Aalto-yliopisto
Kemian tekniikan
korkeakoulu

School of Chemical Technology
Degree Programme in Chemical, Biochemical and Materials Engineering

Lauri Vallin

The Simulation of Continuous Casting of Steel Using Finite Element Method

Master's thesis for the degree of Master of Science in Technology submitted for inspection, Espoo, 5.12.2016.

Supervisor **Ari Jokilaakso**

Instructor **Ari Kruskopf**

Aalto University
School of Chemical
Technology

Aalto University, PO. BOX 11000, 00076 AALTO
www.aalto.fi

Abstract of master's thesis

Author Lauri Vallin

Title of thesis The Simulation of Continuous Casting of Steel Using Finite Element Method

Department Department for Materials Science

Thesis supervisor Ari Jokilaakso

Thesis advisor(s) / Thesis examiner(s) Ari Kruskopf

Date 5.12.2016 **Number of pages** 94 **Language** English

Abstract

Most steel production lines include continuous casting at some point. The steel manufacturers are constantly searching for solutions to give them a competitive edge and decrease production costs. Simulating the continuous casting process can reveal risks of cracking in the process and give other crucial information about the process parameters without expensive and time-consuming pilot testing. Simulating the crack initiation requires modeling of two phenomena; heat transfer in the molten and solidified steel and mechanical strains and stresses in the solidified shell. The purpose of this study was to simulate the behavior of both the thermal and mechanical aspects of the continuous casting process to enable the prediction of crack initiation in the billet to avoid the economic losses caused by a rupture in the billet during the casting process.

The literature study presents the mathematical basis of the finite element formulation for the simulation of continuous casting, and presents different methods of simulation presented in literature. The literary study also includes an in-depth explanation of equations used to model the thermal and mechanical behavior of steel in the continuous casting process. This foundation was used to create a Matlab code, which was used to simulate the continuous casting process, and the results of the code were compared with similar models in literature. The crack initiation analysis predicted correctly the highest risk of hot tear cracking to the phase transition front. The heat transfer model was determined to function well in comparison to other simulations previously performed in literature despite being less sophisticated. The elastic model functioned well, but the plastic material model could not be finished within the scope of this work. The simulation program shows promise but has to be further improved before industry users can adopt it.

Keywords Continuous Casting, Steel, Simulation, Finite Element Method

Aalto-yliopisto
Kemian tekniikan
korkeakoulu

Aalto-yliopisto, PL 11000, 00076
AALTO
www.aalto.fi

Diplomityön tiivistelmä

Tekijä Lauri Vallin

Työn nimi Teräksen Jatkuvavalun Mallintaminen Elementtimenetelmällä

Laitos Materiaalitekniikan laitos

Työn valvoja Ari Jokilaakso

Työn ohjaaja(t)/Työn tarkastaja(t) Ari Kruskopf

Päivämäärä 5.12.2016 **Sivumäärä** 94 **Kieli** Englanti

Tiivistelmä

Suurin osa teräksen tuotannosta perustuu jatkuvavaluun. Teräksen valmistajat etsivät jatkuvasti uusia ratkaisuja, jotka antaisivat heille kilpailullisen edun tai laskisivat kustannuksia. Jatkuvavaluprosessin simulointi voi paljastaa halkeamisriskin ja antaa muuta olennaista tietoa prosessin parametreistä ilman kalliita tehdaskokeita. Halkeamien syntymisen ennustaminen vaatii kahden olennaisen ilmiön mallintamista; sulan ja jähmettyneen teräksen lämmönsiirron ja jähmettyneen teräksen mekaanisten jännitysten ja venymien mallintamisen. Tämän työn tarkoitus oli simuloida sekä valukappaleen lämmönsiirtoa että mekaanisia ominaisuuksia jatkuvavaluprosessissa, jotta halkeamien syntymistä voitaisiin mallintaa. Halkeama valukappaleessa voi johtaa koko valuprosessin pysäyttämiseen, mikä voidaan luotettavalla halkeamien syntymisen ennustamisella välttää.

Kirjallisuusosiossa esitetään matemaattinen perusta elementtimenetelmän käyttöön jatkuvavalun mallintamisessa, sekä eri lähestymistapoja jatkuvavalun mallintamiseen. Kirjallisuusosio sisältää myös lämmönsiirron ja mekaanisen käyttäytymisen mallintamiseen käytettyjen kaavojen tarkastelun, mitä käytettiin perustana Matlab koodin luomiseen. Matlab koodilla mallinnettiin jatkuvavalua, ja tuloksia verrattiin samankaltaisen mallin tuloksiin kirjallisuudesta. Matlab ohjelma ennustaa oikein halkeamien syntymisen suurimman riskin faasimuutosrintamaan. Lämmönsiirtomallin todettiin toimivan hyvin kirjallisuuteen verrattuna, vaikka se onkin yksinkertaisempi. Mallin mekaanisen osuuden elastisuuden mallinnus toimi hyvin, mutta plastisen käyttäytymisen mallinnusta ei voitu viimeistellä tämän työn aikarajoitusten puitteissa. Matlabohjelma toimii lupaavasti, mutta se vaatii vielä parannuksia ja korjauksia ennen kuin sitä voidaan hyödyntää teollisuudessa.

Avainsanat Jatkuvavalu, Teräs, Mallinnus, Elementtimenetelmä

Contents

Nomenclature

		Unit
[**x**]	Matrix (any variable can replace x)	
$[]^T$	Matrix transpose	
x	Column vector	
$\mathbf{x}^T$	Row vector	
x_y	Component of vector: the component direction is in the subscript.	
∇	Gradient operator	
$\dot{x}$	Time derivative of a variable	
x^e	The variable applies to a single element	
b	Bodyforces	N
[**B**]	Spatial derivatives of the field variables, strain-displacement matrix	
c_p	Heat capacity	J/(kgK)
[**C**]	Heat capacity mass matrix	
d, **u**	Vector of displacements in an element, Vector of global displacements	m
[**D**]	Stress-Strain matrix	
E	Young's modulus	Pa
F, **f**	Global force vector, element force vector	N
g	traction forces	N
h, h_{ef}	Heat transfer coefficient of convection, effective heat transfer coefficient in radiation	W/m^2K
H, [**H**]	Enthalpy, Enthalpy mass matrix	J
i	Iteration step index	
[**I**]	Unit matrix (also called identity matrix)	
[**J**], |J|	Jacobian matrix, Determinant of the Jacobian matrix	

k, [**κ**]	Heat conductivity, heat conductivity matrix	W/(mK)
[**K**],[**k**]	Global stiffness matrix, element stiffness matrix	
$[\mathbf{L}]$	Derivative operator	
$\bar{n}$	Surface unit vector	
N, [**N**]	Shape functions, Shape function matrix	
p	Thickness of the cross section in z direction.	m
q	Heat flux	W/m^2
Q	Heat source term	
r, **R**	Element and Global force vector in heat conduction	
t, Δt	time, time-step	s
T	Temperature in Kelvin	K, °C
v	Poisson's ratio	
V	Volume of the domain	m^3
x, y	Spatial coordinates	
X	Recrystallized portion of the element	

Greek letters

δ	Kronecker delta	
ε, **ε**	Strain, strain vector	
ε_{rela}	Relative permittivity	
ξ, η	Isoparametric coordinates	
σ, **σ**	Stress, stress vector	Pa
σ_{SB}	Stefan-Boltzmann constant	
Γ	Boundary of the domain	
ρ	Density	kg/m^3
φ, w	Weight functions	

1. Introduction

Steel manufacturers are constantly in the search of new methods to gain an edge in the competitive market. One method to increase productivity, which is getting more and more interest [1], is to optimize the production line using modeling. This way the cooling of the process can simultaneously be optimized for best possible temperature distribution for hot rolling, and hot cracking and various other harmful phenomena can be avoided when the casting process is modeled prior to casting – or in the case of continuous casting, modeled in real time [2, 3].

During steel continuous casting process, a crack in the solidifying shell can lead to significant economic losses and even danger to factory personnel as molten steel pours out of the crack in the shell. Cracks in the shell are not the only point of interest in continuous casting when modeling, however, as both continuous casting machine producers and users are interested in quality optimization, which means increasing the casting speed and decreasing the production costs. In addition, further expanding the feasible alloying options and improving the alloying accuracy, and more accurate control of the segregation of the elements are all areas of interest.

The aim of this study was to create a Matlab code, which is able to predict the crack initiation in the continuously casted steel billet. Predicting the crack initiation can help avoid economic losses, which would be caused by a rupture in the billet. The prediction of cracks requires the simulation of the thermal and mechanical behavior of the continuously casted steel billet. This work was meant to supplement the previous studies on the heat transfer and microstructure development of continuous casting [4-6] with a more dedicated focus on the mechanical analysis of continuous casting. Matlab was chosen as the code environment due to its accessibility and the possibility to easily adapt sections of the working final model into other already existing continuous casting models.

The modeling of the mechanical behavior includes the stresses and strains caused by the ferrostatic pressure of the molten cone within the billet, the supporting rolls, and gravity. In addition, creep, thermal strains and strains caused by phase transitions are also taken into account. The computational efficiency of the model is not treated in this study, only the accuracy and the reliability of the model.

2. Continuous Casting of Steel

In this chapter, the basics of continuous casting and the various phenomena associated with it are shortly explained.

In continuous casting, molten steel is poured into a bottomless mold, where it solidifies as in Figure 1. The caster can be curved as in Figure 1, or it can be vertically or horizontally straight – this work focuses on the curved set ups. The cooling can be divided into primary cooling in the mold, secondary cooling through the use of water sprays, and tertiary cooling, where the billet cools due to airflow and radiation without purposeful cooling [7].

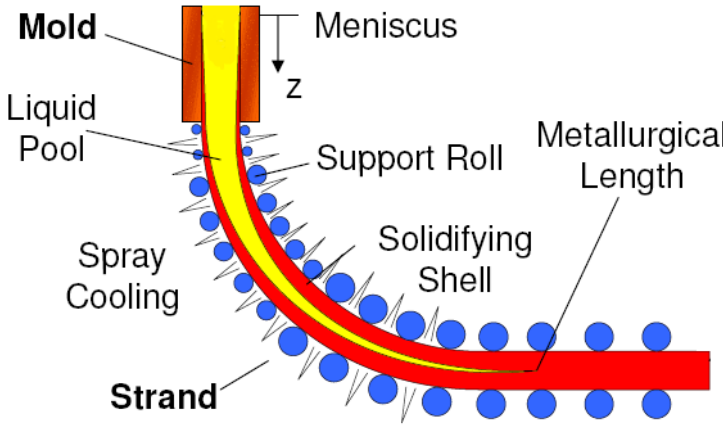

Figure 1 Continuous casting process [8].

The mold is water cooled and its purpose is to cool the molten steel rapidly enough for it to have a solid shell below the mold. In the secondary cooling zone below the mold, water jets and radiation further cool the shell. Other heat transfer during the process are the conduction of heat in the shell from the solid-liquid interface into the boundary where the water jets cool the shell, and the convection within the steel melt cone, the length of which can be 10-20 meters below the mold depending on steel composition and the billet geometry. Heat transfer in the direction of the casting is minimal in relation to other directions.

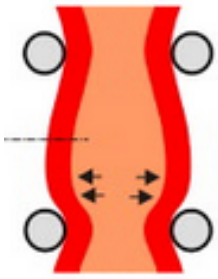

Figure 2 Concept image of bulging due to ferrostatic pressure between the rolls [9].

Below the mold, the billet is supported by rolls, which prevent excessive bulging of the shell due to ferrostatic pressure as in the concept image in Figure 2 [9]. Due to the fact that the billet

moves in relation to the rolls, the maximum bulge is after the halfway point between two rolls [9].

Stresses and strains are generated during casting due to five primary factors: the cooling of the billet generates thermal strains, bulging of the shell between the supporting rolls as in Figure 2, bending of the billet in curved casting machines, creep, and phase transitions. Because parts of the billet are relatively close to the melting point during most of the casting, the yield stress is very low, which in turn leads to plastic response. High temperatures cause the creep to be extremely rapid, which together with plastic material behavior dictate the behavior of the billet during the casting.

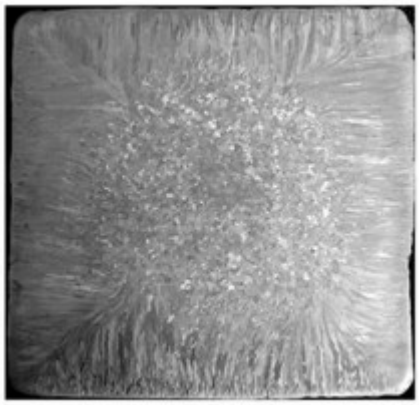

Figure 3 Cross section of a cast billet [10].

The microstructure of the billet consists of columnar grains near the edges of the billet, and equiaxed grains in the center, such as in Figure 3. Microstructure of the billet can also be affected by segregation, which means an uneven distribution of elements in the billet. Segregation can be caused by bulging or solidification shrinkage in the late stages of solidification, in which case it is called centerline segregation [10]. Another cause for segregation is the different solubility of elements in liquid and solid phases; this can be problematic especially in the mushy zone in the interface between the solid and liquid phase [7].

3. Literary Review

This chapter includes the theoretical background for the modeling of continuous casting as well as a short summary of how continuous casting has been modeled in the past.

The structure of this chapter is as follows: first the finite element method is summarized shortly, followed by an overview of literary works concerning the modeling of continuous casting. After these sections, the heat conduction and elasticity problems are discretized into finite element form. The treatment for multiple phases and plasticity are in the sections for temperature and elasticity modeling, respectively.

3.1 The Finite Element Method

The finite element method (FEM) is an efficient numerical method for solving physical problems in many different scientific fields, including but not limited to geology, aeronautical engineering, civil and mechanical engineering and metallurgy. The treatment in this chapter is limited to the point of view of metallurgy, specifically the modeling of heat transfer and stresses and strains.

FEM is based on the idea that the domain of interest is divided into small elements, which consist of nodes where the variables of interest (temperature, displacement, stress or strain) are approximated. Mathematically, FEM is used to solve a partial differential equation and will include all the assumptions of this said model [11]. Hence, the choice of an appropriately accurate mathematical model is necessary.

The steps of FEM analysis is the choice of mathematical model, the discretization of that model into the form solvable by FEM, and the application of FEM into that form of the model [11]. The application itself can still be divided into several sub steps: the division of the domain into elements (1), forming of the governing equations for the elements (2), assembling the equations into a system of equations (3), solving the equations (4) and post processing (5) [11].

Only the basic principles of FEM are explained in this section. In finite element method, the domain of interest is divided into small elements, as mentioned before. The properties of these elements, such as temperature, enthalpy, Young's modulus and so on, are calculated or known at the nodes, but they can be interpolated anywhere in the element. These nodes are always

present in the corners of the element, and depending on the element type used, can also be present at the edges and within the element [12]. A simplified element grid is presented in Figure 4, where a quadrilateral domain (in blue) is divided into four quadrilateral elements. This and the following sections will always use quadrilateral elements as examples as they were used in actual modeling in section 4. The circles highlight the nodes at the corners of the element. The corner nodes of the mesh only belong to one element. The other boundary nodes belong to two elements and the nodes within the domain belong to four elements, contributing to the properties of all of them.

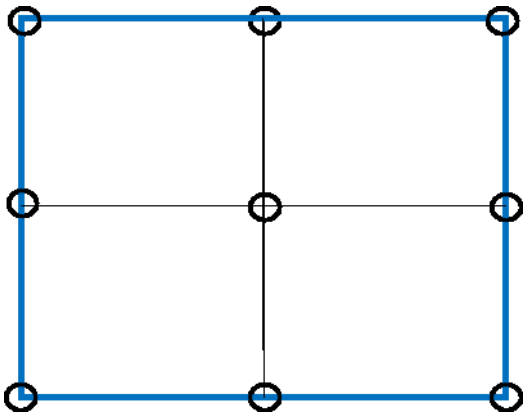

Figure 4 A simplified element grid with four quadrilateral elements and four nodes in each element.

Above in Figure 4 the elements are symmetric rectangular shapes: this is not a requirement and in practice there is very little limitation on the shape of the elements. However, the elements are more accurate the closer their height and width dimensions are to each other – that means that square elements are the most accurate two dimensional quadrilateral elements and equilateral triangles would be the most accurate triangular elements [12, 13].

When modeling linear phenomena, such as small displacements or strains in elastic range or the conduction of heat, FEM practically means solving a system of linear equations. When modeling includes more complicated phenomena, such as heat conduction with phase change or radiation at the boundaries, the equations are no longer linear and the solutions for the equations have to be iterated. These iteration schemes will be shortly explained in the following sections when iteration is necessary.

Finite element method usually includes an error when compared to analytical answer on itself, despite the fact that in many problems the analytical answer can be reached in practice using FEM. The possible sources of error and inaccuracies are summarized here.

Modeling errors are differences between the physical system and the mathematical model [12]. These are difficult to avoid completely as simplifications in mathematical model always include some modeling errors. In this model, plane strain assumption in continuous casting could be called modeling error, as in reality the billet is not infinite in length as the plane strain assumes.

Discretization error means the error caused by representing the infinite number of degrees of freedom in continuous mathematical or physical model as a finite number of degrees of freedom in FEM [12].

The material parameters used will always include errors, as they are fit to some experimental data. The experiments themselves will include error, and as the results of those experiments are generalized into a linear relationship, for example, the generalization will introduce additional error.

Round-off error is caused by the fact that computer programs treat only a finite number of decimals [12]. Double accuracy primarily used in this work, for example, saves numbers with 16 decimal places. The round-off error is usually orders of magnitudes smaller than any other error in modeling.

Two additional error types exist, and those are derived from the previous three: Inherited error is the sum of previous discretization and round-off errors at any point during computation, and manipulation error refers to round-off errors during an algorithm [12]. Though round-off errors are small, manipulation errors can be significant during computations. During FEM modeling, inherited error can be significant when adding the thermal conductivities or stiffness's of the elements into global matrix, and manipulation errors are produced by solving the linear systems of equations [12].

A few conclusions about the errors in finite element analysis, the derivation of which can be found elsewhere [12]: the error in displacements, or temperatures, is the smallest near the nodes and the error in stresses or strains is the smallest within the element near the Gauss integration points (appendix) [12]. In addition, singularities can and will occur in elastic and

especially in plastic analysis near corners and crack tips – these singularities usually manifest as infinitely increasing stresses as the element size is decreased [12]. However, usually these singularities do not prevent convergence towards the correct solution [12].

3.2 The Modeling of Continuous Casting

This chapter includes a short summary of how continuous casting has previously been simulated in literature. The nature of continuous casting itself is not treated in detail; the focus of this chapter is on the modeling of the continuous casting process.

Continuous casting includes various complicated phenomena, which make it difficult to model. These include gas bubbles in the melt, slag-steel interactions, dissolved elements, bending and straightening, material behavior near melting point and the turbulent flow caused by the entry nozzle [1, 14, 15]. The modeling of all of these and other phenomena in a single model is extremely challenging and not even necessary. It is common in literature to focus on few phenomena of interest and significantly simplify or even completely ignore the rest [1]. What follows is a short summary of the simplifications used and practical choices made in literature.

Complete 3D models of the continuous casting process are rare in literature, though not non-existent. Such treatments are often more focused on the computational speed of the modeling than on the actual model itself, such as the model of Klime et al [16], which focused on using GPU resources in modeling in addition to CPU. Another 3D example is the model by Si et al [17], where they treated the molten steel with a finite difference model and the solid shell with a finite element method.

2D models of continuous casting are far more common, and they can be divided into two categories: slice models as in Figure 5, and side-view models as in Figure 6 [2]. In the slice method, a cross-section of the billet, or a small part of it, is treated as it moves through the process. This point of view is enabled by the fact that thermal gradients are negligible in the casting direction when compared to other directions [2]. This methodology, however effective in thermal analysis, prevents accurate handling of bulging [2]. The slice method applies the Lagrangian frame of reference, as the FEM grid moves along with the object of interest.

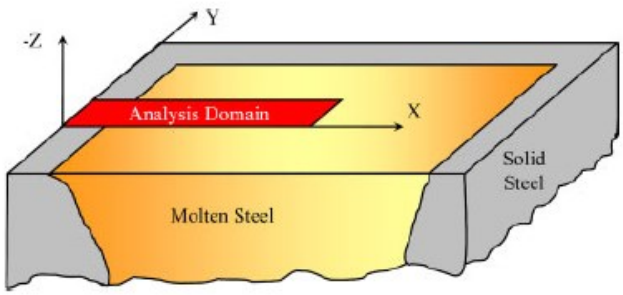

Figure 5 Concept image of the Cross section method, or Lagrangian frame of reference [18]. The analysis domain highlighted in red can only be a small area as in this figure or cover the whole cross section, depending on the focus of the study.

Models of the slice method often focus in the mold area or the primary cooling section, and such work has been performed by Koric et al [18], Heung Nam Han et al [19] and Li et al [3, 26], just to name a few.

The other point of view, the side view, applies the Eulerian frame of reference, where the object of interest moves in relation to the grid as in Figures 6 and 7. This methodology enables more accurate modeling of the bulging of the billet between the supporting rolls [2]. On the other hand, the computational expense is far greater as there are far more elements than in the slice method.

Figure 6 shows how the modeled domain expands as a function of time when simulating the beginning of continuous casting. In the beginning (t=0 min in Figure 6) the molten steel is poured into the continuous casting machine, where its expansion is limited by a nominal casting speed. As the casting process progresses, the billet cools and solidifies and after 24 minutes a significant portion of the billet is completely solid. In a steady state process, however, there will be a molten cone present within the billet up to 2/3 of the length of the casting machine, depending on the casting machine and steel grade produced.

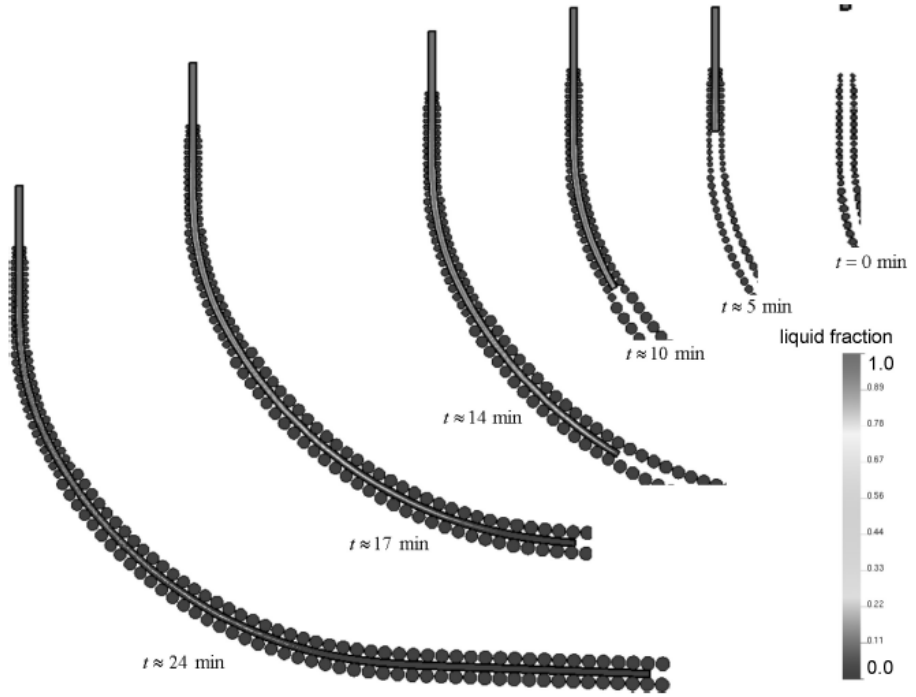

Figure 6 An example image of a side view model [2]. This view was from the work of Bellet et al, where they created a model capable of simulating non-steady state situations, such as the beginning of modeling in this figure.

Figure 7 is a more detailed image of a situation similar to in Figure 6. The molten cone is visible as a pressure free zone in the middle of the billet. The pressure (and the stress) is the highest under the rolls and in the mushy zone.

Models which have used this point of view include Fachinotti et al [20], which focused on the modeling of macro segregation and the solidification of binary alloys in addition to studying the mushy region during the secondary cooling and how the bulging effects the mushy region. Kelly et al [21] studied the mold and the interactions between the mold and the billet, including the effects of the air gap between them. Others who used this method include Janik et al [22] and Kajitani et al [23].

Phenomena which are caused by the contact between the mold and the billet include friction and the strains which it causes, possible imperfections in the shell and non-uniform temperature gradients which may lead to a different temperature histories for the shell, which can lead to different shell thicknesses in different locations [1, 14, 19, 24]. The air gap between the shell

and the mold due to the shrinkage of the billet and the formation of the gap has been simplified in various methods and it has also been the primary focus of some studies [21]. Many studies use special boundary conditions to account for the imperfect contact between the mold and the billet [18, 24], and some authors have taken the mold flux film into account through thermal resistance calculations [19].

