the XTFEM code is limited to either eight node brick elements (C3D8 in Abaqus) or twenty node brick elements (C3D20). However, many industrial components have a complicated geometry that necessitates a mesh consisting of several element types such as brick, tetrahedral, wedge, and shell. A mixed mesh allows one to create quality meshes without greatly multiplying the number of elements and nodes. The plate and hat stiffener model, for instance, could have been meshed and simulated much more efficiently if the spot welds were meshed with solid elements while shell elements were used for the plate and hat stiffener.

4. Multiple Materials: At present, the XTFEM code can handle only one material in the entire component. On the other hand, most industrial components are comprised of multiple materials. This is especially germane for many thermal fatigue situations such as electronic surface-mount devices where mismatches between coefficients of thermal expansion create cyclic stresses and, thus, fatigue damage.

5. Thermal Convection and Radiation: Currently, the only types of boundary conditions available in the code for the thermal TFEM are surface heat flux and fixed temperature boundary conditions. On the other hand, many cases of thermo-mechanical fatigue involve convection and thermal radiation. The plate and hat stiffener model in section 5.3 above is an excellent example. When the AFRL first simulated the experiment in Abaqus, they included both convection and thermal radiation since both modes of heat transfer were significant.

6. No Temperature Boundary Condition: Unlike Abaqus which can simulate a transient, thermal model without a prescribed temperature boundary condition (i.e., it can assume all surfaces are insulated), the XTFEM code cannot handle such conditions since it would make the space-time stiffness matrix singular. Removing this constraint would allow the modeling of problems for which there is no prescribed temperature surface, a condition which, at times, can be more realistic.

7. Contact Nonlinearity: Many thermo-mechanical problems involve surfaces which start out separated from each other but, due to thermal expansion, come into contact and transmit forces. Again, the case of the plate and hat stiffener comes to mind. The original spot welds were actually much thinner than they were modeled in section 5.3.1. In reality, the plate and the hat stiffener would have come into contact, transmitted forces, and conducted heat.

8. XFEM for Crack Growth: Currently, the XTFEM code implements crack growth by way of element deletion. This method has the problem of lost energy due to the disappearance of mass (Belytschko et al. (2014), p.644). Furthermore, one would have to know where the crack would initiate and grow beforehand in order to properly mesh the component for crack growth (Belytschko et al. (2014), p.644), but for most industrial components, one does not know that apart from expensive experimentation or other fatigue algorithms which the XTFEM code seeks to replace (see chapter 3). Lastly, element deletion can generate stress singularities by creating sharp, reentrant corners where an element used to be (see section 5.3.4 above). Therefore, the best way to implement crack initiation and propagation is the extended finite element method (XFEM) which was discussed briefly in section 2.4.1. The chief advantage of XFEM is that the accurate modeling of a crack is independent of the mesh. Furthermore, there is no loss of mass/energy during crack growth and no stress singularities. XFEM would thus avoid the major pitfalls of element deletion.

9. Random Loading: Actual components in the field rarely undergo constant amplitude, constant frequency loading. Rather, the loading history is more often than not random, and being able to simulate random loading would allow the XTFEM code to tackle the vast majority of common fatigue problems seen in industry.

APPENDIX

PROOFS IN THE TWO-SCALE MODEL

This appendix gives proofs for some of the assertions made in chapter 4.

A.1 Proof: $\sigma^{\mu}_{ij}\sigma^{\mu}_{ij} = \left\langle \sigma^{\mu}_{ij} \right\rangle^{+} \left\langle \sigma^{\mu}_{ij} \right\rangle^{+} + \left\langle \sigma^{\mu}_{ij} \right\rangle^{-} \left\langle \sigma^{\mu}_{ij} \right\rangle^{-}$

