



**Politecnico
di Torino**

Politecnico di Torino

Laurea Magistrale in Ingegneria Aerospaziale

**Anno Accademico 2023/2024
sessione di laurea Dicembre 2023**

VIRTUAL MANUFACTURING AND SIMULATION OF METAL CASTING PROCESSES

Relatori:

Marco Gherlone
Andrea Cini (UC3M)
Antonio Raimondo (UC3M)

Candidato:

Tiziano Guglielmini

Contents

1	Abstract	3
2	Thesis introduction	3
3	Casting optimization	4
4	Literature review	6
5	Physical process	10
5.1	Filling phase	11
5.2	Solidification phase	11
5.3	cooling phase	12
6	Numerical method	12
6.1	Discretization	13
6.2	Finite Element Method (FEM)	13
6.3	Finite Volume Method (FVM)	14
6.4	Finite Difference Method (FDM)	14
6.5	Coupled Eulerian Lagrangian (CEL)	15
6.6	Arbitrary Lagrangian Eulerian (ALE)	15
6.7	Smoothed Particle Hydrodynamics (SPH)	16
7	Simulation	16
7.1	Fluid and solid simulation	17
7.2	Solid/liquid interface	19
7.3	Volume of fluid	20
7.4	Mushy zone	21
7.5	Boundary condition	23
7.6	Thermo-mechanical model	24
7.7	Abaqus theory	26
7.7.1	Eulerian formulation	26
7.7.2	Thermal equations numerical scheme	27
7.7.3	Mechanical equations numerical schemes	28
7.7.4	Equation of State	29

8	Model definition	29
8.1	Inspirecast	30
8.1.1	Model	31
8.2	Abaqus	34
8.2.1	Part	35
8.2.2	Properties	35
8.2.3	Assembly	39
8.2.4	Step	39
8.2.5	Interaction	40
8.2.6	Load	40
8.2.7	Mesh	42
8.2.8	Job	43
8.2.9	Visualization	43
9	Results	43
9.1	Inspirecast	43
9.1.1	Filling Phase	44
9.1.2	Solidification Phase	47
9.1.3	Demolding	51
9.2	CEL	53
9.3	Eulerian	58
9.3.1	Filling phase	58
9.3.2	Cooling phase	65
9.4	Lagrangian model	67
9.5	Comparison Inspirecast Abaqus	72
9.6	Conclusions	76
10	Bibliography	78

1 Abstract

This thesis investigates the development of a FEM model for aluminum casting virtual manufacturing. These kinds of simulations are employed in the industries to develop an optimal mold geometry to reduce manufacturing-induced defects in the final piece. Manufacturing process simulation allows to speed up time to market and reduce cost related to testing requiring suitable facilities and a test for each design change. FEM simulation allows the user to go through the heuristic phase much quicker with a significant cost saving. Several commercial software performing casting simulations such as InspireCast are available, but they don't disclose the computational methodology to obtain results and few parameters are tunable. This work is focused on developing a model within the FEM commercial software Abaqus framework for aluminum gravity casting by optimizing various simulation parameters. The FEM method is used to solve the problem with Eulerian, Lagrangian and CEL (Coupled Eulerian-Lagrangian) formulation. The developed FEM model was assessed and validated against InspireCast results.

2 Thesis introduction

This thesis marks the culmination of my master's degree in Aerospace Engineering, representing the culmination of the knowledge acquired throughout my academic journey at Politecnico di Torino and the experiences garnered during the Erasmus+ project conducted at Universidad Carlos III Madrid. The foundation for this work was laid during the degree courses at Politecnico di Torino, where I gained a comprehensive understanding of the theoretical and practical aspects of Aerospace Engineering. The multidisciplinary curriculum equipped me with the necessary tools and insights to delve into advanced topics within the field. Additionally, the invaluable experience gained during the Erasmus+ project at Universidad Carlos III Madrid played a pivotal role in shaping the trajectory of this research. The collaborative and international academic environment at Universidad Carlos III Madrid provided a unique perspective and enriched my understanding of aerospace engineering, contributing significantly to the development of this thesis.

This thesis delves into the creation of a Finite Element Method (FEM) model for virtual manufacturing of aluminum casting. Such simulations play a crucial role in industries by facilitating the development of an optimal mold geometry, thereby mitigating manufacturing-induced defects in the final product. Utilizing computer simulations proves to be a more cost-effective, time-efficient, and resource-friendly approach compared to traditional methods. Although various commercial software exists for such simulations, they often lack transparency in disclosing their computational methodologies.

This research specifically employs Altair's Inspirecast software as a benchmark, representing the current state-of-the-art in the market. The main objective of this master's thesis is to create a model for the casting manufacturing process of a piston using Computer-Aided Engineering (CAE) software, specifically Abaqus. Subsequently, the aim is to compare

the simulation results regarding velocity, temperature, filling time and defects with those obtained from the commercial software Inspirecast. This comparative analysis seeks to gain insights into the behavior of the model, discerning its strengths and limitations. The key advantages of adopting the approach of creating a model in Abaqus include the ability to choose the equations and material behavior, providing a detailed understanding of how each parameter influences the process. In contrast, Inspirecast automates numerous operations, reducing user control over critical parameters and offering limited insight into the mathematical models and equations employed by the software to solve the problem. Before initiating the development of a model, it is crucial to comprehensively understand the intricacies of the process under investigation. This requires conducting various studies to understand the equation sets required for material modeling, determining material parameters, defining boundary conditions, and selecting the appropriate numerical scheme. An initial overview of the casting process is necessary to comprehend the physical aspects involved. Additionally, a study of the functionality of commercial software is essential to identify actions that need to be replicated in the creation of the FEM model in Abaqus. Casting involves pouring molten metal into a mold to achieve the desired final shape. After solidification, the part undergoes refinement and heat treatment. To ensure defect-free and high-performance parts, it is crucial to either test or simulate the casting process. The final properties of the part are influenced by factors such as material composition, heat treatment, and the manufacturing process. Simulating the casting process enables the identification of parameters affecting the end product, allowing for precise control and the development of an optimal manufacturing process. Specifically, in casting, parameters like material composition, pouring temperature, pouring velocity, mold geometry (including the filling system), material, mold temperature, cooling rate, and casting type (gravity, high/low pressure, centrifugal, in air or vacuum, etc.) can be controlled. Tuning these parameters is essential to achieve a defect-free manufacturing process, preventing issues like improper filling, porosity, shrinkage, incorrect microstructure, and inclusion [42]. The structure of the thesis unfolds as follows: an introduction to casting optimization and commercial software (Chapter 3), a comprehensive literature review (Chapter 4), an explanation of the physical process (Chapter 5.2), an exploration of the numerical method (Chapter 6), an in-depth analysis of the simulation process (Chapter 7), an introduction on the model definition (Chapter 8), an examination of the commercial software Inspirecast (Chapter 8.1), a detailed account of the Abaqus simulation (Chapter 8.2), and the presentation and discussion of results (Chapter 9).

This

3 Casting optimization

The industry has increasingly adopted commercial software for simulating casting processes, as these tools enable cost-effective optimization by running various models without the expense associated with physical experiments. The experimental phase of prototyping typically involves constructing a mold with a complete filling system. Even optimizing the existing system necessitates creating a new mold. Furthermore, the experimental phase involves acquiring a furnace, the required alloy for the part, and machinery for

conducting mechanical tests on the final part to characterize its performance. These factors contribute to substantial costs, limiting the number of optimization steps feasible. Typically, the experimental test is subsequent to a computer-based optimization process to validate the simulation results before proceeding with the production.

While each commercial software has its unique characteristics, the fundamental steps for simulating the casting process in casting software include:

1. **Design Phase:** The design process can be carried out within the software itself, or alternatively, the CAD part can be imported.
2. **Material Selection and Properties:** In commercial software, material selection involves either choosing from a library or manually inputting material properties such as density, viscosity, melting temperature, heat capacity, latent heat, and Young modulus. Some of these properties may be temperature-dependent.
3. **Casting Process Selection:** Different casting processes have unique characteristics and are better suited for specific results. The software typically models various casting processes such as sand casting, low and high-pressure casting, die casting, and centrifugal casting.
4. **Mold Cavity Design:** Designing mold cavities is a critical part of a successful casting process. It involves the definition of components like runner, riser, core, pouring cup, vents, and cooling system. Many commercial software offer a quick design function, allowing users to easily select the entry point for the molten metal to quickly analyze the filling without the need to design the filling system.
5. **Mold and Cast Part Material Selection:** Defining mold's external dimensions, along with mold and cast material thermal and mechanical properties, is essential. Several software applications have an automatic function simplifying material selection. Mold heating or cooling during the process can also be specified.
6. **Casting Parameter Selection:** Before initiating the simulation, certain parameters need to be input into the software, such as pouring temperature, filling time or inlet velocity, mold temperature, and pressure for low/high-pressure casting.
7. **Simulation Execution:** To run the simulation, parameters for mesh, including element size, must be defined. Users may choose to simulate only solidification or include the filling process and to cycle the process. The simulation can then be launched.
8. **Results Analysis:** The simulation generates outputs, including the time history of part and mold temperatures, shrinkage, velocity, and the identification of porosity and defects.

These commercial casting simulation software applications are typically utilized as black boxes, where users are not required to comprehend their internal workings. Users simply input data and analyze the generated outputs. Notable examples of such software include MAGMASOFT [43], ProCAST [44], Inspirecast [45], and FLOW-3D [46].

Generally, these tools are based on either the finite volume method or the finite element method. They possess the capability to internally generate meshes without relying on external software. These applications solve the system of Navier-Stokes equations alongside solid constitutive equations and models for solidification

4 Literature review

Presented here are some notable contributions in the field of casting simulation to illustrate the current state of the art. In the work by Venkatesan et al. [3], a C++ code is developed to address the 2D heat problem in the solidification of pure metals and alloys. The solution employs temperature as a field variable with temperature-dependent thermal properties. The enthalpy method, incorporating latent heat, models phase change in the system. The presence of an air gap between the mold and the metal, resulting from solidification shrinkage, impacts the boundary condition, with interface elements simulating this air gap. The program utilizes the Finite Element Method (FEM) formulation for the heat problem, calculating grain structure by considering its dependence on thermal gradients.

Chang et al.'s papers [4] demonstrate the integration of a cellular automaton method into a Finite Volume Method (FVM) for solving the solidification structure problem in aluminum alloy centrifugal casting. The study focuses on understanding how casting parameters influence solidification structure. A continuous nucleation model is employed, utilizing Gaussian distributions that correlate increased grain density with local alloy undercooling. The velocity of a dendrite tip is computed using the KGT model, where different alloy compositions yield distinct coefficients in the growth velocity equation. The cellular automaton (CA) method determines the cell state based on nucleation and growth velocity models, with nucleation sites chosen randomly for the initial condition. Phase change within a cell occurs through nucleation or by the growth of an adjacent already solid cell. The heat problem is solved in one-dimensional polar coordinates using FVM due to the chosen geometry. The rotation of the mold induces a reduction in the air gap dimensions created by shrinkage, affecting the boundary condition. The coupling between grain structure (computed with CA) and macroscopic heat transfer (utilizing FVM) involves first computing the temperature within the cell with FVM, followed by structure computation from the temperature with CA. Additionally, the latent heat resulting from the cell's change of state is incorporated. Then, a modified CA is proposed to account for the behavior of solutes during solidification.

The article by Seetharamu et al. [6] presents a Finite Element Method (FEM) formulation for a 2D solidification problem. The heat problem is addressed for an ingot within a mold, and the heat transfer coefficients in the boundary conditions are dependent on the air gap created by shrinkage. Solving the heat problem yields the temperature field, from which the stress distribution is computed using an elasto-viscoplastic model. The viscoplastic strain emerges only when the stress exceeds a yield limit.

Chijioke et al.'s paper [10] explores the solidification process of an Al-Cu alloy under different cooling conditions. The study involves a 2D static casting simulation with three zones: liquid, solid, and mushy. The model incorporates the Navier-Stokes equations,

with viscosity being a function of the cell's volume fraction. The momentum equation introduces terms that account for drag in the mushy zone, specifically for the fluid part. Initially, there is no relative velocity between the solid and liquid phases, resulting in no drag force input in the momentum equation. However, as the grains grow, they form a structure, and at this point, the flow in the mushy zone can be considered similar to a flow in a porous medium (following Darcy's law). The energy equation is dependent on the volume fraction, and a microscopic model is introduced as a term dependent on the latent heat of solidification. An equation for solute movement within the domain is also included. The model is solved using ANSYS Fluent, incorporating user-defined functions based on the pressure-velocity coupling SIMPLEC for the source term of the latent heat equation. Three cases of cooling on the bottom face are simulated to analyze the results. The paper by Pericleous et al. [11] introduces a mathematical model for mold filling and solidification based on the Navier-Stokes set of equations. The momentum equation incorporates a source term dependent on the permeability of the mushy zone, which is further influenced by the liquid fraction. The liquid fraction, acting as a state variable defining the presence of fluid in a cell, is a function of temperature. To model the free surface (liquid/gas interface), the mass continuity equation is employed, and the GALA algorithm is utilized alongside the SEA method, employing the scalar Φ to represent the liquid volume fraction. This method adjusts the density within the computational cell based on the material it contains. Although numerical smearing may occur due to advection, the implementation of the TVD scheme with staggered control volumes helps mitigate this issue. The heat equation is expressed in the enthalpy formulation, facilitating the straightforward addition of latent heat for phase change. The source term S^h in the equation includes the effects of viscosity dissipation, heat from fluid bulk motion, heat transfer at the wall, and phase change. The problem's solution follows the SIMPLEC procedure, an algorithm involving a guessed pressure field to initially solve the problem, followed by a correction to ensure continuity.

In the work by Hu et al. [15], various analytical and numerical methods are explored. Analytical approaches involve certain assumptions, such as infinite or semi-infinite regions or constant properties within each phase, as seen in Neumann and Stefan's methods. To impose flux instead of temperature, approximations like the Goodman method are introduced. For solving pure heat problems with phase changes, a fixed grid method is employed, where the Finite Difference Method (FDM) is used to determine temperature, and the moving boundary (interface) is situated between two cells. However, fixed grid methods have limitations on how much the boundary can move in a timestep, which can be addressed by significantly reducing the timestep or employing variable grid methods. The former tracks the boundary using Stefan's equation in a finite difference form and adjusts the grid to follow it. Various latent heat methods are then outlined and briefly explained. The apparent heat capacity method includes latent heat in the material's heat capacity within the temperature range of phase change. The effective capacity method incorporates latent heat in the form of an apparent capacity, determined by the choice of a temperature profile between nodes. The heat integration method addresses melting cases, resetting the control volume temperature to the melting point when it surpasses this threshold. The amount of heat caused by this operation is stored, and the process is repeated until the stored heat equals the latent heat. From this point, the control volume

is free to change temperature. The source-based method involves adding a source term in the energy equation to account for phase change. The enthalpy method relies on the fact that the relationship between temperature and enthalpy is a function of latent heat, represented in step or linear forms for pure materials and alloys, respectively.

In the study by Caraeni et al. [19], the domain is divided by an interface between liquid and solid phases. The heat problem is modeled considering both convection and conduction, with the heat flux jump derived from Stefan conditions. The fluid is described by the Navier-Stokes equations with the Boussinesq approximation, and at the interface, the movement caused by shrinkage (change of density) is imposed. A fixed mesh is utilized, and a level set function tracks the interface, following its extension velocity. In the momentum equation, a penalization term is added to enforce a zero velocity in the solid. The problem is solved using Finite Element Method (FEM), incorporating enriched nodes to account for the step change in field variables across the interface with Extended Finite Element Method (XFEM).

In the study by Verkaik et al. [21], the equations utilized for modeling include the Navier-Stokes equations for the fluid, adopting an Eulerian formulation with Arbitrary Lagrangian-Eulerian (ALE) elements to consider mesh movement. For the solid phase, the model employed is that of a neo-Hookean solid in Lagrangian form. The equations are solved using Finite Element Method (FEM) and a Coupled Overlapped Domain (COD) method. This method involves a background fixed mesh of spectral elements, where the solid can move freely. Background nodes are deactivated when covered by the solid, and those that aren't fully covered remain active. An additional FEM solid mesh is introduced to couple the two domains and account for sharp changes in properties.

In the work by Teng et al. [22], a model is presented wherein both the solid and fluid are described using an Eulerian formulation. Stress in the momentum equation is computed for each solid element using either a hyper-elastic or elasto-plastic model, while for fluid elements, zero stress is imposed. A fixed grid is employed, with a separate coordinate system for each solid present. By introducing particles into each solid and assessing their relative distances, collisions can be computed, and reaction forces applied to the solid.

Koric et al. [23] compare the solutions obtained using an in-house code, CON2D, and Abaqus. The model encompasses the energy equation for temperature field computation, with the solid constitutive equation and a strain model (considering elastic, inelastic, and thermal types). Local nodal integration for stress computation is performed in the subroutine UMAT. The mushy zone can be described using an elastic-perfectly plastic model or a rapid creep function to impose zero fluid stress when the temperature is above the melting point.

In the study by ZHOU and QI [26], the Extended Finite Element Method (XFEM) is employed to simulate the interface between solid and liquid. XFEM offers the advantage of additional shape functions localized only in regions with discontinuities. This feature allows for larger elements and eliminates the need for an adaptive mesh to follow discontinuities. The interface is tracked using a level set function with its zero isosurface representing the interface, and its evolution is computed through its velocity.

Cleary et al. [27] utilize a Smoothed Particle Hydrodynamics (SPH) method, well-suited for complex flows due to its Lagrangian formulation and absence of a grid. The pressure is computed from the equation of state and is then used to calculate particle motion with

the SPH momentum equation. The boundary is modeled with boundary particles interacting with those in the domain. The solutions are compared with MAGMAsoft results. In Wang et al.'s work [30], a non-isothermal phase change problem is modeled, dividing the domain into three zones: solid, liquid, and mushy. The enthalpy-based heat equation is solved across the entire domain, with the enthalpy being a step function dependent on temperature to accurately model the three zones. The model is tested in both 2D and 3D problems and compared with ANSYS solutions when analytical ones are unavailable, demonstrating good agreement.

Trovant and Argyropoulos [32] address a phase change problem while considering how properties change with temperature. Equations for energy, momentum, and continuity are written in axisymmetric cylindrical coordinates. The enthalpy-temperature relation introduces a discontinuity for pure material during phase change, making the energy equation nonlinear. Viscosity is set to an arbitrarily high value for the solid, and an exponential law is introduced to eliminate the viscosity discontinuity between the solid and liquid domains. Shrinkage is influenced by thermal expansion in both the liquid and solid phases and the density change during phase change. The Finite Difference Method (FDM) with the SIMPLER algorithm is employed, featuring a staggered mesh with temperature at the center and velocities at the interface of the computational cell.

Shepel and Paolucci [33] develop a model that simulates the filling process, recognizing its significant impact on the final casting results. The solid phase is modeled as a highly viscous material, with viscosity dependent on the solid fraction. Turbulent viscosity is assumed constant everywhere based on the eddy viscosity model. The free surface is tracked using the Volume of Fluid (VOF) method, with the function F representing the fractional volume of a cell. Cooling channels for water are incorporated in the mold, contributing to the definition of heat transfer in the mold. The considered casting process is cyclic, to first obtain the steady state of the mold temperature field. Boundary conditions are derived by analyzing the mold material and its surface coating. The air gap is not considered due to insufficient experimental data to model variations in the heat transfer coefficient, so it is assumed to be constant. The mesh is generated in ANSYS, and the problem is solved with FIDAP.

In the work by Rouboa and Monteiro [34], the authors analyze heat transfer in a mold by considering the energy equation in enthalpy form. The latent heat for phase change is introduced as a source term, and all parameters are functions of the solid fraction and are computed using the lever rule. The problem is solved using both Finite Difference Method (FDM) and Finite Volume Method (FVM) with a multi-block mesh.

Jalali and Najafi [37] investigate solidification in a pipe under different boundary conditions. The model is based on the Navier-Stokes equations, with the enthalpy and liquid fraction used to incorporate latent heat for solidification. The study explores the impact of changing conditions at the inlet and wall temperature on the solidification process.

Thomas [38] delves into the complexity of casting modeling, focusing on the solid constitutive equation where strain is composed of elastic, inelastic, and thermal components. Modeling the inelastic term, especially at high temperatures, poses challenges, often addressed by using a viscoplastic model due to difficulty distinguishing between creep and plasticity effects. The modeling complexity intensifies during solidification, with evolving microstructures potentially inducing material anisotropy. Phase change significantly al-

ters mechanical properties and material density. The interaction with the mold introduces both mechanical and thermal effects, including stress from friction, heat absorption, and the formation of an air gap due to shrinkage. Modeling both solid and liquid phases typically involves temperature-dependent properties such as viscosity, Young's modulus or Poisson's ratio, with large variations near the liquidus and solidus temperatures. In stress models, convective flow effects are often simplified by increasing thermal conductivity with temperature. When fluid effects are crucial, solving the fluid equations becomes necessary but it is computationally expensive. An alternative approach is to separate the simulation into the mechanical problem and thermal problem, utilizing the appropriate models without excessive assumptions.

In [39], Venkatesan et al. simulate the heat problem in casting using the enthalpy equation and accounting for the air gap between the metal and the mold. This air gap influences thermal flux between the two components and is modeled with coincident nodes. Latent heat is incorporated to consider solidification. The solver, implemented in a C++ code, employs Finite Element Method (FEM) and is validated through simulations of an infinite slab and an aluminum alloy casting. The grain structure is predicted by correlating the thermal gradient with the resultant structure.

Ravindran and Lewis [41] develop a FEM formulation of the Navier-Stokes equation to model the casting process. The metal front is computed using the pseudo-concentration advection equation. The phase change is modeled using the apparent heat capacity, dependent on the solid fraction and latent heat. Two models for inter-dendritic flow are introduced, one for the equiaxed region and one for the columnar region. Temperature and front tracking are solved with an explicit scheme, while the flow is solved with an implicit scheme with a longer timestep, coupled during the simulation.

5 Physical process

The casting process can be divided into three distinct phases. The first phase is the filling stage, where molten metal flows through designated ducts to reach the mold cavity, which represents the negative space of the final cast part. The second phase, known as solidification, commences when the cavity is completely filled. During this stage, the material begins to cool down by exchanging heat with the mold. Considering an alloy, as the material passes through the temperature range between solidus and liquidus it starts to solidify and concludes this phase as a solid. The final phase is the cooling process. In this stage, the now completely solid material continues to exchange heat with the mold or, if extracted from the mold, with the surrounding air. It gradually reaches thermal equilibrium with the environment. The last two phases, solidification and cooling, could also be considered as a single phase, as the liquid metal inside the cavity is static at the beginning of the solidification phase, so the process could be considered purely thermal apart from the volumetric shrinkage.

5.1 Filling phase

Prior to the filling phase, the material is heated beyond its melting point and, depending on the casting process type, is poured or pressed into the mold. The design of the mold is crucial to facilitate rapid filling, preventing premature solidification before the cavity is completely filled. To mitigate the risk of premature solidification, the material is heated well above the melting point, typically by over 100 °C. This precaution is necessary because the boundary layer tends to solidify instantaneously upon contact with the cold mold, and if the temperature is not sufficiently high, other parts of the flow may also start to solidify, potentially blocking the rest of the material flow. During this filling phase, the material is entirely molten and behaves as a highly viscous liquid and the typical properties of a metal as Young modulus and Poisson ratio are not valid during this stage.

5.2 Solidification phase

The primary physical phenomenon to model is solidification, which occurs within the molten pool of material. The process initiates with the nucleation of atoms, leading to the formation of the crystal structure. If these nuclei reach a critical size, they develop into grains. Subsequently, the grains grow in various directions, influenced by the temperature gradient [1]. The temperature at which this phase change occurs is composition-dependent. In the case of a pure metal, solidification takes place at a constant temperature known as the solidification temperature. This constancy arises because, during the phase change, latent heat is released, maintaining the temperature at a constant level.

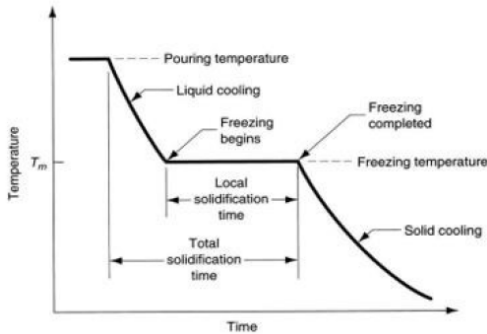


Figure 1: cooling curve of pure metal [2]

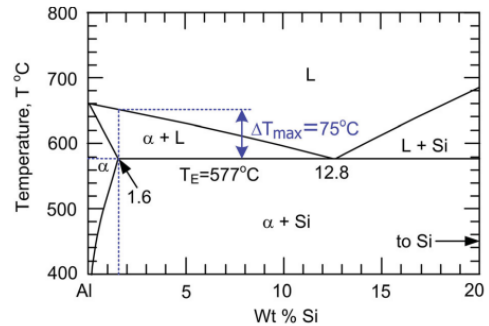


Figure 2: phase diagram of an Al Si alloy

In the case of alloys, the phase transformation is more intricate. This transformation occurs over a temperature range, which is contingent on the alloy's composition. The two extreme temperatures within this range are referred to as the liquidus temperature and solidus temperature. The liquidus temperature signifies the lowest temperature at which the entire alloy is in a liquid state, while the solidus temperature denotes the maximum temperature at which all the material is entirely solid. The composition of the liquid phase plays a crucial role in determining the final composition of the material. This is due to the fact that during solidification, the alloy does not solidify uniformly at once; instead, it solidifies over different times. Consequently, during this process, the

various elements dissolved in the alloy are not evenly distributed throughout the solid matrix. This leads to the formation of grains with varying compositions and distinct matrix structures.

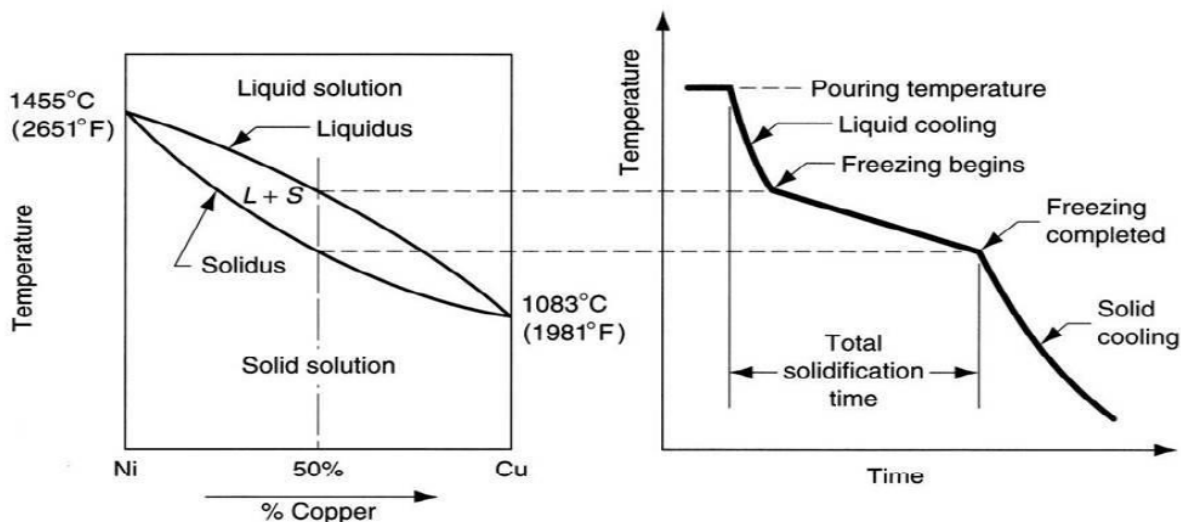


Figure 3: phase diagram and cooling curve of an Al Cu alloy [2]

5.3 cooling phase

In the final stage of the casting process, the material fully occupies the cavity volume and is completely solid, resulting in a velocity of zero for the entire material. The cast part continues to exchange heat with the mold or the surrounding air, gradually approaching an equilibrium temperature with the environment. The cooling process starts from the external surfaces and progresses towards the center of the part. This temperature gradient creates variations in local material properties due to distinct crystal structures, generating undesired residual stresses in the final part. Therefore, mold design must consider the cooling phase, for example by incorporating cooling channels and heat sinks to facilitate even solidification of the part.

At the beginning of this stage, the material exhibits solid properties, typically with low stress resistance, as the temperature remains close to the solidus point. With time, the material gradually acquires its typical behavior and properties.

6 Numerical method

To numerically solve the equations in the model, the continuous domain must undergo discretization through a definition of a mesh. Subsequently, a numerical method is selected to solve the discretized equations. The primary numerical methods include Finite Element Method (FEM), Finite Volume Method (FVM), and Finite Difference Method (FDM), briefly explained below. These three methods could be solved with different

formulations of the numerical problem such as Lagrangian and Eulerian. Furthermore, these two formulations can be combined to derive formulations more suitable for specific types of problems, such as Arbitrary Lagrangian-Eulerian (ALE) and Coupled Eulerian-Lagrangian (CEL). Another specific formulation derived from the Lagrangian approach is Smoothed Particle Hydrodynamics (SPH), which, while accurate, is computationally demanding.