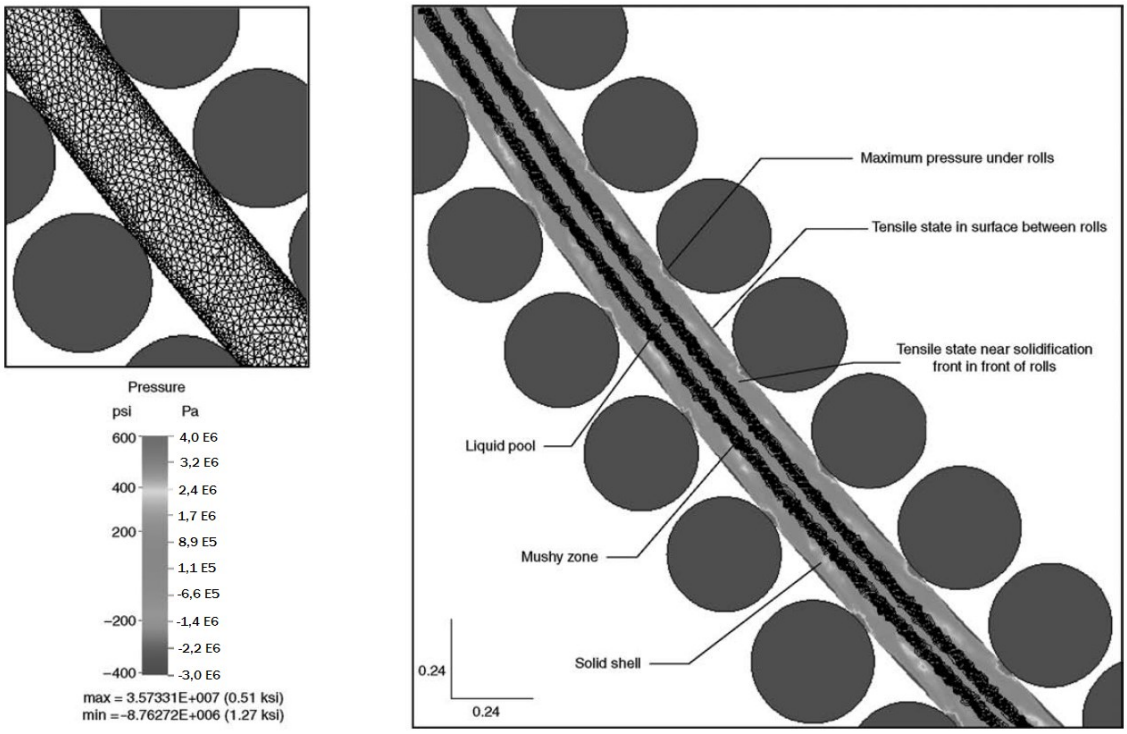

Figure 7 An example of a side view model. This view is from the work of Brian et al [25], in which they introduced a thermomechanical model capable of hot tearing analysis. The elements are extremely small, as can be seen in the upper left image.

The supporting rolls are usually treated as boundary conditions in mechanical analysis, though some authors have modeled the behavior of the rolls as well. Janik et al [22], for example, modeled the rolls, which enabled more accurate modeling of the roll-slab contact and the stresses that developed in the shell during casting [22]. In thermal analysis, the contact with the supporting rolls and its effect on the cooling can be omitted [2], though detailed work has been performed in how to include that into the continuous casting models [26].

The density of steel is generally assumed constant to maintain constant mass in thermal modeling [3]. In mechanical analysis, some change in density is taken into account in the form of thermal linear expansion [3, 11]. Poisson's ratio is usually assumed constant [3, 23, 24].

The liquid elements can be ignored completely [2, 27], but that is not the only possible course of action. The molten elements are sometimes given a certain yield strength (such as 0.01 MPa) and elastic modulus, (such as 14 GPa) to enable some crude form of fluid flow modeling [3, 27]. In some cases, extremely rapid creep is installed into the liquid elements to enforce negligible liquid strength [3, 27]. This rapid creep treatment enables the simplified modeling of the liquid flows, making the prediction of hot-tear cracks more accurate [27].

Usually the models in literature do not take microstructure into account, i.e. they do not model the forming of dendritic arms during solidification. This leads to a variety of necessary simplifications. For example, "solid material" is not entirely straightforward definition, as the solidifying dendrite arms start to touch each other when the solid fraction reaches 70 %, which is sometimes used as the definition for solid when determining if the element is treated as liquid or solid [3, 27]. This method, obviously, requires the code to treat the different phases as fractions. However, simple definition of solid through temperature, solidus and liquidus is still more common than this method. In addition, despite the pillar-like structure of the billet or slab, the material is generally assumed to be isotropic [18, 27].

In addition to these, a number of other phenomena, such as thermal distortion of the mold, solidification of the shell in non-equilibrium and non-uniform removal of the super heat from the solid-liquid interface [1, 25, 27] and still other related aspects have been studied, but they are well beyond the scope of this study.

3.3 The Modeling of Heat Transfer

This chapter includes the FEM formulation for temperature, a short discussion on the treatment of phase change and the boundary conditions used in literature.

A transient heat conduction problem can be expressed as differential equation

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) \tag{1}$$

where T is temperature, t is time, ρ is density, c_p is the heat capacity and k the thermal conductivity. This equation is multiplied by scalar function φ and integrated over the volume V in equation (*2*) [12].

$$\int \rho c_p \frac{\partial T}{\partial t} \varphi \, dV = \int \varphi \nabla \cdot (k \nabla T) \, dV \tag{2}$$

The right hand side term is reformulated in equation (*3*).

$$\int \rho c_p \frac{\partial T}{\partial t} \, dV = \int \nabla \cdot (\varphi \mathrm{k} \nabla T) \, dV - \int \nabla \varphi \cdot k \nabla T \, dV \tag{3}$$

The first term on the right hand side is changed into a surface integral over surface Γ with Gauss theorem [12]. The $\bar{\mathrm{n}}$ in equation (*4*) is the surface unit normal vector.

$$\int \rho c_p \frac{\partial T}{\partial t} \varphi \, dV = \int \bar{\mathrm{n}} \cdot (\varphi \mathrm{k} \nabla T) \, d\Gamma - \int \nabla \varphi \cdot k \nabla T \, dV \tag{4}$$

Here the surface integral over Γ represents different boundary conditions. If the temperature is known at the boundary (known as Dirichlet boundary condition), the surface integral simply vanishes [12], which is assumed at this point. The other boundary conditions are treated later.

Now the heat equation becomes

$$\int \rho c_p \frac{\partial T}{\partial t} \varphi \, dV = - \int \nabla \varphi \cdot k \nabla T \, dV \, . \tag{5}$$

Now we discretize the equation by inserting polynomial approximations in equation (*6*) into the equations to replace the scalar function φ. At this step the domain V is approximated by n elements.

$$\begin{aligned} \varphi &= N_j \\ T &= \sum_{i=1}^{n} N_i \, T_i \\ k &= \sum_{i=1}^{n} N_i \, k_i \end{aligned} \tag{6}$$

Here the shape functions N_i are a function of the spatial coordinates only. Isoparametric elements are generally used in the actual modeling. Isoparametric elements and their use is explained in the appendix. Inserting equations (*6*) into equation (*5*), we get

$$\sum_{i=1}^{n} \int \rho c_p \frac{\partial N_i T_i}{\partial t} N_j \, dV = - \sum_{i=1}^{n} \int \nabla N_j \cdot (\mathrm{N_i k_i} \nabla N_i T_i) \, dV. \tag{7}$$

The time derivative on the left hand side of equation can be treated in many different ways. These are well documented in many sources [2, 12]. In this case implicit time integration is used, as in equation (*8*). Here the superscript *n+1* refers to the new time step where temperatures are to be solved and n the values of the previous time step where values are known. Δt is the time difference between the two time steps.

$$\frac{\partial T_i}{\partial t} = \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{8}$$

By inserting the Implicit Euler backward integration in (*8*) to (*7*) we get equation (*9*).

$$\sum_{i=1}^{n} \int \rho c_p \frac{T_i^{n+1} - T_i^n}{\Delta t} N_i N_j \, dV = -\sum_{i=1}^{n} \int \nabla N_j \cdot \left(\mathrm{N_i k_i} \nabla N_i T_i^{n+1}\right) dV \tag{9}$$

Let us move the unknown temperatures T_i^{n+1} to the left hand side and the known T_i^n to the right hand side.

$$\left(\sum_{i=1}^{n} \int \rho c_p \frac{1}{\Delta t} N_i N_j \, dV + \sum_{i=1}^{n} \int \nabla N_j \cdot (\mathrm{N_i k_i} \nabla N_i) \, dV\right) T_i^{n+1} = \sum_{i=1}^{n} \int \rho c_p \frac{T_i^n}{\Delta t} N_i N_j \, dV \tag{10}$$

Equation (*10*) can also be marked using matrix notation as

$$\left(\frac{1}{\Delta t}[\boldsymbol{C}] + [\boldsymbol{K}]\right) \boldsymbol{T}^{n+1} = \frac{1}{\Delta t}[\boldsymbol{C}] \boldsymbol{T}^n \tag{11}$$

which will primarily be used in this work. Here $[\boldsymbol{C}]$ is called mass matrix, and $[\boldsymbol{K}]$ is called the stiffness matrix.

Previously the boundary term was ignored as it was assumed to be zero, as it does in the case of Dirichlet boundary condition – which means that the temperature is known and prescribed at the boundary. Here the other boundary conditions are discretized into Finite Element form, such as equation (*11*). The dependency of heat conduction coefficient k on temperature is not taken into account in these treatments to enable clearer notation.

Neumann boundary condition specifies a known heat flux q over the boundary Γ in equation (*12*).

$$-(k\nabla T) = q \tag{12}$$

As before, let us multiply with scalar function φ and integrate over the boundary Γ.

$$\int -k(\varphi\nabla T)\, d\Gamma = \int \varphi q\, d\Gamma \tag{13}$$

Discretizing with shape functions, equation (*13*) becomes

$$\int \varphi q\, d\Gamma = \int N_j q\, d\Gamma. \tag{14}$$

Using matrix notation equation (*14*) becomes

$$-\int N_j q\, d\Gamma = \int \boldsymbol{N} q\, d\Gamma. \tag{15}$$

Convection boundary condition, which is also called Robin boundary condition, means that the structure of interest is either cooled or heated through a medium, such as water or air. The efficiency of convection boundary condition is dictated by both the convective heat transfer coefficient h, and the temperature difference between the medium T_{ext} and the temperature of the boundary T_i. Convective heat transfer adds terms both on the right hand and left hand sides of equation (*11*), as it has an effect both on the stiffness matrix of the element and on the source vector [12].

Robin boundary condition specifies a convective heat transfer boundary (*16*).

$$\bar{n} \cdot (k\nabla T) = h(T - T_{ext}) \tag{16}$$

Multiplying with scalar function φ and integrating over the boundary Γ we receive equation (*17*).

$$\int \bar{\mathrm{n}} \cdot (\varphi \mathrm{k}\nabla T)\, d\Gamma = \int \varphi h(T - T_{ext})\ \ d\Gamma \tag{17}$$

Now equation (*17*) is discretized using shape functions in equation (*6*).

$$\int \varphi h(T - T_{ext})\ \ d\Gamma = \sum_{i=1}^{n} \int N_j h(N_i T_i - T_{ext})\ \ d\Gamma \tag{18}$$

Here the sum is necessary only on the boundary, as the driving force of the convective cooling is the temperature difference of the surface and the surrounding. Dividing into multiples of known and unknown forms of T (T_{ext} and T_i^{n+1}) we get equation (*19*).

$$\sum_{i=1}^{n} \int N_j N_i h T_i^{n+1} \, d\Gamma + \int -N_j h T_{ext} \, d\Gamma = \int \boldsymbol{N} h \boldsymbol{N}^T \, d\Gamma \, \boldsymbol{T}^{n+1} - \int \boldsymbol{N} h T_{ext} \, d\Gamma \tag{19}$$

Here T_{ext} is simply a scalar value of the temperature of the cooling medium. The convective heat transfer coefficient h is usually treated as known based on some experiments [22], though some authors prefer to calculate it themselves using different parameters, such as the water velocity from the cooling sprays and the hydraulic diameter of the slot [19, 24]. In this work, different values for h were used according to other literary works, which are listed in section 5.

The fourth kind of boundary condition is radiation. In this work radiation boundary condition was treated in combination with convection, as both are present on the boundaries of the cast billet or slab. The form for radiation boundary condition is

$$-\bar{n} \cdot (k \nabla T) = \sigma_{SB} \varepsilon_{rela} \{(T^{n+1})^4 - T_{ext}^4\} \tag{20}$$

where σ_{SB} is the Stefan-Boltzmann constant (5.670×10^{-8} W m^{-2} K^{-4}) and ε_{rela} is the relative permittivity. The equation (*20*) is problematic as it includes fourth-order exponent of the unknown T^{n+1}; but it can be manipulated into more manageable form in equation (*21*) [28].

$$\begin{aligned} -\bar{n} \cdot (k \nabla T) &= \sigma \varepsilon \{(T_i^m)^3 + (T_i^m)^2 T_{ext} + T_i^m T_{ext}^2 + T_{ext}^3\} \left(T_i^{m+1} - T_{ext}\right) \\ &= h_{ef} \left(T_i^{m+1} - T_{ext}\right) \end{aligned} \tag{21}$$

Here the superscript *m* refers to the previous iteration step and *m+1* to the next iteration step the temperature of which is to be solved. At the end of iteration, when convergence is achieved, T_i^{m+1} practically equals T_i^m (appendix, 8.2). The effective convective term, h_{ef}, is only dependent on the previous iteration step, and there is only a first-order term of the current iteration step *m+1* in the right hand term, enabling the equations to be solved with linear solving algorithms [28].

This method of using radiation boundary condition is iterative and requires several iterative laps to reach convergence.

As there is both convection and radiation on the boundaries of the slab or billet, the effective heat transfer coefficient of radiation h_{ef} can simply be added to the heat transfer coefficient of convection from the cooling water sprays, which is simply marked as h.

Now that we have a computable form for all boundary conditions, the final equation can be presented as in equation (*22*).

$$\left([\boldsymbol{K}] + \frac{1}{\Delta t}[\boldsymbol{C}]\right)\boldsymbol{T}^{m+1} = \boldsymbol{R} + \frac{1}{\Delta t}[\boldsymbol{C}]\boldsymbol{T}^{n} \tag{22}$$

Here $[\boldsymbol{K}]$ is the previously mentioned stiffness matrix, sometimes called diffusivity matrix or heat conduction matrix, $[\boldsymbol{C}]$ is a mass matrix which includes enthalpy or heat capacity and $\boldsymbol{R}$ includes properties from boundary conditions – these are further explained below. $\boldsymbol{T}^{m+1}$ is the new temperature vector which is being solved.

The global diffusivity matrix is formed using the diffusivity matrices of the elements, which are generated using equation (*23*) [12, 15]. The assembly of the element specific matrices into a global matrix is explained in the text books of Cook et al and Fish et al [12, 29], and will not be elaborated here. The global matrices $[\boldsymbol{C}]$ and $\boldsymbol{R}$ are generated in a similar fashion.

$$[\boldsymbol{k}] = \int [\boldsymbol{B}]^T[\boldsymbol{\kappa}][\boldsymbol{B}]\, dV + \int \boldsymbol{N}h\boldsymbol{N}^T\, d\Gamma \tag{23}$$

The matrix $[\boldsymbol{B}]$ is includes the spatial derivatives of the shape functions [12, 13]. In the case of temperature, it can be expressed as in equation (*24*) assuming a four-noded quadrilateral element [12].

$$[\boldsymbol{B}] = \begin{bmatrix} \frac{\partial N_2}{\partial x} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \frac{\partial N_3}{\partial y} & \frac{\partial N_4}{\partial y} \end{bmatrix} \tag{24}$$

In the case of two dimensional analysis, this can be marked as equation (*25*) [12, 13, 29].

$$[k] = \int\int [\boldsymbol{B}]^T[\boldsymbol{\kappa}][\boldsymbol{B}]p\, dx\, dy + \int \boldsymbol{N}h\boldsymbol{N}^T\, d\Gamma \tag{25}$$

Where p is the thickness in the z direction. In practice, isoparametric formulation is used (appendix). In the case of isotropic heat conduction, which is always assumed in this work, $[\boldsymbol{\kappa}]$ is

$$[\boldsymbol{\kappa}] = k\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}. \tag{26}$$

Here k is the heat conduction coefficient, which is a function of temperature [12].

$\boldsymbol{N}$ is the shape function matrix, which is presented in equation (*27*).

$$\boldsymbol{N} = [N_1 \quad N_2 \quad N_3 \quad N_4]^T \tag{27}$$

The shape functions $N_{1\text{-}4}$ are functions of the coordinates x and y, which in practice depend on the integration point used in Gauss quadrature [12], which is explained in appendix 8.3.

The mass matrix $[\boldsymbol{c}]$ of a single element is generated using equation (*28*) [12].

$$[\boldsymbol{c}] = \int \boldsymbol{N}\rho c_p \boldsymbol{N}^T \tag{28}$$

The expression used in this work for **R** of a single element is

$$\boldsymbol{r} = \int \boldsymbol{N} q \, d\Gamma + \int \boldsymbol{N} h T_{ext} \, d\Gamma. \tag{29}$$

Here q is associated with Neumann boundary condition and h is the heat convection coefficient. T_{ext} is the temperature of the convective material which cools or heats the structure on the boundary Γ. q is always zero in this work, as only convective, symmetry and radiation boundary conditions are used in this work.

The formulation for the temperature modeling will change slightly from equation (*22*) when multiple phases are present during modeling. The iron rich end of the iron-carbon phase diagram is presented in Figure 8 [*29*]. It is obvious from Figure 8 that at least two phase transitions occur for steels during continuous casting, which has an approximate temperature range of 900-1600 °C. These phase transitions are liquid to austenite (L->γ) and austenite to ferrite (γ->α). In the case of low carbon steels (such as the blue line in Figure 8), delta ferrite δ will also be present in high temperatures between the austenite and liquid phases.

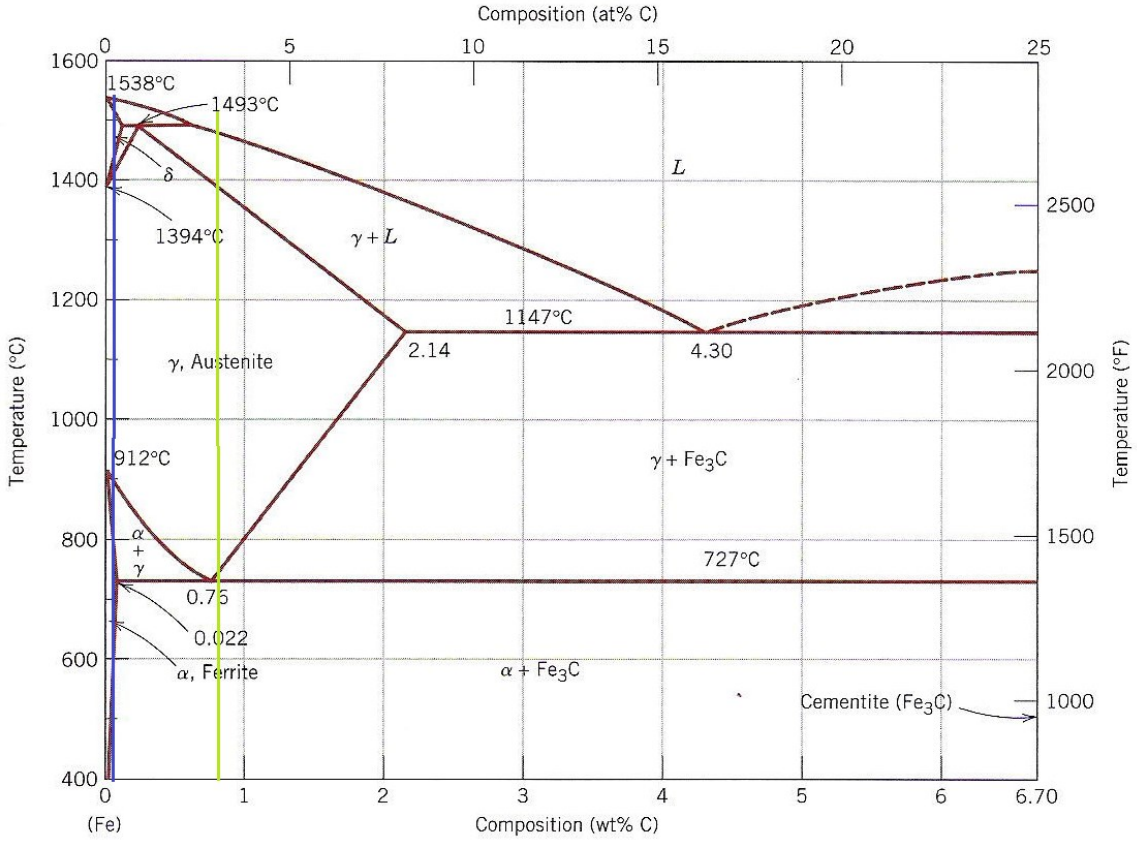

Figure 8 Iron Carbon phase diagram [31]. The green and blue lines represent roughly 0,85 w-% Carbon and roughly 0,02 w-% carbon steels respectively .

When treating simulation problems with several phases, the elemental grid can either be fixed or variable. In fixed grids, the elements and nodal coordinates are created only once, and the phase changes occur within the elements – a single element can then include both solid and liquid phases, for example [32]. On the other hand, in variable grid methods the grid is changed in each iteration step so that the phase boundary always coincides with an element boundary [32]. The variable grid methods are computationally much more expensive and they are more difficult to program. Hu et al [32] presented that a fixed grid in combination with the enthalpy method can be sufficient when modeling phase change, despite the fact that a variable grid would be more accurate. Fixed grid has been used in this work.

In the enthalpy method, enthalpies, H or [**H**], replace the heat capacities, c_p or [**C**] in the FEM formulation. Enthalpy is a function of temperature only, which enables its use without drastically changing the equation (*11*).

The enthalpy formulation used in this work is based on the article of Swaminathan et al [33]. Only Dirichlet boundary condition is treated here, as full enthalpy formulation with precise treatment of Robin or Neumann boundary conditions would lead to excessively long and unnecessarily complex equations.

Let us form a volume integral (*30*), which has already been multiplied with weight function φ.

$$\int \frac{\partial \rho H}{\partial t} \varphi \, dV = -\int \nabla \varphi \cdot (k \nabla T) \; dV \tag{30}$$

The enthalpy is discretized similarly in relation to time as temperature was in the previous section using backward Euler differentiation.

$$\int \frac{\partial \rho H}{\partial t} \varphi \, dV = \int \rho \frac{\left(H_i^{n+1} - H_i^n\right)}{\Delta t} \varphi \, dV \tag{31}$$

Let us use index m to describe the iteration step. The iteration superscript m is looped within each time step. At the end of a single time step, when convergence (appendix 8.2) has been achieved, H_i^{n+1} is equal to H_i^{m+1} and H_i^m. Within the m-iteration loop, the enthalpies H_i^{m+1} are candidates for the new value of H_i^{n+1}.