$$\begin{aligned}
\sigma^{\mu}_{ij}\,\sigma^{\mu}_{ij} &= \left(\left\langle \sigma^{\mu}_{ij} \right\rangle - \left\langle -\sigma^{\mu}_{ij} \right\rangle\right)\left(\left\langle \sigma^{\mu}_{ij} \right\rangle - \left\langle -\sigma^{\mu}_{ij} \right\rangle\right) \\
&= \left(\left\langle \sigma^{\mu}_{ij} \right\rangle^{+} - \left\langle \sigma^{\mu}_{ij} \right\rangle^{-}\right)\left(\left\langle \sigma^{\mu}_{ij} \right\rangle^{+} - \left\langle \sigma^{\mu}_{ij} \right\rangle^{-}\right) \\
&= \left\langle \sigma^{\mu}_{ij} \right\rangle^{+}\left\langle \sigma^{\mu}_{ij} \right\rangle^{+} - 2\left\langle \sigma^{\mu}_{ij} \right\rangle^{+}\left\langle \sigma^{\mu}_{ij} \right\rangle^{-} + \left\langle \sigma^{\mu}_{ij} \right\rangle^{-}\left\langle \sigma^{\mu}_{ij} \right\rangle^{-}
\end{aligned} \tag{A.1}$$

Since $\langle x \rangle^{+} \langle x \rangle^{-} \equiv 0$, the middle term of equation A.1 becomes zero leaving us with:

$$\therefore \sigma^{\mu}_{ij}\,\sigma^{\mu}_{ij} = \left\langle \sigma^{\mu}_{ij} \right\rangle^{+}\left\langle \sigma^{\mu}_{ij} \right\rangle^{+} + \left\langle \sigma^{\mu}_{ij} \right\rangle^{-}\left\langle \sigma^{\mu}_{ij} \right\rangle^{-} \tag{A.2}$$

A.2 Proof: $\left\langle tr\left(\sigma^{\mu}_{ij}\right)\right\rangle^{+} - \left\langle tr\left(\sigma^{\mu}_{ij}\right)\right\rangle^{-} = tr\left(\left\langle \sigma^{\mu}_{ij} \right\rangle^{+}\right) - tr\left(\left\langle \sigma^{\mu}_{ij} \right\rangle^{-}\right)$

$$\begin{gathered}
\left\langle tr\left(\begin{bmatrix} \sigma^{\mu}_{11} & 0 & 0 \\ 0 & \sigma^{\mu}_{22} & 0 \\ 0 & 0 & \sigma^{\mu}_{33} \end{bmatrix}\right)\right\rangle^{+} - \left\langle tr\left(\begin{bmatrix} \sigma^{\mu}_{11} & 0 & 0 \\ 0 & \sigma^{\mu}_{22} & 0 \\ 0 & 0 & \sigma^{\mu}_{33} \end{bmatrix}\right)\right\rangle^{-} = \\
tr\left(\begin{bmatrix} \langle\sigma^{\mu}_{11}\rangle & 0 & 0 \\ 0 & \langle\sigma^{\mu}_{22}\rangle & 0 \\ 0 & 0 & \langle\sigma^{\mu}_{33}\rangle \end{bmatrix}\right) - tr\left(\begin{bmatrix} \langle-\sigma^{\mu}_{11}\rangle & 0 & 0 \\ 0 & \langle-\sigma^{\mu}_{22}\rangle & 0 \\ 0 & 0 & \langle-\sigma^{\mu}_{33}\rangle \end{bmatrix}\right)
\end{gathered} \tag{A.3}$$

Using forward slashes to indicate either/or with the value to the left of the slash being either zero or the variable if positive and the value to the right of the slash being either zero or the variable if negative:

$$\begin{aligned}
\left(\sigma^{\mu}_{11} + \sigma^{\mu}_{22} + \sigma^{\mu}_{33}\right)/0 - 0/\left(-\sigma^{\mu}_{11} - \sigma^{\mu}_{22} - \sigma^{\mu}_{33}\right) &= \langle\sigma^{\mu}_{11}\rangle - \langle-\sigma^{\mu}_{11}\rangle + \langle\sigma^{\mu}_{22}\rangle - \langle-\sigma^{\mu}_{22}\rangle + \langle\sigma^{\mu}_{33}\rangle - \langle-\sigma^{\mu}_{33}\rangle \\
\left(\sigma^{\mu}_{11} + \sigma^{\mu}_{22} + \sigma^{\mu}_{33}\right)/\left(\sigma^{\mu}_{11} + \sigma^{\mu}_{22} + \sigma^{\mu}_{33}\right) &= \sigma^{\mu}_{11}/0 - 0/\left(-\sigma^{\mu}_{11}\right) + \sigma^{\mu}_{22}/0 - 0/\left(-\sigma^{\mu}_{22}\right) + \sigma^{\mu}_{33}/0 - 0/\left(-\sigma^{\mu}_{33}\right) \\
\left(\sigma^{\mu}_{11} + \sigma^{\mu}_{22} + \sigma^{\mu}_{33}\right)/\left(\sigma^{\mu}_{11} + \sigma^{\mu}_{22} + \sigma^{\mu}_{33}\right) &= \left(\sigma^{\mu}_{11} + \sigma^{\mu}_{22} + \sigma^{\mu}_{33}\right)/\left(\sigma^{\mu}_{11} + \sigma^{\mu}_{22} + \sigma^{\mu}_{33}\right) \\
tr\left(\sigma^{\mu}_{ij}\right) &= tr\left(\sigma^{\mu}_{ij}\right)
\end{aligned} \tag{A.4}$$

$$\therefore \left\langle tr\left(\sigma_{ij}^{\mu}\right)\right\rangle^{+} - \left\langle tr\left(\sigma_{ij}^{\mu}\right)\right\rangle^{-} = tr\left(\left\langle\sigma_{ij}^{\mu}\right\rangle^{+}\right) - tr\left(\left\langle\sigma_{ij}^{\mu}\right\rangle^{-}\right) \tag{A.5}$$

A.3 Proof: $\left(\sigma_{kk}^{\mu}\right)^2 = \left(\left\langle\sigma_{kk}^{\mu}\right\rangle^{+}\right)^2 + \left(\left\langle\sigma_{kk}^{\mu}\right\rangle^{-}\right)^2$

Using $\left\langle tr\left(\sigma_{ij}^{\mu}\right)\right\rangle^{+} - \left\langle tr\left(\sigma_{ij}^{\mu}\right)\right\rangle^{-} = tr\left(\left\langle\sigma_{ij}^{\mu}\right\rangle^{+}\right) - tr\left(\left\langle\sigma_{ij}^{\mu}\right\rangle^{-}\right)$ (see proof A.2 above), we have:

$$\begin{aligned}
\left(\sigma_{kk}^{\mu}\right)^2 &= \left(\left\langle\sigma_{kk}^{\mu}\right\rangle^{+} - \left\langle\sigma_{kk}^{\mu}\right\rangle^{-}\right)^2 \\
&= \left(\left\langle\sigma_{kk}^{\mu}\right\rangle^{+}\right)^2 + 2\left\langle\sigma_{kk}^{\mu}\right\rangle^{+}\left\langle\sigma_{kk}^{\mu}\right\rangle^{-} - \left(\left\langle\sigma_{kk}^{\mu}\right\rangle^{-}\right)^2
\end{aligned} \tag{A.6}$$

Using the fact that $\langle x\rangle^{+}\langle x\rangle^{-} \equiv 0$, the middle term of equation A.6 becomes zero leaving us with:

$$\therefore \left(\sigma_{kk}^{\mu}\right)^2 = \left(\left\langle\sigma_{kk}^{\mu}\right\rangle^{+}\right)^2 + \left(\left\langle\sigma_{kk}^{\mu}\right\rangle^{-}\right)^2 \tag{A.7}$$

REFERENCES

Argyris, J. H. and D. W. Scharpf (1969). Finite elements in time and space. *Nuclear Engineering and Design 10*, 456–464.

ASTM (2005). ASTM 1049-85: Standard practices for cycle counting in fatigue analysis.

Bak, B. L., A. Turon, E. Lindgaard, and E. Lund (2016). A simulation method for high-cycle fatigue-driven delamination using a cohesive zone model. *International Journal for Numerical Methods in Engineering 106*(3), 163–191.

Barbu, L. G., S. Oller, X. Martinez, and A. Barbat (2015). High cycle fatigue simulation: A new stepwise load-advancing strategy. *Engineering Structures 97*, 118–129.