Each method and formulation comes with its own set of advantages and disadvantages, and the choice is heavily dependent on the specific problem being modeled and the desired results. In this work, the Finite Element Method (FEM) is employed in Abaqus with an Eulerian formulation due to its accuracy in tracking material flows, which is essential for simulating the filling phase of the casting process. Subsequently, a model is developed using the Coupled Eulerian-Lagrangian (CEL) formulation to showcase its capabilities in modeling both the mold as a solid and the flow of molten metal. Additionally, a model with a Lagrangian formulation is implemented to simulate the thermal aspects during the solidification and cooling phases. This choice is driven by its swiftness and the absence of the need to consider flows inside the material during these particular phases.

6.1 Discretization

The problem at hand is represented by differential equations within a continuous domain, but prior to solving it numerically, discretization is necessary. There are two approaches to discretize the problem: Eulerian and Lagrangian. The Eulerian approach involves computing the variation in time of physical properties at a fixed point. In contrast, the Lagrangian approach solves the problem by tracking a particle and computing the variation around it.

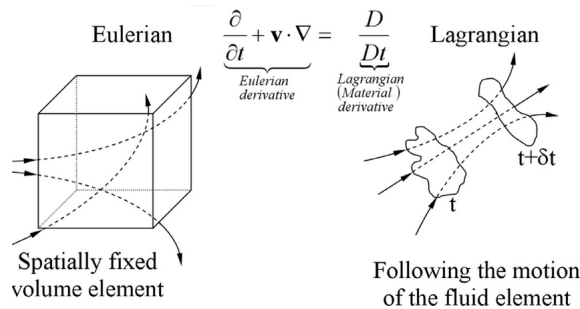


Figure 4: Eulerian vs Lagrangian [31]

6.2 Finite Element Method (FEM)

The Finite Element Method (FEM) involves discretizing the domain into elements composed of nodes and edges, where the properties values are stored. The method then approximates the unknown function over the element domain. These approximations form a system of algebraic equations that can be expanded to consider all the degrees of freedom in the problem and subsequently solved. The shapes of these functions, which define the properties within the element, depend on the number of nodes, the shape of

the element, and the properties of the element. In the context of the heat equation, the FEM formulation is expressed as [6]:

$$KT + C\dot{T} = F \quad (6.1)$$

In the given context, where K represents the conductivity matrix, C stands for the capacity matrix, T is the vector of temperature, \dot{T} is the vector of temperature derivatives with respect to time and F is the vector of external loads, inclusive of both thermal loads and forces. In stress and strain analysis, the Finite Element (FE) formulation is as follows:

$$\sigma = K \cdot d \quad (6.2)$$

Where σ is the stress vector, K is the stiffness matrix and d is the vector of displacement of the nodes.

6.3 Finite Volume Method (FVM)

The Finite Volume Method (FVM) relies on solving conservative laws in integral form. Specifically, for conservative discretization, the sum of the solution in a sub-domain Ω_i with $i = 1, 2, 3 \dots n$ is equivalent to the solution of the entire domain $\Omega = \sum_{i=1}^n \Omega_i$. A general conservation law is expressed as:

$$\frac{\partial}{\partial t} \int_{\Omega} U d\Omega = - \int_S \vec{F} \cdot \vec{n} dS + \int_{\Omega} Q_v d\Omega + \int_S \vec{Q}_s \cdot \vec{n} dS \quad (6.3)$$

6.4 Finite Difference Method (FDM)

The Finite Difference Method (FDM) is a numerical technique employed to solve partial differential equations (PDE). This method relies on the Taylor series expansion to solve derivatives by computing finite differences from the truncation of the limit that defines the derivative. The approach involves discretizing the domain into small control volumes with constant properties inside. For instance, the first-order forward 1st derivative in x can be expressed as:

$$\frac{\partial u}{\partial x} = \lim_{x \rightarrow 0} \frac{u(x + \Delta x) - u(x)}{\Delta x} \quad (6.4)$$

By eliminating the limit and expressing $u(x + \Delta x)$ as a Taylor expansion, the partial derivative of u with respect to x in the point i is obtained as:

$$\frac{\partial u}{\partial x} \Big|_i = \frac{u(x + \Delta x) - u(x)}{\Delta x} + \varepsilon \quad (6.5)$$

Considering the truncation error ε , it is proportional to Δx^n , where n represents the order of the formula, which, in this case, is first order

6.5 Coupled Eulerian Lagrangian (CEL)

The Coupled Eulerian Lagrangian method, also known as CEL, is a modeling approach that facilitates the interaction between two bodies described in distinct formulations. Typically, the problem is formulated with an Eulerian fixed domain containing a Lagrangian deformable body free to move. Initially, the Lagrangian domain is constrained not to overlap with the Eulerian material instance but only with void regions [49]. The assignment of material to a specific zone within the Eulerian domain is achieved through the volume fraction method, which utilizes a variable to determine whether an element contains material or not. To model the interaction between the two bodies, various interaction properties, depending on the problem, must be defined. The CEL formulation allows for large displacements with great material tracking accuracy without encountering the errors associated with classic Lagrangian formulations.

6.6 Arbitrary Lagrangian Eulerian (ALE)

The Arbitrary Lagrangian-Eulerian (ALE) formulation combines the advantages of both methods, offering accurate material tracking typical of the Lagrangian method and the mesh regularity characteristic of the Eulerian method. This formulation utilizes two reference coordinate systems and meshes, one for each formulation along with an additional arbitrary coordinate system and mesh [28] [35]. The supplementary ALE mesh moves independently of the material, resulting in a velocity disparity between the material and the mesh, dependent on how the ALE mesh is moving. Two specific cases represent extremes: when the mesh moves with the material (Lagrangian) and when the mesh remains stationary (Eulerian). This velocity difference constitutes the advective term, representing the material flow through the mesh. In the ALE formulation, it is crucial to compute the ALE mesh velocity with respect to the Eulerian mesh that tracks it, as well as the relative velocity of the material with respect to the ALE mesh. This flexible mesh movement approach helps maintain less deformed element shapes, facilitating easier computations. The algorithm for solving ALE equations divides the timestep into two phases. The first, a Lagrangian phase, computes the variations in velocity and internal energy as the mesh moves with the material. The second phase calculates the transport of mass, momentum, and internal energy across cells in the advection phase. This approach enhances surface tracking accuracy and improves transport computations without necessitating an extremely fine mesh.

The complexity of the ALE method lies in choosing the mesh update at each timestep—by defining the velocity or displacement of the mesh nodes—since this choice significantly impacts the results. Mesh regularization approaches, maintaining the mesh as regular as possible to minimize distortion errors, offer one solution. Another method involves adaptive mesh movement to enhance accuracy, such as refining the mesh by moving nodes toward zones with high gradients.

6.7 Smoothed Particle Hydrodynamics (SPH)

The Smoothed Particle Hydrodynamics (SPH) is a Lagrangian method that discretizes the domain without the need for a mesh. It assigns values of the quantity field to the centers of elements, now considered particles, and establishes relations between these particles based on the constitutive equation [27, 47]. The evolution of a quantity in a particle depends solely on neighboring particles within a certain radius $2h$.

$$A(\mathbf{r}) = \sum_b m_b \frac{A_b}{\rho_b} W(\mathbf{r} - \mathbf{r}_b, h) \quad (6.6)$$

Here, W represents the interpolating kernel, h is the interpolation length, A_b is the value of the quantity A at \mathbf{r}_b , and m_b and ρ_b are the mass and density of neighboring particles, respectively.

Boundary conditions can be imposed by introducing boundary particles that interact with other particles through forces. This method proves useful for simulating free surface flow, as its Lagrangian nature ensures automatic tracking of the flow. Additionally, since particles contain all computational information and do not rely on a mesh, issues related to element deformation are avoided.

7 Simulation

To conduct numerical simulations, it is essential to have models that accurately describe material behaviors while remaining simple enough for analysis and computation within a reasonable timeframe. The model needs to account for fluid metal behavior during pouring, solidification, and the subsequent behavior of the solid material during cooling. In the liquid phase, an accurate mathematical model consists of the three Navier-Stokes equations governing mass, momentum, and energy conservation. The boundary conditions include the no-penetration condition of the mold by the fluid and the results of the heat problem solved at the material mold interface. Heat exchange occurs through convection, conduction, and radiation for the fluid, while the solid experiences only radiation and conduction [34]. The heat problem within the mold can be simulated, or boundary conditions on the fluid can be derived from theory or experimental results. It's noteworthy that during pouring, the mold is already filled with air, although not all models account for this factor.

In the case of metals, the solid is denser than the liquid, causing the material to shrink inside the mold during phase change, creating an air gap between the mold and the part. This affects final dimensions, tolerances, and heat flux during the casting process [3].

After the solidification, the material constitutive equations are the generalized Hook's law, relating strain and stress, and Fourier's law, relating heat flux and temperature. Boundary conditions involve imposing temperatures or heat fluxes at the interface and preventing penetration. At the solid-liquid interface, there are transfers of mass, momentum, heat, and species, influenced in part by the phase change. Mass transfer is

proportional to the velocity of the interface related to the phase change, while momentum, heat, and species transfers are proportional to mass transfer. Additional sources of momentum, heat, and species exchange include convection and diffusion. For momentum, this involves dissipative stress caused by the flow moving around a solid for the liquid part, following Darcy’s law [8] and a solid moving in a liquid, where a drag force is applied. The momentum takes the form of a coefficient multiplied by the difference in velocity between the two phases. Heat and species transfers are modeled as dissipative stresses with proportional coefficients multiplying the difference in heat or concentration between the solid and the interface.

7.1 Fluid and solid simulation

To numerically simulate phenomena involving both liquid and solid phases, a mathematical formulation of the problem is required to describe the distinct behaviors of the two phases in a manner that is both accurate and straightforward to model and solve. The model must incorporate a method for distinguishing between the domains and solving different equations or additional terms to consider their distinct physical behaviors. Various approaches exist to achieve this:

- One approach is to model both phases as a liquid, requiring the solution of the Navier-Stokes equations throughout the entire domain [17]. In this method, viscosity is a function of temperature. For temperatures above the liquidus, it adopts the normal viscosity of the fluid. For temperatures below the solidus, the viscosity becomes infinite. Between these temperature limits, viscosity takes on a value between the normal viscosity and infinity, effectively simulating the coexistence of solid and liquid within the cell [32].

$$\mu = \begin{cases} \mu_l(T) & T > T_l \\ 0 < \mu < \infty & T_s < T < T_l \\ \infty & T < T_s \end{cases} \quad (7.1)$$

In this approach, where T_s represents the solidus temperature, T_l the liquidus temperature, and μ denotes viscosity. A similar strategy involves treating the solid as an extremely viscous medium. However, in this case, viscosity in the mushy zone is correlated with the solid fraction rather than directly with temperature [33].

- Another approach to model the interaction between solid and liquid involves considering the entire domain as a fluid. In the solid region, penalization terms are added to the momentum equation to enforce rigid body modes inside the solid [19,20]. The penalization term, $\beta\chi_s(u_i - u_l)$, comprises the penalization parameter β (typically greater than 10^6), the characteristic function of the solid χ , and u_i representing the rigid body velocity (average of velocity and angular velocity of the solid), while u_l denotes the flow velocity.
- Simulation can be conducted using two distinct software, one for Computational Fluid Dynamics (CFD) and another for stress and strain analysis, with coupling mechanisms in place [18]. The coupling process exchanges information between the fluid simulation and the structural analysis. Specifically, the fluid loads on

the structure from the previous time step, expressed as pressures, are transmitted to the Finite Element Method (FEM) simulator to compute displacements for the current time step. Subsequently, these displacements are conveyed to the CFD simulation to calculate fluid loads at the current time step. This iterative process continues, emphasizing the importance of synchronizing time steps for accurate coupling between the two simulations.

- The Immersed Boundary Technique builds upon the Arbitrary Lagrangian-Eulerian (ALE) method and employs a dual-mesh approach—comprising a background Eulerian mesh for the fluid and a Lagrangian mesh for the solid [?,25]. The interaction between the two involves surface forces \vec{F} and body forces \vec{B} calculated within the Lagrangian mesh. These forces are then transferred to the fluid through appropriate boundary conditions in the momentum equation. The mass and momentum equations are solved while considering the movement of the solid with local velocity and adhering to traction boundary conditions (no-slip) to couple the two domains. A similar approach, the Coupled Overlapping Domain technique (COD) [21], also employs a dual-mesh structure. In addition, it can activate or deactivate nodes in the background based on the Lagrangian mesh’s position, addressing the issue of fictitious fluid in the overlapped zones encountered in the Immersed Boundary approach. The solid-fluid coupling is accomplished by imposing the condition $v = \dot{u}$ at the boundary, where v represents fluid velocity and u denotes solid nodal displacement. In both methods, the algorithm revolves around solving for velocity and pressure in the fluid (Eulerian mesh), computing forces in the solid (Lagrangian mesh), transmitting them to the fluid, and ultimately updating the Lagrangian mesh.
- The coupling of a deformable solid with an incompressible fluid can be achieved through an Eulerian formulation for both components [22]. In this approach, the Eulerian solid employs a material space coordinate field from which the displacement field \mathbf{u} is computed. This displacement field is then used to calculate the deformation gradient \vec{F} using the expression:

$$\vec{F} = \left(\vec{I} - \frac{\partial \vec{u}}{\partial x} \right)^{-1} \quad (7.2)$$

Here, \vec{F} and a constitutive model are utilized to compute the internal forces within the solid.

To facilitate computations and apply appropriate models, the position of the solid is tracked. This tracking mechanism also enables the handling of contact between solids. The coupling is achieved by solving the momentum equation, incorporating an elasto-plastic/hyper-elastic model for the stress tensor σ in the solid domain and setting $\sigma = 0$ in the fluid domain. The entire system is solved using the Finite Element Method (FEM), where stiffness and mass matrices are computed, accounting for the presence of either solid or liquid. Velocity, displacement, and force values are stored at the nodes, while pressure is stored at the center of the cells.

- One approach to simulate the solidification process involves solving the solid constitutive equation to obtain the stress tensor $\vec{\sigma}$, considering the strain rate as

$$\dot{\epsilon} = \dot{\epsilon}_{el} + \dot{\epsilon}_{ie} + \dot{\epsilon}_{th} \quad (7.3)$$

Here, the subscripts el, ie, and th respectively denote elastic, inelastic, and thermal components.

To account for the liquid zone, a constitutive law is chosen such that, when the temperature is higher than the solidus temperature, it results in negligible strength and stress [23]. This is achieved in the elastic-perfectly plastic model by imposing a yield stress $\sigma_y = 0.03$ MPa to eliminate the strength of the liquid. Alternatively, the rapid creep rate function is employed, imposing a viscoplastic relation that generates strain as a penalization term proportional to the difference between the equivalent stress and the yield stress. All equations are solved using the Finite Element Method (FEM) solver ABAQUS, with the implementation of user subroutines.

- The Extended Finite Element Method (XFEM) offers a solution for tracking the liquid-solid interface without the need for continuous mesh updates, as opposed to the classical Finite Element Method (FEM). XFEM achieves this by enriching nodes and elements close to the discontinuity with additional basis functions to capture the desired features [26]. A level function is employed to track the interface, locating it where the function has a zero value.

The FEM formulation of the problem must satisfy both mechanical equilibrium, represented as $[K]\{u\} = \{F\}$, and thermal equilibrium, given by $[K_T]\{T\} = \{Q\}$, where $[K]$ and $[K_T]$ are stiffness matrices, u is displacement, T is temperature, and Q is the thermal force. The matrices $[K]$ and $[K_T]$ exhibit discontinuities due to the interface, and the elastic modulus matrix $[D]$ and the thermal conductivity k must be appropriately chosen to reproduce solid or liquid behavior.

7.2 Solid/liquid interface

During the casting process, a binary system of solid and fluid forms. The two phases exhibit distinct properties, leading to a discontinuity at the interface. Moreover, as the material undergoes solidification, changes in properties result in density variations, inducing natural convection that shifts the interface. Consequently, tracking the interface becomes imperative. The Stefan condition represents the heat condition at the interface, yielding the normal velocity of solidification, denoted as the interface velocity:

$$V_n = \frac{1}{\rho_s L_f} \left[k_s \left(\frac{\partial \phi}{\partial n} \right)_s - k_l \left(\frac{\partial \phi}{\partial n} \right)_l \right] \quad (7.4)$$

Here, ϕ represents temperature, subscripts s and l denote solid and liquid, n is the normal vector, and k is the conductivity [36]. Neumann expanded upon Stefan's work [16], considering two phases in the study of the melting of an infinite slab with non-constant temperature. This purely thermal problem provides an analytical solution applicable to rectangular coordinates [15]. Interface tracking methods vary between Lagrangian and Eulerian approaches. The former employs an evolving mesh that tracks the free surface by moving nodes, while the latter uses a fixed mesh with an additional variable φ that signifies the state of the interface.

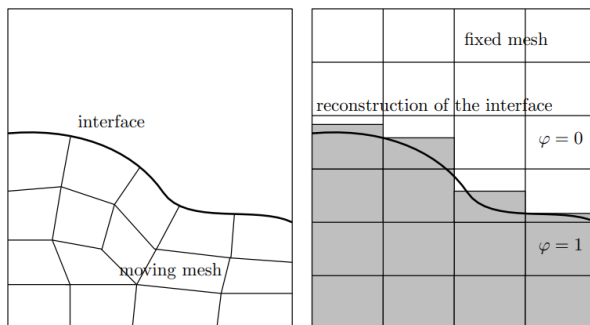


Figure 5: Lagrangian approach (left) and Eulerian approach (right) [35]

Additionally, alternative methods exist for tracking the surface [35]. The Particle-in-Cell (PIC) method, for instance, adopts a meshless approach utilizing massless particles (markers) that move with the flow, as illustrated in Figure 6. Another approach employs a level set function φ , where a value of zero indicates the interface, and positive or negative values delineate the two sides of the surface, as depicted in Figure 7.

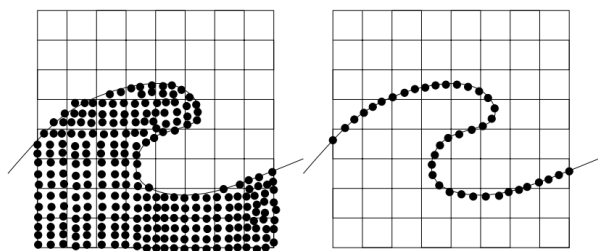


Figure 6: PIC [35]

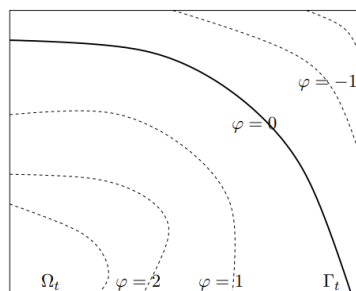


Figure 7: level function [35]

Another approach involves using the pseudo-concentration function [41], where a function F varies between -1 and 1 across the interface, following a first-order pure advection equation.

$$\frac{\partial F}{\partial t} + (\vec{u} \cdot \nabla) F = 0 \quad (7.5)$$

7.3 Volume of fluid

The Volume of Fluid (VOF) method is employed to model multiphysical phenomena, particularly the interface of phase-change materials and the simultaneous presence of one or both phases in the computational domain [7, 33]). The volume fraction, denoted as α , characterizes the distribution of fluids within a cell:

$$\alpha = \begin{cases} 1, & \text{cell contains fluid A} \\ 0 < \alpha < 1, & \text{cell contains interface} \\ 0, & \text{cell contains fluid B} \end{cases}$$

The transport equation for α is given by:

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) + \nabla \cdot (\mathbf{u}_r \alpha (1 - \alpha)) = 0$$

where \mathbf{u}_r represents the relative velocity.

To extend this model beyond structural phase changes, thermal analysis is incorporated. The specific enthalpy equation accounts for the sensible heat of the fluid and the latent heat of the material going through phase change:

$$h = \int_{T_{\text{ref}}}^T c_p dT + \alpha \gamma(T) L = c_p (T - T_{\text{ref}}) + \alpha \gamma(T) L$$

where α serves as an auxiliary variable defining the cell composition, T_{ref} is the reference temperature, c_p is the specific heat at constant pressure, and $\gamma(T) \in [0, 1]$ is the melt fraction. The melt fraction, a function of temperature, is defined as:

$$\gamma(T) = \begin{cases} 0, & T < T_{\text{sol}}, \text{ (solid)} \\ \frac{T - T_{\text{sol}}}{T_{\text{liq}} - T_{\text{sol}}}, & T_{\text{sol}} \leq T \leq T_{\text{liq}}, \text{ (mushy zone)} \\ 1, & T > T_{\text{liq}}, \text{ (liquid)} \end{cases}$$

Here, T_{sol} and T_{liq} represent the solidus and liquidus temperatures, respectively. In this temperature range, some parts of the metal are solid, while others remain liquid, and this is modeled by incorporating a term in the momentum equation derived from Darcy's law ([8]) that describes the slow movement of fluid in a porous medium ([7]).

7.4 Mushy zone

During the solidification process, a mushy zone emerges in the domain, representing the coexistence of the two phases, liquid and solid. Several methods exist to model solidification, one of which is the continuous nucleation model. This model describes heterogeneous nucleation in both the melted pool and at the mold surface using a Gaussian distribution, with undercooling acting as the triggering factor for nucleation:

$$n(\Delta T) = \int_0^{\Delta T} \frac{dn}{d(\Delta T)} d(\Delta T) \quad (7.6)$$

Here, n is the grain density, and ΔT is the local undercooling. The automaton cell method is employed to model microstructure, assigning parameters such as temperature, state (liquid or solid), and crystallographic orientation. These parameters are updated at each step, considering nucleation and growth kinetics, grain density, and dendritic growth (modeled with KGT), and then coupled with the macroscopic model.

To address phase change in the mushy zone's heat problem, the enthalpy form of the energy conservation equation is often utilized [29, 37, 39], in some model the enthalpy is a

piecewise function [30]. The enthalpy equation incorporates latent heat of solidification as a source term [11]:

$$S_h = \frac{\partial}{\partial t}(\rho f_L L) - \nabla \cdot (\rho \vec{u} f_L L) \quad (7.7)$$

Here, L is the latent heat, and f_L represents the liquid fraction, indicating the presence of solid, liquid, or mushy zones. The solid or liquid fraction is typically a function of temperature [41], following the lever rule for binary alloys:

$$f_l = \frac{T - T_l}{T_l - T_s} \quad (7.8)$$

for $T_s \leq T \leq T_l$, where T_s is the solidus temperature and T_l is the liquidus temperature. f_l has a value of one in the liquid zone and zero for the solid zone. More complex models, such as the one presented in [11], introduce equations like:

$$f_s = \frac{(T_l - T) + \frac{2}{\pi}(T_s - T_l) \left(1 - \cos \left[\frac{\pi(T - T_l)}{2(T_s - T_l)} \right] \right)}{(T_l - T_s) \left(1 - \frac{2}{\pi}\right)} \quad (7.9)$$

Solidification in the mushy zone introduces additional terms to the momentum equation due to the movement of forming solid and liquid flow in the closing space around:

$$\vec{S} = \frac{\mu}{K} \vec{u} \quad (7.10)$$

Here, K is the permeability of the mushy zone, and μ is viscosity. Other models, like the one in [41], distinguish between columnar interdendritic flow and equiaxed mushy regions, introducing terms like:

$$\vec{S} = K_0 \frac{f_s^2}{(1 - f_s)^3} \vec{u} \quad (7.11)$$

Here K_0 represents the permeability coefficient, and f_s denotes the solid fraction. This term is derived from Darcy's law, capturing the interaction between the fluid flow and the mushy zone. On the other hand, in the case of equiaxed mushy regions, instead of introducing an additional term like in Equation 7.11, the viscosity is modified. This modification is based on a function involving the viscosity of both solid and liquid phases and the solid fraction 7.12. The solid viscosity is adjusted to a sufficiently large value to effectively bring the velocity to zero within the solid region. This ensures that the viscosity modification contributes to modeling the distinct behavior in equiaxed mushy zones.

$$\mu_{\text{mush}} = \mu_l \left(\frac{\mu_s}{\mu_l} \right)^{f_s} \quad (7.12)$$

In another model for the mushy zone [9], a two-phase approach is taken for an isotropic mushy zone where both solid and liquid phases can move. Conservation equations for mass, momentum, and energy are formulated, accounting for phase transformations caused by mass transfer. The mass conservation equation is the sum of the ones for the solid and liquid and is defined as:

$$-(\rho_s - \rho_g) \frac{\partial g_s}{\partial t} - g_s \left(\frac{\partial}{\partial t} + \vec{v}_s \cdot \nabla \right) \rho_s = \rho_l \nabla \cdot (g_l \vec{v}_l) + \rho_s \nabla \cdot (g_s \vec{v}_s) \quad (7.13)$$

Here g_i is the volume fraction of phase i that is computed by the lever rule for a binary alloy, ρ_i is the density of phase i and \vec{v}_i is the velocity of phase i . The first term is the solidification shrinkage the second one is the cooling contraction and both contribute to the source term on the right side.

Some approximations are made (see [9]) and the conservation equation for the energy with contributes from solid and liquid is written as:

$$-(g_s \rho_s C_s + g_l \rho_l C_l) \frac{\partial T}{\partial t} + (g_s \rho_s C_s \vec{v}_s + g_l \rho_l C_l \vec{v}_l) \cdot \nabla T = \nabla \cdot [(g_s \lambda_s + g_l \lambda_l) \nabla T] + L \Gamma \quad (7.14)$$

Here, C_i is the heat capacity, λ is the heat conductivity, L is the latent heat, and Γ represents the mass transfer caused by phase transformation. The index $i=[s,l]$ defines the solid or liquid phase.

The momentum equation undergoes a similar treatment. In this context, the equations for elastic and inelastic strain, denoted by ε , are derived under the assumption that thermal-induced deformation is exclusively present in the solid phase. The momentum exchange between the two phases is governed by Darcy's law [8]:

$$\vec{M} = \frac{1}{K(g_l)} g_l^2 \mu (\vec{v}_l - \vec{v}_s) \quad (7.15)$$

Here, μ represents viscosity, and K denotes permeability. The stress-strain relationship for the mushy zone follows the creep law, with the parameters n and k being functions of the volume fraction of the liquid phase and the volume fraction of the coherent liquid phase:

$$g_s \vec{\sigma}_s = k \varepsilon_s^n \quad (7.16)$$

Yet another model [10] describes the entire domain as a mixture of solid and liquid, employing only the conservation equations for mass, momentum, and energy. This approach considers the material as a continuous blend of the two phases.

7.5 Boundary condition

Boundary conditions during casting are contingent on the chosen model, encompassing both thermal and mechanical aspects. Mechanical boundary conditions vary significantly

based on the chosen model and the materials involved. For the liquid phase, a no-penetration condition with the solid is generally imposed. Additionally, conditions of slip (for inviscid fluids) or no-slip (for viscous fluids) are considered. As for the solid phase, boundary conditions include no penetration with the mold and no slip, often expressed as normal traction forces 7.17 and tangential traction forces 7.18 [41].

$$\sigma_n = -p + 2\mu \frac{\partial u_n}{\partial n} \quad (7.17)$$

$$\sigma_\tau = \mu \left(\frac{\partial u_n}{\partial \tau} + \frac{\partial u_\tau}{\partial n} \right) \quad (7.18)$$

Considering n and τ as the normal and tangential vectors, and μ as the viscosity, thermal boundary conditions can be expressed either as Dirichlet conditions by imposing a temperature (at the interface with the mold or with the air) or as Neumann conditions by imposing a heat flux at the interface.

Throughout the solidification process, the density of the specimen increases, leading to shrinkage. This results in a change in the thermal boundary condition with the mold [34, 39]. Initially, when the material is in direct contact with the mold, heat exchange occurs over the entire surface through conduction 7.19. Subsequently, as an air gap begins to develop, heat exchange differs because only the asperities are in contact, and eventually, it occurs exclusively through the air gap [12]. This gap can be modeled using diffusion 7.20 or Newtonian heat transfer 7.21 (refer to [34]).

$$\left(\frac{\partial \phi}{\partial \vec{n}} \right)_{m1} = \left(\frac{\partial \phi}{\partial \vec{n}} \right)_{m2} \text{ and } \phi_{m1} = \phi_{m2} \quad (7.19)$$

$$K_m \left(\frac{\partial \phi}{\partial \vec{n}} \right)_m = K_s \left(\frac{\partial \phi}{\partial \vec{n}} \right)_s \quad (7.20)$$

$$K_m \left(\frac{\partial \phi}{\partial \vec{n}} \right)_m = h_i(\phi_m - \phi_s) \quad (7.21)$$

In the context where Φ represents temperature, m , s , and a stand for the mold, the solid, and the environment, K is the thermal conductivity, h is the Newtonian heat transfer coefficient, and \vec{n} is the normal vector at the boundary. The convective heat transfer can also be expressed as a function of the Nusselt, Grashof, and Prandtl numbers (Nu , Gr , and Pr respectively) [40].