Enthalpy can be linearized in relation to temperature as in equation (*32*) [32].

$$H_i^{n+1} = H_i^{m+1} = H_i^m + \left(\frac{dH}{dT}\right)_i^m dT = H_i^m + \left(\frac{dH}{dT}\right)_i^m \left(T_i^{m+1} - T_i^m\right) \tag{32}$$

This is implemented into equation (*31*).

$$\int \rho \frac{\left(H_i^{n+1} - H_i^n\right)}{\Delta t} \varphi \, dV = \int \rho \frac{\left(H_i^m + \left(\frac{dH}{dT}\right)_i^m \left(T_i^{m+1} - T_i^m\right) - H_i^n\right)}{\Delta t} \varphi \, dV \tag{33}$$

Let us separate the variables T_i^{m+1} and T_i^m into different sides. Here T_i^{m+1} is the unknown variable and T_i^m is the last known value.

$$\int \rho \frac{\left(\frac{dH}{dT}\right)_i^m T_i^{m+1}}{\Delta t} \varphi \, dV = -\int \rho \frac{H_i^m - H_i^n + \left(\frac{dH}{dT}\right)_i^m T_i^m}{\Delta t} \varphi \, dV - \int \nabla \varphi \cdot (k \nabla T) \; dV \tag{34}$$

Equation (*34*) above is discretized into finite element form by using shape functions for all variables as in equation (*35*).

$$\begin{aligned} T_i &= \sum_{i=1}^{l} N_i T_i \\ \varphi &= N_i \\ H^m &= \sum_{i=1}^{l} N_i H_i^m \\ H^n &= \sum_{i=1}^{l} N_i H_i^n \\ k^m &= \sum_{i=1}^{l} N_i k_i \end{aligned} \tag{35}$$

As before, the shape functions N_i are functions of x and y.

$$\begin{aligned} &\sum_{i=1}^{l} \int \rho \frac{N_i \left(\frac{dH}{dT}\right)_i^m N_i T_i^{m+1}}{\Delta t} N_j \, dV \\ &\qquad = -\sum_{i=1}^{l} \int \rho \frac{N_i H_i^m - N_i H_i^n + N_i \left(\frac{dH}{dT}\right)_i^m N_i T_i^m}{\Delta t} N_j \, dV \\ &\qquad - \sum_{i=1}^{l} \int \nabla N \cdot \left[N_i k \nabla N_i T_i^{m+1} \right] dV \end{aligned} \tag{36}$$

At this point, it is simpler to move on to matrix notation. Equation (*36*), which is valid for any point in an element, is in matrix notation in equation (*37*), which is valid globally. In addition, we move all $\boldsymbol{T}^{m+1}$ terms to the left hand side of the equation.

$$[\boldsymbol{H}]_1 \, \boldsymbol{T}^{m+1} + [\mathbf{K}] \boldsymbol{T}^{m+1} = [\boldsymbol{H}]_2 \boldsymbol{T}^m \tag{37}$$

$$[\boldsymbol{h}]_1 = \int \rho \frac{\boldsymbol{N} \left(\frac{dH}{dT}\right)_i^m}{\Delta t} \boldsymbol{N} \; dV \tag{38}$$

$$[\boldsymbol{h}]_2 = \int -\rho \frac{\boldsymbol{N} H_i^m - \boldsymbol{N} H_i^n + \boldsymbol{N} \left(\frac{dH}{dT}\right)_i^m}{\Delta t} \boldsymbol{N} \; dV \tag{39}$$

Global matrices $[\boldsymbol{H}]_1$ and $[\boldsymbol{H}]_2$ have been assembled from elemental matrices $[\boldsymbol{h}]_1$ and $[\boldsymbol{h}]_2$. $[\mathbf{K}]$ is the global heat conduction matrix assembled from the local matrices $[\mathbf{k}]$, which has previously been defined in equation (*23*). Equation (*37*) can be reformed into

$$([\boldsymbol{H}]_1 + [\mathbf{K}])\,\boldsymbol{T}^{m+1} = [\boldsymbol{H}]_2\boldsymbol{T}^{m}. \tag{40}$$

Now the new iteration value m+1 for enthalpy can be solved using equation (*32*) [33].

This new enthalpy is calculated in every node. As the derivative of the enthalpy-temperature plot changes as a function of temperature, it is possible that the new iterated value of temperature is not on the plot. Let us imagine a situation as in Figure 9. We must fix the temperature back to the plot from point A. This can be done by solving for T_i^{m+1} from equation (*41*).

$$H_i^{m+1} - H_0 = \left(\frac{dH}{dT}\right)_2 \left(T_i^{m+1} - T_0\right) \tag{41}$$

Here H_0 and T_0 denote known values between phases, such as at the intersection of linear sections 2 and 3 in Figure 9. The $\left(\frac{dH}{dT}\right)_2$ used is from the curve, as in point A in Figure 9, the $\left(\frac{dH}{dT}\right)_2$ from the steeper curve, marked "2", would be used. The equation for fixing T_i^{m+1} back to the curve as in Figure 9 is

$$T_i^{m+1} = \frac{H_i^{m+1} - H_0}{\left(\frac{dH}{dT}\right)_2} + T_0. \tag{42}$$

This method of fixing temperature can be replaced by fixing the enthalpy back to the plot if there is a phase-change area where the slope previously was much steeper than currently. In these situations, it is better to change the enthalpy as in point B in Figure 9, because if the temperature was altered it might change by dozens of degrees and cause problems in further iteration laps.

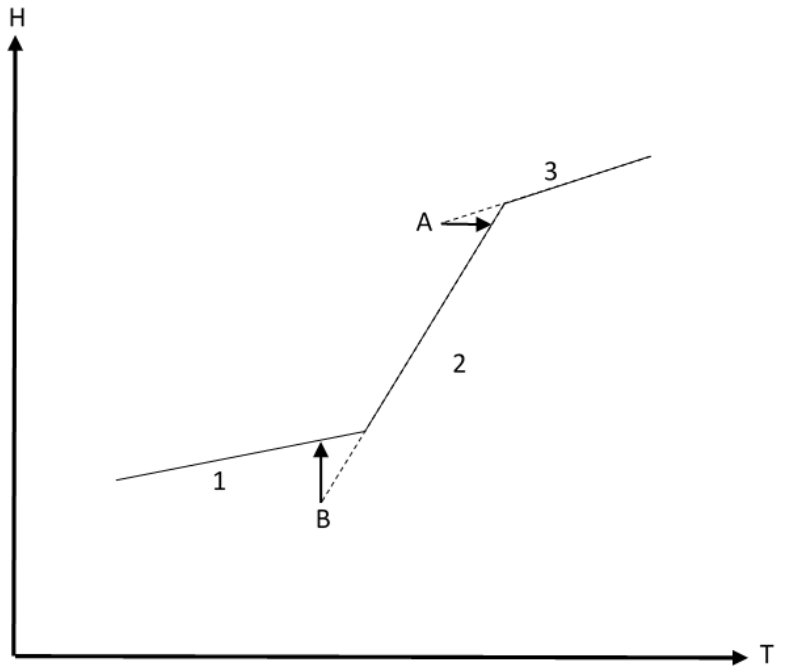

Figure 9 Enthalpy – temperature plot. The dashed lined represents temperature solutions in modeling and the arrows the fixing step. Generally, the temperature is altered in the fixing step, as in point A, but after phase change where there was a very large heat capacity dH/dT, enthalpy is altered instead, as in point B.

The different phases affect all data, be it mechanical or thermal properties. Most of this data is easy to implement for different phases and require no explanation here, but thermal conductivity is slightly more problematic.

The thermal conductivity of steel is a function of temperature, which is easily treated in practice by having it be a polynomial or a piecewise linear function of temperature. In liquid phase, however, both thermal conduction and convection are present, convection being by far the dominant term [24]. Previously in this section it was mentioned that this work uses fixed grid method, where the mesh is not changed depending on the location of the liquid-solid interface. To account for convection, the thermal conductivity of the liquid phase is manually increased. The thermal conductivity of the liquid phase is commonly multiplied by a number in the range 2-6 [22, 24, 27].

3.4 The Modeling of Elasticity, Plasticity and Creep

Four major factors cause strain in continuous casting: external mechanical loads, creep, heat expansion and phase transformations [11]. The treatment of phase transformations can be combined with heat expansion, as the volume change of austenite-to-ferrite corresponds with a change of 150°C in temperature [2], but this still leaves three separate strains to be dealt with. In addition to several sources of strain, yielding and plastic material behavior has to also be considered, as the yield strength of steel is extremely low in the temperatures of continuous casting.

In Figure 10, a stress strain plot of a tensile strength test is presented. Elastic material behavior occurs between point 0 and 1. Elastic region behaves in linear fashion, which enables its rather straightforward modeling [12]. When an external load is removed, if the material response stayed in the elastic region, the material will return to its original shape.

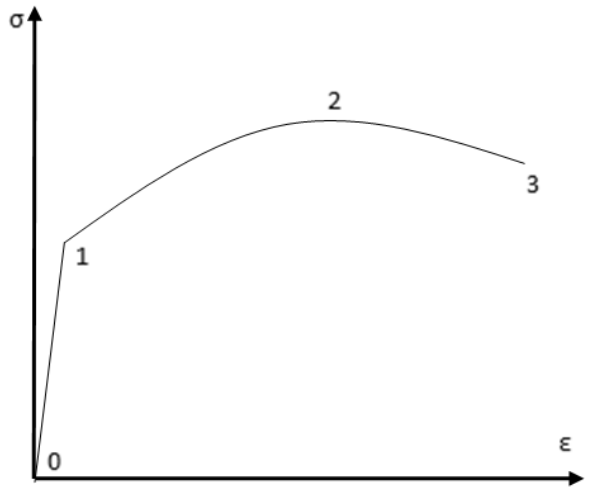

Figure 10 A simplified stress-strain plot of a one dimensional tensile strength test. Point 1 signifies the yield stress, point 2 ultimate tensile strength and point 3 the fracture point.

Plastic material behavior beyond point 1, on the other hand, means that when the external load is removed, the material no longer returns to its original shape [34]. In the plastic range between points 1 and 3 in Figure 10, the material behavior is nonlinear [12]. Creep is the slow deformation of material under constant load due to high temperature [30]. Concept image of creep is

presented in Figure 11 [30]. Creep can be divided into three phases: primary creep between points 0 and 1, linear secondary creep between 1 and 2 and tertiary creep beyond point 2. Of these three usually only the secondary creep is modeled, as the behavior of both primary and tertiary creep is highly complex [35].

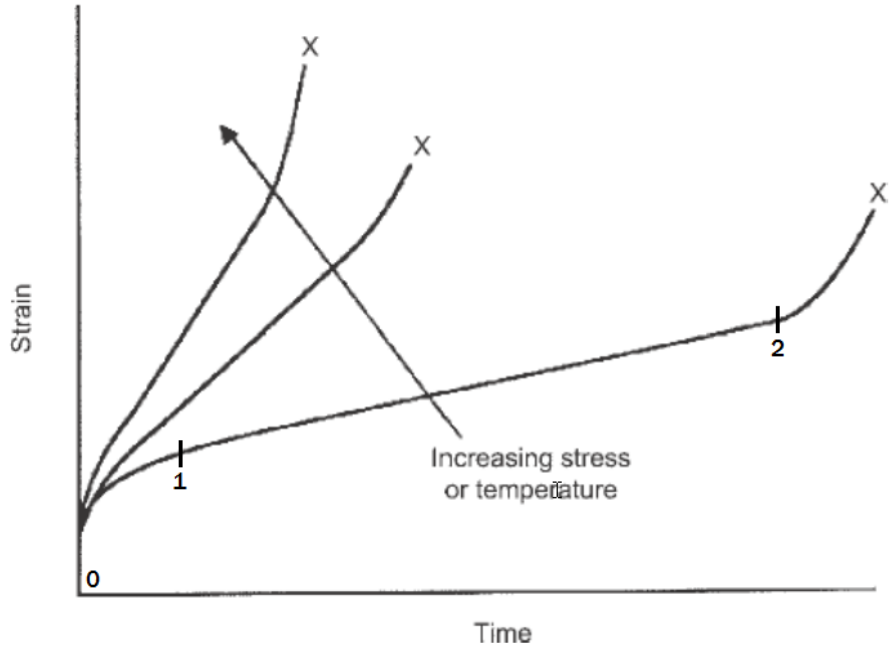

Figure 11 The effect of stress or temperature on creep strain rate [30]. The strain rate increases with increasing stress and temperature on all areas of creep: primary (0-1), secondary (1-2) and tertiary (2-X).

Creep is dependent not only on stress or temperature but also on strain rate which is caused by external loads [35]. The effect of external strain rate is similar to the effect of stress, where higher strain rates cause more significant creep strain [35]. Creep is also highly dependent on the elemental composition and microstructure, as both of these affect the dislocation movement mechanisms immensely [35]. A more in-depth treatment of creep is beyond the scope of this work, but interested readers are referred to the work of Frost et al [35].

3.4.1 Finite Element Formulation for Mechanical Problems

In mechanical problems, the stresses σ and strains ε are usually the primary point of interest. These are used to determine the load, which would cause material failure, which again can take

the form of cracks or yielding and deformation of material. Neither stresses nor strains, however, can be directly computed using finite element analysis, but first the displacements of the domain have to be calculated.

In the modeling of continuous casting, it can be assumed that the strains and displacements remain small, which eliminates the need to consider geometrical nonlinearities. As cracks will form during solidification when strains exceed only a few percent, the small displacement assumption will not be compromised [18].

Here only the most basic equations related to mechanical FEM analysis are presented, a more in depth treatment can be found in the appendices.

The global system of equations is presented in equation (*43*).

$$[\boldsymbol{K}]\boldsymbol{d} = \boldsymbol{F}. \tag{43}$$

Here $[\boldsymbol{K}]$ is the global stiffness matrix, $\boldsymbol{d}$ is the displacement vector and $\boldsymbol{F}$ includes the nodal forces. This equation is used to solve for the displacements $\boldsymbol{d}$, which again can be used to calculate the strains (*47*) and stresses (*48*).

The elemental stiffness matrices are generated with equation (*44*). The procedure of combining element matrices into global matrices is not presented here, but it is well documented in the text books of Cook et al [12] and Fish et al [29].

$$[\boldsymbol{k}^e] = [\boldsymbol{k}] = \int [\boldsymbol{B}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B}]\, dV \tag{44}$$

In the case of the nodal forces, the effect of external boundary forces, internal forces and body forces are separated into different vectors, which can be determined using equations (*45*).

$$\begin{aligned} \boldsymbol{f} &= \boldsymbol{f}_{bound} + \boldsymbol{f}_{int} + \boldsymbol{f}_{body} \\ \boldsymbol{f}_{bound} &= \int [\boldsymbol{N}]^T \boldsymbol{g}\, dS \\ \boldsymbol{f}_{int} &= \int [\boldsymbol{B}]^T [\boldsymbol{D}]\boldsymbol{\varepsilon_0}\, dV - \int [\boldsymbol{B}]^T \boldsymbol{\sigma_0}\, dV \\ \boldsymbol{f}_{body} &= \int [\boldsymbol{N}]^T \boldsymbol{C}\, dV \end{aligned} \tag{45}$$

$\boldsymbol{f}_{bound}$ defines the forces which affect the boundaries of the element. These usually are traction forces, supportive forces or simple loads on top of the structures. These forces in vector $\boldsymbol{g}$ have to be known at the nodes. If they are not, an expression for them has to be calculated. In

continuous casting, the ferrostatic pressure of the molten cone is taken into account as a boundary force, as in

$$\{f\}_{bound} = \int [\boldsymbol{N}]^T \rho |\boldsymbol{g}| l \bar{n} \tag{46}$$

where ρ is the density of molten steel, $\boldsymbol{g}$ is the gravitational acceleration and l is the height of the molten pool. The direction of the force is handled with $\bar{n}$, which is either 1 or -1, defining the force to be perpendicular to the solidifying shell.

$\boldsymbol{f}_{int}$ describes internal forces caused by initial stresses or strains. In this model, these internal forces are only present as corrective loads in plastic analysis to improve the convergence properties of the analysis [11, 29]. In practice, that means introducing the stress term of $\boldsymbol{f}_{int}$ into all elements, where $\boldsymbol{\sigma_0}$ is the stress vector computed during the previous iteration step. The inelastic and thermal strains are similarly taken into account using the form for strains $\boldsymbol{\varepsilon_0}$.

$\boldsymbol{f}_{body}$ are forces which are enacted on the entire volume of the element. The most significant of these forces is gravity, but in other situations outside of this work, this vector can also include for example electromagnetic forces. $\boldsymbol{C}$ is the force vector of those forces, similarly to $\boldsymbol{g}$ in boundary forces.

The global system of equations (*43*) can now be solved for the displacements, which enables the calculations of strains with equation (*47*).

$$\boldsymbol{\varepsilon} = [\boldsymbol{B}]\boldsymbol{d^e} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \dots & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & \dots & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \dots & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}_{3*8} \begin{Bmatrix} u_{x1} \\ u_{y1} \\ \vdots \\ u_{x4} \\ u_{y4} \end{Bmatrix}_{8*1} \tag{47}$$

Here the vector $\boldsymbol{d^e}$ includes the displacements of the nodes in the element, and $[\boldsymbol{B}]$ is the strain-displacement matrix. The connection between stress and strain is

$$\boldsymbol{\sigma} = [\boldsymbol{D}]\boldsymbol{\varepsilon} \tag{48}$$

where $\boldsymbol{\varepsilon}$ is the strain vector, and the expression for matrix $[\boldsymbol{D}]$ depends on the number of dimensions treated. In this work, the form for $[\boldsymbol{D}]$ is

$$[\boldsymbol{D}] = \frac{E}{(1+v)(1-2v)} \begin{bmatrix} 1-v & v & 0 \\ v & 1-v & 0 \\ 0 & 0 & \frac{1-2v}{2} \end{bmatrix}. \quad (49)$$

In simple linear case $[\boldsymbol{D}]$ is Young's modulus E; in this case the equation (*48*) is known as Hooke's law. It dictates that stress is linearly proportional to strain. This is only true within the elastic range of a material, and some material exhibit nonlinear behavior even in the elastic range.

The two dimensional simplification used in this work is called plane strain. Plane strain is a simple method of reducing the amount of computable directions. In plane strain analysis, the z-direction is not treated at all, and it is assumed that $\varepsilon_z = \gamma_{yz} = \gamma_{zx} = 0$ [12]. Technically ε_z can also be nonzero in plane strain analysis as long as it is constant. When the z direction is treated in some rudimentary method, it is called general plane strain.

Equation (*43*) cannot be solved without the implementation of boundary conditions. In the case of mechanical analysis, the most basic boundary conditions are prescribed displacements on the boundary – this known value generally being zero – and known forces on the boundary, which are the boundary forces $\boldsymbol{f}_{bound}$. The prescribed displacements are treated in following manner: the rows and columns corresponding to that degree of freedom at that node are marked zero in the stiffness matrix, and the diagonal is set to 1 [29]. In this treatment of boundary condition, also the corresponding force in the force vector is set to zero [29].

In addition, rigid body motion of the domain has to be prevented as well. This can be done by prescribing the displacements of at least two nodes, or three nodal displacement components to zeros [29]. This will prevent rotational or translational movement [29].

3.4.2 Plasticity and Creep in Finite Element Modeling

Efficient modeling of creep and plasticity have been combined in the work of Li et al [27], who used constitutive equations developed by Kozlowski et al [36] to model continuous casting. Kozlowski et al [36] had developed several equations which combined creep and plastic strain into an inelastic strain. Their work was based on the tensile test measurements of Peter Wray [37] and the creep data of Suzuki et al [38]. Similar constitutive equation, called the Anand constitutive model, has also been used in continuous casting modeling [24]. It is worthwhile to note that some models do not treat elasticity in high temperatures at all, as the elastic region is so small in the temperatures near the melting point [2].

The work of Kozlowski et al has been previously used in continuous casting modeling by at least Koric et al [18] and Li et al [3, 27], who also created another constitutive equation to model delta ferrite, as the model created originally by Kozlowski et al [36] was meant for austenite only. This work adopts the method used by Li et al [3, 27] to model plasticity and creep.

Before going into detail into the modeling of plasticity and creep, a few common principles of FEM plastic analysis should be considered. The displacements are calculated at the nodes using equation (*43*) in the previous section. This leads to the stresses and strains being most easy to calculate at the nodes. Plastic behavior affects the stiffness properties of the element, however, and evaluating the stresses only at the nodes which are in the corners of the element is a poor approximation of the condition of the element. For this reason, the material properties of each element are determined by a few sampling points within that element – typically these points are the gauss quadrature integration points discussed in appendix 8.3 [12].

In each sampling point, a variety of variables (stress, total strain, creep strain, plastic strain) is saved and their history is tracked, as plasticity is history dependent. There is some debate in how many sampling points should be used in each element; some believe that in the case of 2D analysis a single point will suffice, while others believe the more accurate analysis enabled by more sampling points is worth the additional computational cost [12]. In this work, a single sampling point was used in the middle of the element. Using only a single sampling point does not prevent achieving the same accuracy that would be achieved with several sampling points, as by reducing the element size the same accuracy can still be achieved [12].

Let us first define the treatment for thermal strains:

$$\Delta\{\varepsilon_{th}\}^{t+\Delta t} = \left(L_E T_i^{t+\Delta t} - L_E T_i^t\right)\{\delta\} \tag{50}$$

where $\{\delta\}$ is the Kronecker delta and

$$L_E = \sqrt[3]{\frac{\rho(T_0)}{\rho(T)}} - 1. \tag{51}$$

The polynome ρ(T) in equation (*51*) is phase dependent and it is presented in the appendix 8.4.

Next, we create a form for the inelastic strains, which include creep and plasticity. Kozlowski et al [36] developed several constitutive equations with varying complexity, accuracy and reliability in their article. The inability of these models to simulate delta ferrite was redeemed by Li et al [3], who created an enhanced power-law model for the modeling of delta ferrite [3]. The model III of Kozlowski et al [36] is

$$\dot{\varepsilon}_{in} = C e^{-\frac{Q}{T}}\left(\sigma - a_e \varepsilon_p^{n_e}\right)^n. \tag{52}$$

The enhanced power law model by Li et al [3, 27] is

$$\dot{\varepsilon}_{in} = 0.1c \left| \frac{\bar{\sigma}}{C_f \left(\frac{T}{300}\right)^{-5.52} (1 + 1000|\varepsilon_P|)^m} \right|^n . \tag{53}$$

In the mushy region, the inelastic strain rate is treated as a flow strain rate, which is

$$\dot{\varepsilon}_{flow} = \begin{matrix} ac\left(|\bar{\sigma}| - \sigma_{yield}\right) & if\ \bar{\sigma} > \sigma_{yield} \\ 0 & if\ \bar{\sigma} \leq \sigma_{yield} \end{matrix} \tag{54}$$

The full forms of these equations with explanations for all the variables are presented in the appendix 8.4.

In the case of the mushy elements, the corrective loads are not used for the flow strain in equations (*43*) and (*45*). The flow strains are only used in relation to the crack prediction analysis, and they are set to zero when an element solidifies.

The equation (*52*) is reliable only between 950 and 1400 degrees Celsius. It stays reliable as long as the loads and strains are small and simulation time is relatively short (it can only model primary creep) [36].

The combination of these models – (*52*) and (*53*) – was used by Li et al [3] in their cross-section 2D model to determine the effect of casting speed on the shell structure as it exits the mold. They determined that the combination of these two equations is reliable and matches experimental data fairly well.

The algorithm for calculating stresses in plastic analysis within each element can be summarized as follows [27]:

1) Estimate the new strain increment based on the displacement increment of the previous time step using

$$\{\Delta\hat{\varepsilon}\} = [\boldsymbol{B}]\Delta\boldsymbol{u}^t \qquad (55)$$

2) Next, the stresses in the element $\boldsymbol{\sigma}^*$, and the corresponding Von Mises stress $\bar{\sigma}^*$, and the deviatoric stress vector $\boldsymbol{\sigma}'$ are calculated. The stresses in the element are

$$\boldsymbol{\sigma}^{*t+\Delta t} = [\boldsymbol{D}]\left(\boldsymbol{\varepsilon}_{tot}^t - \boldsymbol{\varepsilon}_{th}^{t+\Delta t} - \boldsymbol{\varepsilon}_{in}^t + \Delta\hat{\boldsymbol{\varepsilon}}\right) \qquad (56)$$

where $\boldsymbol{\varepsilon}_{in}^t$ was defined in step 8 of previous time step and $\boldsymbol{\varepsilon}_{tot}^t$ is the total strain accumulated at this point. The Von Mises stress is

$$\bar{\sigma}^{*,t+\Delta t} = \sqrt{\frac{1}{2}\left(\left(\sigma_x - \sigma_y\right)^2 + (\sigma_x - \sigma_z)^2 + \left(\sigma_z - \sigma_y\right)^2 + 2\tau_{xy}^2\right)} \qquad (57)$$

where σ_x, σ_y, σ_z and τ_{xy} are components of the stress. The deviatoric stress vector is

$$\boldsymbol{\sigma}^{*\prime} = \left\{\sigma_x - \frac{1}{3}\sigma_m \quad \sigma_y - \frac{1}{3}\sigma_m \quad \sigma_z - \frac{1}{3}\sigma_m \quad \tau_{xy}\right\} \qquad (58)$$

where σ_m is the sum of the x, y and z components of $\{\sigma^*\}^{t+\Delta t}$. These three stresses are used to define the inelastic strain rate in step 4) and the direction of the strain vector in step 6).

3) Two differential equations are solved simultaneously for $\dot{\varepsilon}_{in}^{t+\Delta t}$ and $\hat{\sigma}^{t+\Delta t}$:

$$\begin{aligned} \dot{\varepsilon}_{in}^{t+\Delta t} &= \varepsilon_{in}^t + f\Delta t \\ \hat{\sigma}^{t+\Delta t} &= \bar{\sigma}^* - 3\mu f\Delta t \end{aligned} \qquad (59)$$

Where f is either the function (*52*) , (*53*) or (*54*) depending on the phase of the element. The equation system (59) is solved using a fully implicit bounded Newton-Raphson integration method introduced by Lush et al [39], which is presented in the appendix 8.6.