Bartsch, T. M. (2003). High cycle fatigue (HCF) science and technology program 2002 annual report. Technical Report AFRL-PR-WP-TM-2004-2040, Air Force Research Laboratory, Propulsion Directorate, Wright-Patterson AFB, OH.

Belytschko, T., W. K. Liu, B. Moran, and K. I. Elkhodary (2014). *Nonlinear Finite Elements for Continua and Structures* (2nd ed.). Hoboken, NJ: John Wiley & Sons.

Bhamare, S., T. Eason, S. Spottswood, S. R. Mannava, V. K. Vasudevan, and D. Qian (2014, February). A multi-temporal scale approach to high cycle fatigue simulation. *Computational Mechanics 53*(2), 387–400.

Bhamare, S. D. (2012). *High Cycle Fatigue Simulation using Extended Space-Time Finite Element Method Coupled with Continuum Damage Mechanics.* Ph. D. thesis, University of Cincinnati.

Bill, R. C. (1986). Micromechanisms of thermomechanical fatigue: A comparison with isothermal fatigue. Technical Report 87331, NASA.

Blevins, R. D., I. Holehouse, and K. R. Wentz (1993). Thermoacoustic loads and fatigue of hypersonic vehicle skin panels. *Journal of Aircraft 30*(6), 971–978.

Cedergren, J., S. Melin, and P. Lidström (2004). Numerical modelling of P/M steel bars subjected to fatigue loading using an extended Gurson model. *European Journal of Mechanics-A/Solids 23*(6), 899–908.

Chaboche, J. (1974). A differential law for nonlinear cumulative fatigue damage. *Materials and Building Research, Paris Institut Technique Du Batiment Et Des Travaus Publies, Annales de l'ITBTP, HS* (39), 117–124.

Chaboche, J. (1977). Sur l'utilisation des variables d'etat interne pour la description du comportement viscoplastique et de la rupture par endommagement. In W. Nowacki (Ed.), *Problèmes Non-Linéaires de Mécanique (Proceedings of French-Polish Symposium, Cracow 1977)*, Warsaw, pp. 137–159. PWN (State Publishing House of Science).

Chaboche, J.-L. (1981). Continuous damage mechanics - A tool to describe phenomena before crack initiation. *Nuclear Engineering and Design 64*, 233–247.

Chaboche, J.-L. (1987). Continuum damage mechanics and its application to structural lifetime predictions. *La Recherche Aerospatiale (English Edition) 4*, 37–54.

Charkaluk, E., A. Constantinescu, H. Maïtournam, and K. D. Van (2009). Revisiting the Dang Van criterion. *Procedia Engineering 1*(1), 143–146.

Charkaluk, E. and L. Rémy (2011). *Fatigue of Materials and Structures: Application to Damage and Design*, Volume 1, Chapter Thermal Fatigue, pp. 271–338. Hoboken, NJ: John Wiley & Sons.

Chessa, J. and T. Belytschko (2004). Arbitrary discontinuities in space-time finite elements by level sets and X-FEM. *International Journal for Numerical Methods in Engineering 61*, 2595–2614.

Coffin, L. (1954a, August). Apparatus for study of effects of cyclic thermal stresses on a ductile material. *Transactions of the American Society of Mechanical Engineers 53*(76), 923–930.

Coffin, L. (1954b, August). A study of the effects of cyclic thermal stresses on a ductile material. *Transactions of the American Society of Mechanical Engineers 53*(76), 931–950.

Cook, R. D., D. S. Malkus, M. E. Plesha, and R. J. Witt (2002). *Concepts and Applications of Finite Element Analysis* (4th ed.). Hoboken, NJ: John Wiley & Sons.

Crossland, B. (1956). Effect of large hydrostatic pressures on the torsional fatigue strength of an alloy steel. In *Proceedings of the International Conference on Fatigue of Metals*, London, pp. 138–149. The Institution of Mechanical Engineers.

Dang Van, K. (1973). *Sur la résistance à la fatigue des métaux. Thèse de Doctorat ès Sciences.* Ph. D. thesis, Scientifique et Technologique l'Armement, Paris.

Dang Van, K. (1993). Macro-micro approach in high-cycle multiaxial fatigue. In D. McDowell and R. Ellis (Eds.), *Advances in Multiaxial Fatigue*, ASTM STP 1991, pp. 120–130. Philadelphia: ASTM International.

