Article

On Numerical Simulation of Casting in New Foundries: Dynamic Process Simulations

Amir M. Horr * and Johannes Kronsteiner ©

LKR Light Metals Technologies, Austrian Institute of Technology, 1210 Vienna, Austria; Johannes.Kronsteiner@ait.ac.at
* Correspondence: amir.horr@ait.ac.at; Tel.: +43-0-50550-6918

Received: 10 June 2020; Accepted: 29 June 2020; Published: 3 July 2020

Abstract: New and more complex casting technologies are growing, and foundries are using innovative methods to reduce cost and energy consumption and improve their product qualities. Numerical techniques, as tools to design and examine the process improvements, are also evolving continuously to embrace modelling of more dynamic systems for industrial applications. This paper will present a fresh approach towards the numerical simulation of dynamic processes using an evolving and dynamic mesh technique. While the conventional numerical techniques have been employed for these dynamic processes using a fixed domain approach, the more realistic evolving approach is used herein to match the complex material processes in new foundries. The underpinning of this new dynamic approach is highlighted by an evolving simulation environment where multiple mesh entities are appended to the existing numerical domain at timesteps. Furthermore, the change of the boundary and energy sources within casting process simulations have rationally been presented and its profound effects on the computational time and resources have been examined. The discretization and solver computational features of the technique are presented and the evolution of the casting domain (including its material and energy contents) during the process is described for semi-continuous casting process applications.

Keywords: casting process; dynamic systems; process simulation; evolving domain; dynamic mesh

1. Introduction

One of the main interests of new foundries is to have reliable and optimised material processes where final quality of products, their lower costs and higher production rates are guaranteed. The power of new numerical techniques for simulating and optimising these processes have already been exploited to avoid costly trial and error cycles. These techniques would open up opportunities for foundries to optimise and tailor their manufacturing process parameters for the required product quality and speed without resorting to a large number of costly experimental works and pilot processes. The numerical process simulations’ aim is to provide a tool for improving and streamlining production developments through process-based predictive modelling. The most crucial issue related to these modelling schemes for material process applications is to produce accurate results in a reasonable computational time while handling the intrinsic material evolution during these processes for realistic predictions.

The introduction of the Finite Element (FE) technique in the second half of the 20th century and its subsequent process simulation tools have revolutionized the world of material engineering design and manufacturing. The concept of splitting and discretizing a complex process domain into simple sub-spaces (so-called grids or elements), where their approximate solutions can be presumed and then assembled into algebraic system of equations, has motivated many applications among material scientists [1–3]. Although the dynamic nature of many foundry processes, where parts are gradually generated in time, can inevitably produce some challenges for these fixed-domain techniques, some
simplified pseudo-dynamic techniques (e.g., element activation technique) have been employed for material processes. As the need for more accurate simulation of these dynamic processes increased and the bar has been raised for shorter computational time (especially for practical and industrial cases) various types of dynamic mesh techniques have been proposed to handle the domain changes during solutions.

The mainstream methodologies within popular foundry process software are either; to represent growing casting domains using an initially deactivated mesh with continuous element activation (at time steps) or; by generating new element layers at physical boundaries by splitting elements [4–13]. The other popular method is to start the casting process simulations with computationally deactivated elements (also called inactive element method) where only activated elements would be employed in matrices’ operations [14]. There are some other techniques which have been developed for Computational Fluid Dynamics (CFD) and free and dynamic boundary problems [15–17] to account for domain evolution and melt flow. However, these methods are either computationally expensive due to large size matrices for practical casting applications or computationally difficult to employ within popular mainstream industrial casting software tools.

The new approach towards simulation of foundry casting processes has recently been proposed using innovative dynamic mesh generation and process-controlled solver restarts [18–23]. The concept of zonal mesh evolution, where a block of new elements is inserted into the growing numerical domain at restarts, is implemented and a scheme for material and energy input/balancing is developed [18]. The application of the technique for industrial semi-continuous casting and Additive Manufacturing (AM) processes have already been investigated where the dynamically growing domains are modelled using appended mesh. The main attractive features of this new technique are that it can treat the growing casting domain with insertion of new element blocks (i.e., starting with small system matrices like inactive element method), while agile enough to be implemented into mainstream industrial software (in this case LS-DYNA® solver version R10.1.0 by LSTC, Ansys Inc, Canonsburg, PA, USA).