4) The scalar stress estimate $\hat{\sigma}^{t+\Delta t}$ is expanded into vector form.

$$\hat{\boldsymbol{\sigma}}^{t+\Delta t} = \hat{\sigma}^{t+\Delta t}\frac{\boldsymbol{\sigma}^{*\prime\, t+\Delta t}}{\bar{\sigma}^{*,t+\Delta t}} + \frac{1}{3}\sigma_m^{*,t+\Delta t}\boldsymbol{\delta}^T \qquad (60)$$

5) Calculate the inelastic strain rate of this time step $\dot{\varepsilon}_{in}^{t+\Delta t}$ (using equation (*52*) or (*53*)) using $\hat{\sigma}^{t+\Delta t}$ and $\bar{\varepsilon}_{in}^{t+\Delta t}$, the latter of which is simply the total of the inelastic strains at time $t + \Delta t$.

6) Change the scalar $\dot{\varepsilon}_{in}^{t+\Delta t}$ into a vector form using

$$\boldsymbol{\varepsilon}_{in}^{t+\Delta t} = \frac{3}{2}\bar{\varepsilon}_{in}^{t+\Delta t}\frac{\boldsymbol{\sigma}^{*\prime}}{\bar{\sigma}^{*,t+\Delta t}} \qquad (61)$$

7) Use equations (*44*) and (*45*) to determine the stiffness and forces of the element, and then solve the displacement increment using equation (*43*). In equation (*45*), the forces due to internal strains $\{\varepsilon_0\}$ are used to take into account the strain increments of previous step, here $\{\varepsilon_0\}$ being the sum of thermal, inelastic and elastic strain within each element.

8) Calculate $\Delta\boldsymbol{\varepsilon}^{t+\Delta t}$ from the displacements using equation (*47*) and update the total strain, total thermal strain and the total stress. Update the inelastic strain, which is used in step 2:

$$\boldsymbol{\varepsilon}_{in}^{t+\Delta t} = \boldsymbol{\varepsilon}_{in}^{t} + \Delta t\boldsymbol{\varepsilon}_{in}^{t+\Delta t} \qquad (62)$$

9) Steps 1-8 are done once for each time step after the temperatures have been calculated. Steps 1-6 are element specific and done to an element at a time, step 7 is global and all elements are treated simultaneously.

3.4.3 Recrystallization

Recrystallization is phenomenon in which new crystal form into the microstructure of deformed steel in high temperatures. The new grains are strain free: in effect recrystallization periodically reduces the strain to zero. A fundamental recrystallization model is beyond the scope of this work, but a simplified model is incorporated.

The equation used to calculate the recrystallized portion X is [40]

$$X = 1 - \exp\left(-0.693\left(\frac{\varepsilon_{in} - \varepsilon_{cr}}{\dot{\varepsilon}\, t_{0.5}}\right)^{n}\right) \quad (63)$$

where ε_{cr} is the critical strain where recrystallization starts, n is the time exponent and $t_{0.5}$ is the time after which half of the microstructure has recrystallized. Mirzadeh et al [41] developed a form for $t_{0.5}$, which is

$$t_{0.5} = \frac{1}{2.25 * 10^{-15} * (\dot{\varepsilon} T^{5.59})^{0.87}}. \quad (64)$$

This form for $t_{0.5}$ was meant for stainless steel, which means it is approximate at best. This model was used, as simple models meant for plain carbon steel and capable of modeling recrystallization near the melting point were not available.

The critical strain ε_{cr} where recrystallization begins was assumed to be 0.5 %, as Kozlowski et al reported that when the inelastic strain reaches 5 % they would assume total recrystallization [36]. The values for time exponent n vary from 1.35 to 3.5 in literature; 3.5 was used in this work [42-44].

3.4.4 Crack Prediction

Cracks in continuous casting usually originate and propagate in the mushy zone [19]. This can be attributed to the ductility loss of the mushy zone due to micro segregation of solute elements at solidifying dendrite interfaces [19]. For this reason, a certain cracking zone can be determined, where the temperature is determined by certain liquid fraction which enables the birth of cracks in the mushy zone [19]. Several different criteria for determining when hot tearing cracks are initiated have been developed in the literature. Perhaps the simplest hot tearing condition is simply the time the node or element spends in the temperature range which has the highest risk of tearing. This risk typically is $0.99T_{melt} - 0.9T_{melt}$ [25]. Another such criterion has been suggested by Won et al. [45], which is

$$\varepsilon_c = \frac{0.02821}{\dot{\varepsilon} * \Delta T_B^{0.8638}} \quad (65)$$

where ΔT_B is the brittle temperature range, which is defined as between the temperatures when 90% and 99% of the solid forms. In this work these temperatures were simply defined as

$$\Delta \mathrm{T_B} = 0.99T_{melt} - 0.9T_{melt} \qquad (66)$$

where T_{melt} is the melting point used in calculations. This range is not the same as the temperatures where 90 % and 99 % solid has formed, but they are an adequate approximation of that range.

In this work the inelastic strain in elements is tested again these crack initiation criteria, and if the strain is over the critical strain or the temperature is within the temperature range the result is saved. At the post processing step of modeling the billet cross section can be examined and the areas with the highest risk of cracking can be studied based on which nodes spent the longest time in the fragile temperature region of which nodes had strain over the critical strain most often.

4. Matlab Model

A Matlab code was written to predict the crack initiation in the continuously casted steel billet. Heat transfer and mechanical strain rates were modeled as presented in the literary study. These enabled the use of crack initiation criteria to predict hot tear cracks in the phase transition front in the billet.

4.1 Simplifications of Continuous Casting

Continuous casting includes various phenomena that are not essential to model to achieve reliable results for the thermomechanical behavior. Some phenomena have to be left out due to their complicated nature, computational cost and the scope of this study. Here is a list of phenomena that were left out of the model.

- The behavior of the molten steel was limited to the computation of the temperatures in the melt. The steel melt is a complicated mixture of inclusions and gas bubbles, and the modeling of its exact behavior would require the solving of the Navier Stokes equations. The exact behavior of the molten steel is not relevant to the stresses in the shell. In addition, the behavior of inclusions were ignored as well. This means that the amount and effect of impurities was assumed to be negligible.
- The movement and vibrations of the slab or billet were not taken into account. The billet was presumed to be in a quasi-steady state, in which the only movement was due to casting speed, which was handled by the manipulation of the boundary conditions of the billet. The contact between the mold and the billet and between the supporting rolls and the billet were assumed perfect in thermal analysis and were used as boundary conditions in mechanical analysis.
- The microstructure of the solidifying shell consists of pillar like structures as in Figure 3, the properties of which are not completely isotropic. This microstructure is ignored, and the behavior of the billet is assumed to be isotropic. The assumption for the solidifying shell to be isotropic can be assumed relatively accurate [26].
- The finite element formulation in this work is designed for small strains only, and is not accurate if strains increase above a few percent. This, however, is not a problem, as

cracks will form in solidification problems before such strains occur, which can be estimated with appropriate damage criteria within the code [18].

Even though the continuous casting of steel is a continuous process, the Matlab code treats it as a batch process. This means that the code performs thermal and mechanical analysis for a cross section that passes through the process as a simplification of the quasi-steady state that is present during production in industrial continuous casting processes.

4.2 The Structure of the Model

A concept image of the structure of the code created for this work is presented in Figure 12. At the beginning of the simulation, the parameters are defined, including the accuracy of modeling – termination parameter β discussed in appendix 8.2 – and the values for various constants such as emissivity and density. The mesh is also generated at this point, as is the initial temperature distribution.

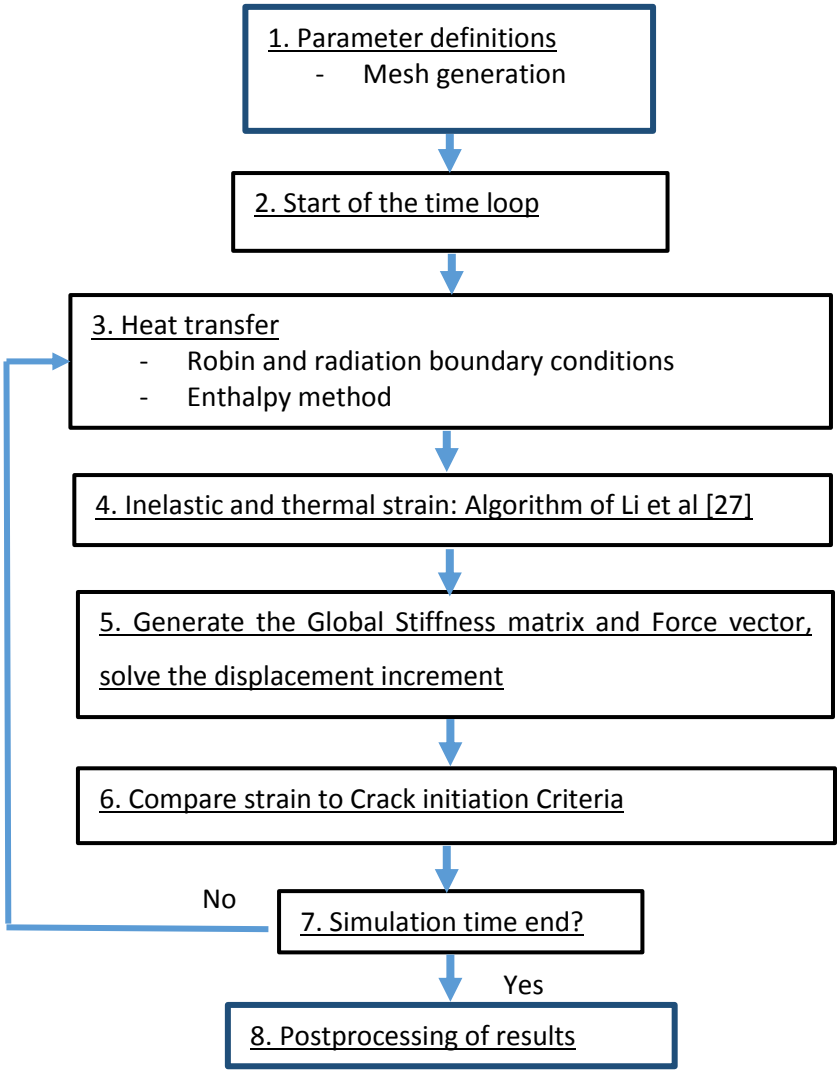

Figure 12 Structure of the Matlab code.

After the definitions, the time loop is initiated. The Heat transfer is calculated first, using the algorithm below. The steps 2-8 take place within a single time step, iterating the temperatures until convergence is achieved.

1) Start the calculation in this time step.
2) Determine the heat conductivity for the current temperatures T_i^m
3) Calculate the effective heat transfer coefficient h_{ef} using equation (*21*).
4) Generate the conduction matrix (equation (*25*)), the enthalpy matrices ((*38*) and (*39*)) and the vector **r** (*29*).
5) Solve the linear system of equations (*37*) for $\boldsymbol{T}_i^{m+1}$.
6) Update the enthalpy values to H_i^{m+1} with equation (*32*) to correspond to the new temperatures. In addition, update the heat capacity $\left(\frac{dH}{dT}\right)_i^m$.
7) Fix the temperature back on to the plot as in Figure 9 on page 28. If the previous phase was liquid, the enthalpy is changed instead of the temperature to minimize the correction step.
8) Compare $\left|\boldsymbol{T}_i^{m+1} - \boldsymbol{T}_i^m\right|$ (appendix 8.2), if the difference is small enough, move on to step 9. Otherwise, return to step 2.
9) Determine H_i^n necessary in equation (*39*).
10) Move to next time step and start from step 1.

The heat transfer calculation is followed by the mechanical analysis, in which the thermal strains are first calculated based on the new temperature distribution, followed by the algorithm of Li et al presented on page 36-37. After the inelastic strain increment of the current time step has been determined, the global stiffness matrix and force vector were generated, and equation (*43*) was solved for the displacements of this time step. The crack initiation condition in equation (67) was checked to see whether any cracks have been born. Next, the simulation is stopped if a predetermined ending criteria is fulfilled, otherwise the simulation of the next time step is initiated. In this work the simulations were stopped when the mold region had been simulated, which was determined through casting speed and the mold length.

4.3 Matlab Continuous Casting Model

There is a concept image of the cross section modeled in Figure 13. Only a quarter was modeled to improve computational efficiency. The bottom and left side of the billet have radiation and convection boundary condition throughout the simulation. The right side and the top have the symmetry boundary condition, meaning that the heat gradient across the boundary is zero.

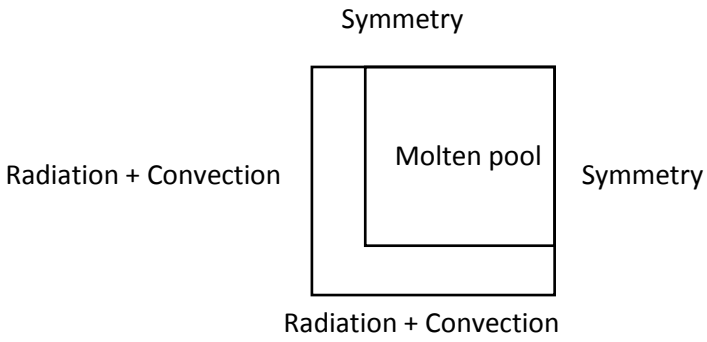

Figure 13 Concept image of the situation modeled.

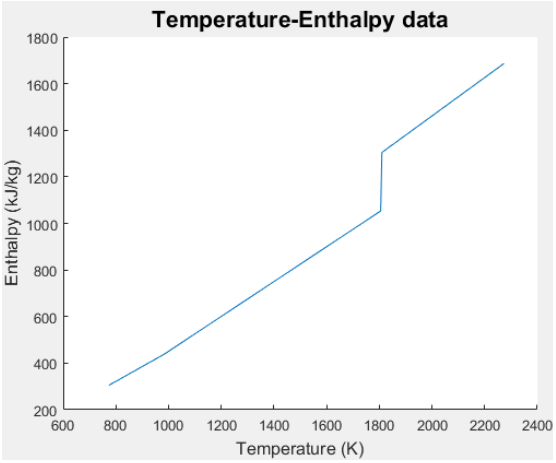

Figure 14 Temperature-enthalpy data for 0.01 % - Carbon steel. The solid-liquid phase transformation takes place over 4.4 degrees kelvin.

The enthalpy data used in modeling is in Figure 14, in which the solid-liquid phase change dominates the figure and other phase changes are not clearly visible. The data for heat conduction coefficient is presented in Figure 15. The heat conduction in the molten steel was multiplied by 5.9 to account for convection within the molten steel [3, 22, 27]. The conductivity in the liquid is constant 200 W/mK and below 1000 K 30.65 W/mK.

The various parameters are presented in Table 1. The mold length, billet size and

heat conduction coefficient were chosen to match those of the work of Li et al [27] to make literary comparison more reliable.

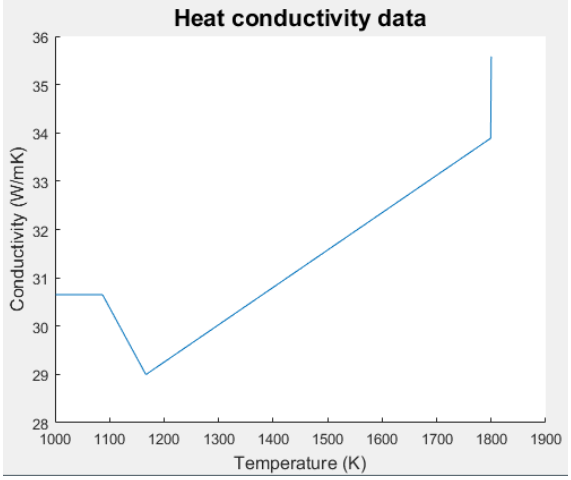

Figure 15 Thermal conductivity as a function of Temperature for a low carbon steel [27]. The conductivity data was slightly altered to match the liquid area into the enthalpy data. The conductivity increases linearly from 33.9 to 200 between 1800 K and 1809.9 K.

The time step for thermal analysis was 0.05 seconds and for mechanical analysis 0.0005 seconds. The temperatures were interpolated between two thermal time steps to calculate the temperatures at mechanical analysis time steps. Mechanical analysis requires shorter time steps due to the thermal stresses, which can be significant even at short time steps and the algorithm used to calculate strain rate can have problems with large stresses at a single time step.

Table 1 Parameters used in modeling.

Parameter	Value	Parameter	Value
Elemental grid	30x30	Casting speed [22, 24]	1.3m/min
Billet size [27]	0.12m x 0.12 m	Heat conduction coefficient h in mold	2000 W/m^{2}K
Thickness of the cross section	$3.5 * 10^{-4}$m	Mold length [27]	0.7 m
Pouring Temperature	1820 K	Density in thermal analysis	7830 kg/m^3
Surface emissivity, ε [19, 22]	0.5	Time step for thermal analysis	0.05 s
Temperature of cooling medium, T_{ext}	300 K	Time step for mechanical analysis	0.0005 s
Carbon content	<0.01 %	Convergence criteria β in heat transfer	10^{-3} K

The boundary conditions in mechanical analysis are described in Figure 16. The boundary condition on the right side and on the bottom of the billet is symmetry similarly to heat transfer analysis. In mechanical analysis, symmetry means that the displacement component in the

direction perpendicular to the plane of symmetry is zero – this means that the displacement in x- direction equals zero on the right side and y- direction on the top.

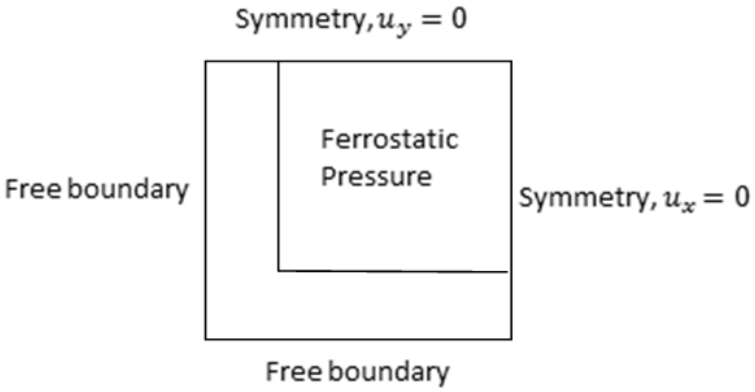

Figure 16 Image of the boundary conditions on the billet. The ferrostatic force acts as a force on the solid shell, forcing it to bulge outwards.

In the mechanical analysis, only elements that had four solid nodes were treated as solid. Nodes that did not belong to a single completely solid element were marked as liquid and their displacements were prescribed as zero, even if those nodes were solid themselves.

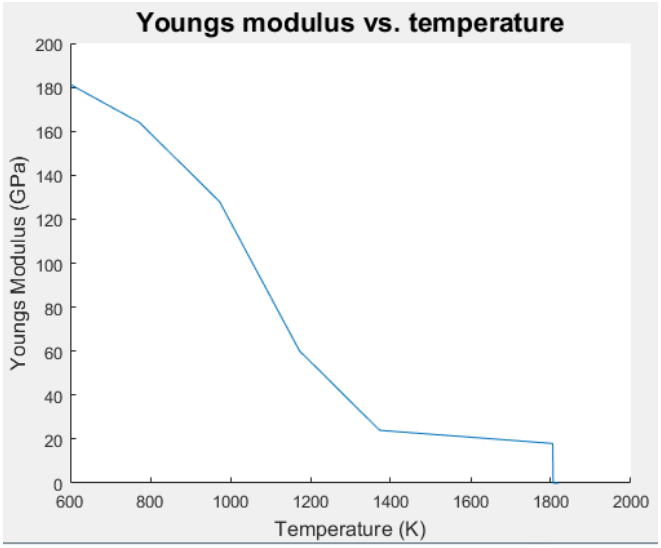

Figure 17 Young's modulus as a function of temperature for a low carbon steel [3]. The Young's modulus is zero for liquid nodes above 1809.9 K, which doesn't affect modeling as only elements which only have solid nodes are treated in the code.

Nodes with prescribed zero displacements were treated as follows: the rows and columns corresponding to that degree of freedom (x- direction on the right boundary and y-direction in the two specific nodes: both directions in liquid nodes) were set to zero in the matrix [**K**] and the corresponding force was set to zero in {F}.

The Young's modulus is presented as a function of temperature in Figure 17. The

Young's modulus does not decrease linearly to zero near the melting point, as very recently solidified nodes could then have extremely low Young's moduli. This would lead to a very low stiffness for their elements, and significant changes in stiffness of adjacent elements can lead to problems when inverting the global stiffness matrix [**K**]. The mechanical analysis was initiated when there was a solid element shell on the three free boundaries. Both the liquid and solid elements were assumed to move at the nominal casting speed.

5. Results and Discussion

5.1 Heat Transfer Model Validation

The thermal analysis section of the code was validated using Comsol Multiphysics version 5.2, which is from hereon called just Comsol. Three Comparison cases were compared between Matlab and Comsol, which included Dirichlet, Robin and Radiation boundary conditions. In addition, one validation case included phase change.

In this comparison it is important to note that Comsol and the Matlab code calculate phase change differently from one another. Comsol uses the apparent heat capacity method and the Matlab code created uses the enthalpy method. The difference of these two is most significant in the mushy region, when the temperature is between the solidus and liquidus. There the derivatives of the temperature-enthalpy curve – meaning heat capacity – is in apparent heat capacity method

$$c_{p-apparent} = \frac{\int_{T_s}^{T_l} C(T,\alpha)dT + H_f}{(T_l - T_s)} \quad (67)$$

where C(T,α) is a heat capacity function, H_f the latent heat of fusion and T_l and T_s are the liquidus and solidus temperatures. In the enthalpy method, the heat capacity in the mushy zone is

$$c_{p-enthalpy} = \frac{dH}{dT} = piecewise\ function \quad (68)$$

which is predetermined in the piecewise linear temperature-enthalpy data. To put it simply, in the apparent heat capacity method the derivative is calculated and in the enthalpy method it is predetermined. Comsol uses the phase fractions (liquid fraction α and solid fraction 1-α) and the heat capacities of the liquid and solid phases to calculate the heat capacity in the mushy region. In the enthalpy method, the derivative is known for the whole temperature range. This difference in phase change calculation can further be elaborated by the fact that the liquid fraction is calculated in different ways. In the enthalpy method, the liquid fraction varies linearly according to equation

$$\alpha = \frac{T - T_s}{T_l - T_s}. \tag{69}$$

On the other hand, in Comsol the changes in liquid fraction have been smoothed by specific smoothing functions as in Figure 18.

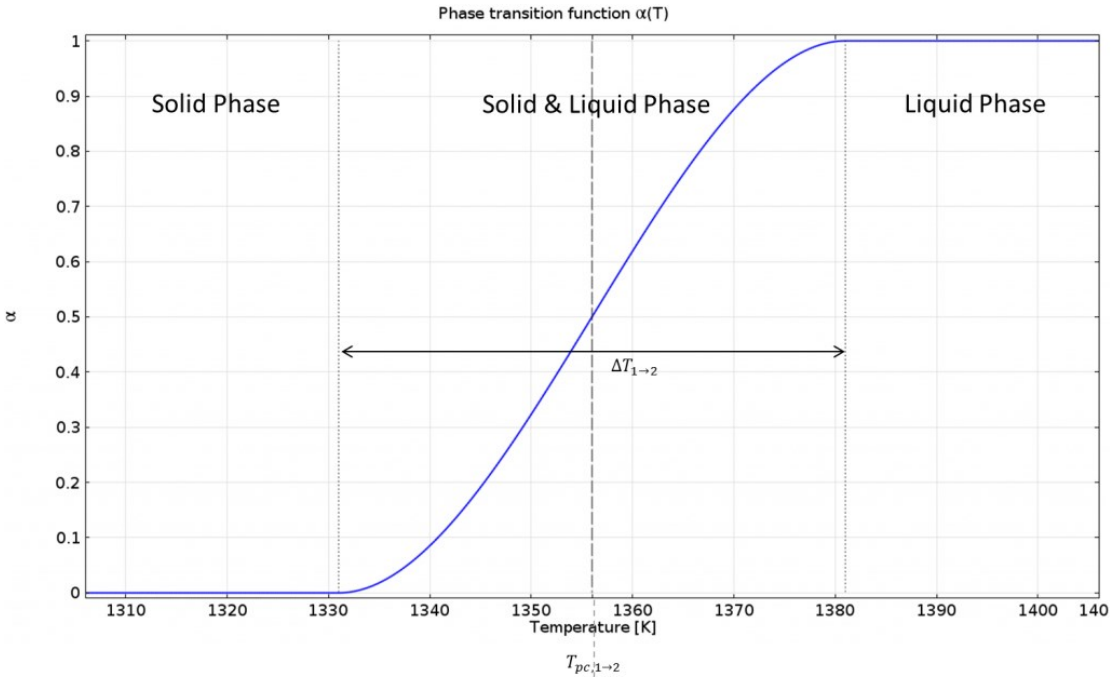

Figure 18 Liquid fraction in Comsol Multiphysics [46]. The temperature range is not the same that was used in validation.