Dang Van, K., B. Griveau, and O. Message (1989). On a new multiaxial fatigue limit criterion: theory and application. In M. Brown and K. Miller (Eds.), *Biaxial and Multiaxial Fatigue*, Volume EGF, London, pp. 479–496. Mechanical Engineering Publications.

Delfour, M., W. Hager, and F. Trochu (1981). Discontinuous galerkin methods for ordinary differential equations. *Mathematics of Computation 36*, 455–473.

Desmorat, R., A. Du Tertre, and P. Gaborit (2015, June). Multiaxial haigh diagrams from incremental two scale damage analysis. *AerospaceLab* (9), 1–15.

Desmorat, R., A. Kane, M. Seyedi, and J. P. Sermage (2007). Two scale damage model and related numerical issues for thermo-mechanical High Cycle Fatigue. *European Journal of Mechanics A/Solids 26*, 909–935.

dos Santos, H. A., F. Auricchio, and M. Conti (2012). Fatigue life assessment of cardiovascular balloon-expandable stents: A two-scale plasticity–damage model approach. *Journal of the Mechanical Behavior of Biomedical Materials 15*, 78–92.

Dowling, N. E. (2013). *Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue* (4th ed.). Upper Saddle River, NJ: Pearson.

Downing, S. and D. Socie (1982). Simple rainflow counting algorithms. *International Journal of Fatigue 4* (1), 31–40.

Dufailly, J. and J. Lemaitre (1995). Modeling very low cycle fatigue. *International Journal of Damage Mechanics 4* (2), 153–170.

Eshelby, J. D. (1957). The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proceedings of the Royal Society of London Series A 241*, 209–229.

Flaceliere, L., F. Morel, and A. Dragon (2007a). Competition between mesoplasticity and damage under HCF–Elasticity/damage shakedown concept. *International Journal of Fatigue 29* (12), 2281–2297.

Flaceliere, L., F. Morel, and A. Dragon (2007b). Coupling between mesoplasticity and damage in high-cycle fatigue. *International Journal of Damage Mechanics 16* (4), 473–509.

Fried, I. (1969). Finite-element analysis of time-dependent phenomena. *AIAA Journal 7*, 1170–1173.

Griffith, A. A. (1921). The phenomena of rupture and flow in solids. *Philosophical Transactions of the Royal Society of London 221*, 163–198.

Hertzberg, R., R. Vinci, and J. Hertzberg (2013). *Deformation and Fracture Mechanics of Engineering Materials* (5th ed.). Hoboken, NJ: John Wiley & Sons.

Ho, S.-Y. (2010). *Structural Failure Analysis and Prediction Methods for Aerospace Vehicles and Structures*, Chapter Aerothermal and Structural Dynamic Analysis of High-Speed Flight Vehicles, pp. 54–84. Bentham eBooks.

Hughes, T. J. R. and J. R. Stewart (1996). A space-time formulation for multiscale phenomena. *Journal of Computational and Applied Mathematics 74*, 217–229.

Hulbert, G. M. and T. J. R. Hughes (1990). Space-time finite element methods for second-order hyperbolic equations. *Computer Methods in Applied Mechanics and Engineering 84*, 327–348.

Hult, J. (1972). Iutam symposium gothenburg 1970. In J. Hult (Ed.), *Creep in Structures 1970*, Berlin. Springer-Verlag.

Irwin, G. (1957). Analysis of stresses and strains near to the end of crack traversing a plate. *ASME Journal of Applied Mechanics 24*, 361–364.

Jiang, H., X. Gao, and T. S. Srivatsan (2009). Predicting the influence of overload and loading mode on fatigue crack growth: A numerical approach using irreversible cohesive elements. *Finite Elements in Analysis and Design 45*(10), 675–685.

Johnson, C. (1984). Error estimates and automatic time step control for numerical methods for stiff ordinary differential equations 1984-27. Technical report, Department of Mathematics, Chalmers University of Technology and the University of Goteborg, Goteborg, Sweden.

Kachanov, L. (1958). Time of the rupture process under creep conditions. *Izvestiia Akademii Nauk SSSR, Otdelenie Teckhnicheskikh* (8), 26–31.