2. Dynamic Simulation Framework

The application of numerical simulation tools for material process chains and their industrial applications have long been promoted with an aim to improve process and product quality, lowering cost and increasing energy efficiency. Casting processes inherently involve multi-phase and multi-physical phenomena (at different length scales) which inevitably raise challenges for their numerical simulations. A number of different simulation frameworks, including separate fluid and mechanical, coupled fluid–mechanical and sequential CFD-mechanical simulations have been proposed by researchers to take into account the multi-physical/phase nature of the casting processes. Increasing computing power and parallel computing have already impacted these process simulations in which it is now possible to include the micro evolution of material during solidification (e.g., grain formation) and to optimize the process (and/or material) parameters for foundry applications. However, some aspects of these dynamic material processes are still required to be upgraded in order to match today’s foundry technologies. One of the main concerns for simulation of these applications is how to represent the growing numerical domain and its corresponding change of the algebraic equations in such way that it would be compatible with existing solver technologies.

As stated earlier, a number of different techniques have broadly been developed for the meshing and discretization of numerical domains associated with different FE and CFD solution schemes for material process applications [18], namely; fixed Lagrangian, updated Lagrangian, pure Eulerian, Arbitrary Lagrangian–Eulerian (ALE) and Mixed Lagrangian–Eulerian (MILE). These solution techniques have extensively been used for simulation of melt flow, casting process and forming applications. In Lagrangian approach, the variation of material confined by the numerical grid is solved at different points in time (i.e., time steps), while for pure Eulerian approach, the evolution of the material flow properties at every point in space, as time varies are recorded. For the simulation of dynamic systems like casting processes, as the casting of billet advances during the process, domain properties may
change in time. The traditional Lagrangian approach towards these numerical solutions is to describe the entire material flow (tracking) by assembling detailed transient histories of each grid cell. While in the Eulerian approach material flow properties are recorded at specified locations depending on their spatial location and time (advection).

Both of these methods have some deficiencies for dynamic process simulations with continuous domain evolution. Subsequently, hybrid and mixed techniques which combine the best features of both solution techniques have rigorously been developed to handle the domain changes. In these techniques, part of the computational mesh may be moved with the material front (i.e., in a Lagrangian fashion for casting), or is fixed in Eulerian manner or even moves in some arbitrarily specified way to give a continuous re-zoning capability. For the casting applications with moving mesh boundaries, different dynamic mesh approaches (e.g., mesh layer splitting, mesh insertion, etc.) have been proposed to trace the elongation of a discretized domain during the simulation [18–21]. Although, these new techniques have already shown some promising results for casting applications, some of the mainstream industrial software tools are unable to implement them (e.g., still based on old activation scheme).

To start the description of the new dynamic mesh technique, let us consider a typical vertical casting process with continuous or semi-continuous nature for aluminium alloys (shown in Figure 1). The process is characterized by its continuous evolution of a cast billet due to solidification of melt within the mould.

Figure 1. Schematic vertical aluminium casting process with its cooling strategy for; (a) start of casting process; (b) initial water cooling; (c) steady state continuous water cooling and; (d) combination of water and secondary air cooling.
The casting starts with filling of the mould along with the initial cooling process. The starting head would then start to move downwards which elongates the cast billet according to the casting speed. The heat energy within the billet is reduced by initial water impinging (to avoid cracking) which is quickly turned into continuous water impingement after a transient period. The trailing water at certain locations on the billet length is wiped off by a specially designed wiper system and the following secondary air cooling is carried out for slower cooling rates. To include these evolutions during the casting process, a directional boundary evolution scheme has been implemented to dynamically generate thermal boundaries and mesh blocks. The technique can be described in the following steps:

- The initial mould filling and primary thermal boundary conditions are modelled using the results of either thermal-CFD melt delivery simulation or thermal boundary assumptions.
- The initial geometry of the main domain (i.e., mould, starting head) and its discretization are defined and initial system matrices are assembled.
- The initial time steps/iterations are solved (using a thermal-mechanical simulation) and based on the casting speed the billet domain coordinates are updated.
- The new discretized grid and its boundaries are adapted by adding a single/multiple layer ‘element-block’ based on the evolution of the domain in vertical direction (including updating coordinates and accounting for shrinkage and thermal deformation).
- The solver is re-started and the system matrices are modified/appended while the input energy is distributed amongst the newly generated grid.
- Thermal time history results (from previous time steps) are mapped to the updated grid and thermal boundaries are updated.
- The energy balance (thermal energy) is achieved through the energy sink/source concept and the previous converged solution is used as a first step for a newly updated domain.
- The simulation scheme continues with the new discretised grid, till the next evolution step is triggered.