The apparent heat capacity method softens the drastic changes in the heat capacity at the phase transitions, which gives the curve a continuous derivative [46]. The enthalpy method used in this work, on the other hand, is a piecewise linear method which has more drastic changes in its heat capacity.

However, these differences in the way of modeling phase change become smaller the wider the phase change area is – the most significant of these areas is the liquid to solid transition, the area of which depends strongly form carbon content, ranging from 5 K to even 100 K with carbon contents of over 1 percent [30].

Table 2 Parameters used in heat transfer model validation. The same density was used for both liquid and solid phases. The billet size and elemental grid refer to the size in figures and the element grid within the figure.

Parameter	Value
Billet size	0.2 m x 0.2 m
Elemental grid	30 x 30
Time step	0.1 s
Heat conduction coefficient, k	30 W/mK
Density of steel, ρ	7830 kg/m^3
Phase change temperature	1760-1810 K
Latent heat	251 kJ/kg
c_p in liquid	600 J/kgK
c_p in solid	450 J/kgK
c_p in mushy zone in Matlab	5020 J/kgK
Convective heat transfer coefficient, h	200 W/m^{2}K
Temperature of cooling medium, T_{ext}	300 K
Surface emissivity, ε	0.5

For the purpose of this treatment, the results of Comsol will be considered correct and all derivation from the results computed with Comsol will be referred to as error. However, it is important to keep in mind the different methods to model phase change when inspecting Figures 27-29, especially near the solidifying fronts.

The validation of the Matlab heat transfer model was done in three ways. First with a Dirichlet boundary condition without phase change, then with Robin and radiation boundary condition without phase change and finally with Robin and radiation boundary conditions and phase change.

First, however, let us inspect the error in a steady state situation when compared to Comsol, which is presented in Figure 19. The boundary conditions are as follows: the temperature is set to 1500 K on the right, 900 K on the upper and lower right corners and 300 K on other boundaries.

As can be seen in Figure 19, the error is very low in steady state solution, on the order of 10^{-6}. The error is highest in the center region of the billet. Additionally, it can be noted that the high error region is slightly on the left, signaling that the solutions near the high temperature boundary on the right are slightly more accurate than the solutions closer to the other boundaries.

Next, let us inspect the results with the Dirichlet boundary condition. The RMS error, Arithmetic mean error and maximum error are presented as a function of time in Figure 20.

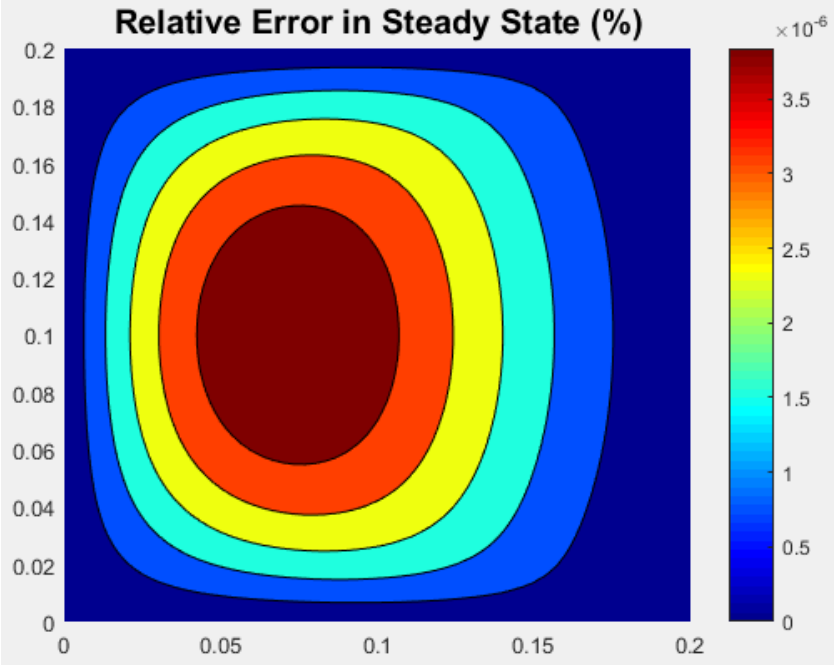

Figure 19 Relative error In Steady State

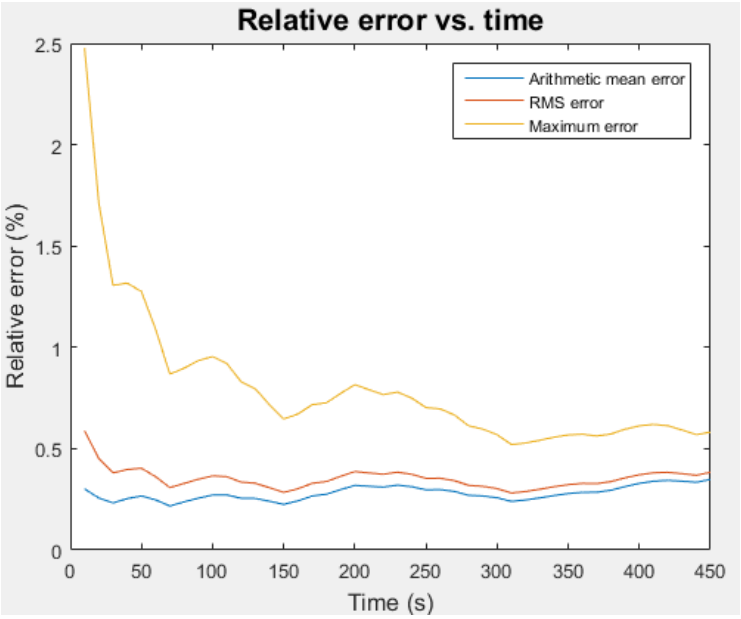

Figure 20 Relative error as a function of time with Dirichlet boundary condition.

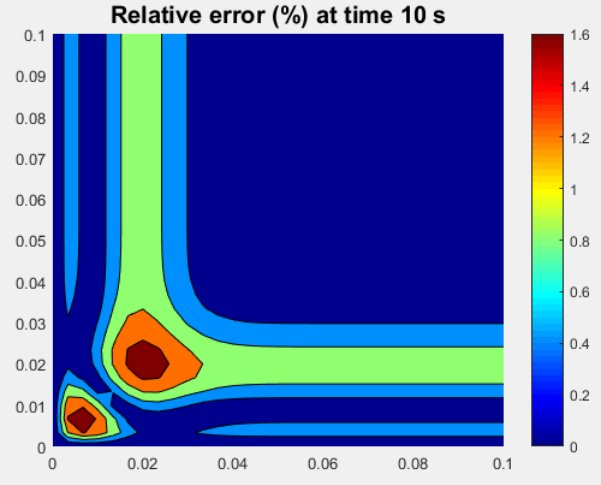

Figure 21 Relative error at time 10 s with Dirichlet boundary condition.

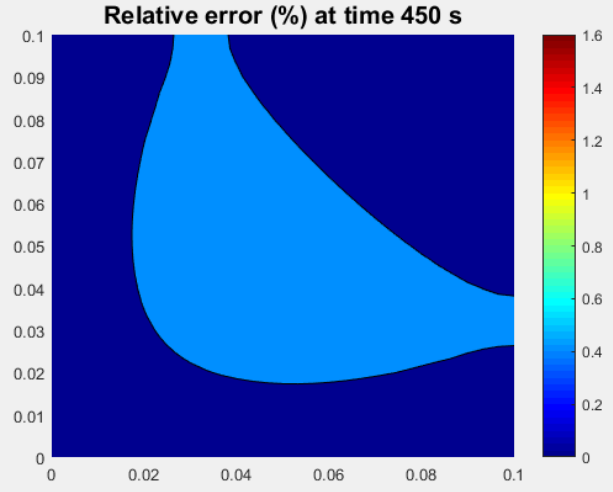

Figure 22 Relative error after 450 s with Dirichlet boundary condition.

In the Dirichlet analysis, the boundaries of the billet were set to the temperature of 300 K, and the right edge was treated as a symmetry plane. C_p was kept constant at 650 J/kgK, otherwise constants are the same as in Table 2. The error is the most significant in the beginning, and it decreases relatively quickly. The maximum error is around the lower left corner of the domain, as is clear in Figure 21. After 450 seconds the error has decreased to a quarter of what it was at the beginning of modeling, as presented in Figure 22.

The reason for the significant error at the beginning of simulation can be erratic fluctuations often present in FEM analysis especially near large gradients. At the beginning of simulation, there is a significant temperature gradient near the corners and at the boundary. This large gradient is caused by the drastic temperature difference between the surface (300 K) and the temperature of the nodes near the surface (near initial temperature 1820 K). Comsol might include some algorithms to decrease such fluctuations, or the fluctuations might simply differ due to different calculation algorithms or discretization's. Simple errors in discretizations, algorithms and coding are also possible in the Matlab code. Nevertheless, all three relative errors decrease as time passes, and after 450 seconds the arithmetic mean error is roughly 0.4 %, and even the maximum error is only roughly 0.6 %. This signals good accuracy when compared to Comsol.

Next, let us examine the behavior of the Matlab code with Radiation and Robin boundary conditions. Again, c_p was kept constant at 650 J/kgK, and otherwise the parameters are presented in Table 2. The RMS, arithmetic mean and maximum error of this validation are presented in Figure 23.

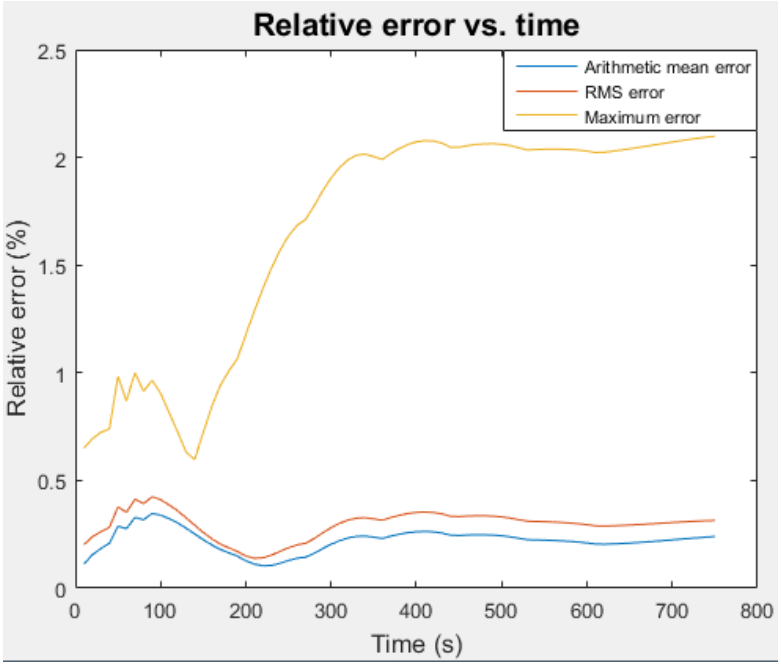

Figure 23 Error as a function of time for Robin and Radiation boundary conditions.

The error behaves quite differently from the case of Dirichlet boundary condition, as the error in the beginning of simulation is now near the minimum error at time 140 s. As before in the case of Dirichlet boundary condition, the maximum error is near the lower left corner at the beginning of simulation as in Figure 24. As time passes, the error near the corners becomes more significant and after 300 seconds the error near the corners is significantly larger than in the other regions of the billet. The error distribution after 750 seconds is presented in Figure 25. After 350 seconds the error becomes almost constant at that time and can be represented with Figure 25. The error is quite low, below 0.2 % through most of the cross section, but reaches roughly 2 % in the lower left corner.

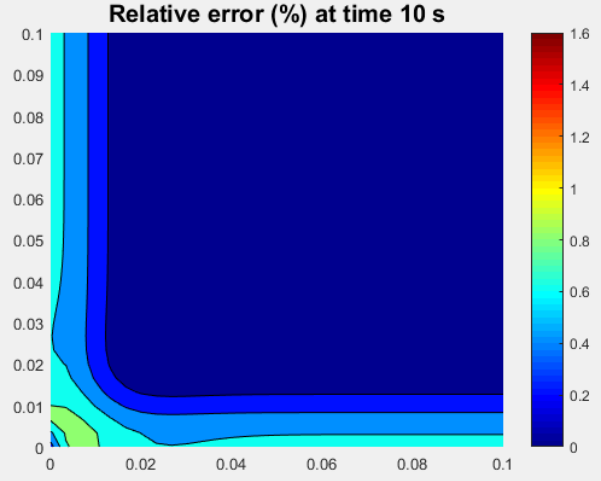

Figure 24 Relative error after 10 seconds in Radiation and Robin boundary condition validation.

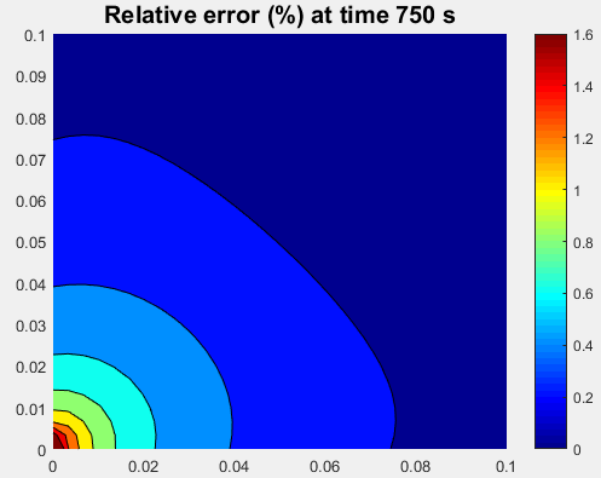

Figure 25 Relative error after 750 seconds in Radiation and Robin boundary condition validation.

A possible reason for this error near the corners could be that Comsol treats the discretization of the corner node differently than the Matlab code. Despite the significant error near the corner, the arithmetic mean and RMS errors are very low, both well below 0.5 %. In conclusion, the Matlab code works with acceptable accuracy in Robin and Radiation boundary condition situations.

The validation for the phase change simulation was performed with a simplified enthalpy data with only two phases, the heat capacities of which are presented along with other parameters in Table 2. The enthalpy in the temperature range 1760-1810 K in the Matlab model was simply a linear line between the enthalpies of the solidus and liquidus, whereas the shape of the data Comsol used is similar to Figure 18. Figure 26 plots the maximum, the arithmetic average and the root-mean square error between the Comsol and the Matlab results.

It is clearly visible from the results that the results for the temperatures diverge from one another as time passes. This can be explained, however, with the different methods to calculate

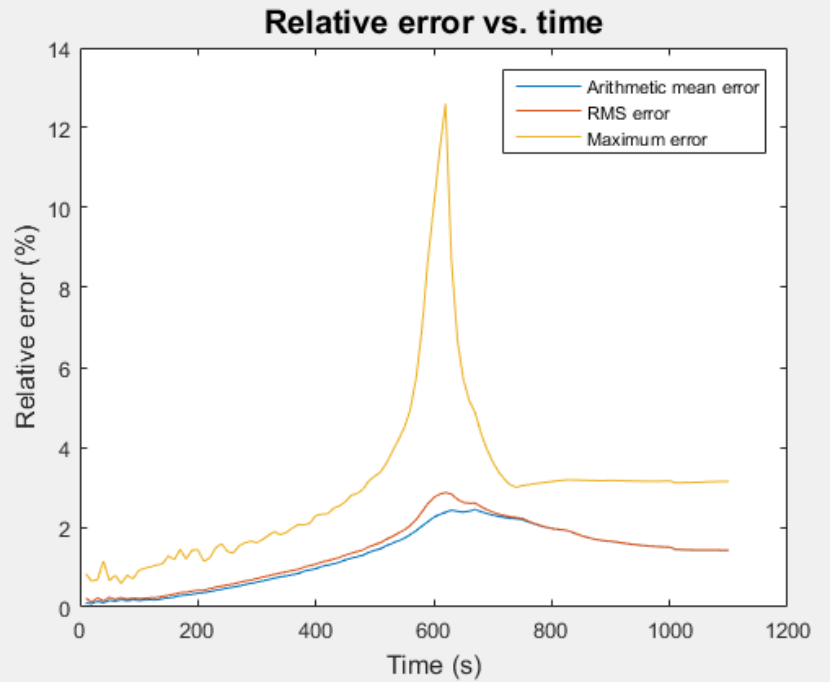

Figure 26 Relative error as a function of time. The peak is located at the time when the phase transition fronts meet in the middle of the billet.

the phase transitions. In fact, if the errors between the Matlab and Comsol results are plotted on to the cross section of the billet, we get a trend as in Figures 27-29.

In Figures 27-29, it is obvious that the highest errors are concentrated on the phase transition front. When 600 seconds have passed, which is close to the maximum peak error in Figure 26, the solidifying fronts meet in the middle of the billet as in Figure 28. The high error in the phase transition front can be explained with the different phase transition modeling methods in Matlab and in Comsol. As the phase transition is not smoothed in the Matlab code, the temperatures of the nodes very near to the phase transition temperatures will be different from those according to Comsol. C_p will be much smaller for Comsol as well, as the smoothing in the

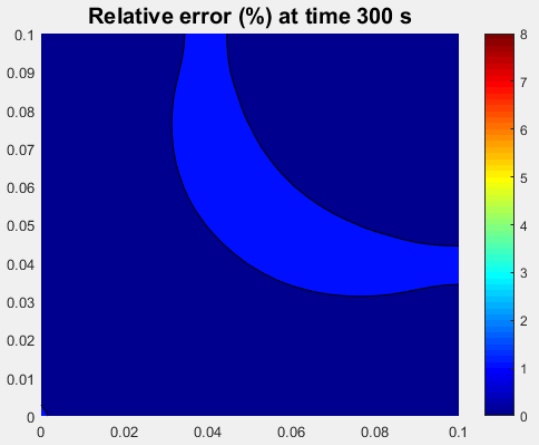

Figure 27 Relative error between the Matlab and Comsol results when 300 simulated seconds have passed.

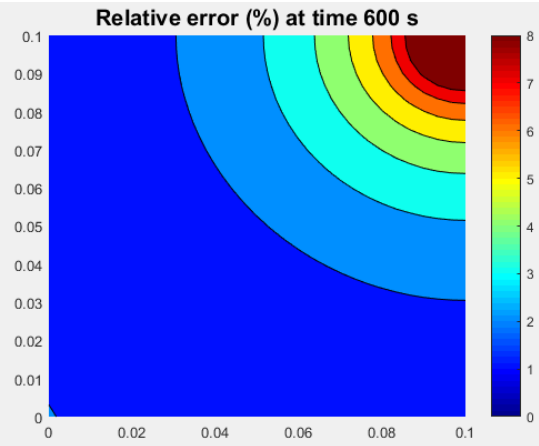

Figure 28 Relative error between the Matlab and Comsol results when 600 seconds have passed

enthalpy-temperature curve effectively reduces c_p near the phase transitions. Those different temperatures and c_p values lead to different thermal gradients, which are the driving force of heat transfer. It seems that the small differences in temperatures and c_p near the phase transition temperatures between Comsol and Matlab stack up and lead to diverging nodal temperatures as the phase transition front progresses.

After the liquid phase has completely solidified, the relative error drops drastically, which is apparent when comparing the errors between Figure 28 and Figure 29.

In addition to significant errors in the liquid-solid interface area, the lower left corner in Figure 29 exhibits noticeable error similarly to the Robin and Radiation validation without phase change. Despite the error being larger in the corner than elsewhere in the boundary, the errors still are below 3 % in Figure 29, showing acceptable accuracy even in the corners.

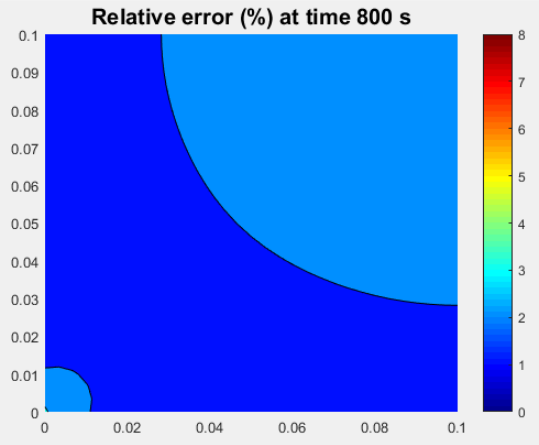

Figure 29 Relative error between Matlab and Comsol results when 800 seconds have passed.

When the mean temperature of the whole billet is compared between the Matlab and Comsol results, as in Figure 30, it is clear that the peak error coincides with the meeting of the phase transitions fronts.

From these figures it is clear that the differences are the most drastic near the phase transition front and the smallest in the molten pool and near the edges.

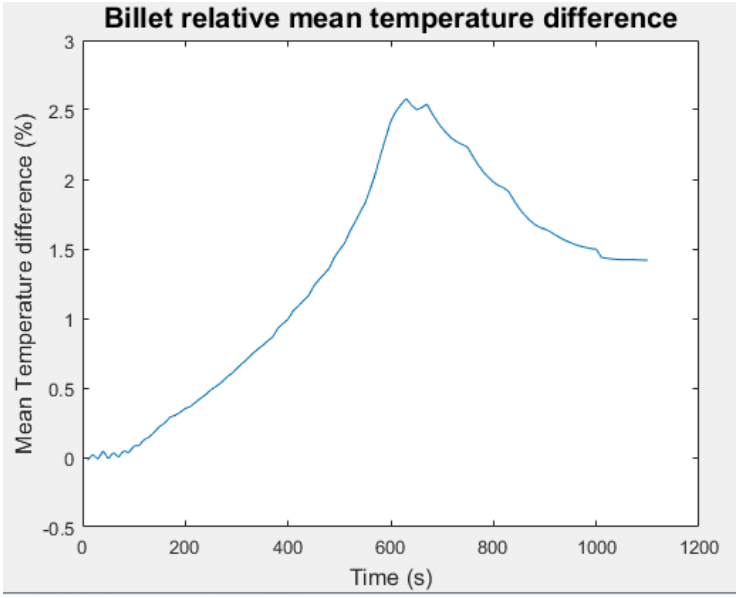

Figure 30 Relative mean temperature difference of the whole billet. The peak difference occurs when the phase transition fronts meet, after that the error decreases again.

5.2 Mechanical Model Validation

The mechanical model was validated using Comsol Multiphysics version 5.2.

Table 3 Constants and Parameters used in the validation.

Billet size (m)	0.2 x 0.1
Elemental grid	30 x 30
Time step (s)	1
Density (kg/m^3)	7830
Conductivity (W/mK)	30
Specific Heat (kJ/kgK)	450
Elastic modulus function (Pa)	$20 \cdot 10^9 - 50 \cdot T^3$
Distributed load N/m^2	10^6
Convective heat transfer coefficient, h (W/m^{2}K)	200
Temperature of cooling medium, T_{ext}	300
Heat conduction coefficient, k (W/mK)	30
Surface emissivity, ε	0.5

The constants used in this validation are presented in Table 3. They were chosen to approximate the more complex situation in actual modeling of the continuous casting.

The validation situation is pictured in Figure 31. The boundary on the right is constricted to prevent rigid body motion and there is a distributed load along the top surface. The other boundaries are free, and the thermal boundary conditions are marked in Figure 31.

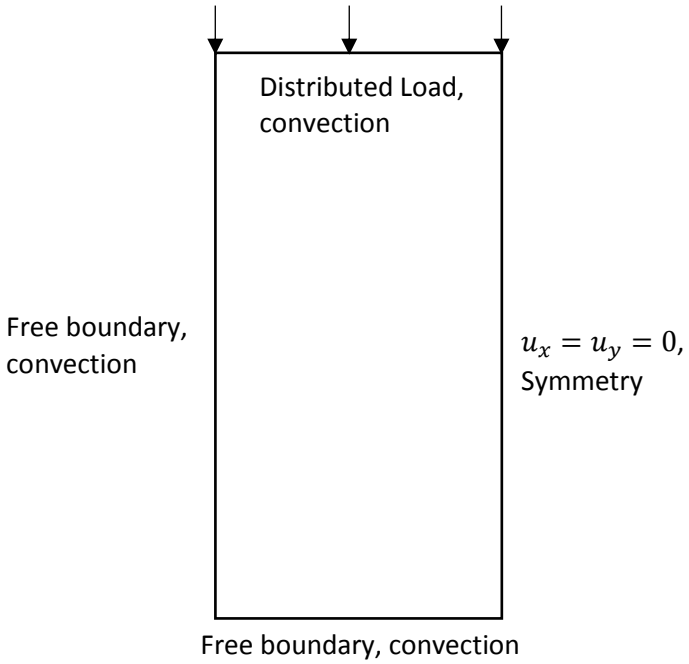

Figure 31 Elastic analysis validation boundary conditions. The boundary conditions are presented as mechanical boundary conditions, thermal boundary conditions.