Karpanan, K. (2016). Critical plane search method for biaxial and multiaxial fatigue analysis. In *Proceedings of the ASME 2016 Pressure Vessels & Piping Conference*, Volume 5, pp. V005T05A011. American Society of Mechanical Engineers.

Kim, K. (2013). High-cycle fatigue simulation for aluminium alloy using cohesive zone law. *Journal of Mechanical Engineering Science 227*, 683–692.

Kim, K. and M.-J. Yoon (2014). Fretting fatigue simulation for aluminium alloy using cohesive zone law approach. *International Journal of Mechanical Sciences 85*, 30–37.

Kröner, E. (1961). On the plastic deformation of polycrystals. *Acta Metallurgica 9*, 155–161.

Krysl, P. (2005, November). *A Pragmatic Introduction to the Finite Element Method for Thermal and Stress Analysis*. Pressure Cooker Press.

Lai, W. M., D. Rubin, and E. Krempl (2010). *Introduction to Continuum Mechanics* (4th ed.). Boston, MA: Butterworth-Heinemann.

Lasaint, P. and P. A. Raviart (1974). *Mathematical Aspects of Finite Elements in Partial Differential Equations*, Chapter On a Finite Element Method for Solving the Neutron Transport Equation, pp. 89–123. Academic Press.

Lautrou, N., D. Thevenet, and J.-Y. Cognard (2009). Fatigue crack initiation life estimation in a steel welded joint by the use of a two-scale damage model. *Fatigue and Fracture of Engineering Materials and Structures 32*, 403–417.

Lee, Y.-L. and M. E. Barkey (2011a). *Metal Fatigue Analysis Handbook: Practical Problem-Solving Techniques for Computer-Aided Engineering*, Chapter Stress-Based Multiaxial Fatigue Analysis, pp. 161–213. Waltham, MA: Butterworth-Heinemann.

Lee, Y.-L. and M. E. Barkey (2011b). *Metal Fatigue Analysis Handbook: Practical Problem-Solving Techniques for Computer-Aided Engineering*, Chapter Stress-Based Uniaxial Fatigue Analysis, pp. 115–160. Waltham, MA: Butterworth-Heinemann.

Lee, Y.-L. and T. Tjhung (2011). *Metal Fatigue Analysis Handbook: Practical Problem-Solving Techniques for Computer-Aided Engineering*, Chapter Rainflow Cycle Counting Techniques, pp. 89–114. Waltham, MA: Butterworth-Heinemann.

Lemaitre, J. (1971). Evaluation of dissipation and damage in metals submitted to dynamic loading. In *International Conference on Mechanical Behavior of Materials in Kyoto, Japan.*

Lemaitre, J. (1985). A continuous damage mechanics model for ductile fracture. *Journal of Engineering Materials and Technology 107*, 83–89.

Lemaitre, J. and R. Desmorat (2005). *Engineering Damage Mechanics: Ductile, Creep, Fatigue and Brittle Failures.* Berlin, Germany: Springer Science & Business Media.

Lemaitre, J. and I. Doghri (1994). Damage 90: A post processor for crack initiation. *Computer Methods in Applied Mechanics and Engineering 115*, 197–232.

Lemaitre, J., J. Sermage, and R. Desmorat (1999). A two scale damage concept applied to fatigue. *International Journal of Fracture 97*, 67–81.

Lestriez, P., F. Bogard, J. L. Shan, and Y. Q. Guo (2007, May). Damage evolution on mechanical parts under cyclic loading. In *AIP Conference Proceedings*, Volume 908, pp. 1389–1394. American Institute of Physics.

Li, X. and N. Wiberg (1996). Structural dynamic analysis by a time-discontinuous Galerkin finite element method. *International Journal for Numerical Methods in Engineering 39*, 2131–2152.

Li, X. and N. Wiberg (1998). Implementation and adaptivity of a space-time finite element method for structural dynamics. *Computer Methods in Applied Mechanics and Engineering 156*, 211–229.

Lu, J., W. Sun, A. Becker, and A. A. Saad (2015). Simulation of the fatigue behaviour of a power plant steel with a damage variable. *International Journal of Mechanical Sciences 100*, 145–157.