7.6 Thermo-mechanical model

The viscoelastic/plastic model is the most comprehensive approach to describe the thermo-mechanical behavior of a metal during casting. It is formulated by a set of five equations.

$$\sigma = D\varepsilon_e \quad (7.22)$$

$$\varepsilon_e = \varepsilon - \varepsilon_T - \varepsilon_v - \varepsilon_{pc} \quad (7.23)$$

$$\varepsilon_T = \alpha \Delta T \quad (7.24)$$

$$\varepsilon_v = f(\sigma_n, T, \varepsilon, s) \quad (7.25)$$

$$\varepsilon_{pc} = \beta f_s \quad (7.26)$$

Here, D denotes the stiffness matrix, ε represents the total strain, and the indices T , v , pc correspond to thermal, viscous, and phase change components, respectively [13]. The thermal strain is defined as the change in temperature multiplied by the coefficient of thermal expansion. The phase change strain is determined by the alteration in solid fraction multiplied by the volumetric contraction. The viscous strain is derived from the law of power creep at low temperatures and follows the Norton-Hoff law.

$$\varepsilon_v = \left(\frac{\sigma_{net}}{K} \right)^{\frac{1}{m}} \quad (7.27)$$

Here, K represents the consistency of the metal, σ_{net} denotes the stress exceeding the threshold where viscoplastic phenomena occur, and m is the strain rate sensitivity. The overall strain can also be considered as the sum of elastic and irreversible strains, both dependent on the phase state. In certain scenarios, the molten phase may be approximated as a solid. For instance, this approximation is applicable when the molten pool is small and appears for a short duration. The phase transformation induces strain in the material, resulting from changes in density. Additionally, there are models for thermal, viscous, and plastic strain [14].

$$\varepsilon_{sol}^{th} = \alpha_{sol} [\theta - \theta_{ref}] I \quad (7.28)$$

$$\dot{\varepsilon}_{mel}^{vis} = \frac{1}{\eta_{mel}^{vis}} \sigma \quad (7.29)$$

$$\dot{\varepsilon}_{sol}^{pl} = \lambda \frac{\partial \Phi}{\partial F_{loc}^{pl}} \quad (7.30)$$

Here, α denotes the isotropic heat expansion coefficient, θ represents the temperature, θ_{ref} is the reference temperature, λ stands for the Lagrange multiplier, and Φ is the yield function. F_{loc}^{pl} represents the local plastic driving force, η_{mel}^{vis} is a viscosity-related material constant, and σ is the stress. All the strain terms (thermal, elastic, and inelastic) can also be expressed in the form of strain rate $\dot{\varepsilon}$ since the generation of stress is a transient phenomenon during solidification [38].

The Helmholtz free energy density $\Psi(\varepsilon, \theta, \nu)$, which depends on stress ε , temperature θ , and another internal variable ν , is employed to model the material [14]. The Helmholtz free energy comprises a thermal part, which is a function of temperature, and a mechanical part, which is a function of strain. The strain can be readily computed from the energy equation.

$$\vec{\sigma} = \rho \frac{\partial \Psi}{\partial \varepsilon} \quad (7.31)$$

During the cooling and solidification process, the material in contact with the mold experiences lower temperatures than the material within. This discrepancy results in distinct thermal histories, giving rise to residual stress and deformation within the material due to the temperature gradient.

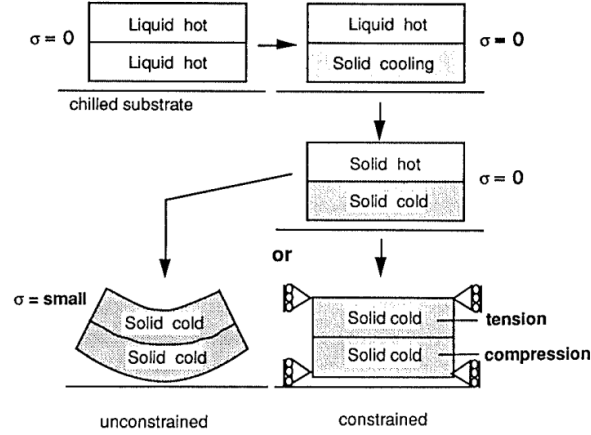


Figure 8: Shell solidification without friction [38]

7.7 Abaqus theory

7.7.1 Eulerian formulation

The Eulerian formulation is based on a fixed grid of elements through which the material flows. This formulation outperforms the Lagrangian approach for problems involving significant material deformation, as the grid remains fixed instead of being integral with the material. In this approach, material is not assigned to an element but can move through the grid, requiring computation of the material boundary at each time increment to track its position.

In Abaqus/Explicit, the Eulerian formulation is based on the volume of fluid method (paragraph 7.3), used to compute the Eulerian volume fraction (EVF). This method enables an Eulerian element to contain multiple materials. For each element, the sum of all material contributions to the volume fraction must equal one, ensuring the element is fully occupied. Otherwise, Abaqus fills the element with a fictitious material, "void," to satisfy the condition $EVF=1$. The material's external boundary is computed based on the EVF, approximating it as a planar face inside the element with $EVF \downarrow 1$. Elements with $EVF=1$ are considered inside the volume occupied by the material, and nodal averaging algorithms can address visualization issues resulting from this approximation.

The Eulerian time incrementation algorithm consists of two phases: a Lagrangian phase followed by an Eulerian phase, known as Lagrange plus remap [57]. In the Lagrangian phase, nodes follow materials, causing element deformation. In the Eulerian phase, grid deformation stops, and elements with greater deformation are re-meshed. Material flow is computed through the relative motion of nodes with respect to the material, and state variables of the material follow the material itself during advection. Second-order advection is the default setting in Abaqus/CAE and considers linear distribution of the

variable in the element, this is computed by quadratic interpolation of the value in the elements and the adjacent ones. A first linear distribution is differentiated by the quadratic function to compute the slope at the middle point. Then it is limited to be inside the minimum and maximum of the value in the adjacent elements. This is done to guarantee that the advection is monotonic and it is called flux limiting.

Eulerian nodes permit the addition of boundary conditions in the form of velocity and acceleration. However, displacement is not allowed during the Eulerian phase, as automatic re-meshing occurs. By default, materials can flow freely inside and outside the domain, depending on the velocity at boundary nodes. If a boundary element is filled with more than one material the material that will flow inside or outside the domain follows the volume fraction of that element.

Elements compatible with Eulerian analysis in Abaqus are EC3D8R and EC3D8RT, based on the Lagrangian elements C3D8R and C3D8RT, respectively, mechanical and thermally coupled elements. These Eulerian elements allow the use of multiple materials inside the element and support the Eulerian transport phase. Strain and temperature are applied uniformly to all materials within an element, with state variables evolving separately according to the volume fraction then the results are averaged.

Eulerian material interaction is, by default, defined as sticky behavior, enabling the transmission of tensile stress at the interface. More complex interactions can be implemented in Eulerian-Lagrangian simulations (CEL). The Lagrangian body will interact with the Eulerian material and in the overlapped region the Eulerian elements will be filled with void. The interface is tracked automatically, allowing the use of a regular Eulerian grid, which is more efficient. The default contact condition for general contact (supported by the Eulerian formulation) is hard contact for normal behavior, preventing penetration. The default tangential behavior is frictionless, meaning there is no exchange of shear forces.

7.7.2 Thermal equations numerical scheme

Fully coupled thermal-stress analyses are supported by Abaqus/Explicit, addressing temperature and displacement fields simultaneously [52]. This approach is essential when stress and temperature distributions are closely interrelated. The thermal component of the problem is integrated using a forward difference scheme in time in Abaqus/Explicit:

$$\theta_{(i+1)}^N = \theta_{(i)}^N + \Delta t_{(i+1)} \dot{\theta}_{(i)}^N \quad (7.32)$$

Here, θ^N represents the temperature at node N, and i refers to the time increment. This is an explicit scheme because it calculates the temperature at the next time step solely based on the temperature value and its derivative at the current time step. The derivative is computed at the beginning of each time step as:

$$\dot{\theta}_{(i)}^N = (C^{NJ})^{-1} \left(P_{(i)}^J - F_{(i)}^J \right) \quad (7.33)$$

In this equation, C^{NJ} is the lumped capacitance matrix, $F_{(i)}^J$ is the internal flux vector, and $P_{(i)}^J$ is the nodal source vector [58].

The forward-difference operator is conditionally stable, and the stability condition limit for the thermal problem can be approximated as:

$$\Delta t \approx \frac{L_{\min}^2}{2\alpha} \quad (7.34)$$

Here, L_{\min} is the minimum element dimension, and $\alpha = \frac{k}{\rho c}$, where k is the thermal conductivity, ρ is the density, and c is the specific heat.

7.7.3 Mechanical equations numerical schemes

Abaqus/Explicit relies on an explicit central-difference time integration method along with the lumped (diagonal) form of the mass matrix [50]. The equations of motion are given by:

$$\dot{u}_{(1+\frac{1}{2})}^N = \dot{u}_{(1-\frac{1}{2})}^N + \frac{\Delta t_{(i+1)} + \Delta t_{(i)}}{2} \ddot{u}_{(i)}^N \quad (7.35)$$

$$u_{(i+1)}^N = u_{(i)}^N + \Delta t_{(i+1)} \dot{u}_{(i+\frac{1}{2})}^N \quad (7.36)$$

Here, u^N represents the degree of freedom on node N, and i is the increment number. This explicit scheme calculates the next increment using the previously computed values of velocity and acceleration.

The use of the lumped mass matrix is crucial for the efficiency of the explicit method, as its diagonal nature simplifies the computation of its inverse. It is used to compute the acceleration at the beginning of the increment as:

$$\ddot{u}_{(i)}^N = (M^{NJ})^{-1} (P^J - I^J) \quad (7.37)$$

Here, M^{NJ} is the mass matrix, P^J is the load vector, and I^J is the internal force vector. In Eulerian elements, nodes may not require mass or inertia if the neighboring elements are void.

The central difference scheme is stable, with stability dependent on the highest frequency of the system. Abaqus introduces a small damping as bulk viscosity to control oscillations at high frequencies. The stable time increment can be expressed in terms of the smallest element dimension and the dilatation wave speed:

$$\Delta t \approx \frac{L_{\min}}{c_d} \quad (7.38)$$

Here, L_{\min} is the smallest element dimension, and c_d is the dilatation wave speed. This condition ensures that material cannot flow across more than one element in a single time increment. This Δt is not a conservative estimation and for 3D problem it is divided by a factor between $\sqrt{3}$ and 1. The reported stable time increments during the simulation can be found in the status file. Further details about the FEM formulation in Abaqus can be explored in the Abaqus guide [59].

7.7.4 Equation of State

For modeling hydrodynamic behavior, Abaqus provides support for various equations. In this study, the Linear $U_s - U_p$ Hugoniot form is employed to characterize both air and aluminum:

$$p_H = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2} \quad (7.39)$$

Here, c_0 represents the bulk speed of sound, and s is the slope of the linear $U_s - U_p$ Hugoniot form of the equation of state. The relationship between particle velocity U_p and shock velocity U_s is defined by the material parameter c_0 as

$$U_s = c_0 + sU_p \quad (7.40)$$

This leads to the linear $U_s - U_p$ Hugoniot in the form

$$p = \frac{\rho_0 c_0^2 \eta}{(1 - s\eta)^2} \left(1 - \frac{\Gamma_0 \eta}{2}\right) + \Gamma_0 \rho_0 E_m \quad (7.41)$$

Here, E_m denotes energy per unit of mass, Γ_0 is a material constant representing the Gruneisen ratio, and η is the nominal volumetric compressive strain.

8 Model definition

This study focuses on developing a Finite Element Method (FEM) model for casting simulation in Abaqus and subsequently analyzing the results by comparison with the commercial software InspireCast.

Throughout the model development phase, various assumptions and decisions were made to carefully select parameters and problem formulations. The objective was to ensure that the simulation time remains sufficiently short to serve as an effective tool for virtual manufacturing, without compromising the accuracy of the results. It is crucial to note that the primary goal of this work is to create a robust FEM casting model and assess its validity and accuracy. The emphasis is on understanding the utilization of the FEM solver Abaqus to simulate the intricate casting process, rather than optimizing mold geometry to reduce defects in final parts. A balance was requested in choosing the geometry, which needed to be complex enough to depict the flow of molten aluminum and capture potential air entrapment during filling and solidification phases. Simultaneously, it had to be simple enough to facilitate the creation of a mesh with an optimal element count and regular shapes, ensuring efficient simulation times.

Below is a general overview of the main assumptions, with detailed explanations of the decisions made during the model's development provided in the subsequent Abaqus section (refer to paragraph 8.2). The initial decision was to employ an Eulerian formulation, offering precision in tracking molten aluminum during filling and the ability to capture

multiple materials, including air. Eulerian simulations show shorter computation times compared to other formulations like Coupled Eulerian-Lagrangian (CEL) and Arbitrary Lagrangian-Eulerian (ALE). However, drawbacks include the absence of the mold in the model, necessitating proper boundary conditions (mechanical and thermal). Abaqus provides specific conditions for mechanical Eulerian boundary conditions in the load module, allowing users to define material flow properties on external surfaces. For the mechanical one more assumption have to be made considering Abaqus capabilities. While Eulerian simulations offer advantages, they also come with limitations in Abaqus, such as the inability to have density as a function of temperature and the inability to import a thermal map for boundary conditions. These limitations impact the model's accuracy in tracking shrinkage during the solidification phase.

To complement this work, a CEL model is also developed as an extension of the Eulerian model. The CEL model simulates the mold with a Lagrangian domain, eliminating the need for assumptions in the thermal problem. However, this formulation requires tracking the overlap of Eulerian and Lagrangian domains, and interaction properties must be defined. While providing enhanced result accuracy, CEL simulations have longer computation times, making them practical only with substantial computational resources. In this work, a simplified CEL model is developed to illustrate its capabilities.

After analyzing the results of the Eulerian model, a Lagrangian model is also developed to monitor the evolution of the thermal problem for both the mold and the cast part throughout the entire solidification and cooling phases. This formulation does not consider the movement of the molten metal still present at the end of the filling phase. However, it provides a significantly faster yet accurate method for tracking the evolution of the temperature field over an extended period, with shorter run times compared to the Eulerian approach.

8.1 Inspirecast

InspireCast, developed by Altair, serves as a casting simulation environment specifically tailored to support the manufacturing design phase. Engineered for user-friendly interaction, it streamlines simulations by automating numerous actions, enabling rapid assessments without the need for extensive parameter configuration. This tool proves invaluable throughout the design process, aiding in the elimination of defects such as air entrapment, cold shuts, and shrinkage in the final product.

Creating a casting simulation in InspireCast involves two main steps: defining the geometry and specifying the casting process. The initial step requires defining the part's geometry, achievable through InspireCast's CAD capabilities or by importing a pre-defined geometry from a dedicated CAD tool. Once the part's geometry is established, subsequent actions involve defining the cast part, material, mold, gates, and other components, along with detailing the casting process. InspireCast allows users to assign the status of a cast part to solid objects, to differentiate other parts such as inlets, risers, and molds. Material selection is facilitated through a built-in library offering various options, including cast-iron, aluminum, brass, copper, magnesium, and steel. Material properties, accounting for fluid and solid characteristics at varying temperatures, include dynamic viscosity, density, conductivity, specific heat, latent heat, solidus, and liquidus temper-

ature, volumetric shrinkage, solid fraction, Poisson ratio, thermal expansion coefficient, Young modulus, and yield stress. Upon defining the material, users must specify the pouring temperature of the metal, with InspireCast providing recommended initial temperatures for each material.

The next step involves designing gates, which can be done by importing a dedicated geometry or through InspireCast’s rapid definition of inlet surfaces without additional geometry requirements. Subsequently, the mold is defined, accompanied by additional features such as risers, cooling systems, and cores. InspireCast streamlines this process by automatically generating mold geometry without requiring extensive user input.

The final step involves defining the casting process, offering choices like gravity, high pressure, low pressure, and general processes. InspireCast suggests initial process parameters, such as filling time, pressure, or velocity, providing flexibility for manual adjustments. Throughout the model definition, InspireCast provides automatic suggestions, allowing users to conduct quick preliminary simulations while also offering the flexibility for expert users to manually fine-tune parameters, making it accessible to both beginners and seasoned professionals.

8.1.1 Model

The chosen geometry remains consistent to facilitate direct comparison between casting simulation results obtained from Abaqus and InspireCast. The model geometry derives from the requirement for a complex geometry that allows for in-depth analysis of flow patterns and potential defects in the cast part. Simultaneously, the geometry is kept sufficiently simple to prevent excessively long simulation run times, especially within Abaqus. The selected design features a piston head with a double inlet channel affixed to the bottom side and a riser on the top surface. To introduce complexity, holes were strategically incorporated on the sides, and two o-ring grooves were added.

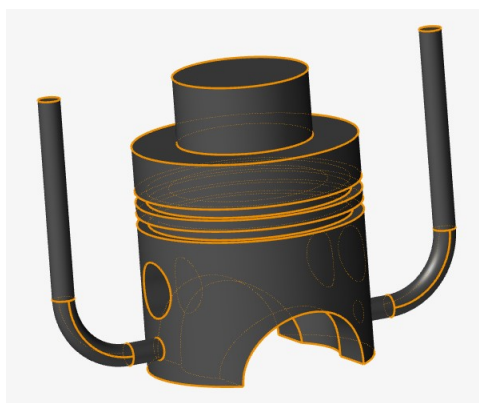


Figure 9: Geometry of the cast part (piston head) with inlet ducts and riser

The geometry was initially created and scaled in the Abaqus CAD environment before being imported into InspireCast. The initial step involved assigning the status of a cast part to this geometry, along with specifying the material and filling temperature. The chosen material is aluminum A7075, an alloy primarily composed of around 5.5% zinc,

with additional elements such as magnesium and copper. This alloy is known for its excellent mechanical properties and corrosion resistance.

The measurement system was set to [mm Kg N s] as the imported geometry was designed in millimeters. InspireCast offers flexibility in selecting different measurement systems based on the design and properties used. It automatically handles conversions without requiring input from the user. The material properties of A7075 in InspireCast are defined over a temperature range from a minimum of 298.15 K (25 °C) to a maximum of 973.15 K (700 °C). The following diagrams illustrate the curve of A7075 material properties as a function of temperature.

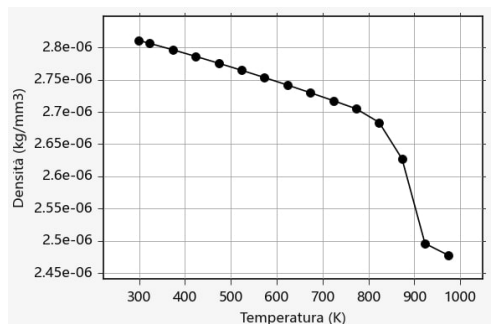


Figure 10: A7075 density curve data from Inspirecast

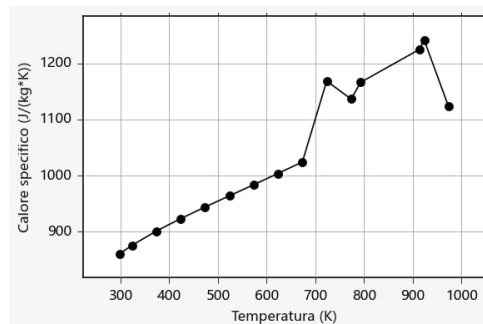


Figure 11: A7075 specific heat curve data from Inspirecast

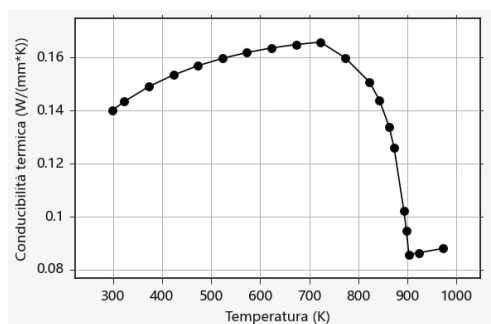


Figure 12: A7075 conductivity curve data from Inspirecast

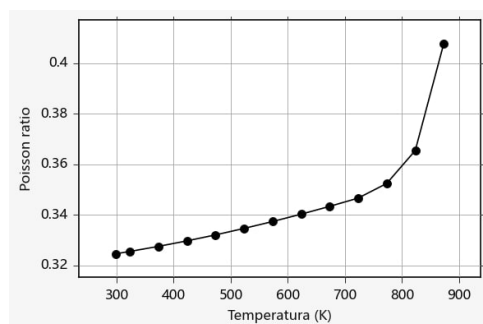


Figure 13: A7075 Poisson ratio curve data from Inspirecast

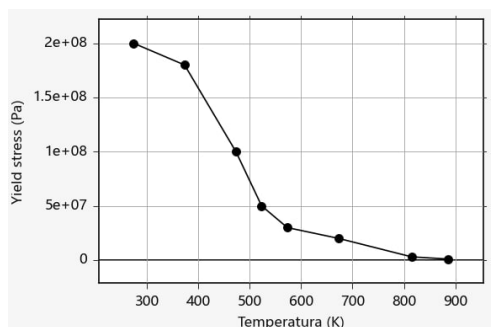


Figure 14: A7075 yield stress curve data from Inspirecast

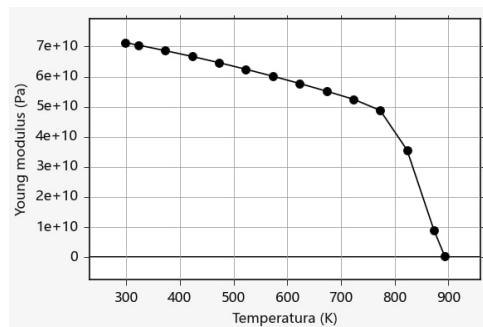


Figure 15: A7075 Young modulus curve data from Inspirecast

It is important to note that the two following curves, the solid fraction and the viscosity are defined only for the temperature interval in which the liquid phase is present as the properties do not have value for the solid state.

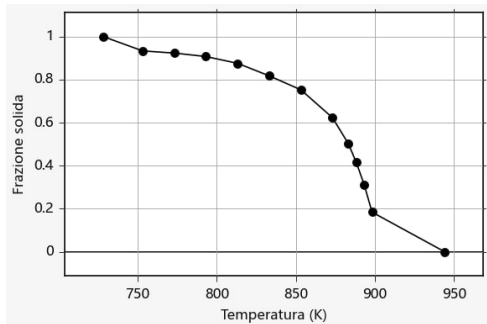


Figure 16: A7075 solid fraction curve data from Inspirecast

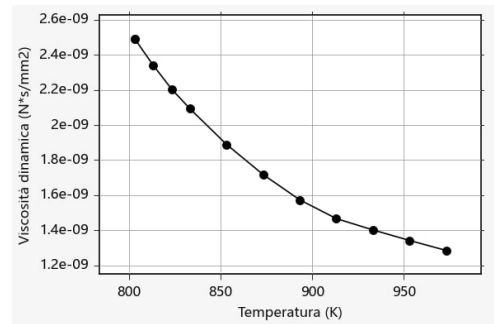


Figure 17: A7075 dynamic viscosity curve data from Inspirecast

Following the definition of geometry and material in InspireCast, the next step involves specifying the inlet by selecting its shape, position, and dimensions. In this simulation, the inlet ducts were pre-defined in the imported geometry. Consequently, the top horizontal faces of these ducts were selected as the inlet. The areas have a diameter of 5 mm, resulting in a total inlet surface area of $1.57 \times 10^{-4} \text{ m}^2$.

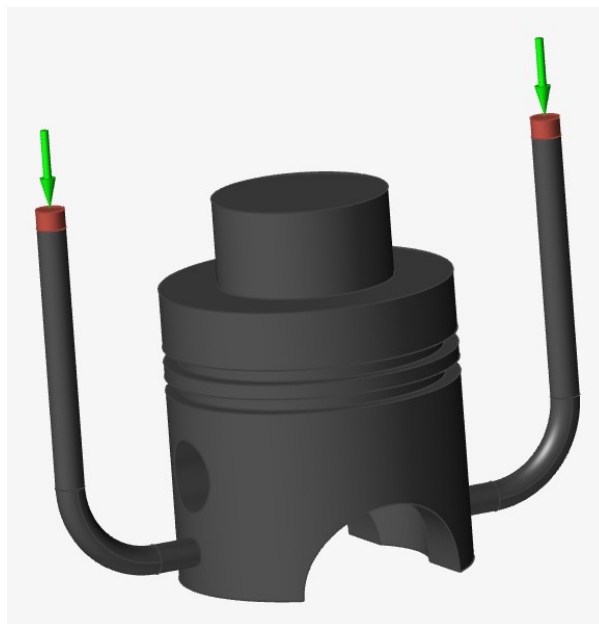


Figure 18: Inlet position in the Inspirecast model

The next step involves defining the mold, which is an automated feature in InspireCast. By selecting the corresponding command, the software automatically generates a mold that encapsulates the part, typically taking on a generic parallelepiped shape.

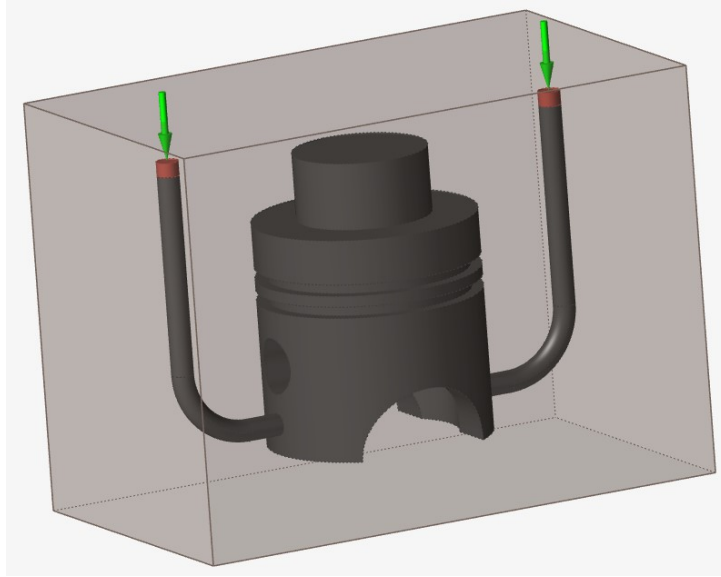


Figure 19: Mold generated automatically by Inspirecast

The final mandatory step to complete the model is defining the casting process, with options including general process, gravity, high pressure, and low pressure. In this case, the general process was chosen, providing the flexibility to set parameters such as inlet velocity or filling time. Specifically, an inlet velocity of 2×10^3 mm/s was selected to align with the same boundary condition applicable in Abaqus.

Before launching the simulation, the user must specify whether to simulate only filling, solidification, or both. Additionally, the mean thickness and the temperature at which the mold will be opened are required inputs. InspireCast automatically selects these parameters; for this work, the mean thickness is 11.72 mm, and the opening mold temperature is set at 673.15 K (400 °C). Upon launching the simulation, InspireCast automatically generates the mesh and proceeds with computing output parameters, including temperature, velocity, solid fraction, mold temperature, porosity, and air entrapment for filling, cooling, and demolding phases.

8.2 Abaqus

In this study, the Abaqus software, a Finite Element Analysis (FEA) tool developed by Dassault Systèmes, is employed for simulating the casting process. Abaqus is renowned for its capability to compute various parameters such as stress, strain, displacement, velocity, and temperature. Specifically, it is used here to simulate a casting process involving the solution of thermal and mechanical equations. The model is created through a series of modules, each allowing the definition of specific aspects of the modeling process:

Part Module: Allows users to import a CAD model or utilize Abaqus CAD capabilities to define part geometry.

Property Module: Enables the definition of materials and their properties, assigning materials to sections specifying the type of parts (solid, shell, etc.) and their characteristics (homogeneous, Eulerian, composite, etc.).

Assembly Module: Involves positioning parts together in a coordinate system and performs Boolean operation on them.

Step Module: Allows the creation of simulation steps, offering various possibilities such as static, dynamic, thermal, electric, implicit and explicit simulations, with the inclusion of field and history output requests.

Interaction Module: Focuses on creating and assigning interactions, constraints, connectors, and fasteners between parts.

Load Module: Allows users to define loads, boundary conditions, and fields.

Mesh Module: Involves defining the mesh, selecting the type and shape of elements, and subsequently meshing the parts.

Optimization Module: Provides the capability to create an optimization process by setting constraints or objectives within the simulation.

Job Module: After defining all model parameters, this module is used to create and manage the job for running the simulation.

Visualization Module: Allows users to visualize the output of the job requested in the step module.

Sketch Module: Provides a platform for drawing sketches.