The mechanical elastic analysis comparison to Comsol is presented in Figure 32. This comparison is limited to y-Displacement. The displacement in x-direction behaves nearly identically, and as strains and stresses are derived from displacements the analysis of displacements is the most straightforward way to analyze the error behavior of the Matlab code.

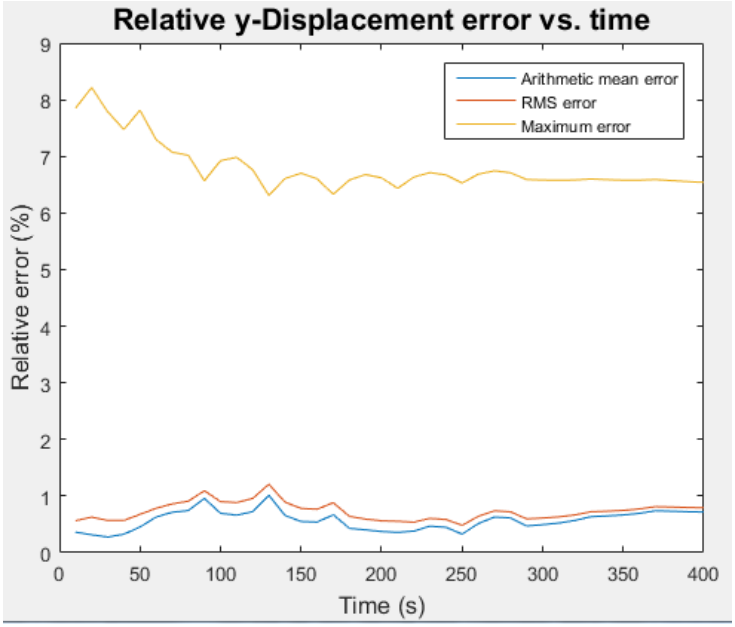

Figure 32 Y- Displacement relative error vs time. The maximum error is quite high, but the RMS and Arithmetic mean errors are more reasonable.

The Maximum error is quite high in the Y-direction displacements in Figure 32, but as can be seen from Figure 33, the maximum error is localized in the lower right corner, near where the displacements are set to zero. Comsol treats the corner elements differently from Matlab, balancing the strict boundary conditions, such as the combination of prescribed displacement and a force on the top right corner. The exact method of this balancing could not be replicated in the Matlab code.

Disregarding the high error peak near the corner, the errors are otherwise below 1 %; in fact often well below 0.5 %. This can be deemed as sufficiently low, as the errors in temperature will introduce errors into displacements as well, due to the different stiffness between Matlab and Comsol. The errors in temperature in this mechanical validation are nearly identical to the errors in Figures 23-25, as the free boundaries had a convective boundary condition in thermal analysis. This means an error of 0.05 % - 0.15 % in the areas outside of the lower and upper left corners.

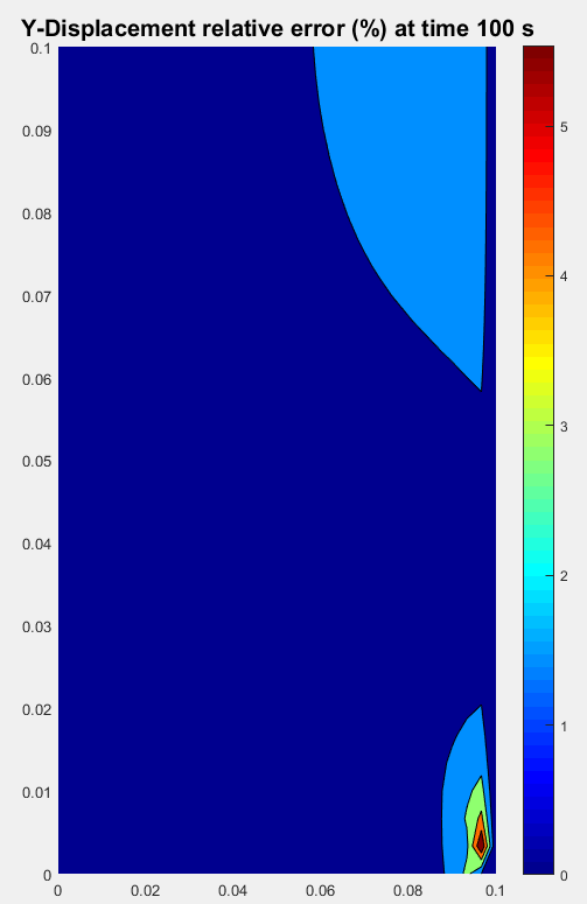

Figure 33 Y-Displacement relative error after 100 seconds of simulation.

Running elastic simulations without the coupling between the temperature and elastic modulus ($E = 20 \cdot 10^9 - 50 \cdot T^3$) will lead to an error distribution very similar to Figure 33. The highest error will in this case be roughly 6.1 percent, located in the nodes in the lower and upper right corner elements similarly to the lower right corner in Figure 33. The cause for the location and magnitude of the error are the balancing mechanisms for the strict boundary conditions in Comsol. The error outside the lower and upper right corners in this situation are primarily well below 0.5 %, mean error in the whole billet being 0.3 %. The mean error in Figure 32 ranges between 0.3 % and 1%, which suggest that the error in temperature introduces 0-0.7 % error into elastic strains. In conclusion, the thermo-elastic model can be determined to work acceptably.

5.3 Simulation of Continuous Casting

The discussion here is limited to the area of the mold, since there is a significant amount of literary data for the temperature and stress distributions at the mold exit. The simulation parameters are presented in section 4.3.

Figure 34 presents the temperature distribution of the billet cross section at the mold exit. The solid-liquid interface is located at roughly 2 centimeters from the billet-mold boundary.

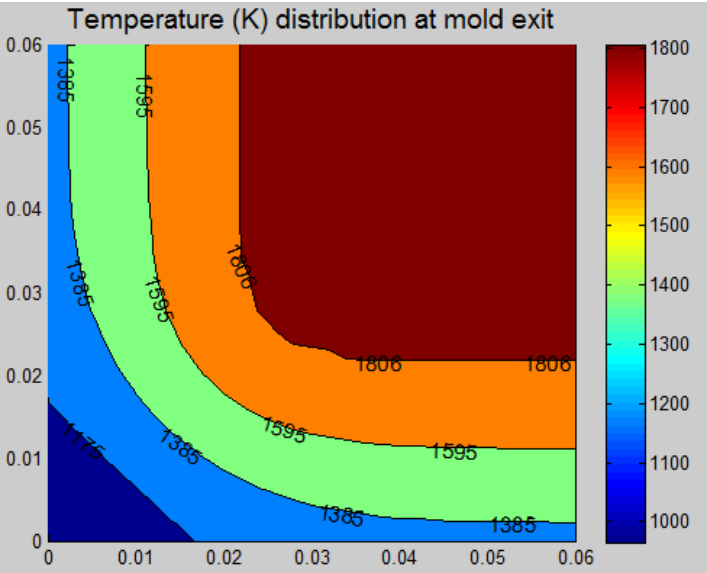

Figure 34 Temperature distribution of the billet at the mold exit.

Another temperature distribution profile, Figure 35, is provided to compare the results to the work of Li et al [27] in Figure 36. When comparing the temperature contours between Figure 35 and Figure 36, it can be noticed that they match relatively well. Figure 36 has lower temperatures in general, which can be attributed to different boundary conditions, as Li et al [3] took the gap between the billet and the mold into account. Their enthalpy data is slightly different as well, including a different solidus temperature of 1510.9°C.

In addition to slightly lower temperatures in the billet, the Figure 36 has a much wider area between temperatures 1500.72°C and 1477.02°C. This is most likely due to different temperature-enthalpy data used in the modeling of the different phases. The lack of a sophisticated modeling of the mushy zone might also have contributed to this difference. The model of Li et al [3, 27] is more sophisticated and treats the mushy zone as a phase fraction

between the solid and the liquid phase, calculating the heat conductivity with phase fractions. The Matlab model, on the other hand, calculates the heat conductivity as a simple mean of the values of the gauss integration points. Other factors may also contribute to this difference, as the temperature range of 1477.02°C -1500.72°C is not actually in the mushy zone itself.

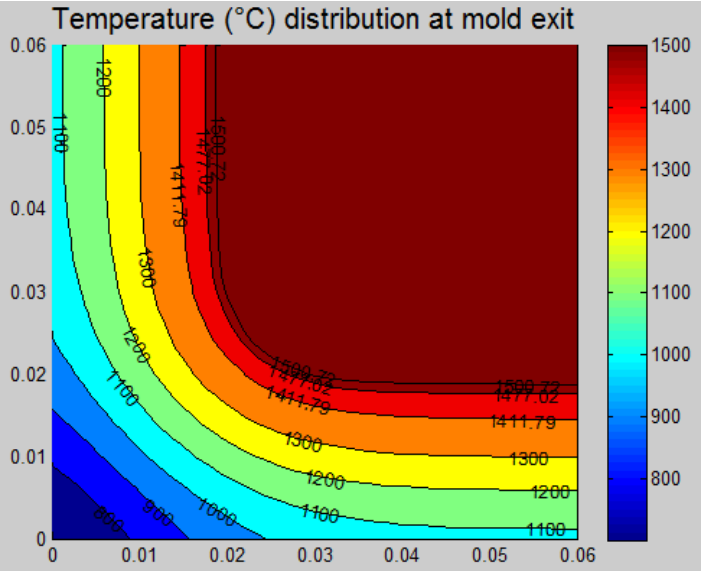

Figure 35 Temperature distribution at the mold exit. A corresponding figure from literature is Figure 36.

The carbon contents are different for the Matlab code (0.01 %) and the work of Li et al [27] (0.04 %). The difference between these percentages is quite small, so the differences caused by carbon content have been neglected in this analysis. The thermal model results matches other literary works besides those of Li et al [3, 27] quite well as well [18, 22, 47].

The temperature model can be determined to work qualitatively in comparison to the literature, at least in the mold region. After the mold, the secondary cooling would probably require more sophisticated methods to approximate the effective heat transfer coefficient than a constant value to accurately simulate the cooling water jets.

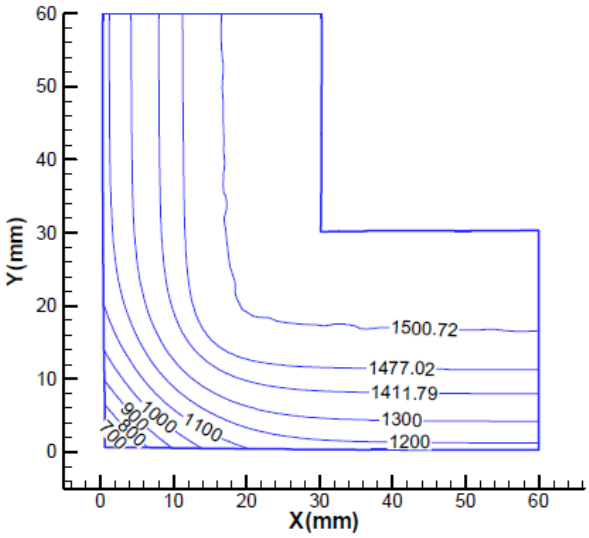

*Figure 36 Thermal contours at mold exit [3]. Casting speed of 2.2 m/min and billet section of 120mm*120mm was used. Mold length was 700 mm, mesh size 0.1mm- 1mm and time step varied from 0.0001 to 0.5 seconds. Pouring temperature was 1540 °C.*

The works of Janik et al [22] and Han et al [47] match the results of the work of Li et al [3, 27] relatively well in the magnitude of strains and stresses. This comparison focuses on the work of Li et al when making comparisons [3, 27] due to the more similar nature of this work and their work.

The plastic analysis did not function when the results were compared to the results from literature, such as Figure 37. The strain distributions were uneven due to recrystallization, which is not modeled at all in the work of Li et al [3]. Without recrystallization modeling, the values for strain are higher and their distribution is more even. However, when recrystallization was not used in the Matlab code, it led to problems with convergence and strains of 40-80 % even before the mold exit, which is unrealistic and had to be balanced with the use of recrystallization.

Increasing the critical strain, which initiates recrystallization, could improve the results to be closer to Figure 37, but the current critical strain takes assumptions of equation (*52*) better into account. The parameters in equation (*52*) are based on the assumption of total recrystallization when the strain reaches 5 %, which is better approximated with the critical recrystallization strain of 0.5 %. In addition, simulation tests with higher recrystallization critical strains yielded no better results – the strains were higher but their distribution was no closer to Figure 37.

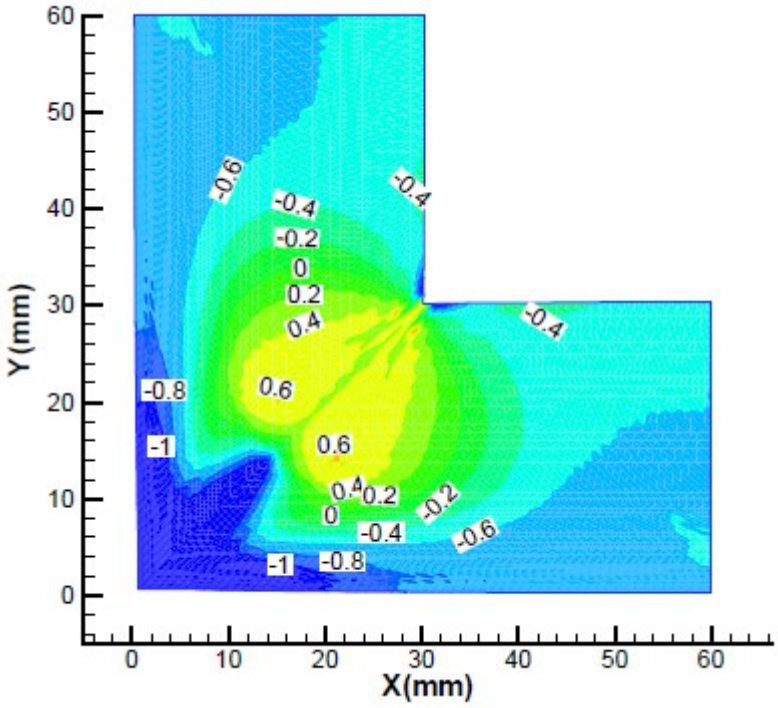

Figure 37 Strain at mold exit [3]. Casting speed of 2.2 m/min was used in this simulation. Modeling parameters were the same as in Figure 36. The unit of strain is percent (%)

Problems in numerical values of strains occurred occasionally during the simulations. The problems were extremely high strain rates in the algorithm of Li et al [27] and divergence in the local algorithm. The high strain rates lead to high inelastic strains, which in turn caused high corrective loads, leading to systematic growth of strains to values of 30-60 %. Divergence in the local algorithm lead non-numerical values for strain, which caused a multitude of other problems in subsequent calculations.

The high strain rates were encountered primarily in elements that were near the solidification front and had recently solidified. Sometimes these elements would have several molten elements next to them, leading to a high ferrostatic force and strain rates of order 10^3-10^6 depending on the inelastic strain and temperature history of the element. As high strain rates would lead to high correcting forces, which affect more elements leading to general problems in the code, the strain rate of elements with several molten elements next to them was manually reduced to a lower value if the strain rate calculated was over 10^{-2}. The strain rates in continuous casting are usually on the order of 10^{-4} [27, 36].

These problems would decrease in occurrence frequency as the element size was reduced, as the problem rises mostly due to the error of representing the solid interface with rectangular elements. In the actual solid-liquid interface in continuous casting, there would be no sharp corners as the ones present in the solid elements immediately after its solidification, but a continuous curved interface which would expand slowly. The fact that the solidification is discontinuous and happens an element at a time causes some error in analysis, especially due to the higher ferrostatic pressure in those elements. These problems with the elements at the solid-liquid interface might be smaller with a triangular mesh used by Li et al [27], but the element shape alone cannot account for the error.

In addition to problems caused by the geometry of the mesh, there were also convergence problems with the local algorithm used in step 3 of the strain rate algorithm (appendix 8.6). These convergence problems led to the algorithm diverging and the values of stress and strain would increase with each cycle of the algorithm until Matlab's capacity to treat them as numbers ended, which would lead to non-numeric values and termination of the simulation. These problems were treated with setting a limit to how high the stress and strain could increase within the algorithm, and manually setting the values of stress and strain if they diverged. The manual set was treated as keeping the values of inelastic strain and stress the same as the inputs into the algorithm. This manual setting of values causes some error, but due to the rare occurrence of the divergence within the algorithm, the error is deemed to be small.

The cause of the divergence problems in the algorithm is not certain. It is possible that at certain strains and stresses the values for the strain rate and the derivatives of stress and strain lead to problematic values and divergence due to the equations (*52*)-(*54*). However, this is unlikely, as Li et al [27] did not report similar problems, and the original article claimed the algorithm to be stable with nearly all stress and strain functions as well. The most likely reason for this divergence are mistakes in coding the algorithm into Matlab.

Finally, recrystallized elements also caused problems for the local algorhitm, as after the element has recrystallized the inelastic strain is lower than at the previous time step, but when calculating the estimates for stress and strain (equations (*116*) and (*117*) in appendix 8.6) the derivatives of the last time step used in the estimates have not changed. Unfortunately, due to

the nature of those derivatives they cannot simply be scaled based on the recrystallized portion X, and no solution for these convergence problems in the local algorithm was found.

In Figure 38 there is an image of the total Von Mises stresses (equation (57)) in the billet. Take note that the magnitude of these stresses is not realistic, as they factor in the total thermal stress even though the mechanical simulation was started when there already was one layer of solid elements; thus, thermal strain had already accumulated before mechanical analysis was started. In addition, Figure 38 includes the total inelastic stress, which during modeling was reduced by recrystallization.

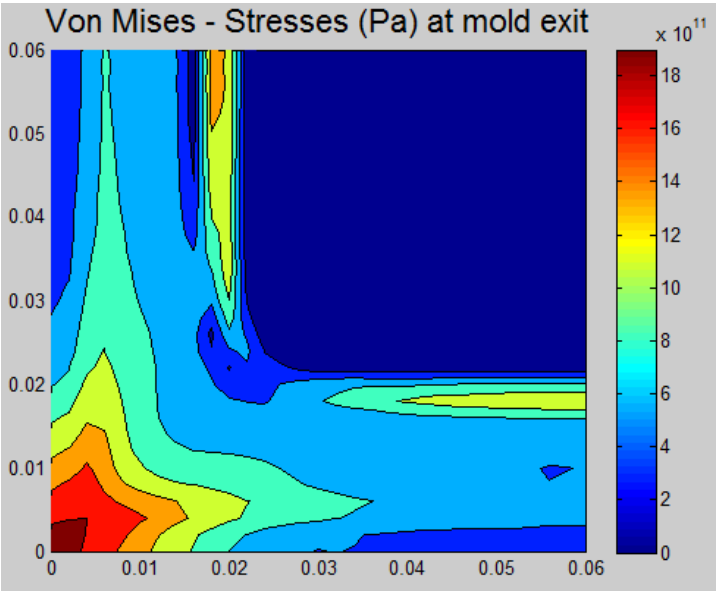

Figure 38 Total Von Mises Stresses at mold exit. The figure includes total thermal stress and total inelastic stress and recrystallization is not taken into account, which explains the high stresses.

Figure 38 shows that despite the poor functionality in comparison to other literature, the Matlab code shows some promising behavior. The high stresses in the lower left corner can be explained by the lower temperature in the corner and as such, higher stress due to cooling of the billet. In addition, there is a natural singularity in the corner, where stresses would likely approach infinity in the corner if the elements size were to be refined infinitely. There is an area of higher stress on the phase transition front, showing where the total inelastic stress is the highest. Figure 38 should be completely symmetric in relation to a diagonal from the lower left corner to the upper

right corner, which is obviously not the case. This can be attributed to the problems in the plastic analysis.

Next, let us examine the crack initiation conditions presented in equations (65) and (66).

At first, let us examine the most rudimentary crack initiation condition as in equation (66). The time steps every node spent in that temperature range were summed and plotted into Figure 39, which shows the highest risk of hot tear cracks is in the phase transition front. This method of plainly summing the time steps spent in the risky temperature region will not highlight the current phase transition front as strongly, as the current front has only just recently solidified.

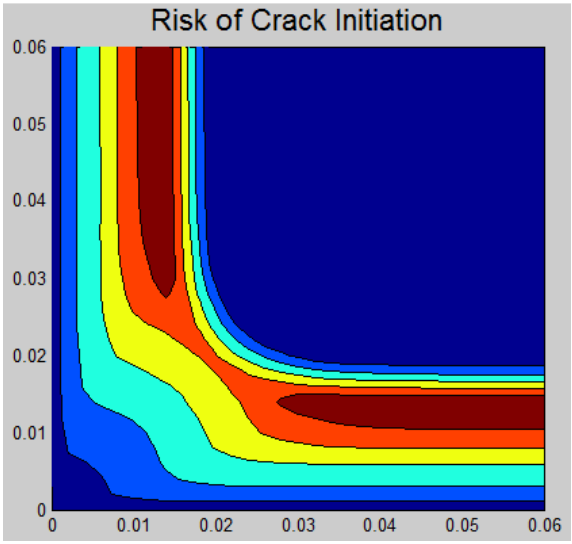

Figure 39 Risk of crack initiation at the mold exit. Dark brown areas have higher risk and dark blue areas have zero risk of crack initiation.

According to Figure 39, the highest risk of hot tear cracking is on the phase transition front, and the corner of the billet has lower risk than the edges of the billet. This is due to the more efficient cooling of the corner, and thus a shorter time period is spent in the temperature range near the solidification point.

Another cracking criterion, equation (65) links the strain rate and the temperature range to the risk of cracking. The results of that cracking criteria are plotted in Figure 40. Figure 40 doesn't show as high of a risk to the current phase transition front as those elements only just solidified, as was the case in Figure 39. Both Figure 39 and Figure 40 predict the highest risk on to the phase transition front, and a smaller in the more rapidly cooled corner. The uneven strain distribution

results contribute to Figure 40, as especially the darkest brown region is uneven and not symmetric as it logically would be.

The results that the areas near the phase transition front are most at risk from hot tear cracking are meant as a confirmation of the hot tear cracking criteria, as hot tear cracks always do initiate in the mushy region between the solid and the liquid. Figure 39 and Figure 40 show that despite the poor correlation of the strains themselves with the results of the literature, the crack prediction functions to some extent.

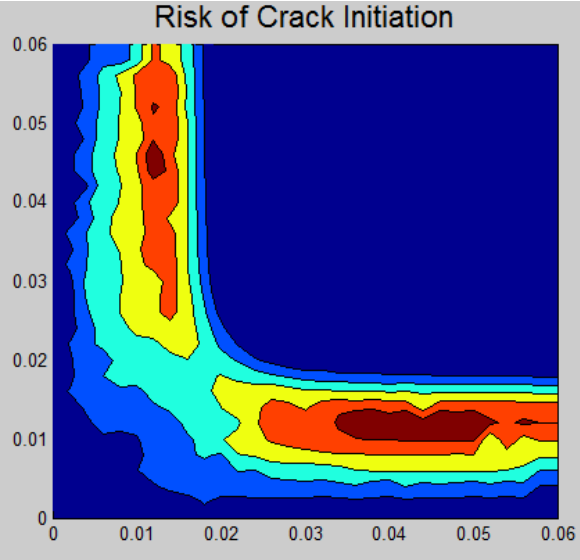

Figure 40 Risk of crack initiation with cracking criteria in equation (65). The figure represents the risk at the mold exit.

If the crack prediction is analyzed within each time step, the region where there is a risk of crack is always on the phase transition front, but the crack analysis cannot compare the regions in order of highest risk. It can only determine if there is risk of material failure or not as the risk prediction criteria is either a temperature range (equation (66)) or if the strain is over certain value (equation (65)).

The crack initiation prediction in its current form is not sophisticated enough to immediately recognize when a crack is initiated and accurately predict in which situations cracks may arise. Instead, it simply provides an approximated risk of the most risky regions. In the future, the crack prediction model could be improved to be capable of analyzing crack initiation more accurately by incorporating microstructure model into the existing Matlab code, as the analysis of hot tear

cracks requires knowledge of the dendrites in the mushy region, which in turn requires microstructure modeling.