Manson, S. (1953). Behaviour of materials under conditions of thermal stresses. techreport 2933, NACA.

Martin, C. and W. Sun (2015). Comparison of transcatheter aortic valve and surgical bioprosthetic valve durability: A fatigue simulation study. *Journal of Biomechanics 48*, 3026–3034.

Matsuishi, M. and T. Endo (1968). Fatigue of metals subjected to varying stress. In *Proceedings of the Kyushu Branch of Japan Society of Mechanical Engineers*, Fukuoka, Japan, pp. 37–40. Japan Society of Mechanical Engineers.

Melenk, J. M. and I. Babuska (1996). The partition of unity finite element method. *Computer Methods in Applied Mechanics and Engineering 139*, 289–314.

Milne, I. (1994). The importance of the management of structural integrity. *Engineering Failure Analysis 1*(3), 171–181.

Miner, M. (1945). Cumulative damage in fatigue. *Journal of Applied Mechanics 67*, A159–A164.

Moës, N., J. Dolbow, and T. Belytschko (1999). A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering 46*, 131–150.

Mohanty, S., S. Majumdar, and K. Natesan (2012, June). A review of stress corrosion cracking/fatigue modeling for light water reactor cooling system components. techreport, Argonne National Laboratory, Argonne, IL.

Murakami, S. (2012). *Continuum Damage Mechanics*, Volume 185 of *Solid Mechanics and its Applications*. New York, NY: Springer.

Oden, J. T. (1969). A general theory of finite elements. II. Applications. *International Journal for Numerical Methods in Engineering 1*, 247–259.

Oller, S., O. Salomón, and E. Oñate (2005). A continuum mechanics model for mechanical fatigue analysis. *Computational Materials Science 32*, 175–195.

Palmgren, A. (1924). Die lebensdauer von kugellagern, (The service life of ball bearings). *Zeitschrift des Vereinesdeutscher Ingenierure 68*(14), 339–341.

Papadopoulos, I. V., P. Davoli, C. Gorla, M. Filippini, and A. Bernasconi (1997). A comparative study of multiaxial high-cycle fatigue criteria for metals. *International Journal of Fatigue 19*(3), 219–235.

Papuga, J., M. Vargas, and M. Hronek (2012). Evaluation of uniaxial fatigue criteria applied to multiaxially loaded unnotched samples. *Engineering Mechanics 19*(2), 99–111.

Paris, P., M. Gomez, and W. Anderson (1961). A rational analytic theory of fatigue. *The Trend in Engineering 13*, 9–14.

Pirondi, A., N. Bonora, D. Steglich, W. Brocks, and D. Hellmann (2006). Simulation of failure under cyclic plastic loading by damage models. *International Journal of Plasticity 22*(11), 2146–2170.

Qian, D. and S. Chirputkar (2014). Bridging scale simulation of lattice fracture using enriched space-time Finite Element Method. *International Journal for Numerical Methods in Engineering 97*, 819–850.

Rabotnov, Y. N. (1969). *Creep Problems in Structural Members*. North-Holland Series in Applied Mathematics and Mechanics. Amsterdam: North-Holland.

Raje, N., T. Slack, and F. Sadeghi (2009). A discrete damage mechanics model for high cycle fatigue in polycrystalline materials subject to rolling contact. *International Journal of Fatigue 31*(2), 346–360.

Reed, R., J. Smith, and B. Christ (1983). *The Economic Effects of Fracture in the United States*, Volume 647. Washington, D.C.: US Department of Commerce, National Bureau of Standards.

Reed, W. and T. Hill (1973). Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory.

Roe, K. L. and T. Siegmund (2003). An irreversible cohesive zone model for interface fatigue crack growth simulation. *Engineering Fracture Mechanics 70*, 209–232.

Safe Technology Ltd. (2002). *fe-safe Fatigue Theory Reference Manual*. Sheffield, UK: Safe Technology Ltd.

Santos, J. L., M. De Freitas, B. Li, and T. Trigo (2003). Fatigue assessment of mechanical components under complex multiaxial loading. In *European Structural Integrity Society*, Volume 31, pp. 463–482. Elsevier.