In the upcoming sections, these modules will be explored in detail, elucidating the step-by-step procedure for creating an Eulerian aluminum casting model in Abaqus.

8.2.1 Part

The initial step in creating a model is to define the geometry. In Abaqus, this is accomplished within the "Part" module. The part list for this model includes inlet ducts, a riser, and a piston head. These parts are individually created and subsequently assembled in the assembly module.

All components are designated as 3D Eulerian in the "Create Part" menu. For the inlet ducts, the swept solid function is employed, specifying both the sweep path and section. The piston head exhibits a more intricate shape, requiring a multi-step creation process. Initially, a cylinder is formed using the "Extrude Solid" command (radius=0.05m, height=0.1m). Subsequently, two circular crowns are generated as o-ring grooves (radius: max=0.055m, min=0.045m, height=0.005m). Another smaller cylinder defines the internal empty volume of the piston head (radius=0.035m, height=0.08m). The riser is a cylinder (radius=0.03m, height=0.13m). Lastly, two small cylinders create the side holes – the first generating two circular holes in the side wall (radius=0.0125m), and the second forming two semicircular holes at the bottom of the piston head (radius=0.03m).

It is noticeable the great dimensions of the riser, that derive from studies performed in previous models and will be explained in the result section.

8.2.2 Properties

The subsequent step in model creation involves defining the materials utilized in the simulation and assigning them to the respective volumes. In Abaqus, material property definition is carried out in the "Properties" module within the "Create Material"

menu. This entails assigning a name and specifying material behaviors. To allow material assignment in subsequent modules, it is necessary to create a section. The section is established by indicating the type of part to which the material will be assigned, in this case, Eulerian. Subsequently, the materials created earlier are selected for assignment. The initial task is to create the materials: aluminum and air. All properties must adhere to the same unit system; in this work, the International System of Units (SI) is employed. The material properties for air are considered constant, as their variation does not significantly impact the results given the other assumptions made for aluminum properties. For air, the chosen Equation of State (EOS) is a linear $U_s - U_p$ Hugoniot equation. This choice addresses issues created by the ideal gas equation during the evolution of the air flow.

Air		
density	1.2	Kg/m ³
sound speed	343	m/s
specific heat	1000	J/Kg K
conductivity	0.026	W/m K
dynamic viscosity	1×10^{-5}	Kg/ m s

Table 1: Air mechanical and thermal properties

The primary assumption that introduces a degree of error into the simulation is the use of a constant value for density due to Abaqus limitation in the Eulerian formulation of the problem. This limitation prevents the accurate assessment of the shrinkage phenomenon during the phase change and solidification. The density of aluminum varies in the range from 2500 kg/m³ at around 1000 K (726.85 °C) to 2800 kg/m³ at 300 K (26.85 °C). A value of 2700 kg/m³ is chosen, as for all temperatures below 800 K (526.85 K), the density exceeds the selected value (refer to the graph in Figure 10).

Aluminium		
density	2700	Kg/m ³
latent heat	3.96×10^4	J/Kg
speed of sound	1500	m/s

Table 2: Aluminium mechanical and thermal properties

During the casting process, aluminum undergoes a phase change from liquid to solid, which needs to be implemented in Abaqus material parameters. The Abaqus Explicit Eulerian simulation only allows for one definition of material behavior, preventing the insertion of two equations – one for the liquid phase and one for the solid phase. The filling process requires a fluid material, so a linear $U_s - U_p$ Hugoniot equation is selected in the Equation of State (EOS) menu. This choice aligns with Abaqus guide information on using this formulation to model incompressible viscous flow governed by the Navier-Stokes equation of motion.

To model the transition from liquid to solid, the assumption is made that a highly viscous material will act as a solid if the simulation time is short enough. Viscosity can be defined as a function of the Solid Fraction parameter [54]. Generally, after the material reaches a Solid Fraction value above 0.7, it can be considered solid, as the forming structure creates constraints to the fluid aluminum flow, which is now less than 30% of the volume. This assumption dictates that the viscosity has a steep curve, increasing rapidly with decreasing temperature in a range from 920 K (646.85 °C) to 870 K (596.85 °C). The value of dynamic viscosity grows by five orders of magnitude from the initial value for fluid aluminum, 1.2×10^{-3} kg/(m s), to a value of 100 kg/(m s). After this significant variation, dynamic viscosity continues to increase until reaching a value of 5000 kg/(m s) for a temperature of 800 K (526.85 °C) and remains constant for temperatures below this threshold.

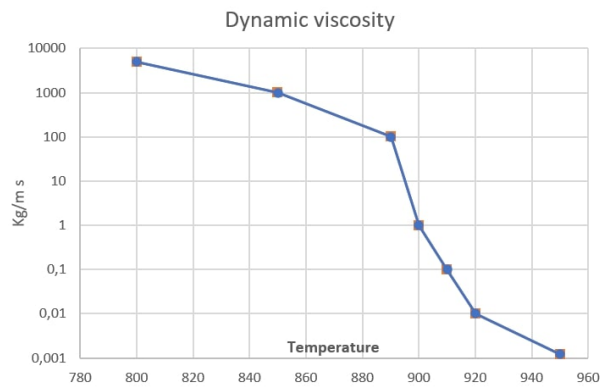


Figure 20: aluminum viscosity

To determine an appropriate viscosity value for modeling the solid phase, several tests were conducted by varying the viscosity and observing the material behavior. To expedite run time, a simple geometry was employed, a cube of aluminum that impact a planar surface under the influence of gravity. Four viscosity values were tested, and the results after $t=0.85$ s revealed significant deformation for viscosity up to 100 kg/(m s), which is already 10^5 times greater than the viscosity of fluid aluminum. At 1000 kg/(m s), the material began to exhibit solid behavior but still showed notable deformations. Ultimately, the model with a viscosity value of 5000 kg/(m s) demonstrated a fully solid behavior (figure 21) and was subsequently chosen as the viscosity for the solid aluminum in the casting model.

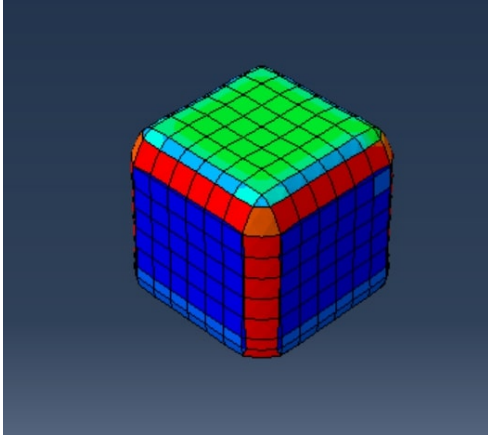


Figure 21: aluminum viscosity equal 5000 kg/(m s)

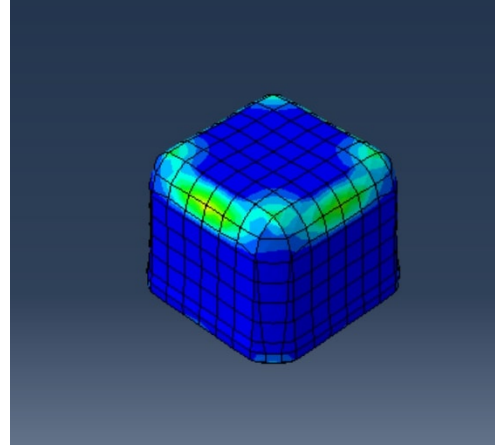


Figure 22: aluminum viscosity equal 1000 kg/(m s)

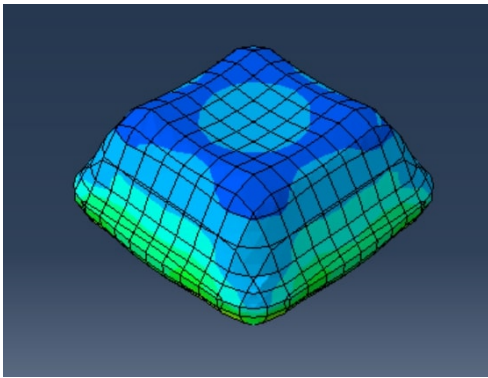


Figure 23: aluminum viscosity equal 100 kg/(m s)

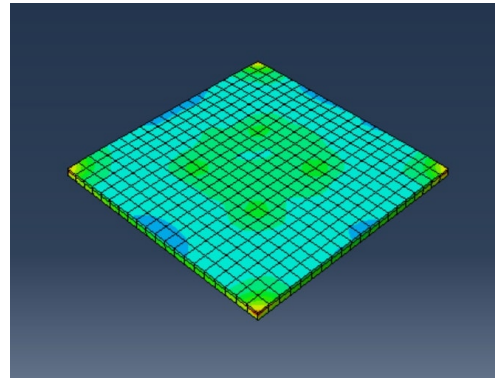


Figure 24: aluminum viscosity equal 10 kg/(m s)

Aluminum conductivity and specific heat data are taken from Inspirecast A7075 material properties.

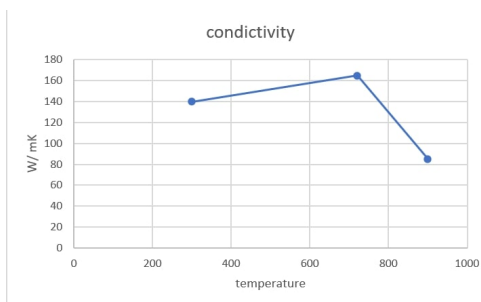


Figure 25: aluminum conductivity

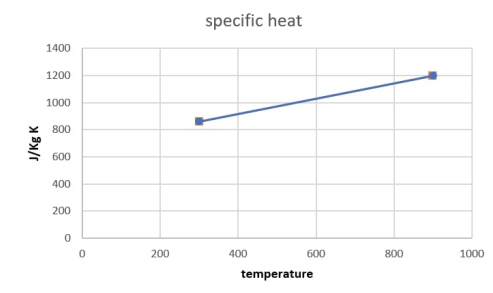


Figure 26: aluminum specific heat

After defining the material properties, an Eulerian section needs to be defined to assign materials to the part. The section is created with the create section command by selecting the solid category and Eulerian type. Subsequently, the two materials, air and aluminum, are assigned to the section.

8.2.3 Assembly

In the assembly module, the previously created parts can be assembled to form complex geometries. All the parts described before, including the inlet ducts, riser, and piston head, are joined together. The piston head is created by performing boolean operations on simpler parts, including an external cylinder, internal cylinder, two circular crowns, a cylinder for the circular holes, and another cylinder for the semicircular holes. Initially, these parts are added to the same space, positioned relative to each other, and then the boolean operations are performed.

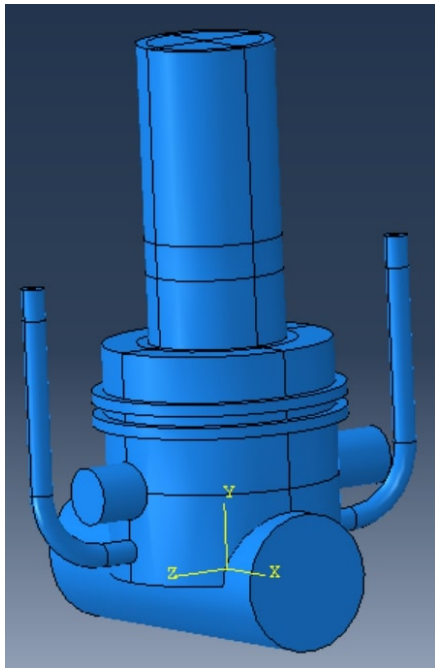


Figure 27: Assembly before boolean operations

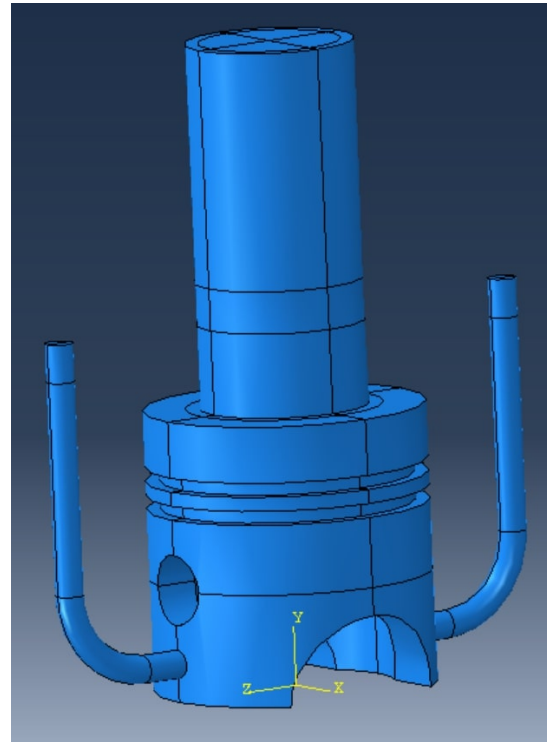


Figure 28: Final assembly

8.2.4 Step

In the Step module, the user specifies the type of analysis to be performed. Abaqus is capable of performing Eulerian analysis only with explicit schemes. For this model, which includes a thermal problem, a Dynamic, Temp-disp, Explicit analysis is chosen. The explicit analysis, although slower than implicit ones due to instability issues in the scheme, is necessary for solving the Eulerian formulation. The casting simulation involves both displacement and temperature degrees of freedom, so Temp-disp analysis is the correct choice to compute the evolution of all degrees of freedom.

Two steps are defined to allow a better definition of Boundary Conditions in the next stages of model definition. Both steps follow the Dynamic, Temp-disp, Explicit procedure but have different durations. The first step is created to simulate the filling phase with a total time of 1.6 s, a duration derived from the Inspirecast simulation. The second step represents the solidification phase and lasts for 4 s. This time is chosen to allow the

solidification of the aluminum but does not represent the entire cooling phase, as FEM simulations have substantially longer run times compared to Inspirecast. The output request is set at 0.01 s, serving as a compromise between result accuracy and memory usage.

8.2.5 Interaction

In the Interaction module, the user can define how different parts interact. General contact is the only option available for models that include Eulerian parts. For purely Eulerian models, interactions are not explicitly defined because Abaqus automatically tracks material surfaces inside the volumes and computes their interactions and energy exchange. In models that include Lagrangian bodies, such as CEL models, interaction properties must be defined to allow Abaqus to compute the exchange of energy between Lagrangian and Eulerian bodies. For Eulerian models, Abaqus, after the initial definition of the material field inside the Eulerian volume, automatically tracks material's external surfaces.

Material position can be defined as a discrete field in this module by creating a part that represents the initial volume of material and its position at $t = 0$ s. Abaqus then computes the volume fraction for each Eulerian element partially or totally overlapped by the part (for more information, refer to Section 7.3). This action needs to be performed for every material involved in the model, as Abaqus has the capability to take into account multiple materials, such as air and aluminum in this model. For simpler geometries or geometries that are partitioned in a manner where a single or multiple partitions are entirely filled with a single material, the material assignment can be made as a uniform field in the load module, as performed in this work.

8.2.6 Load

In this module the user could define loads, boundary conditions and fields.

- **Loads**

Loads can only be added in the user-defined step and not in the initial one already present in Abaqus. In this work, only gravity load is defined with a value of -9.8 m/s^2 along the Y-axis, applying to the entire model for every step of the simulation.

- **Boundary Conditions**

Boundary conditions are essential as the absence of the mold needs to be substituted by appropriate mechanical and thermal boundary conditions. The first condition to define is the non-penetrability of the mold by the material. On all the Eulerian external surfaces where the mold is supposed to be, an Eulerian boundary condition of "inflow=none" and "outflow=none" is imposed (see Figure 31). In addition, the circular crown on the top surface is closed to resolve numerical errors that arise at sharp corners. The remaining surfaces are the two inlets and the riser (outlet). For the inlets, the imposed conditions are "inflow=free" and "outflow=equilibrium" for the filling step and "inflow=none" and "outflow=none" for the cooling step to stop

the aluminum from continuing to flow inside the domain. For the riser, the Eulerian boundary conditions for both steps are "inflow=free" and "outflow=equilibrium" as the air can continue to flow inside the domain for the entire duration of the simulation. With Eulerian boundary conditions, Abaqus computes the material that enters or exits the domain using information about normal velocity to the boundary and the material on the elements at the boundary. This means that if the material has a negative normal velocity at an open boundary, the same material continues to flow inside the domain, as it happens at the inlet. For a positive normal velocity, the material exits the domain, and the "equilibrium" condition selected allows the material to flow outside the volume while maintaining a zero-order continuous stress across the surface by imposing artificial traction forces [53].

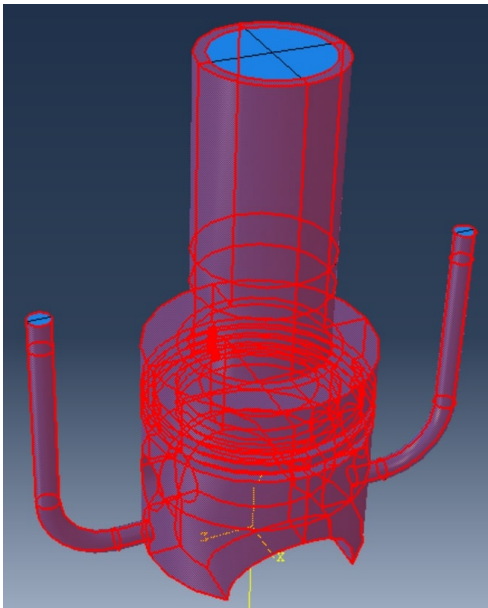


Figure 29: BC with no inflow or outflow

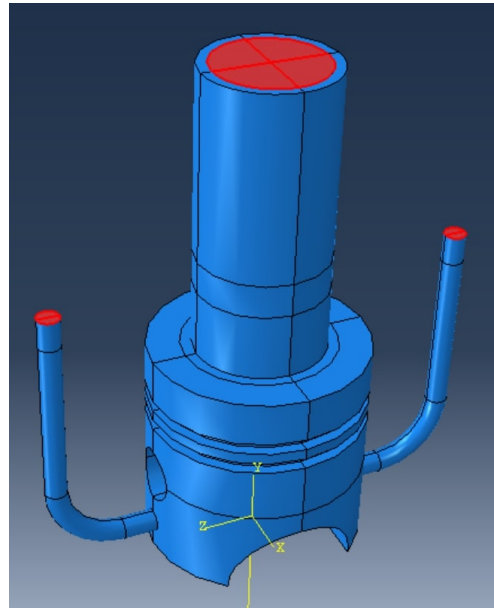


Figure 30: Inlet BC

On the inlet surfaces, a boundary condition is also imposed to define the inlet velocity at -2 m/s along the Y-axis. This is only imposed for the filling step and is modified to a value of 0 m/s for the cooling step to stop the inflow of aluminum inside the domain.

To simulate the boundary layer on all external surfaces (representing the mold surface), the material is considered to stick to the surface. The velocity magnitude along the three axes is defined as $v_1 = v_2 = v_3 = 0$ from the initial step for all the steps.

To impose boundary conditions for the thermal problem, some assumptions have to be made. Firstly, there is a need to consider Abaqus capabilities and how the mold acts during the casting process. Abaqus allows the definition of surface heat flux (Neumann BC) as a load or the imposition of the temperature at the surface (Dirichlet BC). The former could be computed by defining all the heat exchanged by the aluminum during its cooling, given the aluminum's thermal properties and the temperature difference. However, applying this load becomes challenging as Abaqus does not distinguish between air and aluminum in the domain. This leads

to subtracting heat from the Eulerian domain based on the load defined, causing the air to cool below 0°C. This issue might be resolved by user-defined load distribution using a subroutine that computes the heat flux based on the material present in the element at the surface. Another challenge is the need to know the mold temperature evolution at each point, as the heat flux depends on the temperature difference between aluminum and the mold.

For this work after several tests and studies, the boundary condition that defines the surface temperature is chosen, assuming that in a simulation of first approximation, the temperature of the mold could be considered as fixed if the mold has some surface treatment and has a significant heat capacitance. The temperature is set for all surfaces except inlets and outlets at 500 K (226.85 °C).

- **Fields**

Material assignment is made through the field manager as a uniform field. For this model, the initial volume of aluminum is a small length of the inlet canals to ensure that all the elements forming the inlet surface are full of aluminum. This ensures that during the simulation, no air or void will enter, as explained in the Boundary Condition paragraph. As that volume is full of aluminum, the value of the volume fraction is set to 1, automatically imposing the volume fraction of air and void to be 0. The process is repeated for the air volume by adding a row to the predefined field creation, selecting the rest of the Eulerian domain, and again assigning the volume fraction of air to be 1, while the volume fraction of aluminum and void is imposed to be 0. The material called "void" is a fictitious material with no properties that Abaqus uses to fill the elements that are not 100% full of materials, enabling the solution of the equations.

The other fields to be defined are the temperature ones for aluminum and air. The aluminum is set at $t = 0$ s with a temperature of 1000 K (726.85 °C), and the air at 310 K (36.85 °C).

8.2.7 Mesh

In this module, the mesh is created. Since the Eulerian volume has a complex geometry, Abaqus cannot mesh it directly. There is a need to divide the volume into simpler shapes that the software can mesh automatically; this action is called partitioning. The Eulerian simulation in Abaqus supports only hex element types from the explicit library, and the selection of the thermally coupled option activates the temperature as a degree of freedom. The element type is "EC3D8RT: An 8-node thermally coupled linear Eulerian brick, reduced integration, hourglass control" (ref. Abaqus elements library).

To create the mesh, a global seed is defined at 0.0028 m, generating a mesh of 47982 elements. The mesh is generated following different rules; the green zones are structured zones, while the yellow ones follow a sweep rule with advancing front.

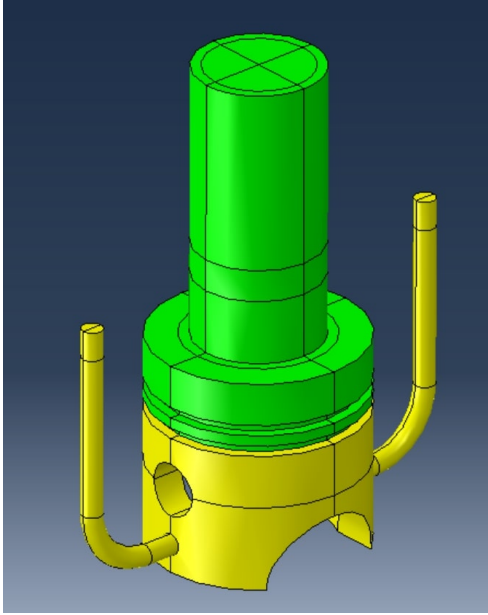


Figure 31: Eulerian domain partitions

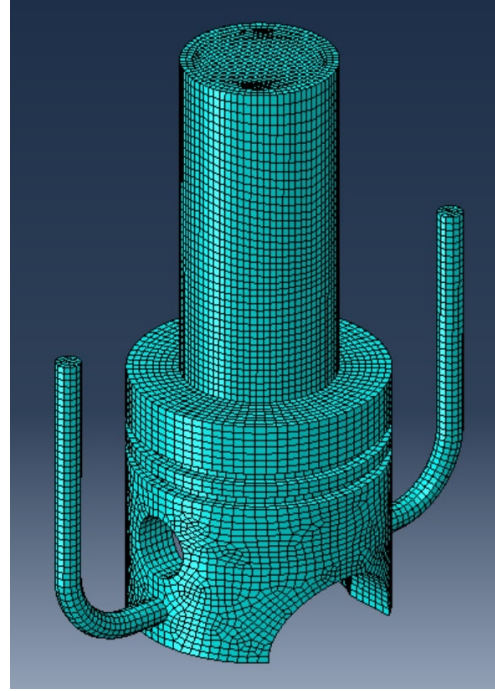


Figure 32: Eulerian domain mesh

8.2.8 Job

Here, with the job manager, the simulation is created and launched. You can choose the number of domains in which the simulation is solved, also known as parallelization, and how many processors are going to be used to solve it. Abaqus then divides the elements and nodes to evenly split them between the domains. When the job is submitted, its status can be tracked in the monitor window, which displays the increment, time, stable increment, total energy, and kinetic energy.

8.2.9 Visualization

In this module, the results of the job can be analyzed by selecting the different outputs requested in the step module. Additionally, single elements can be queried to analyze output values with more precision. It is also possible to create plots of various data.

9 Results

9.1 Inspirecast

The previously developed model in Inspirecast (refer to paragraph 8.1.1) is executed on a PC with the following specifications: Intel Core i7-1360P 2.2GHz processor and 16GB of RAM. The program's efficiency is evident in the time required to complete the simulation on this non-workstation machine with limited processing power. The runtime

for this model, characterized by a simple geometry, is 3:58 minutes. This showcases the software's effectiveness during the design phase, providing a swift evaluation of the manufacturing process without the need for iterative design and feasibility loops.

The simulation is segmented into three distinct steps: filling, cooling, and demolding. According to Inspirecast results, the aluminum successfully fills the mold in 1.75 seconds, followed by a cooling phase lasting 24.80 seconds. Subsequently, the demolding phase and final cooling take an additional 234.82 seconds.

9.1.1 Filling Phase

In the filling phase, the molten aluminum is introduced through the specified inlet, rapidly occupying the entire chamber volume within a brief period of 1.79 seconds. The inlet velocity is defined at -2000 mm/s along the Y axis. Due to the influence of gravity, acting along the Y axis with an acceleration of -9800 mm/s^2 , the aluminum flows downward through the inlet duct. It achieves a peak velocity magnitude of approximately 3300 mm/s at the midpoint of the duct, precisely at the intersection with the piston head (highlighted in red in the subsequent images).

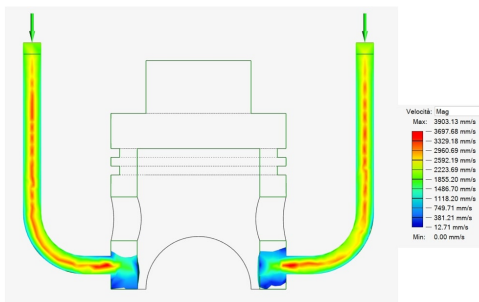


Figure 33: velocity results $t_1 = 0.1s$

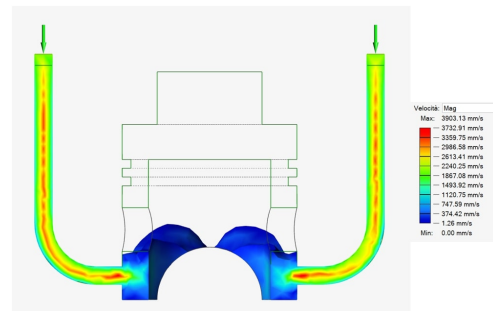


Figure 34: velocity results $t_1 = 0.3s$

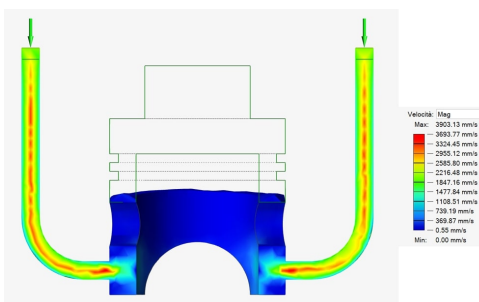


Figure 35: velocity results $t_1 = 0.7s$

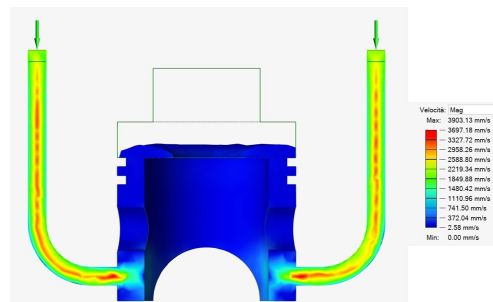


Figure 36: velocity results $t_1 = 1.1s$

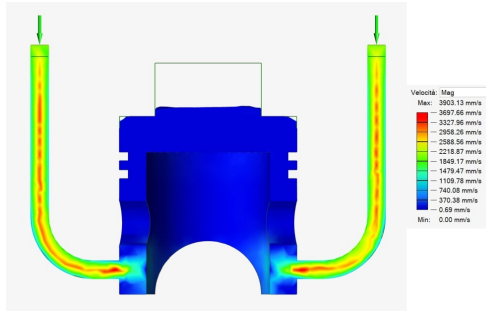


Figure 37: velocity results $t_1 = 1.55s$

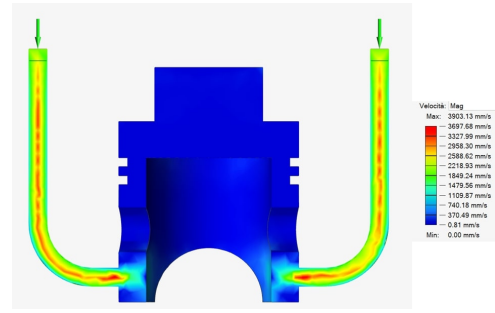


Figure 38: velocity results $t_1 = 1.79s$

The velocity magnitude at selected points is plotted below to illustrate its evolution during the filling phase. It is important to note that Inspirecast does not compute the velocity of the air present in the mold; the plots begin when the aluminum reaches that specific zone of the mold. Inspirecast only calculates the pressure for the air.