The work of Li et al [3] places the highest risk of failure away from the phase transition front much closer to the surface of the billet, as can be seen in Figure 41. Their work suggests that after the billet has solidified to be thicker than 10 mm the stiffness of the billet is enough to prevent critical strain in the mushy region as well, which again lowers the risk of failure. It is worthwhile to note that the failure risk analysis of Li et al [3] is slightly more complex than the one in this work, partly enabled by their more sophisticated treatment of the mushy region. This would suggest that the crack prediction analysis of the Matlab code is correct to place the high risk on the phase transition front, but after the billet shell thickness is over 10 mm the code should no longer detect risk of failure. A possible cause for the high risk at the phase transition front even after the shell thickness exceeded 10 mm are the uneven strain distributions provided by the plastic analysis of this work. The high strain rates at the phase transition front in combination with the cyclic recrystallization makes the strain history vastly different from the work of Li et al [3], making comparisons with Figure 41 more difficult.

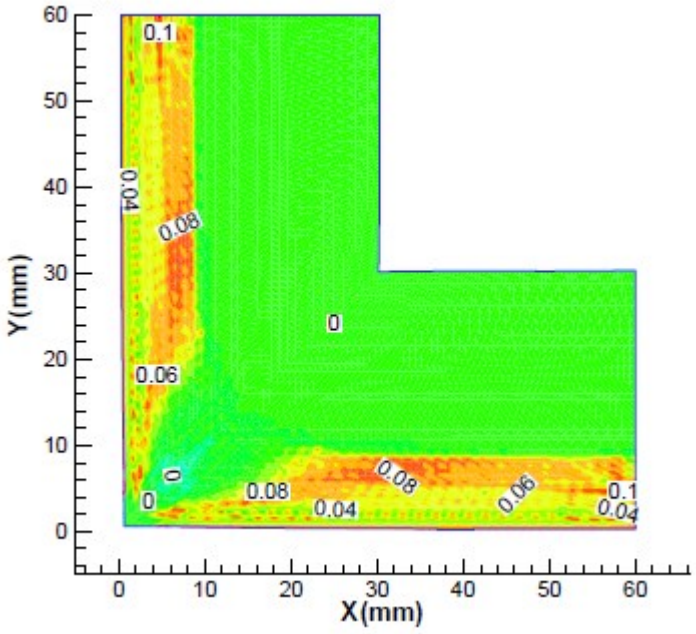

Figure 41 Damage strain at mold exit [3]. The higher values signal higher risk of failure.

Other works regarding hot tear cracking often place the highest risk of failure to the border of austenite and ferrite phase or other intergranular boundaries [48], however these could not even be considered in this work, as the microstructure was not modeled. The crack initiation analysis of the Matlab code should possibly be altered in the future to take the phases into account, which would necessitate the modeling of the microstructure of the billet.

6. Conclusions and Recommendations

The purpose of this work was to create a Matlab code capable of simulating both the thermal and mechanical aspects of the Continuous Casting of steel. Such simulations would expand the previous works into the subject, and enable the prediction of hot tear cracks during the process.

As was presented in the literary study, there exist various approaches to simulating the continuous casting of steel. This work incorporated a Lagrangian frame of reference by modeling a cross section of a steel billet in the mold region of the continuous casting process. The results were compared to the simulation results of the original works that had used the fundamental mechanical analysis tools also used in this work.

The heat transfer model functioned well when compared to results from Comsol Multiphysics. The greatest error when compared to Comsol was located in the corner, which could be caused by different discretization of the corner node in Comsol than in Matlab. However, even the errors in the corners were relatively low, staying under 3 %. The error in phase change modeling was higher in comparison to Comsol, reaching a maximum of 14 % when the phase transition fronts met in the middle of the billet. After that, the error dropped drastically, signifying that the different methods of modeling phase transition was the primary cause of high errors in comparison to Comsol.

The heat transfer modeling of continuous casting using the Matlab code was determined to work acceptably. Despite being significantly less sophisticated than the thermal analysis models in literature, the temperature contours corresponded fairly well to the ones in literature. The most significant differences between the model in this work and the ones in literature are the modeling of the gap between the billet and the mold, and the modeling of the mushy region as a phase fraction of the solid and the liquid. Additionally, the differences in the enthalpy data and elemental grid size and shape also contribute to the differences between the results of this work and literature.

The thermo-elastic analysis of the Matlab code functions well, the maximum error being roughly 6.5 % and the errors being below 1 % in most of the simulated region. These errors include some error from the heat transfer analysis. In addition, Comsol uses different discretization for the

corner node, or it uses balancing algorithms near the strict constraints, which are not used in the Matlab code.

The plastic analysis of the billet within the mold does not work as well as the thermos-elastic analysis. The strains are much higher than in the literature unless recrystallization is incorporated into the model. When recrystallization is also used, the strains are lower than in the comparable literary works, and their distribution does not resemble the literary results. Recrystallization reduces the strains which also affects the distribution, and can be accounted for some of the error when compared to literature, but there are some other problems with the code that necessitate the use of recrystallization to reduce the strains.

It was determined that the most likely cause for the very high inelastic strains when recrystallization isn't present are mistakes in the incorporation of either the local stress-stain algorithm or the general mechanical strain rate algorithm. The local stress-strain algorithm showed convergence problems with certain values of stress and strain as well as recrystallized elements, the first problem of which is not mentioned elsewhere in literature, suggesting coding mistakes. The strain rate algorithm had problems with elements in the phase transition front that had several molten elements next to them. These elements would have very high strain rates, leading to very high inelastic strains and problems in convergence and massive corrective forces. The strain rate of such elements in the phase transition front was for those reasons manually lowered. This problem decreased in occurrence when the element size was reduced.

Unfortunately reducing the element size significantly was not an option. The computational efficiency of the Matlab code is poor enough to make it not worth the additional computational cost to decrease the element size. On a normal desktop computer, it takes nearly 40 hours to simulate the mold region with 900 elements if the mushy region is not modeled. If the mushy region is also treated, that time is increased to nearly 150 hours. Increasing the number of elements, which would be beneficial especially in the mechanical analysis, leads to extremely long modeling times. The poor computational efficiency could also be improved by translating the Matlab code into C code, for example, which would be highly recommended if sections are added into the code to make it more sophisticated in some aspects, such as adding microstructure prediction.

In conclusion, the following list of changes and additions to the code is suggested before it can be used reliably in industry applications.

1) Reworking the inelastic strain rate calculations until the results match more closely with literature.
2) Phase fraction based phase transition calculations should be considered in the heat transfer analysis.
3) The effect of the gap between the billet and the mold should be incorporated in the thermal analysis. This gap could be calculated in the mechanical analysis.
4) General plane strain should be added to mechanical analysis, to enable the calculation of strains in the casting direction.
5) If crack initiation analysis is of special interest, microstructural modeling should be added to the code.
6) Improving the computational efficiency of the code.

7. References

[1] Thomas, B.G. Modeling of the Continuous Casting of Steel—Past, Present, and Future. Metallurgical and Material Transactions, 2002. Vol 33:6. P. 795-812 .DOI: 10.1007/s11663-002-0063-9

[2] Bellet M. & Heinrich A. A Two-dimensional Finite Element Thermomechanical Approach to a Global Stress–Strain Analysis of Steel Continuous Casting. ISIJ International, 2004. Vol 44:10. P. 1686-1695. DOI: 10.2355/isijinternational.44.1686

[3] Li C. & Thomas B.G. Maximum casting speed for continuous cast steel billets based on sub-mold bulging computation. Steelmaking conference proceedings, 2002. Vol 85. P 109-130.

[4] Louhenkilpi S. & Laitinen E. & Nieminen R. Real-time simulation of heat transfer in continuous casting. Metallurgical Transactions B, 1993. Vol 24:3. P. 685-693. DOI:10.1007/BF02673184

[5] Miettinen J. & Louhenkilpi S. Calculation of thermophysical properties of carbon and low alloyed steels for modeling of solidification processes. Metallurgical transactions B, 1994. Vol 25:6. P. 909-916. DOI: 10.1007/BF02662773

[6] Louhenkilpi S. & Miettinen J. & Holappa L. Simulation of microstructure of As-cast steels in Continuous casting. ISIJ International, 2006. Vol 46:6. P. 914-920. DOI: 10.2355/isijinternational.46.914

[7] Irving M.R. Continuous Casting of Steel. Maney Publishing, 1993. ISBN 978-0-901716-53-8.

[8] Copper Mould Used In Steel Billet Casters, Jay Market Creators Pvt Ltd. Available online 21.1.2016 13:00. http://coppermouldtubes.com/

[9] Jabria K. & Godoya E. & Dumura D. & Mouchetteb A. & Beleb B. Cancellation of bulging effect on mould level in continuous casting: Experimental validation. Journal of Process Control, 2011. Vol 21:2. P. 271-278. DOI:10.1016/j.jprocont.2010.10.020

[10] Miyazaki M. & Muraio T. & Isobe K. Formation mechanism and modeling of Centerline Segregation. Nippon steel technical report no. 104, 2013.

[11] Kojic M. & Bathe K.-J. The 'Effective-stress-Function' algorithm for thermo-elasto-plasticity and creep. International Journal For Numerical Methods in Engineering, 1987. Vol 24:8. P. 1509-1532. DOI: 10.1002/nme.1620240808

[12] Cook R.D. & Malkus D.S. & Plesha M.E. Concepts and Applications of Finite Element Analysis, 3rd edition. USA, John Wiley & Sons: 1989. ISBN 0-471-84788-7

[13] Hakala M.K. Lujuusopin Elementtimenetelmä. Espoo, Finland; Otakustantamo, 1980. ISBN 951-971-285-1

[14] Thomas B.G. Issues in Thermal-mechanical modeling of Casting Processes. ISIJ International, 1995. Vol 35:6. P. 737-743. DOI: 10.2355/isijinternational.35.737

[15] Seetharamu K.N. & Paragasam R. & Quadir G.A. & Zainal Z.z. & Prasad B.S. & Sundararajan T. Finite element modeling of solidification Phenomena. Sadhana, 2001. Vol 26:1. P. 103-120. DOI: 10.1007/BF02728481.

[16] Klime L. & Stetina J. A rapid GPU-based heat transfer and solidification model for dynamic computer simulations of continuous steel casting. Journal of Materials Processing technology, 2015. Vol:226. P. 1-14. DOI: 10.1016/j.jmatprotec.2015.06.016

[17] Si H.M. & Cho C. & Kwahk S.Y. A hybrid method for casting process simulation by combining FDM and FEM with an efficient data conversion algorithm. Journal of Materials Processing Technology, 2003. Vol 133: 3 P. 311-321. DOI: 10.1016/S0924-0136(02)01008-7

[18] Koric S. & Hibbeler L.C. & Thoams B.G. Explicit coupled thermo-mechanical finite element model of steel solidification. International Journal For Numerical Methods in Engineering, 2009. Vol 78:1. DOI: 10.1002/nme.2476

[19] Han H.M. & Lee J.E. & Yeo T. & Won Y.M. & Kim K.H. &Oh K.H. & Yoon J.K. A Finite element model for 2-dimensional slice of cast strand. ISIJ International, 1999. Vol 39:5. P 445-454. DOI: 10.2355/isijinternational.39.445

[20] Fachinotti V.D. & le Corre S. & Triolet N. & Bobadilla M. & Bellet M. Two-phase thermo-mechanical and macrosegrecation modelling of binary alloys solidification with amphasis on the

secondary cooling stage of steel slab continuous casting processes. International journal For numerical methods in engineering, 2006. Vol 67:10. P. 1341-1384. DOI: 10.1002/nme.1664.

[21] Kelly J.E. & Michalek K.P. & O'Connor T.G. & Thomas B.G & Dantzig J.A. Initial Development of Thermal and Stress Fields in Continuously Cast steel billets. Metallurgical and materials Transactions, 1988. Vol 19: 10. P. 2589-2606. DOI: 10.1007/BF02645486

[22] Janik M. & Dyja H. & Berski S. & Banaszek G. Two-dimensional thermomechanical analysis of continuous casting process. Journal of Material Processing Technology, 2004. Vol 153-154. P. 578-582 DOI:10.1016/j.jmatprotec.2004.04.129

[23] Kajitani T. & Drezet J.M. & Rappaz M. Numerical Simulation of Deformation-Induced Segregation in Continuous Casting of Steel, 2001. Vol 32:6. P. 1479-1491. DOI: 10.1007/s11661-001-0236-1

[24] Song J. & Cai Z. & Piao F. & Zhu M. Heat Transfer and Deformation Behavior of Shell Solidification in Wide and Thick Slab Continuous Casting Mold. Journal of Iron and Steel Research, 2014. Vol 12:1. P 1–9. DOI:10.1016/S1006-706X(14)60112-6

[25] B.G. Thomas & M. Bellet. Modeling of Stress, Distortion, and Hot Tearing. ASM Handbook, Volume 15: Casting 2008. ASM handbook Committee. P. 449-461. DOI: 10.1361asmhba0005238

[26] Hardin R.A. & Liu K. & Kapoor A. & Beckermann C. A. Transient simulation and dynamic spray cooling control model for continuous steel casting. Metallurgical and Materials Transacions, 2003. Vol 34:3. P. 297-306. DOI: 10.1007/s11663-003-0075-0.

[27] Li C. & Thomas B.G. Thermomechanical Finite-Element Model of Shell Behaviour in Continuous Casting of Steel. Metallurgical and Material Transactions, 2003. Vol 35:66. P. 1151-1172. DOI: 10.1007/s11663-004-0071-z.

[28] Hämäläinen J. & Järvinen J. Elementtimenetelmä virtauslaskennassa. Jyväskylä, Finland: CSC – Tieteellinen Laskenta Oy, 1994. ISBN 952-9821-07-7.

[29] Fish J. & Beltschko T. A First course in Finite Elements. Southern Gate, Chichester, England: John Wiley & Sons, 2007. ISBN 978-0-470-03580-1.

[30] Hosford W.F. Elementary Materials Science. ASM International 2013. Electronic ISBN 978-1-62870-220-0.

[31] Iron- Carbon phase diagram. Available online 11:00 18.04.2016: http://web.mit.edu/course/3/3.11/www/vg/fe_c.gif

[32] Huy H. & Argyropoulos S.A. Mathematical modelling of solidification and melting: a review. Modelling and Simulation in Materials Science and Engineering, 1999. Volume 4:4. P. 371-396. DOI: 10.1088/0965-0393/4/4/004

[33] Swaminathan C.R. & Voller V.R. On the enthalpy method. International journal of Numerical Methods for Heat & Fluid flow, 1993. Vol 3:3. P. 2333-244. DOI: 10.1108/eb017528

[34] Carter G.F. & Paul D.E. Materials Science and Engineering, 9th edition. ASM International 1991. Electronic ISBN 978-1-62198-301-9.

[35] Frost H.J. & Ashby M.F. Deformation-Mechanism Maps, The Plasticity and Creep of Metals and Ceramics. Oxford, New York, USA: Pergamon Press 1982. ISBN: 978-0080293387

[36] Kozlowski P.F. & Thomas B.G. & Azzi J.A. & Wang H. Simple Constitutive Equations for Steel at High Temperature. Metallurgical and Materials Transactions, 1992. Vol 23:903. DOI: 10.1007/BF02675567

[37] Wray P.J. Plastic Deformation of Delta-Ferritic Iron at Intermediate Strain Rates. Metallurgical and Materials Transactions, 1976. Vol 7:11. P. 1621-1627. DOI: 10.1007/BF02817878

[38] Suzuki T. & Tacke K.H. & Wünnenberg K. & Schwedtfeger K. Creep properties of steel at continuous casting temperatures. Ironmaking and Steelmaking, 1998. Vol 15:2. P. 90-100.

[39] Lush A.M. & Weber G. & Anand L. An implicit Time-integration procedure for a set of internal variable constitutive equations for isotropic elasto-viscoplasticity. International Journal of Plasticity, 1989. Vol 5: 5. P. 521-549. DOI:10.1016/0749-6419(89)90012-0

[40] Madej L. & Sitko M. & Pietrzyk M. Perceptive comparison of mean and full field dynamic recrystallization models. Archives of Civil and Mechanical Engineering 2016. Vol 16:4. P. 569-589. DOI:10.1016/j.acme.2016.03.010

[41] Mirzadeg H. & Najafizadeh A. The rate of dynamic recrystallization in 17-4 PH stainless steel. Materials and design, 2010. Vol. 31:10. P. pages 4577-45583. DOI: 10.1016/j.matdes.2010.05.052

[42] Bianchia J.H. & Karjalainen L.P. Modelling of dynamic and metadynamic recrystallisation during bar rolling of a medium carbon spring steel. Journal of materials Processing Technology, 2005. Vol 160:3. P. 267-277. DOI:10.1016/j.jmatprotec.2004.06.016

[43] Xu D. & Zhu M. & Tang Z. & Sun C. Determination of the Dynamic Recrystallization Kinetics Model for SCM435 Steel. Journal of Wuhan university Technology – Material Science and Engineering, 2013. Vol 28:4. P. 819-824. DOI 10.1007/s11595-013-0775-5

[44] Jonas J. & Quelennec X. & Jiang L. & Martin E. The Avrami kinetics of dynamic recrystallization. Acta materialia, 2009. Vol 57:9. P. 2748-2756. DOI: 10.1016/j.actamat.2009.02.033

[45] Won Y. & Yeo T.J. & Seol D. & Oh K. A New Criterion for Internal Crack Formation in Continuously Cast Steels. Metallurgical and Materials Transactions, 1999. Vol 31: 4. P. 779-794. DOI: 10.1007/s11663-000-0115-y

[46] Bannach N. Phase Change: Cooling and Solidification of Metal. Comsol Blog, August 12, 2014. Available online 29.3.2015 12:00 https://www.comsol.com/blogs/phase-change-cooling-solidification-metal/

[47] Han Z. & Cal K. & Liu B. Prediction and Analysis on Formation of Internal Cracks in Continuously Cast Slabs by mathematical Models. ISIJ International, 2001. Vol 41: 12. P. 1472-1480. DOI: 10.2355/isijinternational.41.1473

[48] B. Mintz & S. Yue & J.J. Jonas. Hot Ductility of steels and its relationship to the problem of transverse cracking during continuous casting. International Material Reviews, 1991. Vol 36: 1. Pp. 187-220. DOI:10.1179/imr.1991.36.1.187

[49] Nikishkov G.P. Introduction to the Finite Element Method. University of Aizu lecture notes 2004. Available online 26.5.2016
http://homepages.cae.wisc.edu/~suresh/ME964Website/M964Notes/Notes/introfem.pdf

8. Appendices

8.1 Isoparametric Variables ξ and η

It is remarkably easier to integrate variables in isoparametric coordinates, where the boundaries of integration can be freely chosen. In addition, the shape of the elements in the isoparametric coordinates do not have to match the element in the real coordinate system, enabling more complex element shapes as in the Figure 42.

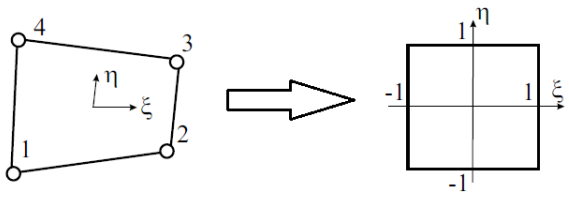

Figure 42 Isoparametric formulation [49]. Despite the fact that the element shape in the normal xy-plane isn't rectangular as on the left, the element in the isoparametric coordinates remains rectangular and easy to handle.

The isoparametric coordinate system used in this work is presented in the Figure 43. Both coordinates ξ and η vary between -1 and 1, reaching those values at the nodes in the corners of the element [12]. Take note that the size of the isoparametric element is actually arbitrary; the limits -1 and 1 are most commonly used due to their easy implementation in integration. The nodes were numbered as in Figure 43.

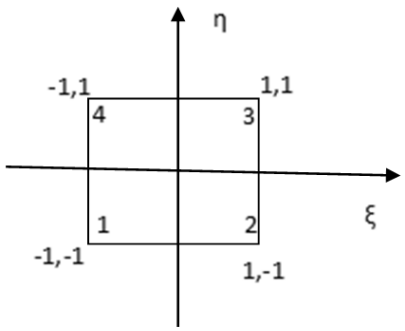

Figure 43 Isoparametric mapping of a quadrilateral element.

The shape functions are the same for isoparametric variables as they would be for x and y. Even though integration is easier in isoparametric coordinates, at some point the treatment has to be transferred to the global coordinates x and y . This is done with the help of a Jacobian, [**J**], as follows:

$$\begin{matrix}\frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y}\end{matrix} = [\boldsymbol{J}]^{-1}\begin{bmatrix}\frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta}\end{bmatrix} \tag{70}$$

where N_i refers to the shape function, x and y refer to the spatial coordinates and ξ and η refer to the isoparametric coordinates. $[\boldsymbol{J}]$ is the Jacobian matrix, which is defined for a 2D case as

$$[\boldsymbol{J}] = \begin{bmatrix}\frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta}\end{bmatrix} \tag{71}$$

The actual shape functions used in this work are

$$\boldsymbol{N} = \begin{Bmatrix}N_1 \\ N_2 \\ N_3 \\ N_4\end{Bmatrix} = \begin{matrix}\frac{1}{4}(1-\xi)(1-\eta) \\ \frac{1}{4}(1+\xi)(1-\eta) \\ \frac{1}{4}(1-\xi)(1+\eta) \\ \frac{1}{4}(1+\xi)(1+\eta)\end{matrix} \tag{72}$$

This assumes that the nodes are numbered as in Figure 43.

When transforming integrals into isoparametric form, the integral has to be multiplied with the determinant of the Jacobian matrix, $[\boldsymbol{J}]$, to account for the difference in the sizes of the elements in the different coordination systems.

$$\int f\, dV = \int\int f(x,y)p\, dxdy = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)p|\boldsymbol{J}|\, d\xi d\eta$$

8.2 Iteration: Convergence and Termination

The model in this work includes iteration of certain unknowns until convergence is achieved. Iteration has been explained well in any basic mathematical text book, and is not explained in detail here. The definition of convergence and termination, however, is of much significance when considering the accuracy of modeling.

This work includes many iterative processes, for example enthalpy and radiation iteration within a single time step in temperature modeling. In these processes, a calculation accuracy, β, has to be decided.

The iteration is terminated inside a time step and calculation is continued in the next time step or procedure once the variable x which is currently being iterated (temperature, for example), no longer changes more than the value β between iteration cycles.

$$\left| x^i - x^{i-1} \right| < \beta \tag{73}$$

Some limitation for β is set by the so-called machine accuracy: double – format numbers are saved with sixteen decimals, meaning that β by definition cannot be smaller than 10^{-15}. Even this accuracy, however, is foolish in practice as model simplifications already introduce errors into calculation that are by far more significant.

β affects calculations in practice as well. Having a very small β leads to significantly more iteration laps and thus longer computation times than having a higher β.

In the validation and results section, the β used in relation to results will always be mentioned.

When solving linear systems of equations, such as in temperature modeling, the new temperatures of the domain are in vector $\boldsymbol{T}^{m+1}$. The termination can then be decided with equation (*74*). Although technically average of absolute values or root mean square calculation can be used instead of maximum in (*74*), the maximum is most robust. The termination can be decided with

$$max|\, \boldsymbol{T}^{m+1} - \boldsymbol{T}^{m}| < \beta \tag{74}$$

where max|| operator chooses the maximum absolute difference of vectors $\boldsymbol{T}^{m+1}$ and $\boldsymbol{T}^{m}$. Once the difference is below β, $\boldsymbol{T}^{m+1}$ and $\boldsymbol{T}^{m}$ are practically equal.

8.3 Numerical integration: Gauss Quadrature

Many equations in this work include integrals, which in this work are handled with Gauss Quadrature numerical integration.

The integral in this chapter is created using isoparametric formulation as in appendix 8.1.

Using gauss quadrature, integrals such as equation (*75*) can be calculated.

$$\int f\, dV = \int\int f(x,y)p\, dxdy = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)p|J|\; d\xi d\eta \tag{75}$$

Here $f(x,y)$ is a function of the spatial coordinates x and y, p is the thickness in z direction and $|J|$ is the determinant of the Jacobian. The integral is reformulated into

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)p|J|d\xi d\eta = [w_1 f(\xi_1,\eta_1) + w_2 f(\xi_2,\eta_2) + w_3 f(\xi_3,\eta_3) + w_4 f(\xi_4,\eta_4)]p|J| \tag{76}$$

where the w_1, w_2, w_3, w_4 are scalar weights and the $f(\xi_1,\eta_1) - f(\xi_4,\eta_4)$ are values of function f in four points. The example presented here is for two dimensional quadrilateral element, for which the weights are always 1 and the integration points are presented in Table 4.