Schijve, J. (2009). *Fatigue of Structures and Materials* (2nd ed.). Dordrecht, Netherlands: Springer.

Siegmund, T. (2004). A numerical study of transient fatigue crack growth by use of an irreversible cohesive zone model. *International Journal of Fatigue 26*(9), 929–939.

Sines, G. (1959). *Metal Fatigue*, Chapter Behavior of metals under complex static and alternating stresses, pp. 145–169. New York: McGraw-Hill.

Sun, C. and Z.-H. Jin (2012). *Fracture Mechanics*. Waltham, MA: Academic Press.

Suresh, S. (1998). *Fatigue of Materials* (2nd ed.). Cambridge, U.K.: Cambridge University Press.

Sziroczak, D. and H. Smith (2016). A review of design issues specific to hypersonic flight vehicles. *Progress in Aerospace Sciences 84*, 1–28.

Takagaki, M. and T. Nakamura (2007). Fatigue crack modeling and simulation based on continuum damage mechanics. *Journal of Pressure Vessel Technology 129*(1), 96–102.

Tchoupou, K. M. T. and B. D. S. Fotsing (2015). Fatigue equivalent stress state approach validation in non-conservative criteria: A comparative study. *Latin American Journal of Solids and Structures 12*(13), 2506–2519.

Wang, C. H. and M. W. Brown (1996). Life prediction techniques for variable amplitude multiaxial fatigue – Part 1: Theories. *Journal of Engineering Materials and Technology 118*, 367–370.

Welsch, G., R. Boyer, and E. Collings (1994). *Materials Properties Handbook: Titanium Alloys*, Chapter Ti-6Al-2Sn-4Zr-2Mo-0.08Si, pp. 337–362. Materials Park, OH: ASM International.

Wöhler, A. (1860). Versuche über die festigkeit der eisenbahnwagenachsen. *Zeitschrift für Bauwesen 10*, 160–161.

Yang, Y., S. Chirputkar, D. N. Alpert, T. Eason, S. Spottswood, and D. Qian (2012). Enriched space-time finite element method: A new paradigm for multiscaling from elastodynamics to molecular dynamics. *International Journal for Numerical Methods in Engineering 92*, 115–140.

Zhang, R., L. Wen, S. Naboulsi, T. Eason, V. K. Vasudevan, and D. Qian (2016). Accelerated multiscale space-time finite element simulation and application to high cycle fatigue life prediction. *Computational Mechanics 58*(2), 329–349.

BIOGRAPHICAL SKETCH

After receiving his bachelor's degree from Texas Tech University in 2006, Ryan T. Schlinkman worked as a petroleum engineer for nine years until being laid-off in the 2014-2015 oil & gas downturn. He decided that he had had enough of the wild economic swings of the O&G industry and decided to get out and start a new career as a mechanical engineer with a focus on simulation and finite element analysis. He returned to school to earn his master's degree at The University of Texas at Dallas, and post-graduation, he will be moving on to start the PhD in Structural Engineering program at the University of California, San Diego.

CURRICULUM VITAE

Ryan T. Schlinkman

July 15, 2019

Contact Information:

Department of Mechanical Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.
Email: rts160030@utdallas.edu

Educational History:

BS, Mechanical Engineering, Texas Tech University, 2006

Employment History:

Computational Sciences Intern, Engility Corp. / DOD HPCMP, May 2018 – August 2018
Mechanical Design Engineer Intern, Smart Start, Inc., May 2017 – July 2017
Project Engineer / Analyst, Phoenix PetroCorp, Inc., August 2006 -August 2015

Professional Recognitions and Honors:

Phi Kappa Phi Honor Society, The University of Texas at Dallas, 2018
Tau Beta Pi Engineering Honor Society, Texas Tech University, 2006
Golden Key International Honour Society, Texas Tech University, 2006
National Society of Collegiate Scholars, Texas Tech University, 2006
Phi Sigma Theta Honor Society, Texas Tech University, 2006
Graduated *cum laude*, Texas Tech University, 2006

Professional Memberships:

American Society of Mechanical Engineers (ASME), 2015 – present