Point 1 (End of Inlet Duct) experiences a high-speed flow resulting from the conversion of potential energy into velocity during the descent of the aluminum, which starts at a velocity of 2000 mm/s.

The flow at point 2 (same height as Point 5) is much slower as it is already within the piston head, where the aluminum flow begins to spread to fill the entire volume.

Points 3 and 6 exhibit smaller velocities due to a constant mass flow at the inlet but an increasing area in the piston head compared to the inlet. According to the conservation of mass flow equation ($\rho v A = \text{const.}$), an increase in area results in a decrease in velocity. At point 4 following the same principle, as the area decreases, the velocity increases.

All the plots exhibit an initial peak, indicating the arrival of aluminum flow in that particular region. The temporal delay in the peaks corresponds to the distance from the inlet, as illustrated by the graph.

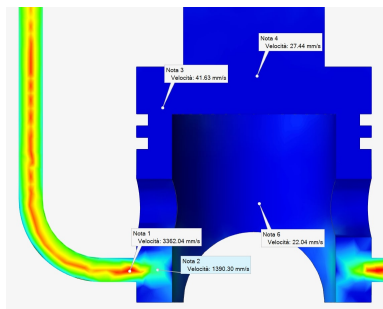


Figure 39: Points for which the velocity is plotted

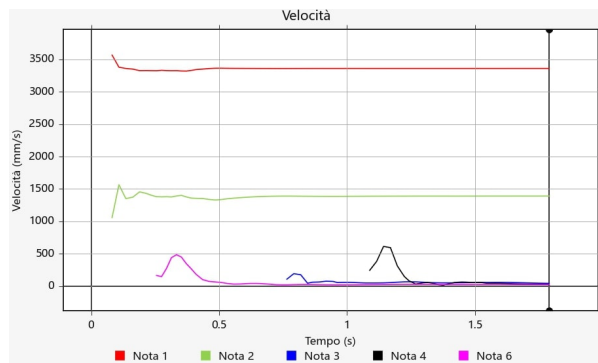


Figure 40: plot of velocity magnitude [mm/s] for the point highlighted in figure 39

The inlet temperature of the aluminum is set at 1000 K (726.85°C). Points 1 and 2, closer to the inlet, exhibit higher temperatures throughout the filling time as new aluminum at 1000 K continuously reaches them from the inlet. When the other three points are reached by the aluminum, the temperature decreases over time as the aluminum exchanges heat with air and the mold. The farther a point is from the inlet, the more

the aluminum temperature decreases. Specifically, point 6 is reached at $t = 0.4$ s, point 3 at $t = 1$ s, and point 8 at $t = 1.42$ s. As the graph shows, point 4 sees the aluminum starting from a temperature of 899 K.

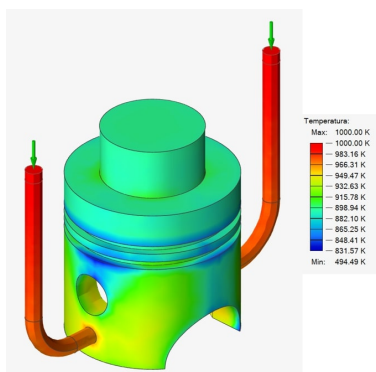


Figure 41: Temperature field a $t_1 = 1.79$ s

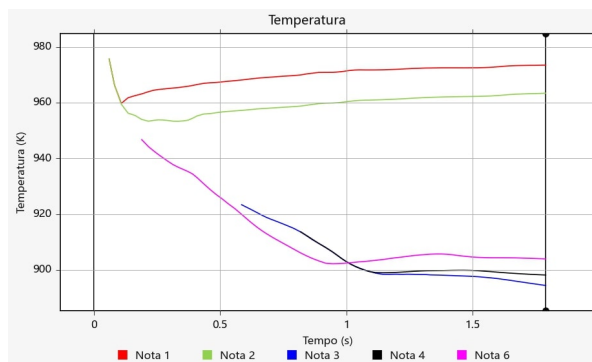


Figure 42: plot of temperature for the point highlighted in figure 39

The points closer to the edges are colder due to the higher contact area with the mold, resulting in a higher heat flux and faster temperature decrease. Points such as those at the bottom corner of the piston head lateral surfaces (highlighted in blue in the image) exhibit a temperature of 844 K at the end of the filling phase. Other colder regions include areas where the aluminum follows a longer trajectory, such as the upper edges of the lateral holes over the ducts, with a temperature of 842 K. These two points are nearly 150 K colder than the material closer to the inlet (see Figure 43).

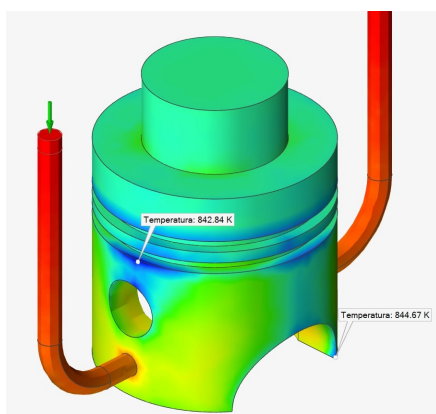


Figure 43: Colder points of piston head at the end of filling phase $t_1 = 1.79$ s

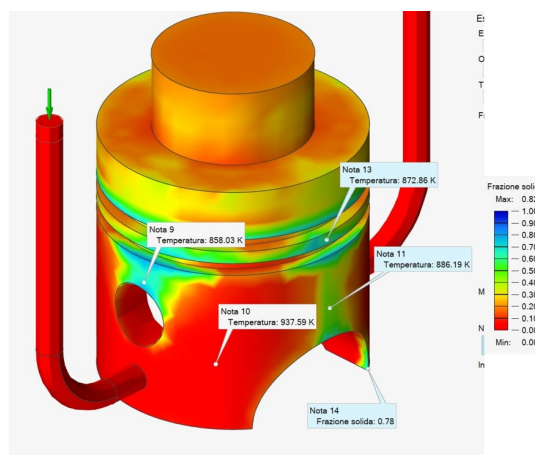


Figure 44: Solid Fraction at the end of the filling phase $t_1 = 1.79$ s

These colder zones of material start to solidify during the filling phase. At time $t_1 = 1.79$ s, some regions (highlighted in blue) are completely solid, indicated by the parameter called Solid Fraction (SF) = 1, meaning that the elements of the mesh are 100% full of solid material. Solid Fraction is directly related to the temperature; in the material definition, the Solidus temperature is specified. In the temperature range below

T_{solidus} , the material is completely solid. Between T_{solidus} and T_{liquidus} , the material is in a mixed phase with some parts solidified and some parts still liquid, resulting in a Solid Fraction with a value between 0 and 1. Finally, above T_{liquidus} , all the material is in liquid form.

The corner point of the piston lateral surface is the first point to solidify; after around 0.6s, its SF is above 0.7, indicating that 70% of the material in the element containing that point is below T_{solidus} . The other points have a similar history, while point 10, the one closer to the inlet duct, remains nearly completely liquid after the filling phase.

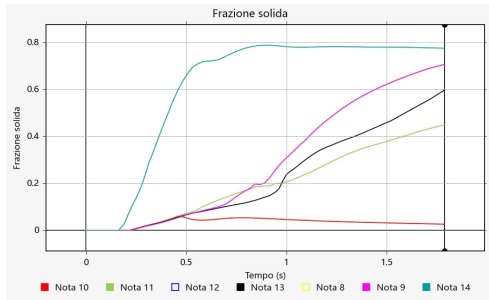


Figure 45: Solid Fraction plot for points in figure 44

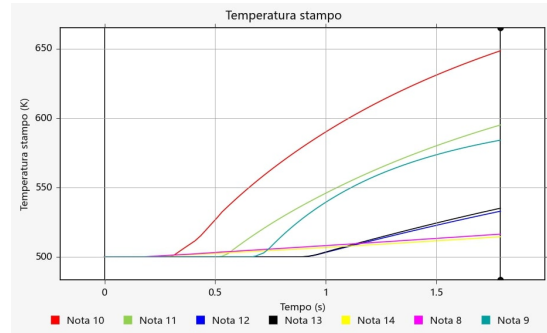


Figure 46: Mold temperature plot for points in figure 44

The two inlet ducts maintain a temperature of 1000 K throughout the entire filling time as new aluminum is continuously poured inside. The volumes of the piston head attached to the ducts exhibit a temperature gradient pointing toward the ducts. The temperatures show a slightly positive trend during the filling phase, primarily influenced by the heat exchange occurring between the material already present in the zone and the incoming, hotter aluminum entering the cavity.

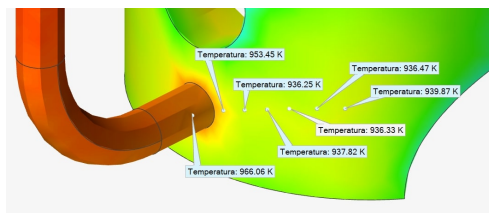


Figure 47: Temperature points to compute gradient plot

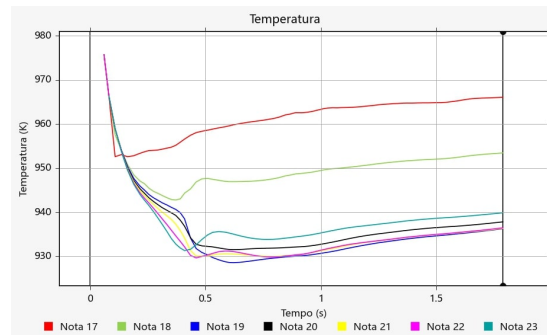


Figure 48: plot of temperature for the point highlighted in figure 48

9.1.2 Solidification Phase

During this phase, Inspirecast computes temperature, mold temperature, solid fraction, and solidification time. With no more aluminum flowing from the inlets, the fluid material

is now considered static. This phase spans a duration of 40.55 s.

Below is the temperature evolution of the part, with a consistent legend for all plots:

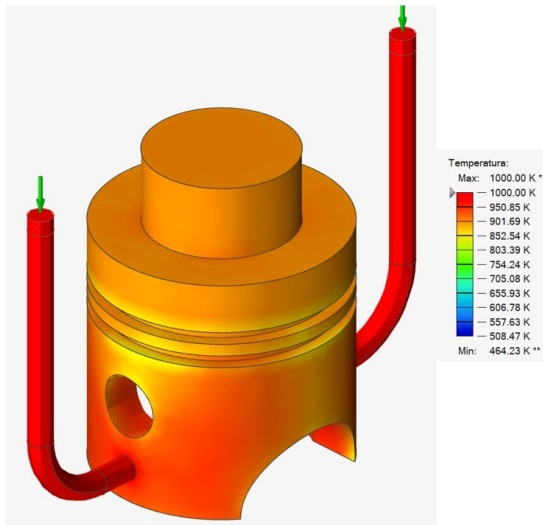


Figure 49: Temperature fields at $t_2 = 0s$ of the solidification phase

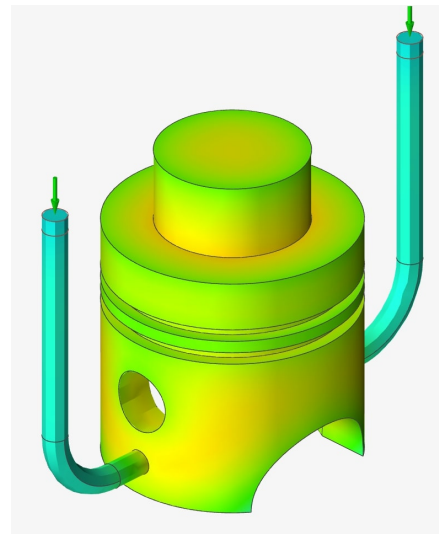


Figure 50: Temperature fields at $t_2 = 5.5s$ of the solidification phase

As previously explained, the corners and edges cool more rapidly due to increased heat dissipation to the mold. Additionally, the inlet ducts, characterized by a high surface-to-volume ratio, tend to cool faster. This is because mass is proportional to volume, and the heat flux exchanged with the mold is proportional to surface area. Consequently, the inlet ducts commence the solidification phase as the hotter regions of the part and conclude as the cooler regions. Refer to the graph in Figure 52.

The top of the piston head, being thicker (20 mm) have greater heat capacitance, remains the hottest part. The volume of the mold surrounding it is confined by the sides of the piston head, so it absorbs more heat compared to other zones of the mold cooling slowly the part as the temperature difference is smaller. [curve per un tratto orizzontali per il calore latente](#)

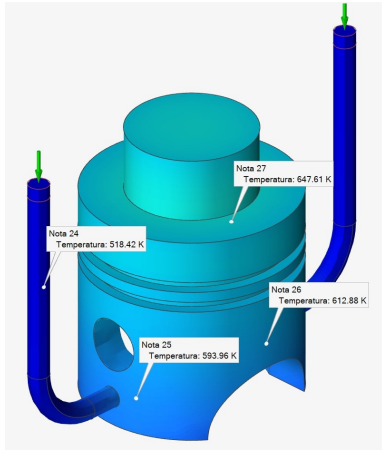


Figure 51: Temperature fields at $t_2 = 40.55s$ of the solidification phase

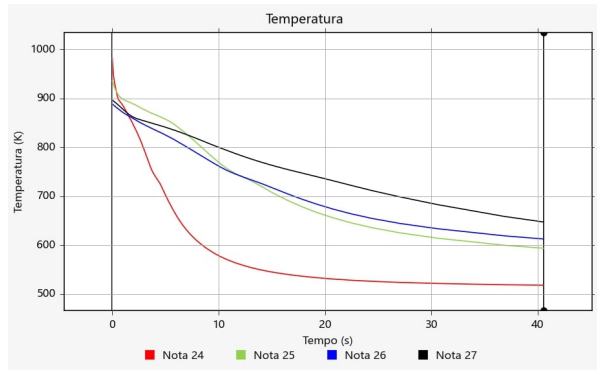


Figure 52: Temperatures plot during solidification phase relative to the points in figure 51

Internal points exhibit the same pattern, points farther from the inlet, like points 35 and 36, tend to have a lower initial temperature. However, these points are situated in a thicker region, resulting in a flatter curve during the initial part of the cooling phase. The increased thickness allows them to retain more heat due to the proximity to a greater volume of hot material. Conversely, points in proximity to or within the ducts, exemplified by points 40 and 41, showcase a higher initial temperature but display a remarkably steep cooling curve, particularly for point 40 located inside the duct.

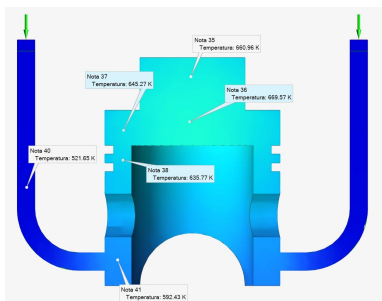


Figure 53: Temperature fields at $t_2 = 40.55s$ of the solidification phase

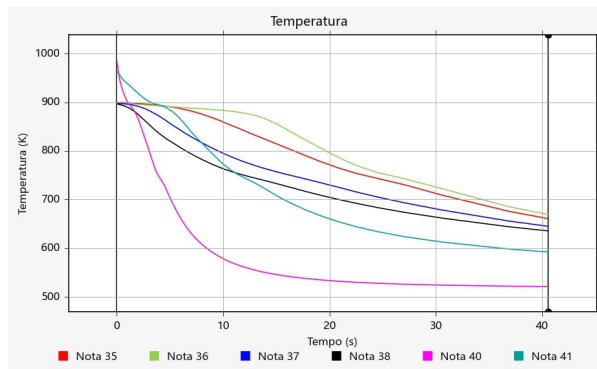


Figure 54: Temperatures plot during solidification phase relative to the points in figure 117

The evolution of mold temperature initiates at a base temperature of 500 K (226.85 °C) at the beginning of the filling phase. As the solidification phase unfolds, the heat transferred from the aluminum to the mold contributes to a rise in mold temperature. In the initial stages of this phase, the warmer regions are in proximity to the intersection of the inlet ducts and the lateral surface of the piston head (depicted in red zones in Figure 55, point 25 in Figure 51). After 5.5 seconds, the ducts exhibit a significant decrease in temperature, hovering just above 585 K (illustrated in light blue zones in Figure 56 and the plot of point 24). At this timestep, the top of the piston head and the edges intersecting with the riser are the warmer zones at around 780 K.

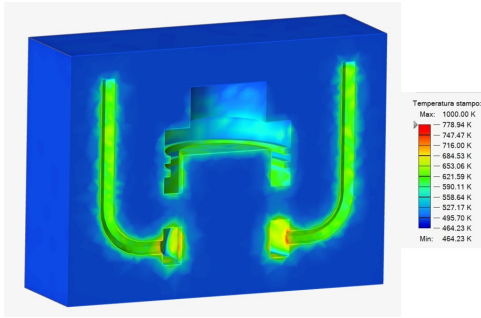


Figure 55: Mold temperature fields at $t_2 = 0.0s$ of the solidification phase

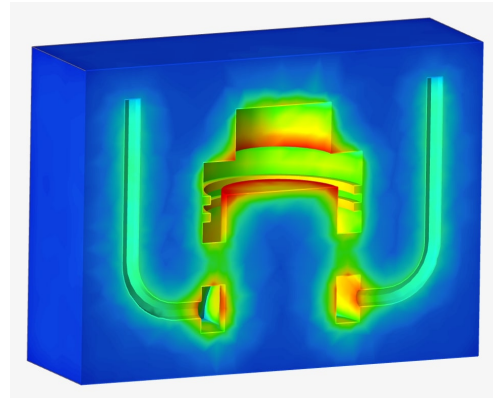


Figure 56: Mold temperature fields at $t_2 = 5.5s$ of the solidification phase

Figure 57 illustrates the heat distribution within the mold, reaching even the farthest zones from the cast part. The hottest regions are found in proximity to the top of the piston head, where a substantial amount of aluminum is present, releasing a significant amount of heat (refer to the plots of points 27 on the top surface and 29 on the inside top surface of the piston head in Figure 58).

The temperature profile of the mold, as depicted in the plots of Figure 58, exhibits a sharp increase initially, as molten aluminum, at 1000 K, is poured into a mold at 500 K. This substantial temperature difference results in a high heat flux. Subsequently, as the material begins to exchange heat with the mold and dissipate heat from the system, the metal undergoes a cooling process.

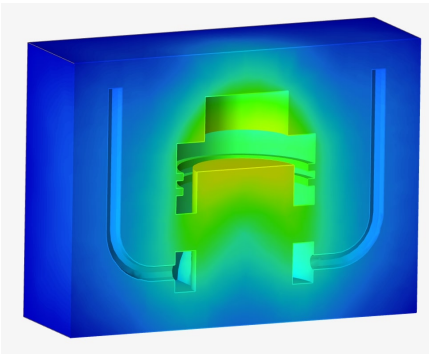


Figure 57: Mold temperature fields at $t_2 = 40.55s$ of the solidification phase

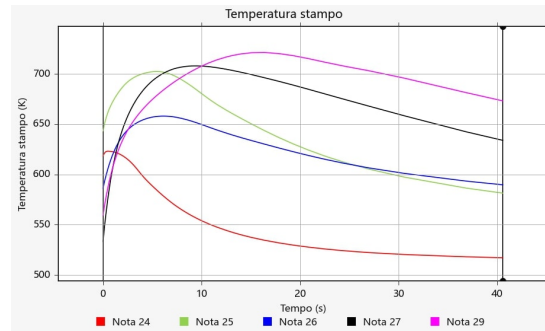


Figure 58: Mold temperature plot of points in figure 51 during the solidification phase

The temperature field of the mold and cast part, at the end of the solidification phase, highlights a distinctive pattern. Specifically, in the upper section of the piston head, where the thickness increases and a riser is present, a significant volume of hot aluminum retains higher temperatures throughout the solidification process compared to other areas of the part also because of the major distance from the external mold surfaces. This phenomenon is observable in both the cast part and the mold, indicated by a prominent red zone with a temperature reading of 673 K.

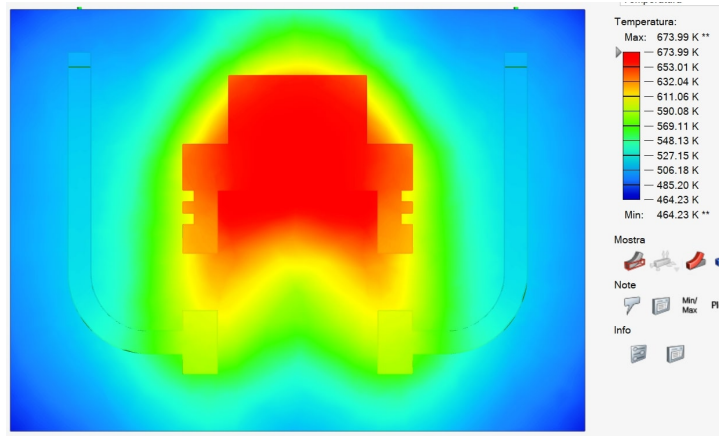


Figure 59: Temperature field in mold and cast part at the end of the solidification phase $t_2 = 40.5 s$

The trend of the solid fraction is the inverse of the temperature trend. Over time, the solid fraction increases to gradually approach 1 as the material undergoes cooling. As the temperature decreases, the solid fraction increases, reaching a plateau of 1 for the entire volume at $t_2 = 30.54 s$ when the maximum temperature of the aluminum is 723 K (449.85 °C). This temperature is below the solidus temperature of 728.15 K (455 °C), indicating complete solidification of all the aluminum.

Subsequently, the cooling phase continues until the entire volume reaches a temperature below 673.15 K, the specified mold opening temperature during the model's development. This threshold is achieved at time $t_2 = 40.55 s$. This precaution is taken to ensure that all the material is fully solidified before initiating the demolding process.

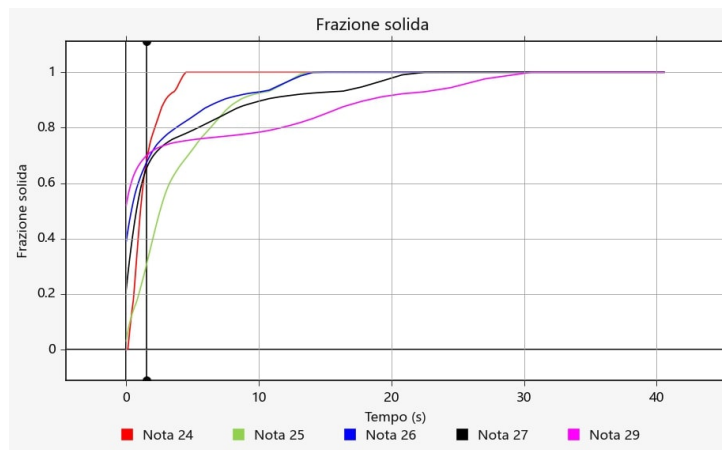


Figure 60: Plot of Solid Factor of points in figure ??

9.1.3 Demolding

This is the lengthiest phase of the simulation, lasting 261.9 s, and it models the cooling of the solidified part exposed to the open air after the mold is opened. In this step, Inspirecast calculates various potential defects in the final part, such as porosity, microporosity, shrinkage volume, and cold shuts, alongside temperature and solid fraction, as

in the preceding steps.

The temperature profiles of all points exhibit similar trends. They tend to approach thermal equilibrium with the air around the part and neighboring points, eventually reaching the same final temperature. This equilibrium is achieved through heat exchange. The hotter zones, represented by points 27 and 29, experience a rapid temperature decrease initially due to the higher temperature differential with adjacent materials (air and aluminum). In contrast, the slightly cooler zones, such as point 25, initially exhibit a positive temperature trend, receiving more heat from the hotter zones than they lose to the air. As the temperature differentials within the aluminum diminish, all zones cool down at a uniform rate. The behavior of the ducts (point 24) is similar, but slower, as they are surrounded only by air, and with a lower temperature differential.

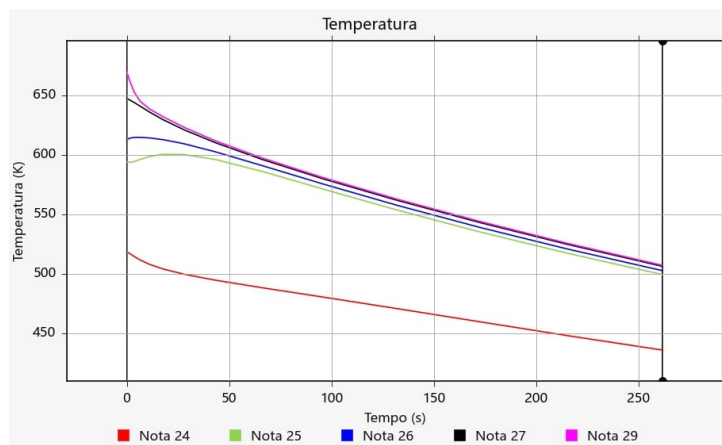


Figure 61: Plot of temperature of points in figure 51

Inspirecast also analyzes potential defects in the part, such as microporosity and porosity. Microporosity is evaluated using the Dimensionless Niyama criterion [56]. This criterion considers local thermal conditions, including the Niyama criterion, defined as the local thermal gradient divided by the square root of the local cooling rate. If this value falls below a certain threshold, dependent on material properties, microporosity is likely to form. The resulting microporosity distribution is presented below, providing a clearer visualization of the affected zones. Porosity, on the other hand, is the ratio of void volume to solid volume.

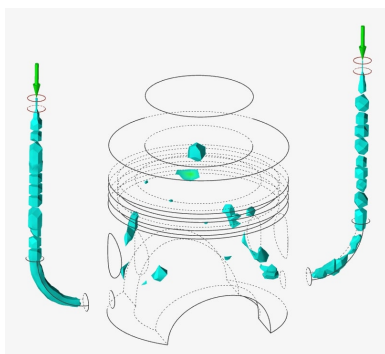


Figure 62: Microporosity of the final part

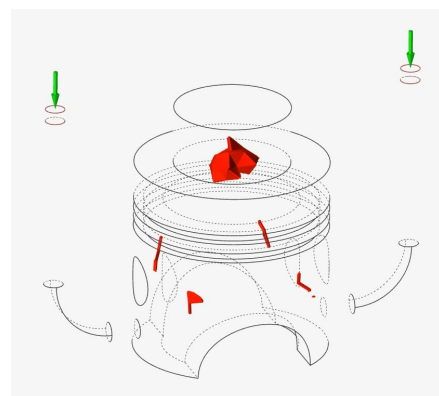


Figure 63: Porosity of the final part

Microporosity primarily forms in the inlet ducts due to the steep temperature decrease and the relatively small volume-to-area ratio compared to other zones. While this type of defect could be mitigated by increasing the diameter of the ducts, it is not a significant concern as it is confined to the ducts and does not affect the final part extensively. Another critical zone affected by microporosity is the volume below the riser. In the same area, porosity also develops for the same reason as microporosity: as the aluminum solidifies, it contracts, and if the riser fails to supply molten metal, a porous and highly stressed zone will form. In these regions, porosity reaches a value of 90%, indicating that the void-to-solid ratio is 0.9, implying that void and solid material occupy nearly the same volume in these areas.

9.2 CEL

To delve deeper into Abaqus' capabilities in modeling a casting process, a Coupled Eulerian Lagrangian (CEL) model was developed. This model is distinguished by its ability to accurately track the interaction between two entities: the solid mold (Lagrangian body) and the fluid aluminum (tracked in the Eulerian domain). In contrast to the Eulerian model, this approach enables a more precise analysis, particularly for the thermal problem. In this case, the mold is treated as a body with its material properties, eliminating the need for approximations through boundary conditions.

To allow the computation of interactions between the material in the Eulerian volume and the Lagrangian body, these two domains must overlap, at least partially. Abaqus requires the definition of interaction properties to compute the exchange of forces and energy between the two domains. However, incorporating these additional interactions significantly increases the simulation time, approximately by a factor of 10 with respect to an equivalent Eulerian model. Consequently, the geometry used for this simulation is simpler than that in the Eulerian model, deviating from the classical casting geometry.

The simplified geometry consists of a Lagrangian body, represented as a circular plate (the mold), and a circular cylinder for the Eulerian domain. The Lagrangian domain completely overlaps with the Eulerian one to ensure interaction. Tests conducted during model development revealed the necessity of eliminating all sharp edges and introducing a very fine mesh near the contact zone, especially in the Eulerian domain.

This fine mesh is crucial because in the portion of the Eulerian domain overlapped with the Lagrangian domain, Abaqus automatically inserts a fictitious material called "void." This material lacks mechanical or thermal properties, but the solver requires it to compute equations in the Eulerian domain. Without taking these measures, during the simulation, some of this void material could penetrate other parts of the Eulerian domain where the Lagrangian body is not present. The chosen geometry facilitates meshing the two parts with a finer mesh close to the interaction zone. This approach helps eliminate numerical problems in this part of the volume while ensuring the generation of regular-shaped elements.