Table 4 Integration points and their cordinates

Integration point	ξ	η
1	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
2	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
3	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
4	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$

In practice this means approximating the integral by calculating the value of the function in several points (sometimes called sampling points or Gauss integration points), and multiplying that value with certain weight. Several different orders of accuracy are available in literature

[12], but in this work the second order accuracy was sufficient in all integrals to reach the same accuracy as in analytical integration.

8.4 Creep and Plasticity Expanded Forms

The densities of ferrite, austenite and delta-ferrite follow the polynomes below [27]:

$$\begin{aligned}
\rho_{ferrite} &= 7881 - 0.324T(°C) - 3 \cdot 10^{-5}T(°C)^2 \\
\rho_{austenite} &= \frac{100\big(8106 - 0.51T(°C)\big)}{(100 - C_{\%})(1 + 0.008C_{\%})^3} \\
\rho_{delta-ferrite} &= \frac{100\big(8011 - 0.47 \cdot T(°C)\big)}{(100 - C_{\%})(1 + 0.013C_{\%})^3}
\end{aligned} \tag{77}$$

where C(%) is the carbon content percentage in steel and $T(°C)$ is the temperature in degrees Celsius.

The constitutive equation for strain rate in austenite in full form is [27, 36]

$$\begin{aligned}
&\dot{\varepsilon_{in}} = Ce^{-\frac{Q}{T}}\big(\sigma - a_e\varepsilon_p^{n_e}\big)^n \\
&C = 46550 + 71400\,C_{\%} + 12000\,C_{\%}^2 \\
&Q = 44650 \\
&a_e = 130.5 - 5.128 * 10^{-3}T \\
&n_e = -0.6289 + 1.114 * 10^{-3}T \\
&n = 8.132 - 1.540 * 10^{-3}T
\end{aligned} \tag{78}$$

The enhanced power law model by Li et al [26] for delta ferrite is

$$\begin{aligned}
&\dot{\varepsilon_{in}} = 0.1c\left|\frac{\bar{\sigma}}{C_f\left(\frac{T}{300}\right)^{-5.52}(1 + 1000|\varepsilon_P|)^m}\right|^n \\
&C_f = 1.3678 * 10^4(C_{\%})^{-0.0556} \\
&m = -9.4156 * 10^{-5}\,T + 0.349501 \\
&n = \frac{1}{1.617 * 10^{-4}\,T - 0.06166}
\end{aligned} \tag{79}$$

The equation for flow strain in the mushy region is [27]

$$\begin{aligned}
&\dot{\varepsilon}_{flow} = \begin{cases} ac\big(|\bar{\sigma}| - \sigma_{yield}\big), & if\ \bar{\sigma} > \sigma_{yield} \\ 0, & if\ \bar{\sigma} \le \sigma_{yield} \end{cases} \\
&a = 1.5 * 10^8\frac{MPa}{s}
\end{aligned} \tag{80}$$

where a is chosen according to the work of Li et al and σ_{yield} is set to 0.01 MPa [27].

The coefficient c is defined as [27]

$$c = \begin{matrix} \frac{\varepsilon_{max}}{|\varepsilon_{max}|}, if\ \varepsilon_{max} \geq \varepsilon_{min} \\ \frac{\varepsilon_{min}}{|\varepsilon_{min}|}, if\ \varepsilon_{max} < \varepsilon_{min} \end{matrix} \quad (81)$$

where ε_{max} is the greatest value in the principal strains and ε_{min} the minimum principal strain.

8.5 Derivation of the Equations in Mechanical Analysis

The displacements are solved in mechanical FEM analysis from equation

$$[\boldsymbol{K}]\boldsymbol{d} = \boldsymbol{F}. \tag{82}$$

$[\boldsymbol{K}]$ signifies the global stiffness matrix, $\boldsymbol{d}$ the displacement vector which is used to calculate the strains and stresses, and $\boldsymbol{F}$ is the global force vector.

The aim of the following treatment is to provide a so-called weak form, which is a differential equation that can be discretized into finite element form. For the weak form we need expressions for forces at the boundary of the domain, forces within the domain and for the material response to loads, i.e. the material stiffness. The treatment here follows the text book of Fish et al [29].

First, let us derive an expression which connects the stresses and forces of a domain. Let us start from an assumption of force equilibrium in xy-plane, as in Figure 44.

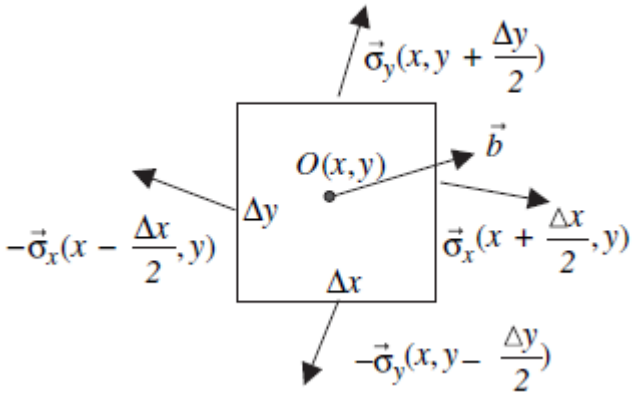

Figure 44 Body force vector on an infinitesimal element [28].

The domain in Figure 44 is of infinitesimal size, for which the equilibrium equation assuming static situation is

$$\begin{aligned} -\sigma_x\left(x - \frac{\Delta x}{2}, y\right)\Delta y + \sigma_x\left(x + \frac{\Delta x}{2}, y\right)\Delta y - \sigma_y\left(x, y - \frac{\Delta y}{2}\right)\Delta x \\ + \sigma_y\left(x, y + \frac{\Delta y}{2}\right)\Delta x + \boldsymbol{b}(x, y)\Delta x\Delta y = 0. \end{aligned} \tag{83}$$

The x and y are the coordinates, Δx and Δy are distances from the origo O in Figure 44 and $\vec{\sigma}_x$ and $\vec{\sigma}_y$ are stress vectors on the surfaces of the element. Take note that this work uses different notation than the work of Fish et al [29], which is why the vector components are marked as σ_x instead of $\vec{\sigma}_x$ as in Figure 44.

Dividing the equation (*83*) by $\Delta x \Delta y$ and taking the limit as $\Delta x \to 0$ and $\Delta y \to 0$, we receive equations (*84*).

$$\begin{aligned} \lim_{\Delta x \to 0} \frac{\sigma_x\left(x + \frac{\Delta x}{2}, y\right) - \sigma_x\left(x - \frac{\Delta x}{2}, y\right)}{\Delta x} &= \frac{\partial \sigma_x}{\partial x} \\ \lim_{\Delta y \to 0} \frac{\sigma_y\left(x, y + \frac{\Delta y}{2}\right) - \sigma_y\left(x, y - \frac{\Delta y}{2}\right)}{\Delta y} &= \frac{\partial \sigma_y}{\partial y} \end{aligned} \tag{84}$$

Combining the two equations (*83*) and (*84*) yields

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \boldsymbol{b} = 0. \tag{85}$$

Equation (*85*) is in component form as follows:

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + b_x &= 0 \\ \frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + b_y &= 0. \end{aligned} \tag{86}$$

Or in more concise format:

$$[\mathbf{L}]^{\mathrm{T}} \boldsymbol{\sigma} + \boldsymbol{b} = 0. \tag{87}$$

Now we have a connection between the stresses **σ** and the forces $\boldsymbol{b}$. $\boldsymbol{b}$ is the body force vector and the first term $[\mathbf{L}]^{\mathrm{T}}\{\sigma\}$ is opened in equation (*88*).$[\mathbf{L}]^{\mathrm{T}}$ is called the symmetric gradient matrix operator: in this case its transpose is used.

$$[\mathbf{L}]^{\mathrm{T}} \boldsymbol{\sigma} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} \tag{88}$$

$[\mathbf{L}]^{\mathrm{T}}$ includes derivatives of the spatial coordinates and $\boldsymbol{\sigma}$ is the stress vector. Now we will define a constitutive relation, which defines the connection between stresses and strains in material:

$$\boldsymbol{\sigma} = [\boldsymbol{D}]\boldsymbol{\varepsilon} \tag{89}$$

where $\boldsymbol{\varepsilon}$ is the strain vector, and the expression for matrix $[\boldsymbol{D}]$ depends on the number of dimensions treated. At this point the exact form of the matrix $[\boldsymbol{D}]$ is, however, quite arbitrary.

We still need a relation between displacements and strains, as only displacements can be solved using FEM. This relation is expressed in equation (*47*).

$$\boldsymbol{\varepsilon} = [\boldsymbol{L}]\boldsymbol{u} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} u_x \\ u_y \end{Bmatrix} \tag{90}$$

Here $\boldsymbol{u}$ is the displacement vector. On any boundary, either the displacement or the traction must be prescribed, but both cannot be prescribed on the same boundary.

Traction boundary condition is defined

$$\boldsymbol{\sigma}\bar{\boldsymbol{n}} = \boldsymbol{g} \;\; on \;\; \Gamma_t \tag{91}$$

where $\bar{n}$ is a surface vector of the boundary and $\boldsymbol{g}$ is the traction force. Similarly, displacement boundary condition is defined as

$$\boldsymbol{u} = \boldsymbol{u}_{known} \;\; on \; \Gamma_u \tag{92}$$

where $\boldsymbol{u}_{known}$ is any known displacement – usually zero.

Now we have two equations; (*93*) and (*94*). They will be used to derive an expression which can be discretized into FEM form.

$$\boldsymbol{\sigma}\bar{\boldsymbol{n}} - \boldsymbol{g} = 0 \tag{93}$$

$$[\mathbf{L}]^{\mathrm{T}}\boldsymbol{\sigma} + \boldsymbol{b} = 0 \tag{94}$$

Let us multiply the equations by a weight function w and integrate over volume V in equation (*94*), and in the case of equation (*93*), over boundary area Γ.

$$\int \left(w[\mathbf{L}]^{\mathrm{T}}\boldsymbol{\sigma} + w\boldsymbol{b}\right) dV = 0 \tag{95}$$

$$\int w(\boldsymbol{\sigma}\bar{\boldsymbol{n}} - \boldsymbol{g})\ d\Gamma = 0 \tag{96}$$

Let us apply Greens theorem to the first term of equation (*95*).

$$\int w\left([\mathbf{L}]^{\mathrm{T}}\boldsymbol{\sigma}\right) dV = \int w\boldsymbol{\sigma}\bar{\boldsymbol{n}}\ d\Gamma + \int [\mathbf{L}]^{\mathrm{T}} w\boldsymbol{\sigma}\ dV \tag{97}$$

Now equation (*95*) has been altered into equation (*98*).

$$\int [\mathbf{L}]^{\mathrm{T}} w\boldsymbol{\sigma}\ dV = -\int w\boldsymbol{\sigma}\bar{n}\ d\Gamma - \int w\boldsymbol{b}\ dV \tag{98}$$

Now we recall the equality $\boldsymbol{\sigma}\bar{\boldsymbol{n}} = \boldsymbol{g}$ in equation (*93*) and substitute it into equation (*98*).

$$\int [\mathbf{L}]^{\mathrm{T}} w\boldsymbol{\sigma}\ \ dV = \int w\boldsymbol{g}\ d\Gamma + \int w\boldsymbol{b}\ dV \tag{99}$$

Substituting (*48*) and (*47*) into equation (*99*), we get

$$\int [\mathbf{L}]^{\mathrm{T}} w[\boldsymbol{D}][\boldsymbol{L}]\boldsymbol{u}\ \ dV = \int w\boldsymbol{g}\ d\Gamma + \int w\boldsymbol{b}\ dV \tag{100}$$

which is called the weak form [29]. Here **u** is now the displacements which we want to solve. The first term product, $[\mathbf{L}]^{\mathrm{T}} w[\boldsymbol{D}][\boldsymbol{L}]$**,** determines the material response to the forces $\boldsymbol{g}$ and $\boldsymbol{b}$. Let us formulate the weak form into

$$\int ([\boldsymbol{L}]w)^{T}[\boldsymbol{D}][\boldsymbol{L}]\boldsymbol{u}\ dV = \int w^{T}\boldsymbol{g}\ d\Gamma + \int w^{T}\boldsymbol{b}\ dV \tag{101}$$

for future benefit [29]. Now we can discretize the weak form (*101*) with finite element formulation. There are two degrees of freedom per node as in equation (*102*) – again, assuming a two dimensional scenario with quadrilateral four-node elements.

$$\boldsymbol{d} = \{u_{x1}\quad u_{y1}\quad u_{x2}\quad u_{y2}\quad u_{x3}\quad u_{y3}\quad u_{x4}\quad u_{y4}\}^{T} \tag{102}$$

The finite element approximation of the trial solution and weight function on each element are (*103*) and (*104*), respectively [29].

$$\boldsymbol{u} = \boldsymbol{u}^{\boldsymbol{e}} = [\boldsymbol{N}^{\boldsymbol{e}}]\boldsymbol{d} \tag{103}$$

$$w^{T} = w^{eT} = w^{eT}[\boldsymbol{N}^{e}]^{T} \tag{104}$$

The superscript e refers to element specific variables. Here the weight functions w, displacements $\boldsymbol{u}$ and shape functions $[\boldsymbol{N^e}]$ are functions of the spatial coordinates x and y only.

$$[\boldsymbol{N}^e] = \begin{bmatrix} N_1^e & 0 & \dots & N_4^e & 0 \\ 0 & N_1^e & \dots & 0 & N_4^e \end{bmatrix} \tag{105}$$

Again, this expression for $[\boldsymbol{N}^e]$ is valid for quadrilateral four-node elements only.

Summing over all the elements (n_e is the number of elements), the weak form can now be written as

$$\sum_{i=1}^{n_e} \left\{ \int \left([\mathbf{L}]_{\mathrm{s,i}} w \right)^T [\boldsymbol{D}] [\mathbf{L}]_{s,i} \boldsymbol{u} \; dV - \int w^T \boldsymbol{g} \, d\Gamma - \int w^T \boldsymbol{b} \; dV \right\} = 0. \tag{106}$$

We already have an expression for the strains, which can now be formulated into

$$\boldsymbol{\varepsilon} = [\mathbf{L}]_S \boldsymbol{u} = [\mathbf{L}]_S [\boldsymbol{N^e}] \boldsymbol{d}^e = [\boldsymbol{B^e}] \boldsymbol{d^e} \; . \tag{107}$$

Where $[\boldsymbol{B^e}]$ is called the strain-displacement matrix and it is presented in equation (*108*). The size of the matrices is in the subscript of the matrices.

$$\begin{aligned} [\boldsymbol{B^e}] = [\mathbf{L}]_S [\boldsymbol{N^e}] &= \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}_{3*2} \begin{bmatrix} N_1^e & 0 & \dots & N_4^e & 0 \\ 0 & N_1^e & \dots & 0 & N_4^e \end{bmatrix}_{2*8} \\ &= \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & 0 & \dots & \dfrac{\partial N_4}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1}{\partial y} & \dots & 0 & \dfrac{\partial N_4}{\partial y} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_1}{\partial x} & \dots & \dfrac{\partial N_4}{\partial y} & \dfrac{\partial N_4}{\partial x} \end{bmatrix}_{3*8} \end{aligned} \tag{108}$$

The derivatives of the shape functions in equation (*106*) are:

$$([\mathbf{L}]_{\boldsymbol{s}} w)^T = ([\boldsymbol{B^e}] w^{\boldsymbol{e}})^T = [\boldsymbol{B^e}]^{\boldsymbol{T}} w^{\boldsymbol{eT}} \tag{109}$$

Substituting (*103*), (*104*), (*107*) and (*109*) into (*106*), we get

$$w^{eT}\sum_{e=1}^{n_e}\left\{\int [B^e]_i^T[D][B^e]_i dV\, d^e - \int [N^e]_i^T g\, d\Gamma - \int [N^e]_i^T b\, dV\right\} = 0. \tag{110}$$

At this point we omit the treatment of w^{eT}, which can be used to further examine the essential and natural boundary conditions. Here we simply refer to the work of Fish et al [29].

We can separate the expressions for the stiffness matrix, body forces and external forces from equation (*110*) into equation (*44*). At this point we stop marking the element matrices with superscripts e; it is separately always marked when a variable is global or specific to an element. Stiffness matrices [**k**] and the forces $\boldsymbol{f}$ will be marked in capital letters for global variables and in small letters for element variables.

$$\begin{aligned} [k^e] = [k] &= \int [B]^T[D][B]\, dV \\ f_{ext} &= \int [N]^T g\, d\Gamma \\ f_{body} &= \int [N]^T b\, dV \end{aligned} \tag{111}$$

When the element matrices are combined into a global system of equations, we have equation (*43*).The procedure of combining element matrices into global matrices is not presented here, but it is well documented in the text books of Cook et al [12] and Fish et al [29].

As only two dimensions are treated here, the integral in the equation (*44*) for $[k]$ and f_{body} is [12]:

$$\int X\, dV = \int\int Xp\, dxdy \tag{112}$$

where p is the thickness of the cross section in the z direction.

Now we have derived an expression for the global system of equations

$$[K]d = F. \tag{113}$$

8.6 Local Algorithm for Solving Stress and Strain

This algorithm is based on the work of Lush and al [39]. In step three of the strain rate calculation this algorithm is used to solve the local stress and inelastic strain at that time step. The algorithm solves the simultaneous equations

$$\begin{aligned} \varepsilon_{n+1} - \varepsilon_n - \Delta t f(\bar{\sigma}_{n+1}, \varepsilon_{n+1}) = 0 \\ \bar{\sigma}_{n+1} - \sigma^*_{n+1} + 3\mu\Delta t f(\bar{\sigma}_{n+1}, \varepsilon_{n+1}) = 0 \end{aligned} \tag{114}$$

Here the subscript n designates the current time step and subscript n the previous time step. $f(\bar{\sigma}_{n+1}, \varepsilon_{n+1})$ is either function in equation (*78*) or (*79*), depending on the current phase. From now on $f(\bar{\sigma}_{n+1}, \varepsilon_{n+1})$ will simply be marked f, with a subscript n or n+1 to denote the time step. Here μ is the so called lame constant, which is defined as

$$\mu = \frac{E}{2(1+v)} \tag{115}$$

Initial estimates for ε_{n+1} and $\bar{\sigma}_{n+1}$ are

$$\begin{aligned} \varepsilon_{n+1} &= \varepsilon_n + \frac{b_1(\bar{\sigma}^*_{n+1} - \bar{\sigma}_n - 3\mu\Delta t f_n) + a_1 \Delta t f_n}{b_2 a_1 + a_2 b_1} \\ \bar{\sigma}_{n+1} &= \bar{\sigma}_n + \frac{b_2(\bar{\sigma}^*_{n+1} - \bar{\sigma}_n - 3\mu\Delta t f_n) - a_2 \Delta t f_n}{b_2 a_1 + a_2 b_1} \end{aligned} \tag{116}$$

Here the parameters a and b are

$$\begin{aligned} a_1 &= 1 + 3\mu\Delta t \frac{\partial f_n}{\partial \bar{\sigma}_n} \\ a_2 &= 3\mu\Delta t \frac{\partial f_n}{\partial \varepsilon_n} \\ b_1 &= \Delta t \frac{\partial f_n}{\partial \bar{\sigma}_n} \\ b_2 &= 1 - \Delta t \frac{\partial f_n}{\partial \varepsilon_n} \end{aligned} \tag{117}$$

This algorithm is a two level iterative scheme. The level 1 iterations are performed for a generic iteration k, where ε^k_{n+1} is tested if it satisfies the upper equation in (*114*).

1) Determine $\bar{\sigma}^k_{n+1}$ with the level 2 iterations.
2) Determine the error E^k_ε:

$$E_{\varepsilon}^{k} = \varepsilon_{n+1}^{k} - \varepsilon_{n} - \Delta t f_{n+1}^{k} \tag{118}$$

3) Compare E_{ε}^{k} to a suitable termination accuracy β. If

$$E_{\varepsilon}^{k} < \beta \tag{119}$$

the scheme has converged and $\varepsilon_{n+1} = \varepsilon_{n+1}^{k}$ and $\bar{\sigma}_{n+1} = \bar{\sigma}_{n+1}^{k}$.

$$\beta = 10^{-5} * \varepsilon_{n} \tag{120}$$

If convergence is not achieved, iteration continues.

4) Calculate the new Newton-Raphson correction

$$\Delta\varepsilon^{k} = \frac{-E_{\varepsilon}^{k}}{1 - \Delta t\left(\frac{\partial f_{n+1}^{k}}{\partial \varepsilon_{n+1}^{k}} + \frac{\partial f_{n+1}^{k}}{\partial \bar{\sigma}_{n+1}^{k}}\frac{\partial \bar{\sigma}_{n+1}^{k}}{\partial \varepsilon_{n+1}^{k}}\right)} \tag{121}$$

5) The estimate for the next iteration is

$$\varepsilon_{n+1}^{k+1} = \varepsilon_{n+1}^{k} + \Delta\varepsilon^{k} \tag{122}$$

6) The initial estimate for the next level 2 iteration is

$$\bar{\sigma}_{n+1}^{k+1} = \bar{\sigma}_{n+1}^{k} + \frac{\partial \bar{\sigma}_{n+1}^{k}}{\partial \varepsilon_{n+1}^{k}}\,\Delta\varepsilon^{k} \tag{123}$$

Level 2 iterations are done for a generic iteration *i*, where the superscript *k* denotes the level 1 iteration round, and *i* denotes the level 2 iteration round. In level 2 iterations, $\bar{\sigma}_{n+1}^{k,i}$ is tested for satisfaction of the lower equality in equation (*114*).

1) Determine error $E_{\sigma}^{k,i}$ associated with $\bar{\sigma}_{n+1}^{k,i}$

$$E_{\sigma}^{k,i} = \bar{\sigma}_{n+1}^{k,i} - \bar{\sigma}_{n+1}^{*} + 3\mu\Delta t f_{n+1}^{k,i} \tag{124}$$

2) Compare $E_{\sigma}^{k,i}$ to a tolerance, if

$$E_{\sigma}^{k,i} < 10^{-5} * \bar{\sigma}_{n+1}^{*} \tag{125}$$

Convergence has been achieved and the following derivative can be calculated to be used in level 1.

$$\frac{\partial \bar{\sigma}_{n+1}^{k}}{\partial \varepsilon_{n+1}^{k}} = \frac{-3\mu\Delta t\left(\frac{\partial f_{n+1}^{k,i}}{\partial \varepsilon_{n+1}^{k}}\right)}{1+3\mu\Delta t\left(\frac{\partial f_{n+1}^{k,i}}{\partial \bar{\sigma}_{n+1}^{k,i}}\right)} \tag{126}$$

If convergence is not achieved, iteration is continued.

3) Calculate the Newton-Raphson correction

$$\Delta\sigma_{NR}^{k,i} = \frac{-E_{\sigma}^{k,i}}{1+3\mu\Delta t\left(\frac{\partial f_{n+1}^{k,i}}{\partial \bar{\sigma}_{n+1}^{k,i}}\right)} \tag{127}$$

4) Either the upper or the lower bound for the stress is updated. If $\Delta\sigma_{NR}^{k,i} < 0$, then

$$\begin{aligned} \bar{\sigma}_{upper} &= \bar{\sigma}_{n+1}^{k,i} \\ \Delta\sigma_{max}^{k,i} &= \frac{1}{2}\left(\Delta\bar{\sigma}_{lower} - \bar{\sigma}_{n+1}^{k,i}\right) \end{aligned} \tag{128}$$

If $\Delta\sigma_{NR}^{k,i} \geq 0$, then

$$\begin{aligned} \bar{\sigma}_{lower} &= \bar{\sigma}_{n+1}^{k,i} \\ \Delta\sigma_{max}^{k,i} &= \frac{1}{2}\left(\Delta\bar{\sigma}_{upper} - \bar{\sigma}_{n+1}^{k,i}\right) \end{aligned} \tag{129}$$

5) Determine which correction is used: if $\left|\Delta\sigma_{NR}^{k,i}\right| > \left|\Delta\sigma_{max}^{k,i}\right|$, then

$$\begin{aligned} \Delta\sigma^{k,i} &= \Delta\sigma_{max}^{k,i}, else \\ \Delta\sigma^{k,i} &= \Delta\sigma_{NR}^{k,i} \end{aligned} \tag{130}$$

6) The estimate for the next iteration is

$$\bar{\sigma}_{n+1}^{k,i+1} = \bar{\sigma}_{n+1}^{k,i} + \Delta\sigma^{k,i} \tag{131}$$

Each time the level 2 iterations are performed, the lower ($\bar{\sigma}_{lower}$) and upper ($\bar{\sigma}_{upper}$) bounds for stress have to be initialized. The initialization was performed as suggested in the article of Lush et al [39].