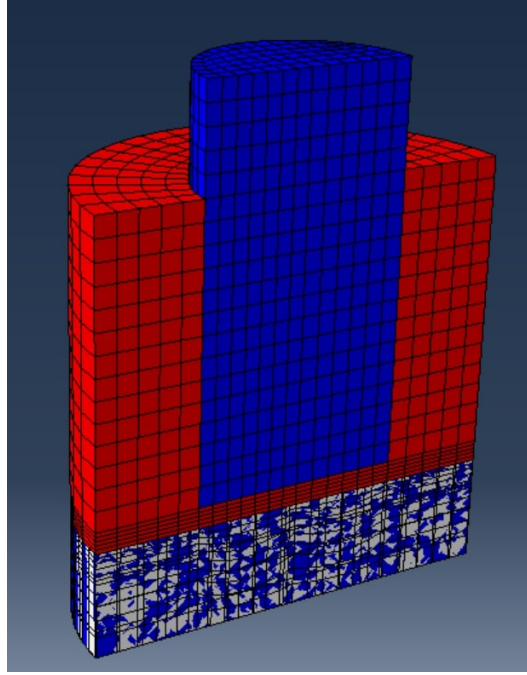


Figure 64: CEL model's geometry section. Aluminum (blue), air (red) and mold (grey)

In the figure above, the volume occupied by aluminum at the beginning of the simulation is highlighted in blue. To expedite the simulation time before contact, aluminum is initially placed in close proximity to the Lagrangian body. In red, the air is represented, and at the bottom, the gray volume denotes the Lagrangian body. As previously mentioned, the Eulerian domain in this zone is entirely composed of void material.

Material assignment in the load module for the initial step involves specifying a uniform field by selecting the cells occupied by aluminum and air and assigning the volume fraction value. All elements in the aluminum cells must start with a volume fraction value of 1, indicating complete material occupancy. The same principle applies to cells occupied by air. Abaqus automatically assigns fictitious void material to the remaining elements without material assignment or incomplete filling. In this case, due to the partitioning of the domain with cells defining the initial volume of air and aluminum, only the elements overlapped with the mold are filled with void.

Boundary conditions are imposed on all surfaces of the Eulerian domain, excluding the top blue circular surface (refer to the figure below) of Inflow and outflow imposed to none to keep the materials inside the volume. On the top surface, Eulerian boundary conditions are defined as inflow=free and outflow=equilibrium, allowing aluminum to enter.

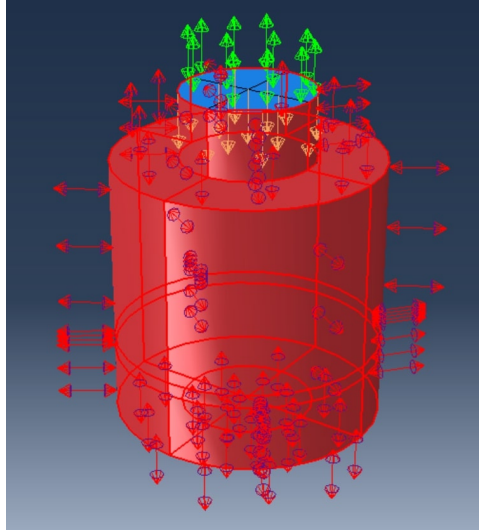


Figure 65: In red the closed boundary, green arrows the open boundary

To enable Abaqus to compute interactions between Lagrangian and Eulerian bodies, contact interactions need to be defined. For the mechanical problem, both tangential and normal behaviors are specified using penalty conditions with a value of 0.2 and hard contact. The former indicates friction between the aluminum and the mold, while the latter ensures that the aluminum cannot penetrate the mold. For the thermal aspect, thermal conductance is defined as pressure-dependent (the only option supported for the Eulerian domain), with a value of 0 W/m K for pressure = 0 and $50 \text{ W/m}^2 \text{ K}$ for all other pressure values [55]. This implies that as soon as the Eulerian material (air or aluminum) comes into contact with the mold, heat conduction begins.

The mesh is generated with a general seed of 0.002 m, and a local seed is imposed to create a finer mesh close to the interaction zone. Local seed is applied to the edges (zone A) that encompass the volume between the aluminum and the mold, with an approximate dimension along the Y -axis of the mesh set to 0.0004 m. In the region of the volume overlapped with the Lagrangian body (zone B), the seed is imposed on the edges with a bias to create a finer mesh near the interaction zone, resulting in an approximate dimension along the Y -axis ranging from 0.002 m to 0.0004 m.

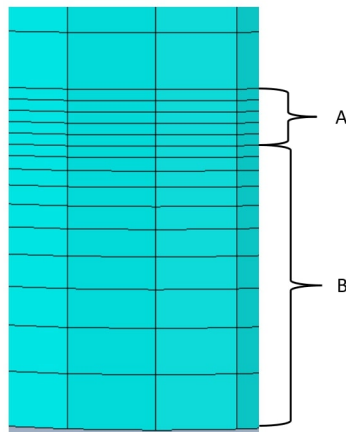


Figure 66: Detail of Eulerian mesh in the interaction zone

The simulation results indicate that at time 0.00 s, the aluminum occupies a columnar volume. However, by time 0.04 s, it spreads to cover a broader volume as more material enters from the open top surface. This material enters with an imposed velocity along the Y -axis with a magnitude of -1 m/s. Despite the applied gravity load, the aluminum tends to adhere to the upper surface of the volume, suggesting that the current model still suffers from numerical problems caused by errors in the definition. A primary hypothesis is that numerical issues arise from the presence of sharp corners, particularly in the vicinity of the convex corner. In this area, the material attempts to separate from the surface due to the acting gravity force. However, due to the imposed boundary condition preventing material entry through the corner, the aluminum remains stuck to the surface even after the corner, resulting in this non-physical behavior.

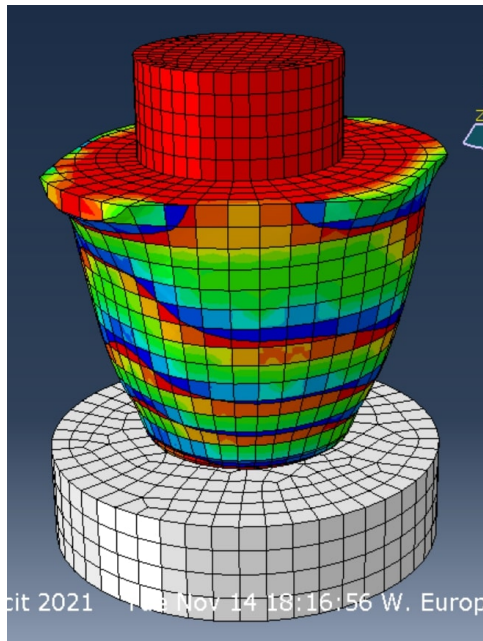


Figure 67: Aluminum at $t = 0.04s$

The outcomes of this simulation serve as the foundation for advancing the model to address existing issues. The initial improvement involves opening a section of the surface to facilitate air escape and prevent an increase in pressure inside the cavity. The second enhancement, informed by insights gained from other simulations, is to enforce zero velocity on all closed boundaries. This condition proves effective in resolving numerical challenges that arise at sharp corners, as observed in the previous simulation.

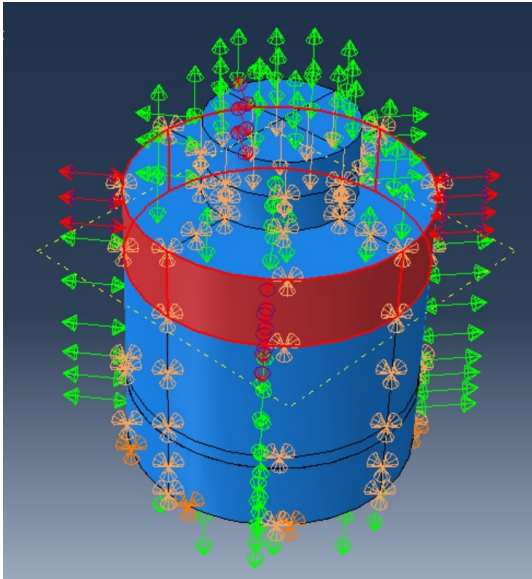


Figure 68: In red the new open boundary

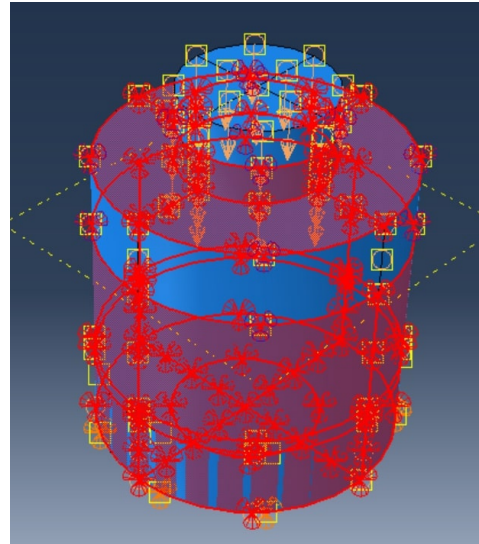


Figure 69: In red the boundary with the imposed zero velocity

The enhancements implemented resulted in improved outcomes, addressing the earlier issue of material adhesion to the upper surface of the domain. Moreover, the material flow is now directed towards the bottom, reflecting a more realistic behavior. The material does not stick anymore through the edge curvature as the imposed zero velocity does not allow the material to move around the edge but it just detaches and falls. To further enhance the results, it is advisable to refine the mesh near the interaction zone along the X and Z axes. Currently, these dimensions align with the global seed, resulting in flat but large elements.

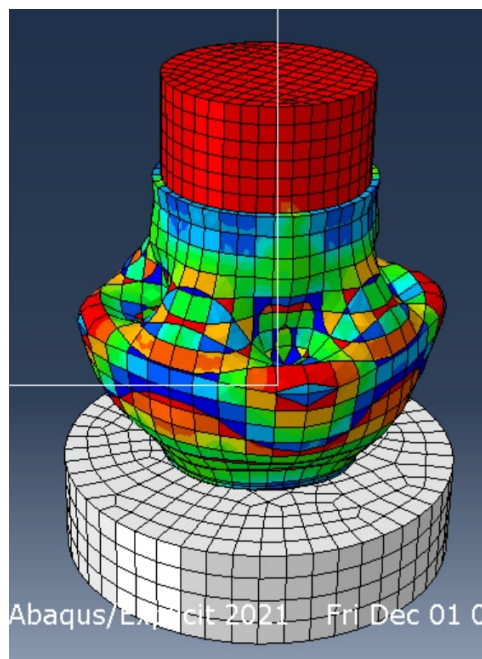


Figure 70: Aluminum at $t = 0.025$ s

9.3 Eulerian

Here are the results of the simulation obtained from the Eulerian model, elaborated extensively in Section 8.2.

9.3.1 Filling phase

The Abaqus simulation runtime for the filling phase, lasting 1.6 s, is 20:42 h. Throughout this phase, molten aluminum enters through the inlets, initiating the process of filling the Eulerian domain. At $t_1 = 0.0$ s, aluminum is initially confined to a small section of the inlet ducts (depicted in red). Driven by gravity, it gradually descends, and, due to the specified boundary conditions (inflow=free), pulls more aluminum into the domain. As the flow reaches the surface of the piston head, it begins to disperse, creating a spray pattern caused by the impact (see Figure 72).

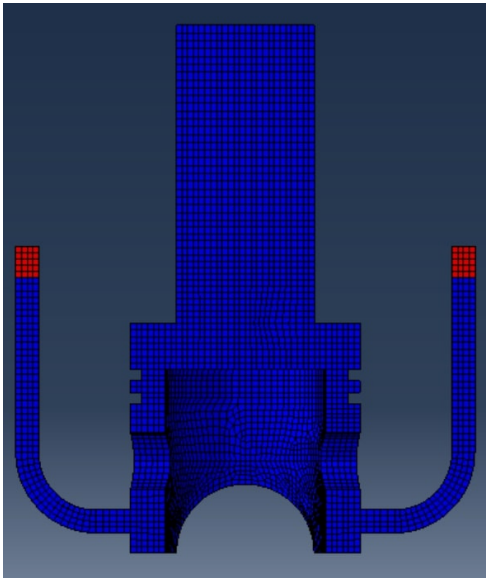


Figure 71: Volume Fraction of aluminum at $t=0.0$ s

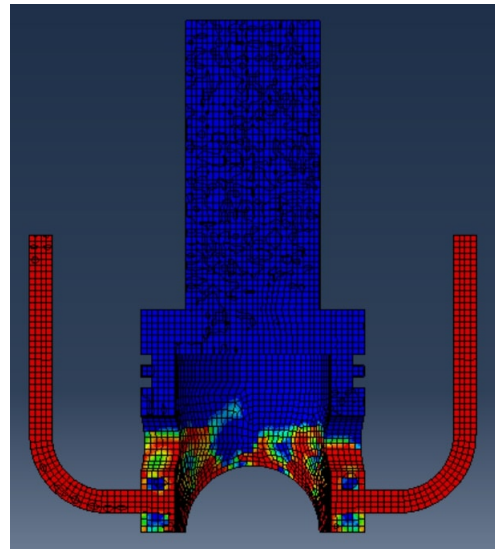


Figure 72: Volume Fraction of aluminum at $t=0.30$ s

Subsequently, the flow decelerates as it enters the larger volume of the piston head compared to the ducts (see Figure 73). Despite the ongoing filling process, certain regions (depicted in blue and green) remain entrapped air within the aluminum volume because of the turbulence.

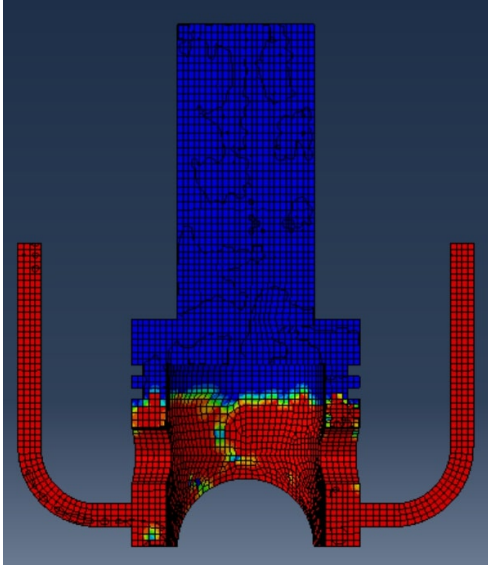


Figure 73: Volume Fraction of aluminum at $t=0.70$ s

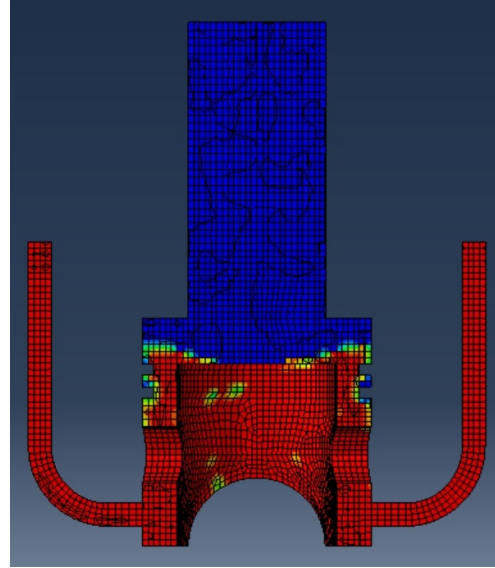


Figure 74: Volume Fraction of aluminum at $t=0.95$ s

At the conclusion of the filling step, the entire domain is occupied by aluminum, apart from residual air pockets, particularly noticeable in the o-ring grooves and on the top of the piston head. The coarseness of the mesh in these intricate features together with the BC of velocity equal to zero for the node at the surface limits the accuracy of tracking air entrapment, a finer mesh will solve the problem.

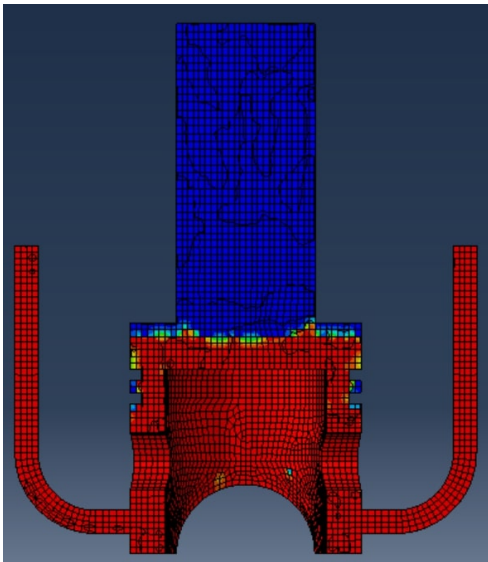


Figure 75: Volume Fraction of aluminum at $t=1.25$ s

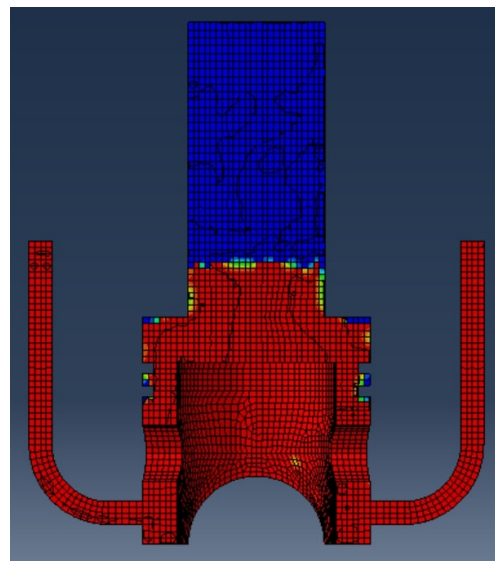


Figure 76: Volume Fraction of aluminum at $t=1.60$ s

By selectively displaying elements with aluminum fill exceeding 50%, we can better illustrate the division of aluminum flow upon contact with the wall at $t_1 = 0.19$ s. Subsequently, $t_1 = 0.47$ s, the flow slows down while filling the cavity, still exhibiting significant turbulence and entrapment of air (blue and green zones).

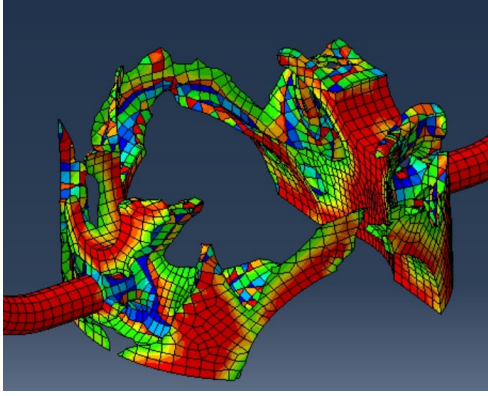


Figure 77: Aluminum at $t=0.19$ s

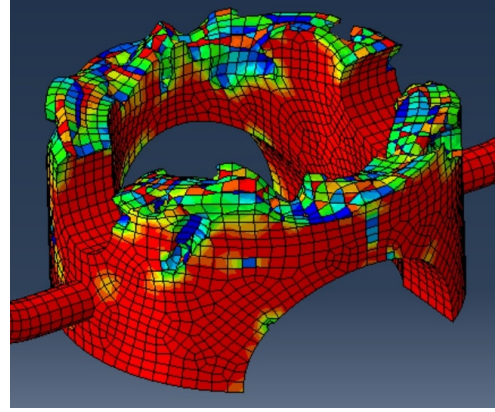


Figure 78: Aluminum at $t=0.47$ s

Plotting the velocity provides insights into the dynamic behavior of specific points during the filling phase. For instance, point 20881, located at the end of the inlet duct, experiences an initial transient phase when filled with air. Subsequently, as aluminum reaches the element, the velocity stabilizes at a constant value of 3.62 m/s for the majority of the step. Point 20529, situated in the piston head lateral surface, exhibits a velocity profile with two distinct peaks. The first peak corresponds to the displacement of air pushed away by the descending aluminum in the duct, while the second peak marks the passage of the aluminum front. The velocity then stabilizes at 2.94 m/s. Examining the point above the semicircular hole in the side wall (45999), the velocity plot fluctuates due to the influence of aluminum turbulent flow reaching this cavity zone, as depicted in Figure 77. Subsequently, the velocity stabilizes around 0.28 m/s during the filling, approximately at $t_1 = 1$ s. Similarly, the point at the corner of the top surface (14273) initially exhibits a near-zero velocity until $t_1 = 0.65$ s, when the air accelerates to exit the mold. The velocity then stabilizes for the remainder of the filling phase. Finally, the point in the riser (4656) shows minimal velocity until around $t_1 = 1$ s when it is reached by the aluminum, resulting in a velocity peak. Subsequent undulations occur before settling at a filling velocity of 0.13 m/s.

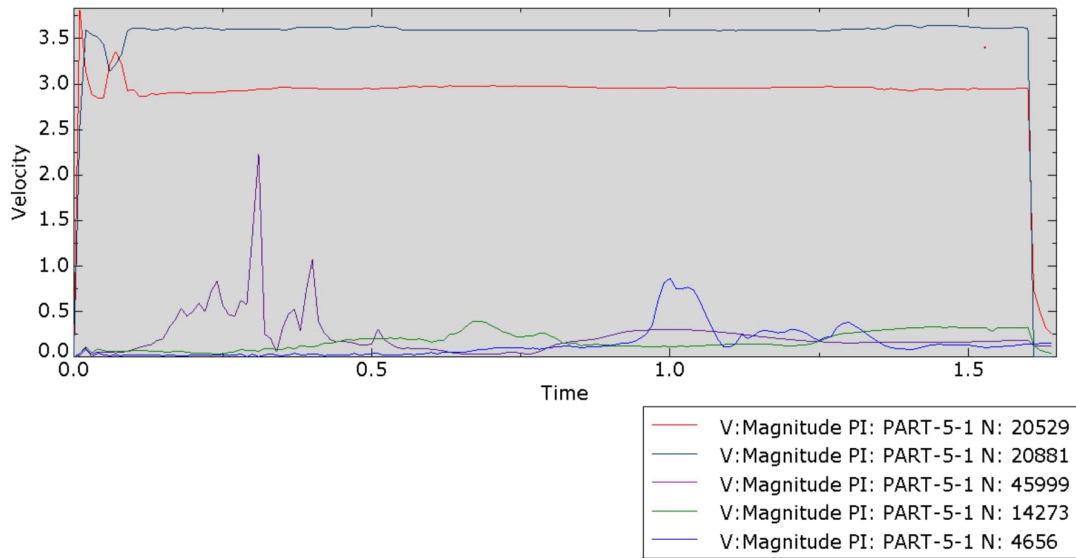


Figure 79: Velocity magnitude plot for the filling phase for point in figure 79

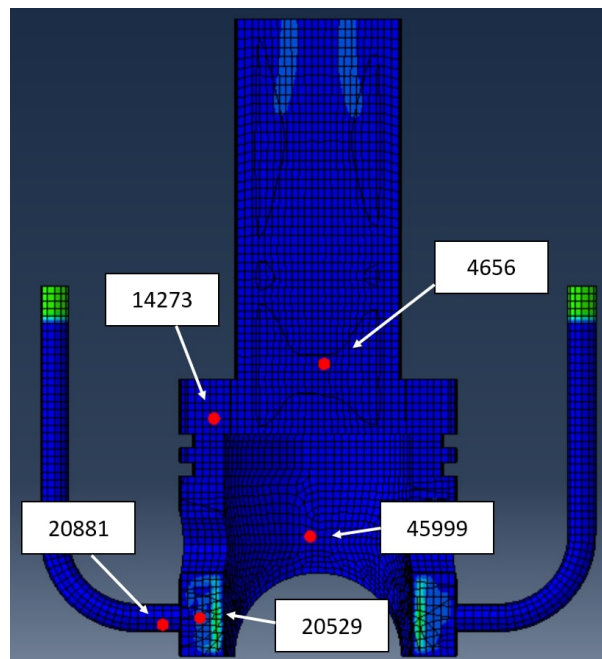


Figure 80: point selected to plot the velocity

The aluminum volume fraction swiftly attains a value of 1 in the initial two points, signifying that they are the first zone of the piston head to be filled with aluminum. As expected, the other points are subsequently reached by the material. The presence of numerous oscillations is attributed to the turbulent flow of aluminum. Since there is no void in the volume at the start of the simulation, the void volume fraction remains zero for all elements throughout the entire simulation. The air volume fraction is then calculated as 1 minus the aluminum volume fraction.

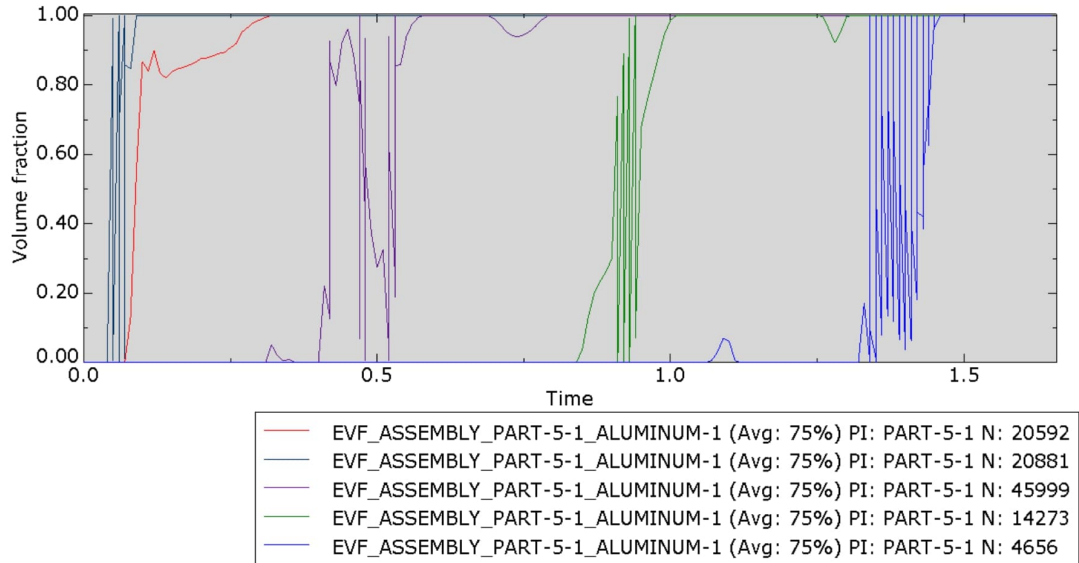


Figure 81: Aluminum volume fraction plot for the points in figure 80

The figure below highlights areas that still contain air. Predominant air entrapped is located in the spaces between the o-ring grooves, where the mesh's coarseness limits accurate tracking of the air as it escapes when the aluminum fills these spaces. Another notable area is the corner of the top surfaces, where the completely flat and horizontal surface makes it challenging for the air to flow away into the riser, also because of the velocity BC imposed on all the external surfaces.

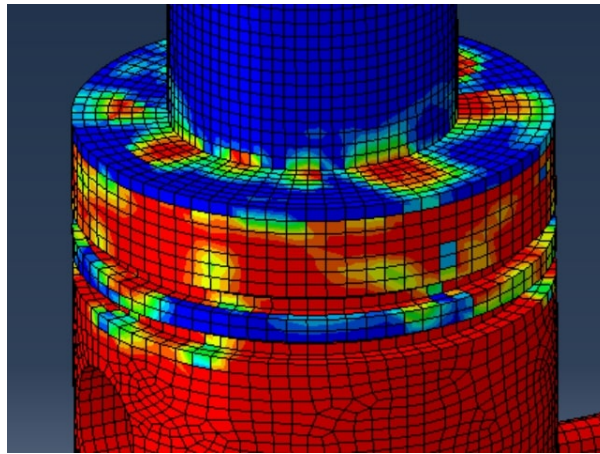


Figure 82: Remaining air (in blue) at $t_1 = 1.60s$

The temperature evolution during this phase is primarily influenced by the aluminum temperature. The molten metal enters the volume at a constant temperature of 1000 K (726.85 °C), while the air, initially at 310 K (36.85 °C), experiences a rapid increase in temperature. No heat flux is present on the lateral surface of the Eulerian domains, indicating that heat exchange occurs solely between the air and aluminum. As seen in Figure 84, the aluminum remains at the inlet temperature of 1000 K, showcasing its high capacitance during the short filling phase. A comparison of the left image highlights

that, where the aluminum is present, the temperature remains at 1000 K, while the surrounding air has undergone a temperature change from the initial 310 K because of the heat exchanged with the aluminum.

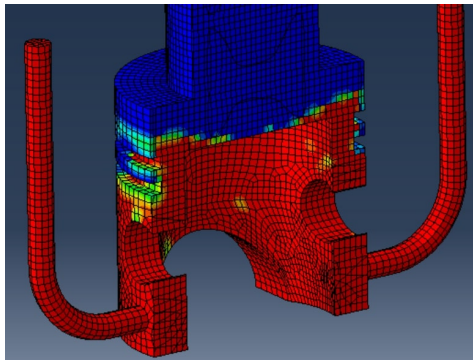


Figure 83: Aluminum at $t=0.95$ s

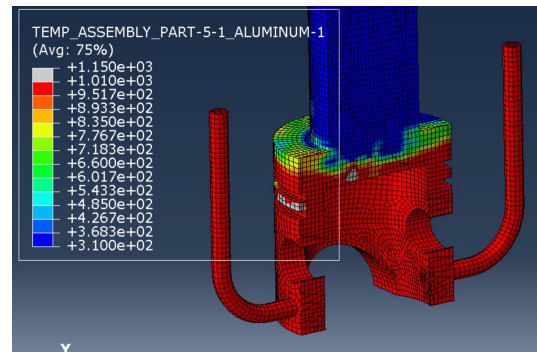


Figure 84: Temperature at $t=0.95$ s

Initially, all points begin at a temperature of 310 K, reflecting the initial air-filled conditions. Subsequently, the temperature begins to rise as the air exchanges heat with the entering aluminum, reaching a uniform temperature of 1000 K across all nodes once they are reached by the molten metal.

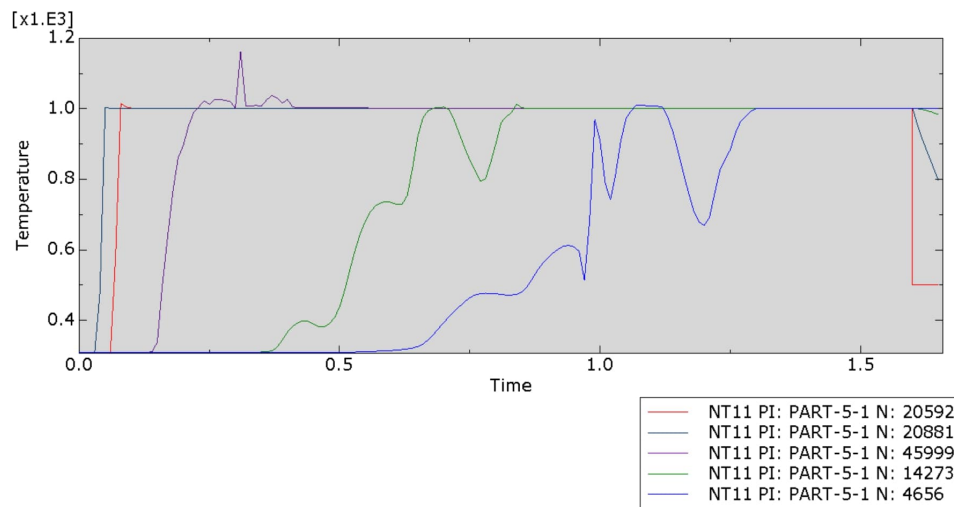


Figure 85: Temperature plot during filling phase

The temperature is nearly homogeneous throughout the entire aluminum domain due to the adiabatic behavior with the external world considered during the filling step. In fact, during the filling step, no thermal boundary conditions are imposed to model the mold. Another contributor to this temperature field is the negligible heat absorbed by the air because of the low conductivity and brief time step.

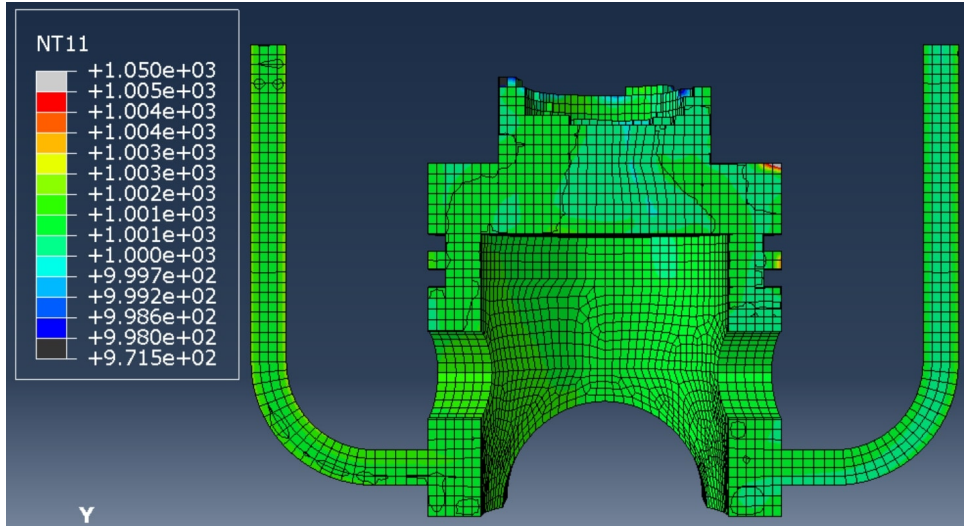


Figure 86: Aluminum temperature field at $t=1.6$ s

The geometry underwent modifications following attempts on a similar design featuring a different riser geometry. Initial simulations yielded comparable results until the point where the molten aluminum reached the top surface of the piston head. Then the material was drawn into the narrower section of the riser, resembling a nozzle, leading to a swift vertical flow of aluminum reaching the outlet surface. This caused the surface to act as an inlet allowing new aluminum material inside the domain. To address this issue, various tests were conducted. Initially, a taller riser was designed to prevent aluminum from reaching the outlet, but this adjustment introduced non-physical outcomes (see Figure 87). After further troubleshooting, the current design was chosen for its larger riser surface area, eliminating the undesirable results.

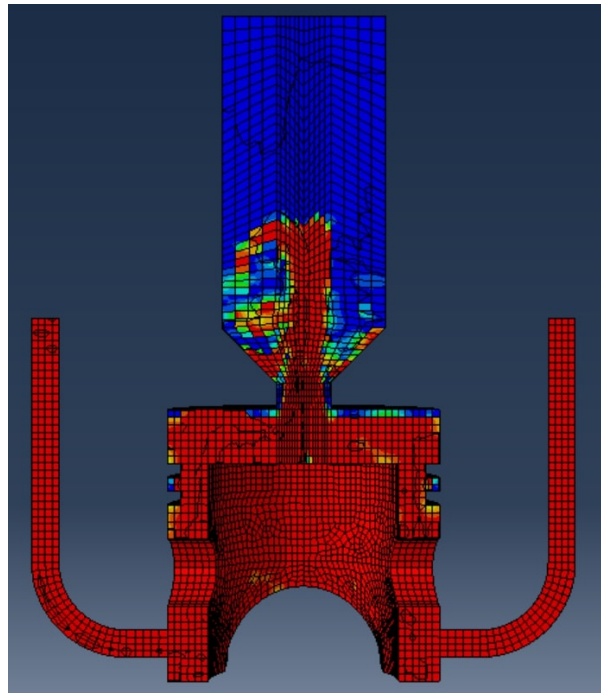


Figure 87: Aluminum volume fraction $t_1 = 1.42s$

9.3.2 Cooling phase

The simulation is conducted in two steps to facilitate the modification of Eulerian boundary conditions on the inlet surface. Specifically, at the beginning of the cooling phase, the boundary conditions are adjusted to inflow=outflow=none to prevent further influx of aluminum into the domain. During this step, thermal boundary conditions as Dirichlet conditions are introduced by fixing the temperature of the external surface. This temperature is assumed to remain constant at 500 K (226.85 °C). The decision to impose a fixed condition is motivated by the absence of data, and alternative values or time-dependent variations would not improve accuracy or validity without supporting data.

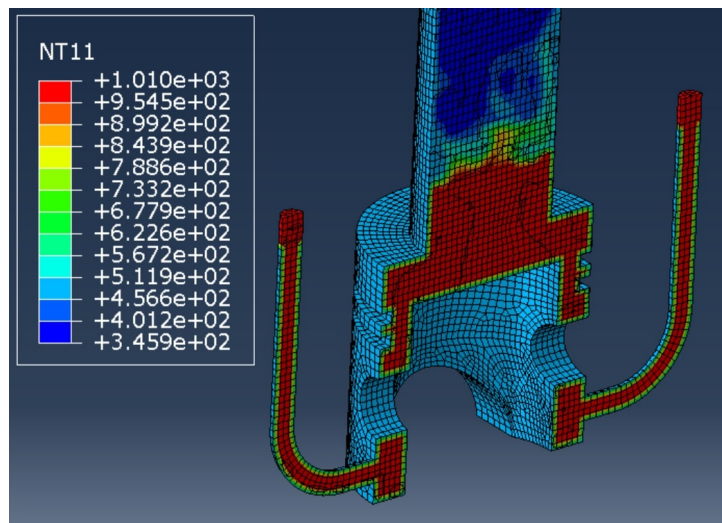


Figure 88: Temperature field at $t_2 = 0s$

It is evident that the air in the riser retains its initial temperature, while the air trapped inside the mold heats up rapidly due to the heat flux from the aluminum. The material within the mold begins to conduct heat with the material at the boundary, where a fixed temperature value is imposed while the core material still remains at 1000 K.

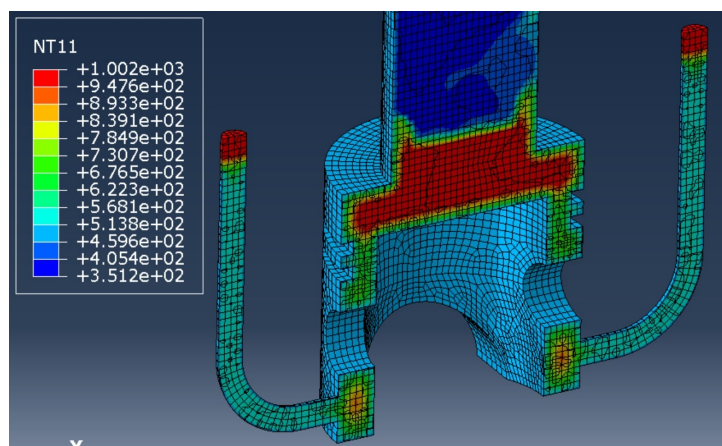


Figure 89: Temperature field at $t_2 = 0.50s$, total simulation time $t=2.1 s$

As time progresses, the aluminum continues to exchange heat with the mold through the imposed boundary conditions, resulting in a temperature gradient directed toward the center. Notably, the inlet ducts, characterized by a larger area-to-volume ratio, cool down more rapidly than the rest of the part, exhibiting a temperature of 569 K in the middle of the duct.

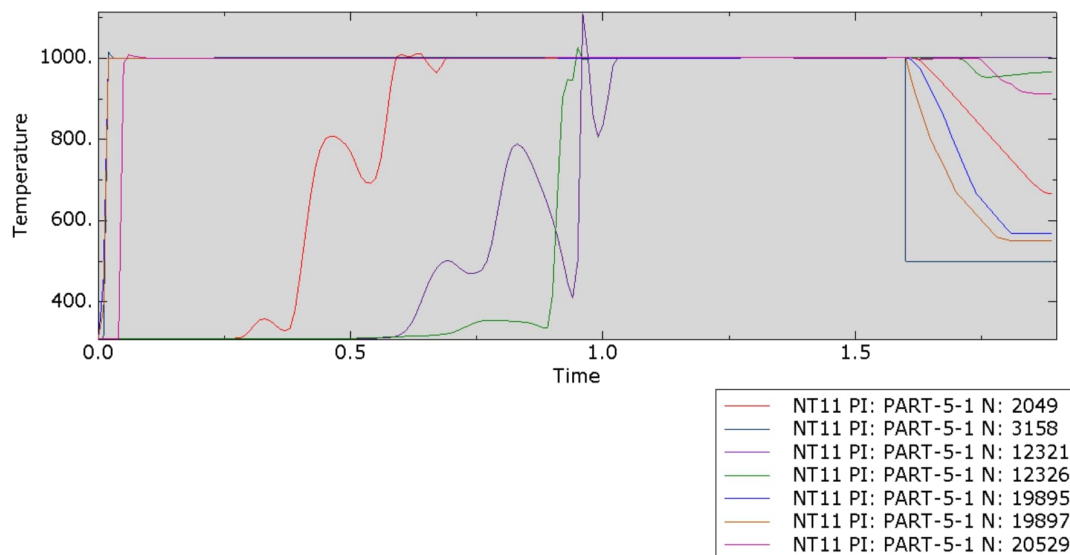


Figure 90: Temperature plot until $t_2 = 0.50s$, total simulation time $t=2.1 s$

The temperature plot in the graph begins at the initiation of the filling phase, and the cooling phase commences at $t=1.6 s$. Notably, point 3158, situated on the boundary, exhibits a step-like behavior due to the imposed boundary condition. Points 19895 and 19897 within the inlet duct display a remarkably steep cooling curve, aligning with the previously explained swift cooling phenomenon in the duct compared to the rest of the mold because of the high surface-to-volume ratio. Additionally, point 2049, positioned near the o-ring groove, experiences distinctive behavior owing to the intricate geometry created by the grooves, resulting in a thinner volume with a substantial area. Conversely, point 20529, located in a thicker zone, requires more time to cool down. Point 12326, situated in proximity to a boundary but inside the thicker portion of the piston head, demonstrates a flatter cooling curve due to its higher heat capacitance for the first portion, then a sharp step because of the heat extracted by the boundary points acting as heat sink and then again the curve tends to flatten thanks to the equilibrium of heat received from the hotter material on top and the colder under. Lastly, point 12321, positioned in the middle of the top of the piston head, does not immediately reflect the influence of the boundary condition during this short period, maintaining a temperature of 1000 K.

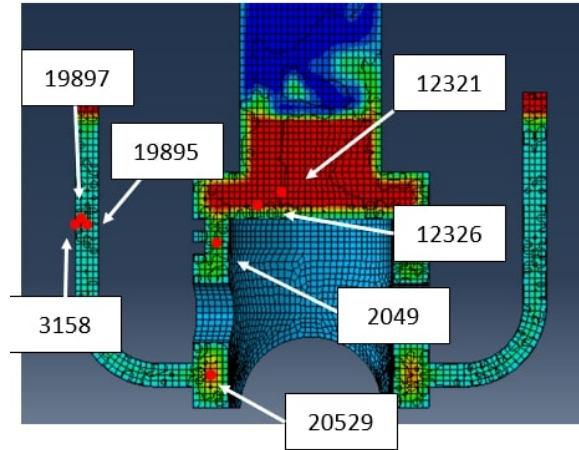


Figure 91: Points with plotted temperature

The simulation concludes at $t=2.1$ s after running for 50 hours and 17 minutes. Notably, during the filling step where the temperature boundary condition was not implemented, the simulation exhibits a stable time increment of approximately $1.015 \times 10^{-6}s$, determined by the dominant mechanical problem. In contrast, during the cooling step, the critical stable time increment is influenced by the thermal problem and amounts to about $4.287 \times 10^{-8}s$. Consequently, the presence of temperature BC implementation in the cooling step leads to a significantly prolonged runtime. Initially, the time required to compute the output field every $1 \times 10^{-2}s$ is 4 minutes and 47 seconds. Towards the end of the simulation, this time increases to 1 hour and 15 minutes, resulting in a computation time escalation of 15.6 seconds.

Due to these challenges, the simulation was prematurely terminated without completing the cooling process. The cooling phase, which lasts 40 seconds according to the Inspire-cast simulation, would require approximately 500 hours of run time in Abaqus.

Furthermore, the simulation results reveal inconsistencies with reality. Surface points display an abrupt temperature step, indicating non-coherent behavior. Points in proximity to these surfaces exhibit an extremely rapid cooling curve, transitioning from 1000 K to temperatures ranging from 600 K to 900 K within approximately 0.5 seconds. This abrupt cooling rate is excessive, especially considering the real casting process where the mold absorbs heat from the aluminum, causing both materials to approach an equilibrium temperature between their initial temperatures causing a slower cooling rate.

9.4 Lagrangian model

Starting from the results of the Eulerian model a Lagrangian simulation was developed using the same geometry to explore the impact of various thermal boundary conditions on the results, considering the shorter run time. Specifically, a surface heat flux was introduced as a load to observe the outcomes. In the Eulerian model, the filling phase was defined as adiabatic with respect to the heat exchanged by the material with the mold, resulting in a temperature field at the end of the step with an almost constant value of 1000 K for all the aluminum. This facilitated the creation of a Lagrangian model

that assumes the mold is already filled with material with a velocity value equal to zero for all the points. The surface heat flux was calculated as the total heat exchanged by the aluminum to decrease from 1000 K to 400 K using the heat equation:

$$Q = Mc_p\Delta T \quad (9.1)$$

The values are approximations, as the overall behavior remains insensitive to the decimal precision in the equation. The total volume of the part multiplied by the density yields a mass (M) of approximately 1.25 kg. The specific heat of aluminum is $c_p = 910 \text{ J}/(\text{kg K})$, and ΔT is 600 K. These values result in a total heat of around $7 \times 10^5 \text{ J}$. This value is then introduced in Abaqus over a duration of 10 seconds as the total heat flux, and the software computes the value per unit area. Initially, this load was applied on all surfaces, but it resulted in the inlet ducts cooling down faster than the rest, leading to unrealistic temperatures below 0 K. To rectify this, the model was enhanced by applying the load exclusively to the piston head surfaces. Additionally, another load was applied to the ducts with a value of $3 \times 10^4 \text{ J}$ computed using the same formula.

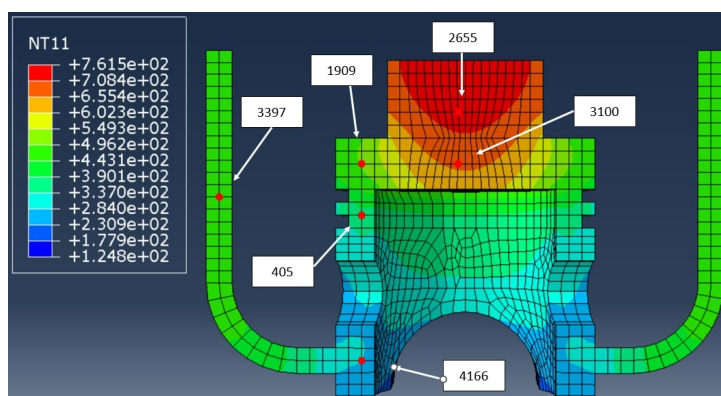


Figure 92: Points selected to plot temperature history

This approach yields superior results compared to the one employed in the Eulerian model, where the temperature was imposed at the surface. The primary drawback of the earlier approach was that the material at the boundary had an externally imposed temperature, resulting in a step-like temperature history. These points then act as temperature sinks and the material inside the mold begins to exchange heat with them. In this new approach, all nodes exhibit a more coherent temperature behavior, although some points still reach an extremely low temperature because of the uniform heat flux distribution. Some zones as the riser are thicker and the flux, dependent on the temperature difference, should have a higher value than in other thinner zones that cool down faster. This leads to an unrealistic scenario of some points that approach and trespass 0 K. Furthermore, the assumption of a constant heat flux generates a nearly uniform descending curve, for every point throughout the entire cooling phase, while a steeper initial phase is expected due to the greater temperature difference with the mold.

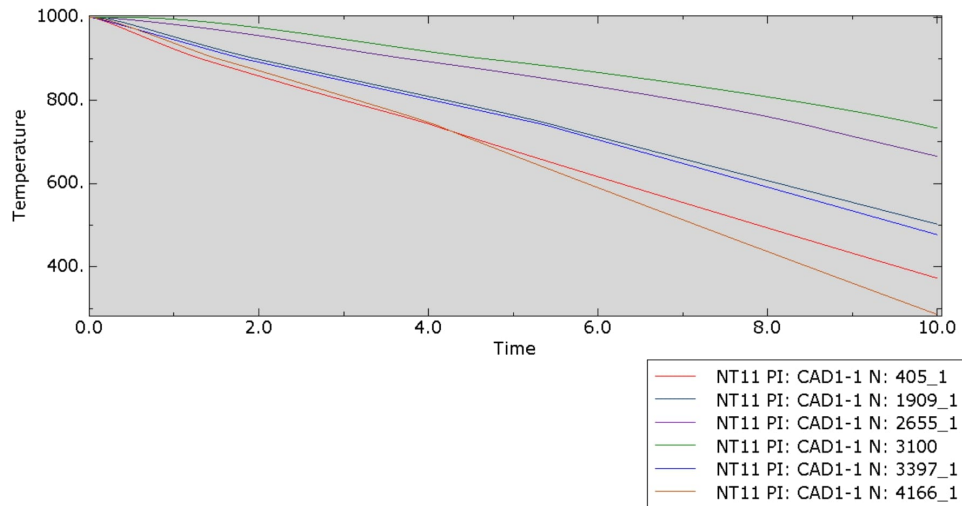


Figure 93: Temperature plot for points in figure 92

This model is constructed with an estimated heat flux that only partially reflects reality, as several assumptions were made about its intensity and, more importantly, its spatial and temporal distribution.

A potential enhancement to this approach involves conducting multiple simulations to refine the definition of the heat flux absorbed by the mold, particularly by specifying it on a zone-by-zone basis rather than as a total flux. This also necessitates an external approximation of the heat exchanged by each region of the part based on its volume and adjacent surface area.

Always under the assumption that the aluminum still shows a uniform temperature of 1000 K at the end of the filling phase and completely fills the cavity, an enhanced model is constructed to consider both the mold and the cast part. In this model, a Lagrangian body is utilized to define the geometry of both instances. The mold geometry derives from the Inspirecast one to be able to later compare the results. Two distinct materials, aluminum for the cast part and steel for the mold, are then assigned to their respective regions within the volume. This approach represents a more comprehensive model, as the mold is explicitly modeled rather than being replaced by simplified boundary conditions. This problem formulation demonstrates a brief run time compared to the Eulerian approach, resolving a step time of 40 seconds in approximately 2 hours and 21 minutes.

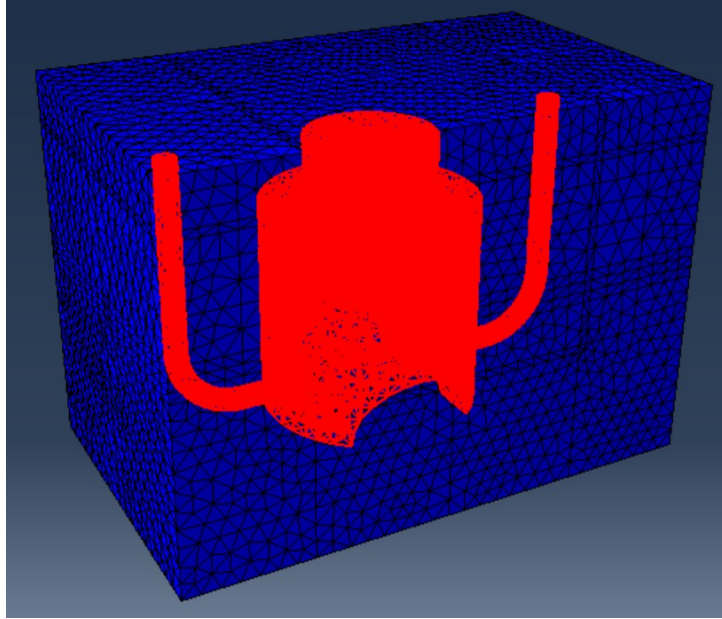


Figure 94: Lagrangian model geometry with the mold (blue) and cast part (red)

In this comparison, the color legend is kept constant to illustrate the cooling process for both the mold and cast part more effectively. The temperature begins at 1000 K for aluminum (red) and 500 K for the steel mold (blue). It concludes with the majority of the mold (blue) at around 507 K, featuring a central hotter zone (light blue) at approximately 593 K.

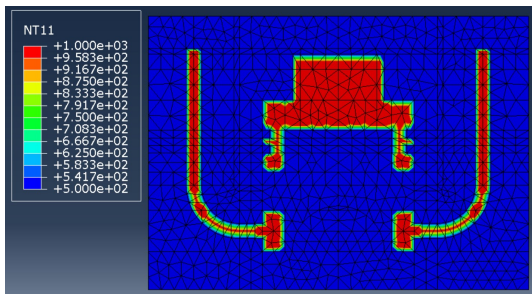


Figure 95: Aluminum temperature field at $t=0.0$ s

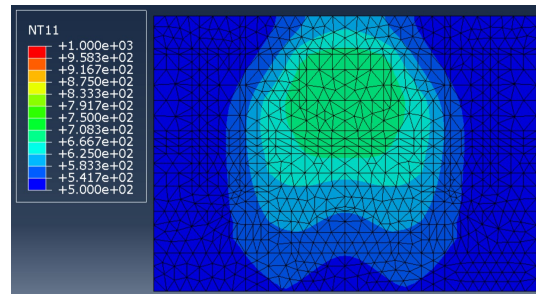


Figure 96: Aluminum temperature field at $t=40$ s

A dynamic legend enhances the comprehension of how heat is exchanged between the mold and aluminum, highlighting the most affected zones. As detailed in earlier sections, the top portion of the piston head, along with the riser, exhibits significant heat capacitance due to its high mass, making it consistently the hottest part of the system. Inlet ducts cool down faster than other areas due to their smaller volume and being enveloped by a substantial portion of the mold. Notably, the heat flux exchanged from the aluminum to the mold is conducted through the mold, causing the temperature to rise also in farther regions over time. The central part of the mold, surrounded by the hot cast part, emerges as the hottest zone at the conclusion of the cooling phase.

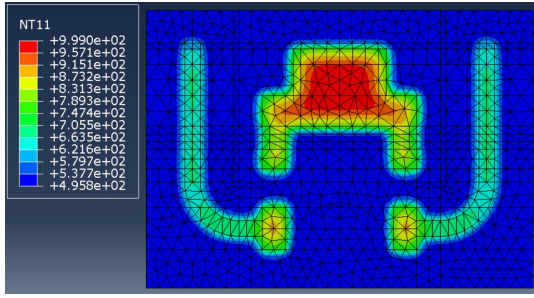


Figure 97: Temperature field at $t=1.06$ s

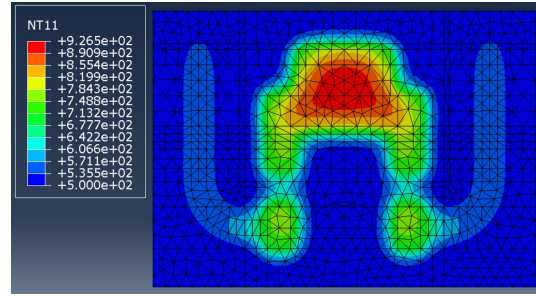


Figure 98: Temperature field at $t=4.61$ s

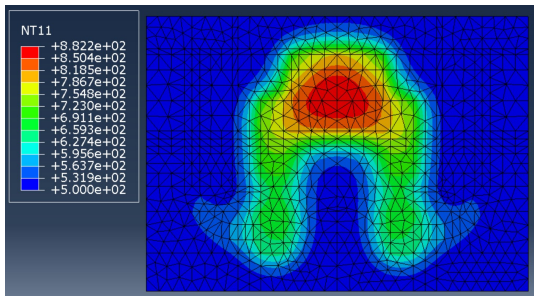


Figure 99: Temperature field at $t=10.04$ s

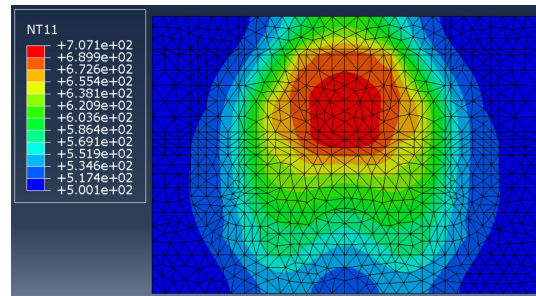


Figure 100: Temperature field at $t=40$ s

The temperature history of aluminum highlights the rapid cooling of the inlet ducts, which could potentially result in defects, particularly in the small volumes that impede the flow of molten material to the region where porosity forms. On the contrary, the red area, representing the hotter section, experiences a slower cooling process. However, due to the considerable thickness of the material, the thermal volumetric shrinkage is important and may lead to a highly stressed area in the middle. This central region is the last to solidify, and insufficient material is available to adequately fill it because of the shrinkage.

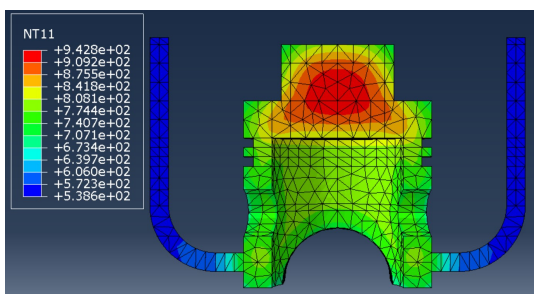


Figure 101: Aluminum temperature field at $t=3.8$ s

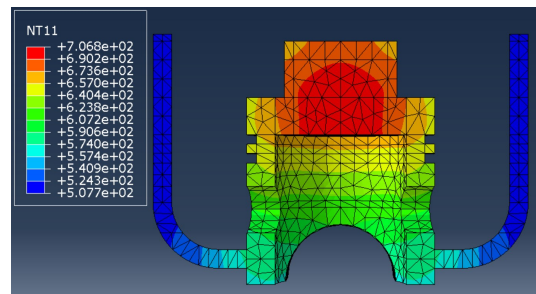


Figure 102: Aluminum temperature field at $t=40$ s

This model demonstrates higher accuracy compared to the Eulerian approach, as the only assumption made here is that, at the end of the filling phase, all the aluminum maintains a temperature of 1000K. The thermal problem is subsequently addressed by calculating the heat exchange between the two components without relying on additional assumptions.

9.5 Comparison Inspirecast Abaqus

The comparison of the two simulations allows us to extrapolate several important information that are useful for further improving the model.

It is noticeable how In Abaqus the aluminum is quicker than in Inspirecast to fill the cavity even if the velocities in the ducts are comparable. This is because to ensure that the material entering the domain is all aluminum a portion of the ducts was previously filled in the model definition giving an advantage in time. Furthermore Inspirecast inlet definition creates a small volume on top of the geometry of the part that imposes another delay with respect to Abaqus.

It is also important to notice how the aluminum flow in Abaqus is more turbulent and at contact with the surface at the end of the duct, it divides and creates several complex flows with respect to Inspirecast's slower and calmer flow dynamic.

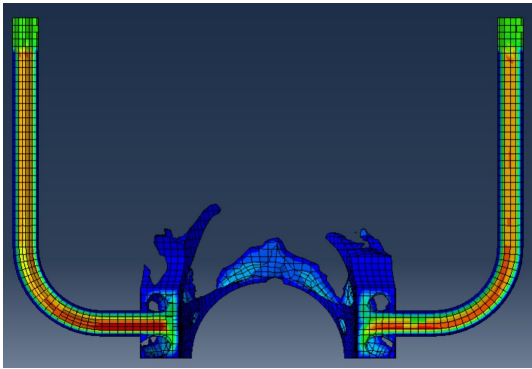


Figure 103: Abaqus aluminum at $t=0.19$ s

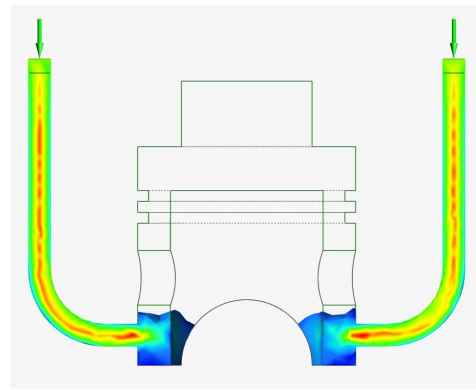


Figure 104: Inspirecast aluminum at $t=0.2$ s

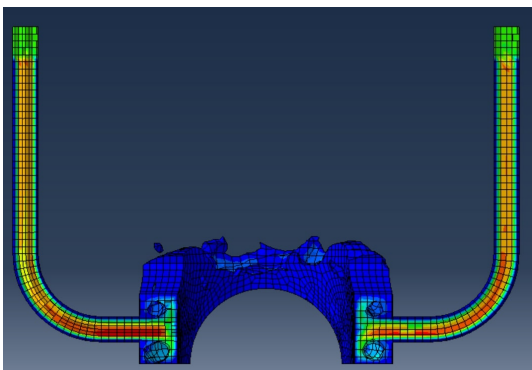


Figure 105: Abaqus aluminum at $t=0.35$ s

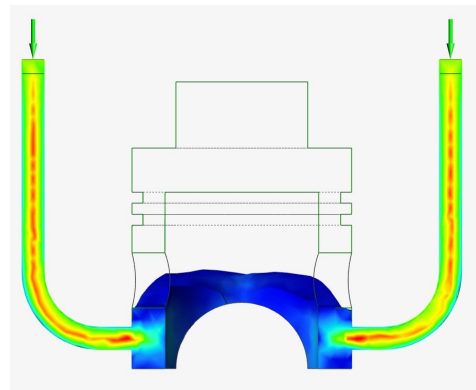


Figure 106: Inspirecast aluminum at $t=0.35$ s

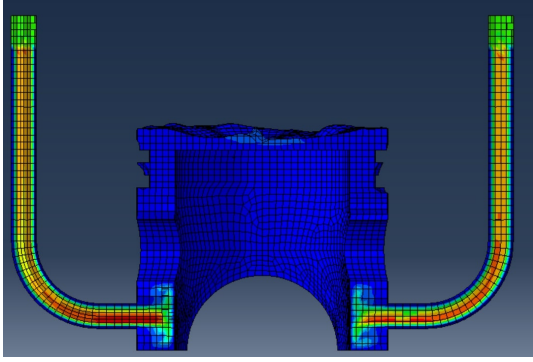


Figure 107: Abaqus aluminum at $t=1$ s

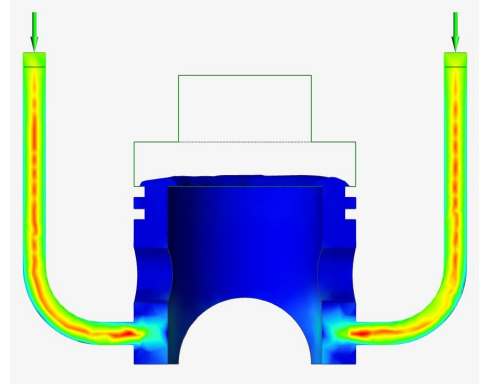


Figure 108: Inspirecast aluminum at $t=1$ s

One of the assumptions embedded in the Abaqus model definition was to incorporate the thermal effects of the mold exclusively during the cooling phase, considering that the mold acts adiabatically in the filling phase. This simplification was necessitated by the absence of data to introduce a spatial and temporal thermal boundary condition that varies together with the aluminum flow entering the cavity. The imposed fixed temperature boundary condition forces the boundary nodes to remain at 500 K, leading to the instantaneous solidification of aluminum. However, due to the coarse mesh, only a few elements inside the ducts, not constituting the external surface, witness the aluminum flow, resulting in a non-realistic behavior. This issue would be mitigated with a very fine mesh, where the majority of the duct area sees the flow, and only a few elements in contact with the surface witness the material solidifying, mirroring reality. This assumption in Abaqus causes the aluminum to remain at 1000 K, owing to the minimal heat flux at the air-aluminum interface during the short duration of the initial phase. In contrast, Inspirecast comprehensively considers all thermal aspects of the mold, extracting heat from the aluminum, resulting in a significant temperature decrease, as deeply explained in the preceding section dedicated to analyzing Inspirecast results.

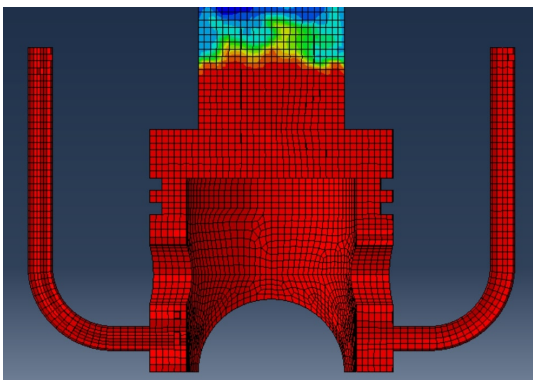


Figure 109: Abaqus aluminum temperature at $t=1.6$ s

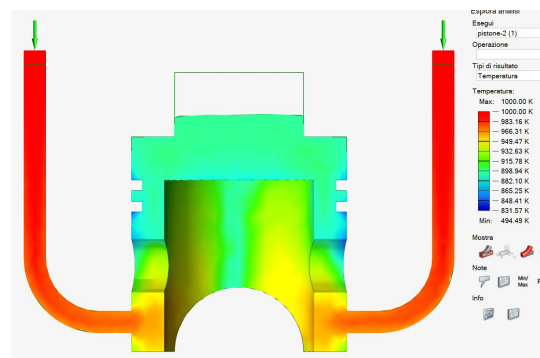


Figure 110: Inspirecast aluminum temperature at $t=1.6$ s

In the following series of images, the disparity between the temperature fields generated by the two software platforms becomes apparent. In the images produced by Abaqus (on the left), the imposed temperature at the boundary generates a substantial

temperature gradient due to the presence of aluminum at 1000 K within the part and the surface temperature set at 500 K. This results in a faster cooling process compared to the Inspirecast simulation. Consequently, in the Abaqus simulation, the green area near the o-ring groove reaches around 650 K at $t=2.1$ s, whereas the corresponding area in the Inspirecast simulation registers a temperature of 893 K, indicating a temperature difference of approximately 240 K.

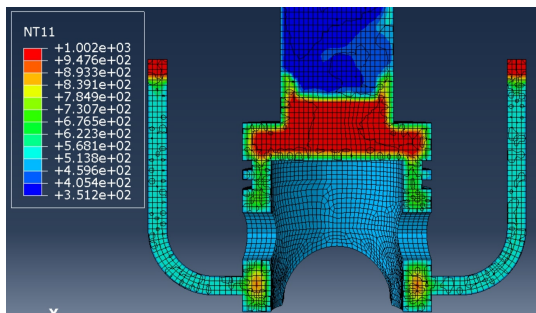


Figure 111: Abaqus aluminum temperature at $t=2.1$ s

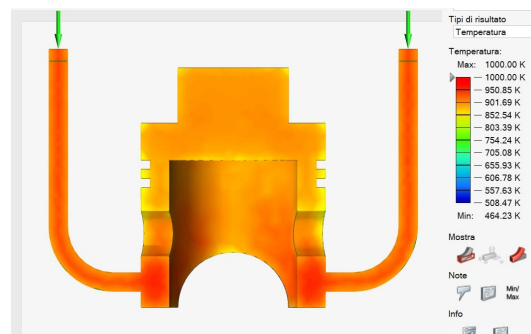


Figure 112: Inspirecast aluminum temperature at $t=2.1$ s from the start and $t_2 = 0.5$ s from the beginning of the solidification step

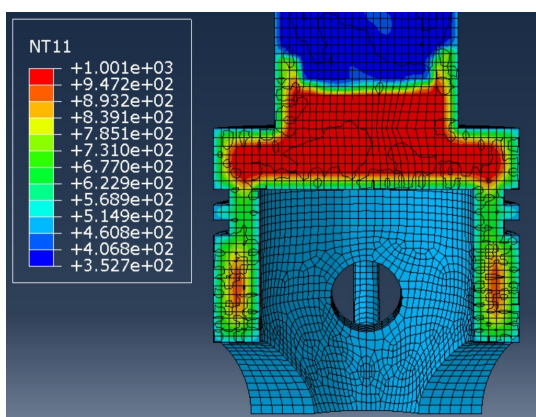


Figure 113: Abaqus aluminum temperature at $t=2.1$ s

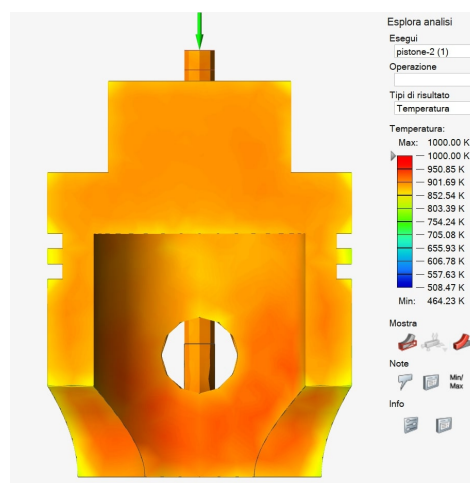


Figure 114: Inspirecast aluminum temperature at $t=2.1$ s from the start and $t_2 = 0.5$ s from the beginning of the solidification step

Here is a comparison of the temperature evolution in the cast part during the solidification phase. Both graphs exhibit similar behavior, but the Abaqus plot appears more compressed to the left, suggesting faster heat dissipation through the mold, likely influenced by the coarse mesh of the model.

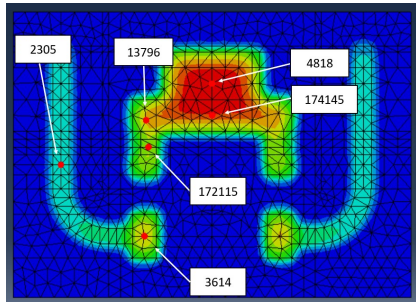


Figure 115: Abaqus aluminum temperature plotted points

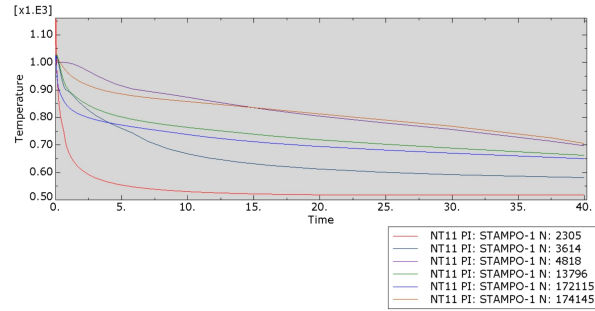


Figure 116: Aluminum temperature curves for points in figure 115

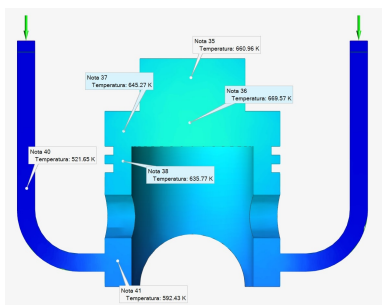


Figure 117: Temperature fields at $t_2 = 40.55s$ of the solidification phase

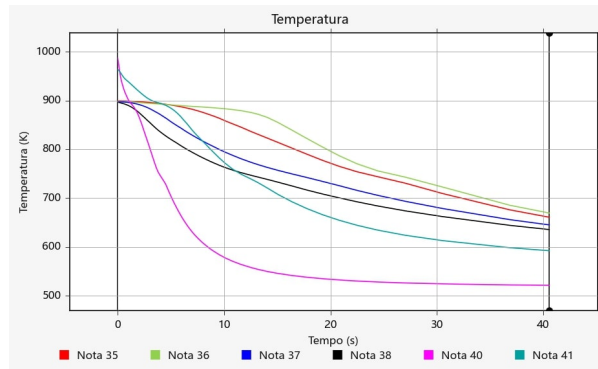


Figure 118: Temperatures plot during solidification phase relative to the points in figure 117

The temperature fields exhibit a comparable shape in the results from both software applications. In the case of a simple geometry, such as the one chosen for this model, predicting the final temperature field based on the initial conditions becomes more straightforward. The outcomes align with the anticipated field, indicating higher temperatures where more aluminum is concentrated. Moreover, this zone is positioned at the center of the mold, necessitating more time for cooling compared to regions closer to the boundary.

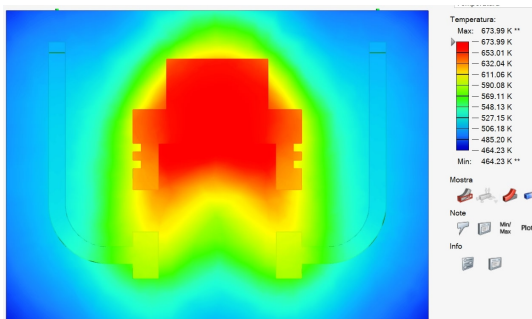


Figure 119: Inspirecast temperature field at $t_2 = 40.5 s$

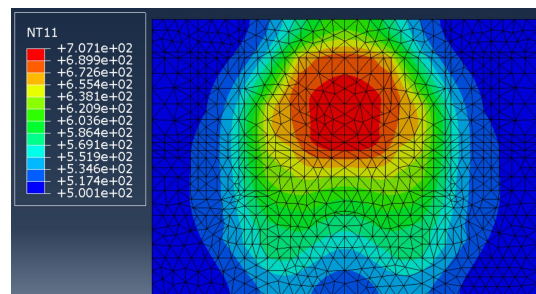


Figure 120: Abaqus temperature field at $t=40 s$

9.6 Conclusions

The utilization of Abaqus for modeling casting phenomena offers the advantage of fine-tuning various parameters in the model and simulation. However, this comes with the drawback of making several assumptions due to the software's limitations. An Eulerian formulation becomes necessary to track fluid aluminum and air during the filling phase and air entrapment during solidification. This formulation imposes certain limitations, such as the need for assumptions in defining boundary conditions to compensate for the absence of the mold in the model. Additionally, material density remains constant and cannot be a function of temperature, and material behavior equations are fixed and cannot vary with temperature.

Addressing the absence of the mold in the model involves creating a Coupled Eulerian-Lagrangian (CEL) model, introducing challenges in defining interactions between Eulerian and Lagrangian domains. Near the interaction zone, if the mesh is not refined enough, "void" tends to penetrate the Eulerian domain occupied by other materials because of numerical errors, requiring a significantly finer mesh. Tests performed show the need to refine the mesh of more than an order of magnitude with respect to the global seed. This refinement substantially increases model runtime (more than an order of magnitude) compared to an equivalent purely Eulerian model.

When opting for a purely Eulerian model, defining mechanical and thermal boundary conditions to substitute the mold introduces challenges. Mechanical boundary conditions are easily implemented through Eulerian BC option in Abaqus, but for the thermal BC assumptions are necessary, degrading model accuracy.

Another limitation lies in the inability to define material density as a function of temperature, preventing the tracking of shrinkage, microporosity, residual stress at the end of solidification and buoyancy effects between solid and liquid phases.

The final limitation pertains to phase change, as Abaqus only allows the definition of a single behavior equation for each Eulerian material. To model aluminum's behavior throughout the casting process, involving a transition from liquid to solid, compromises are needed, such as approximating solid behavior by defining viscosity as a function of temperature during the filling phase. This is feasible as the simulation time is short and a highly viscous media could be assumed as solid.

In comparison with specialized casting simulation software like Inspirecast, Abaqus exhibits significant limitations, particularly in simulation runtime, which is approximately 1000 times longer, not accounting for the time needed for model setup, especially in the load and mesh modules.

To optimize runtime, two distinct models are developed: one based on the Eulerian formulation to accurately track the flow of aluminum during the filling phase, and another based on the Lagrangian formulation to precisely monitor the evolution of the temperature field during the solidification and cooling stages. The latter model operates on the assumption that during the short time of the filling stage, the aluminum retains its pouring temperature, enabling the definition of a homogeneous temperature in the Lagrangian model. This assumption is partially supported by the Eulerian simulation, even though it considers adiabatic filling conditions. Inspirecast indicates a temperature drop of nearly 100 K in some small regions at the boundaries. Despite this, the Lagrangian model could

serve well as a first approximation or when considering specific mold materials or surface treatments that reduce heat exchange.

The accuracy of Inspirecast has not been directly tested in this study due to the unavailability of facilities for performing aluminum casting. However, previous studies comparing results from virtual casting environments and experimental results highlight the capabilities of such software [60]. Inspirecast, classified as a medium-level virtual casting environment, ensures good accuracy with short simulation times and relatively low costs. Nevertheless, it comes with limitations for advanced users, as certain operations, such as mold creation, are automated and not customizable by the user. An RX analysis on the cast part, when compared with simulation results, reveals accuracy in identifying the position of defects, such as porosity in the final part, but tends to overestimate their dimensions [61].

In general, Inspirecast serves as a valuable tool during the design phase, particularly in defining the running system to eliminate defects. However, in the author's opinion, a final experimental test is imperative to validate the designed geometry. This practice contributes to a reduction in design time and cost, ensuring a more robust and reliable outcome.

Developing an in-house Finite Element Method (FEM) model for virtual manufacturing in Abaqus is an insightful exercise. This hands-on approach not only enhances proficiency in utilizing Abaqus but also contributes to a deeper understanding of the complexity involved in virtual manufacturing simulations. The acquired knowledge can be instrumental in refining and optimizing future simulations for more accurate and reliable results.

Different assumptions during the model definition focus the solution of the problem on different aspects of the process leading to various results and grades of accuracy. However, the practical utility of this in-house model, in the author's opinion, is significantly limited, primarily due to the extended run time required for simulations.

10 Bibliography

References

- [1] O Ayer and O Kaya 2021 J. Phys.: Conf. Ser. 2130 012021
- [2] Learn Casting and Solidification with Altair Inspire CastTM
- [3] A. Venkatesan, V.M. Gopinath, A. Rajadurai, Simulation of casting solidification and its grain structure prediction using FEM,
- [4] S. R. CHANG, J. M. KIM and C. P. HONG, Numerical Simulation of Microstructure Evolution of Al Alloys in Centrifugal Casting
- [5] Khan, M.A.A. Sheikh, A.K. Comparative study of simulation software for modelling metal casting process.
- [6] Finite element modelling of solidification phenomena K N SEETHARAMU1, R PARAGASAM1, GHULAM A QUADIR1, Z A ZAINAL1, B SATHYA PRASAD2 and T SUNDARARAJAN2
- [7] Investigation of the Solid-Liquid Phase Change in the Presence of Gas Phase: Numerical Modeling and Validation Ravi Govindram Kewalramani, Sebastian Pose, Ingo Riehl, Tobias Fieback
- [8] A. D. Brent, V. R. Voller, and K. J. Reid, "Enthalpy-porosity technique for modeling convection-diffusion phase change: Application to the melting of a pure metal," Numerical Heat Transfer, vol. 13, pp. 297-318, 1988.
- [9] Two-Phase Modeling of Mushy Zone Parameters Associated with Hot Tearing IVAR FARUP and ASBJØRN MO
- [10] Micro-macro model for the transient heat and fluid transport in solidification structure evolution during static casting processes Chijioke P. Egole a,* , Henry E. Mgbemere b , Gbeminiyi M. Sobamowo c , Ganiyu I. Lawal b
- [11] Free surface Navier-Stokes flows with simultaneous heat transfer and solidification/melting K. A. Pericleous, M. Cross, G. Moran, R Chow and K. S. Chan
- [12] Finding Boundary Conditions: A Coupling Strategy for the Modeling of Metal Casting Processes: Part I. Experimental Study and Correlation Development MICHAEL TROVANT and STAVROS ARGYROPOULOS
- [13] Finding Boundary Conditions: A Coupling Strategy for the Modeling of Metal Casting Processes: Part II. Numerical Study and Analysis MICHAEL TROVANT and STAVROS ARGYROPOULOS
- [14] A computational phase transformation model for selective laser melting processes Isabelle Noll1 · Thorsten Bartel1 · Andreas Menzel1,2

- [15] : Henry Hu and Stavros A Argyropoulos 1996 Modelling Simul. Mater. Sci. Eng. 4 371
- [16] Fixed-Grid Modelling of Solid-Liquid Phase Change in Unstructured Meshes Using Explicit Time Schemes Short title: Fixed-Grid Modelling of Solid-Liquid Phase Change P.A. Galionea,c, O. Lehmkuhla,b, J. Rigolaa , A. Olivaa,
- [17] A Volume-Averaged Two-Phase Model for Transport Phenomena during Solidification J. NI and C. BECKERMANN
- [18] MODELING FLUID STRUCTURE INTERACTION FOR AEROSPACE APPLICATIONS Andrey A. Aksenov Kirill A. Iliine, and Vladimir V. Shmelev
- [19] Fluid-structure interaction: Extended-FEM approach to solidification Daniela Caraeni , Vincent Casseau , Wagdi G. Habashi
- [20] A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies M. Coquerelle a,b, G.-H. Cottet a,
- [21] An overlapping domain technique coupling spectral and finite elements for fluid–structure interaction, A.C. Verkaika,, M.A. Hulsenb, A.C.B.Bogaerds c, F.N. van de Vosse
- [22] Eulerian Solid-Fluid Coupling Yun Teng David I.W. Levin† Theodore Kim‡
- [23] Efficient thermo-mechanical model for solidification processes Seid Koric,† and Brian G. Thomas‡
- [24] A fixed-mesh method for incompressible flow–structure systems with finite solid deformations Hong Zhao a , Jonathan B. Freund a,b,, Robert D. Moser
- [25] Interpolation functions in the immersed boundary and finite element methods Xingshi Wang · Lucy T. Zhang
- [26] Treatment of discontinuous interface in liquid-solid forming with extended finite element method ZHOU Ji-ming 1,2, QI Le-hua 1,2
- [27] Flow modelling in casting processes Paul Cleary a,, Joseph Ha a, Vladimir Alguine b, Thang Nguyen b
- [28] ALE formulation for Fluid structure interaction problems M. Souli a,1, A. Ouahsine b,* , L. Lewin
- [29] AN ENTHALPY METHOD FOR CONVECTION/DIFFUSION PHASE CHANGE V. R. VOLLER M. CROSS AND N. C. MARKATOS
- [30] A precise time-domain expanding boundary element method based on enthalpy formulation for non-isothermal phase change problems Zihao Wang, Weian Yao , Xiaofei Hu

- [31] Shadloo, M. S. Le Touzé, David Oger, Guillaume. (2016). Smoothed particle hydrodynamics method for fluid flows, towards industrial applications-Motivations, current state, and challenges.
- [32] Mathematical modeling and experimental measurements of shrinkage in the casting of metals, Michael Trovant and Stavros A. Argyropoulos
- [33] Numerical simulation of filling and solidification of permanent mold casting, Sergey V. Shepel, Samuel Paolucci
- [34] numerical simulation of the aluminium alloy solidification in complex geometries, Elisieu Monteiro, Abel Rouboa
- [35] Numerical Simulation of Two-Phase Free Surface Flows Alexandre Caboussat
- [36] Computation of Solid-Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids H. S. Udaykumar, R. Mittal,† and Wei Shyy
- [37] Numerical Modeling of the Solidification Phase Change in a Pipe and Evaluation of the Effect of Boundary Conditions A. Jalali and A.F. Najafi
- [38] Issues in Thermal-Mechanical Modeling of Casting Processes Brian G. THOMAS
- [39] Simulation of casting solidification and its grain structure prediction using FEM A. Venkatesana,, V.M. Gopinathb, A. Rajaduraib
- [40] Heat Transfer Model of Directional Solidification by LMC Process for Superalloy Casting Based on Finite Element Method LIU CAO, DUNMING LIAO, YUZHANG LU, and TAO CHEN
- [41] Finite element modelling of solidification effects in mould filling K. Ravindran, R.W. Lewis
- [42] Casting Simulation of Air Disc Brake, Shankar V Patil Shivakumar D Malagi
- [43] <https://www.magmaflow.com/en/solutions/magmaflow/>
- [44] <https://www.esi-group.com/products/casting>
- [45] <https://www.altair.com/products/product-pages/inspire-cast>
- [46] <https://www.flow3d.com/modeling-capabilities/>
- [47] Particle-based Ice Freezing Simulation Yunbin Miao Shanghai Jiao Tong University Shuangjiu Xiao Shanghai Jiao Tong University
- [48] PHASE-FIELD SIMULATION OF SOLIDIFICATION1 W. J. Boettinger, J. A. Warren,1 C. Beckermann, and A. Karma
- [49] ABAQUS/CAE user's guide
- [50] ABAQUS user guide https://help.3ds.com/2021/english/dssimulia_established/SIMACAEANLRefMap/simaanl-c-expdynamic.htm?contextscope=all

- [51] ABAQUS user guide https://help.3ds.com/2021/english/dssimulia_established/SIMACAEMATRefMap/simamat-c-eos.htm?contextscope=all#simamat-c-eos-idealgas
- [52] ABAQUS guide coupled thermal-stress analysis https://help.3ds.com/2021/english/dssimulia_established/SIMACAEANLRefMap/simaanl-c-couptempdisp.htm?contextscope=all
- [53] ABAQUS user guide https://help.3ds.com/2021/english/dssimulia_established/simacaecaerefmap/simacae-t-lbibceditorseulerianbc.htm?contextscope=all&highlight=eulerian+boundary+condition&id=&analyticsContext=search-result&analyticsSearch=eulerian%20boundary%20condition&myapp=false
- [54] Viscosity of high crystal content melts: Dependence on solid fraction Antonio Costa
- [55] Modelling of Thermal Gap Conductance in Casting using Finite Element Analysis Shukri Afazov
- [56] Altair guide for Inspirecast https://2022.help.altair.com/2022.2/cast/en_us/topics/cast/analysis/result_types_r.htm
- [57] ABAQUS guide https://help.3ds.com/2021/english/dssimulia_established/simacaeanlrefmap/simaanl-c-euleriananalysis.htm?contextscope=all&highlight=eulerian+multi+material&id=&analyticsContext=search-result&analyticsSearch=eulerian%20multi%20material&myapp=false
- [58] ABAQUS guide, thermal equation https://help.3ds.com/2021/english/dssimulia_established/simacaeanlrefmap/simaanl-c-couptempdisp.htm?contextscope=all&highlight=thermal+scheme&id=&analyticsContext=search-result&analyticsSearch=thermal%20scheme&myapp=false
- [59] Abaqus guide FEM https://help.3ds.com/2021/english/dssimulia_established/SIMACAETHERefMap/simathe-c-deformation.htm?contextscope=all&contextscope=all
- [60] Software Simulation and Experiment for Production of a Pillow Block Using Sand Casting Process Raharjo et al.
- [61] Simulation and validation of high-pressure die casting of an electric motor housing for automotive application Gianluigi Lanzoni