To handle the thermal-mass issues during casting, the water jet or spray-impingement cooling schemes are common practices in mainstream foundry processes (i.e., continuous and semi-continuous casting processes). Hence, the numerical simulation of cooling rates and thermal boundaries during casting processes are essential. The process of cooling and its modelling schemes are established to achieve a deeper understanding of the thermal evolution and temperature fields within the cast billet (including thermal stresses) during industrial processes. For the simulation purpose, the melt flowing out of the mould can be considered as a viscous liquid which is confined in a thin solidified shell, formed during the initial cooling process (e.g., mould contact-based and/or convective air cooling schemes). Various techniques have already been developed for the modelling of cooling processes which can be summarised as:

- Pure experimental methods where results of experimental trials are used (and sometimes extrapolated) for Heat Transfer Coefficient (HTC) calculations using a set of process conditions (water flow rates, billet surface temperatures, etc.).
- Analytical-experimental technique where analytical and empirical parameters are fitted using results of carefully designed experimental trials.
- Numerical techniques based on CFD simulation where bubble dynamics and flow regimes are considered using multi-phase fluid techniques.
- Hybrid numerical-analytical-experimental techniques where analytical techniques (using experimentally fitted parameters) are used to enhance the multi-phase CFD simulation for HTC calculation. The accuracy and agility of these hybrid techniques can be improved with the help of Machine-Learning (ML) and Artificial Intelligence (AI) for data fitting and extrapolation.

The initial trials of these hybrid cooling techniques have shown some promising results for impingement cooling simulations, although many more in-depth research works have to
be conducted to prove their potential capabilities [24]. In the research work herein, a hybrid numerical-analytical-experimental technique has been initiated to calculate the required HTC for casting simulations [22,24]. A parallel analytical-experimental scheme has firstly been carried out to develop a modified Chen model [25] which can be employed during CFD simulations. In the second phase, the modified analytical model has been implemented in the CFD software (Ansys CFX, version 17.0, Ansys Inc, Canonsburg, PA, USA) for the calculation of cooling curves [26]. Recently, some efforts have also been initiated to improve the accuracy of the result using Genetic-Algorithm Symbolic Regression (GASR) techniques [27] where multi-dimensional data fitting capabilities are used to improve analytical modelling [24]. The underlying research is aimed at bringing together the physical and analytical cooling concepts with data process capabilities of ML and AI for cooling regimes. The combination of physical, analytical and smart data-driven methods can essentially be used to find an optimised multi-parameter solution for the analytical formulation of flow regime and bubble dynamics during cooling. The GASR technique has enthusiastically been employed to form a multi-dimensional search space developed for parameter fitting. In the recent work, the calculation of bubble drag forces during the impingement cooling on the hot billet surface has been calculated using insertion of scenario-based micro CFD simulation results into a smart GASR tool [24]. Figure 2 shows the schematic features of the proposed hybrid cooling framework.

![ML-Based Hybrid Numerical-Analytical-Experimental Framework](image)

**Figure 2.** Hybrid numerical-analytical-experimental framework for cooling process and HTC modelling.

Although some analytical and numerical approaches for cooling systems during material processes have already been developed, the number of parameters and their definitions into real-world processes has always been awkward. The numerical approaches for cooling processes are mostly based on multi-phase fluid evolution (e.g., boiling regimes), its flow dynamics and contact between the coolant and the hot surfaces. However, the computational time and resources required for the detailed numerical simulations of multi-phase cooling processes are beyond the tolerable limits of the practical industrial applications. Hence, the combination of simpler analytical models with fitting power of smart data-driven techniques (for parameters) and measured data from a limited number of carefully set-up experimental trials have been used herein to calculate the required HTC values. To help simplify the analytical cooling scheme, a hybrid modelling technique is used to combine the physical equations with data-fitted terms. These data are obtained using micro-scale CFD simulations (e.g., for bubble...
dynamics) and/or experimental measurements. The hybrid technique has willingly been used herein to form a practical HTC estimation tool for the transient casting process simulation [24].

3. Dynamic Casting Simulation

To resemble conditions during foundry processes, several dynamic and transient simulation techniques have recently been employed and special attention has been paid to the multi-physical/scale nature of these processes. For casting simulations, many of the conventional computational techniques, developed for other branches of engineering sciences, have also been adapted and employed for the modelling and optimisation of these processes. Combinations of state-of-art innovative material models (including continuous and discrete material models), reliable numerical methods and high-performance computing tools can be used to explore what technical solutions are viable for casting processes. These combined schemes can also be enhanced by employing data-driven modelling where ML and AI fitting and learning capabilities are used to imitate the essence of whole industrial casting processes in modern foundries.

The computational aspects of dynamic generative systems like; casting processes, for full-scale industrial applications is revitalize herein using the concept of evolving domain and dynamic mesh technique [18]. The initial starting point for the proposed Dynamic Mesh Technique (DMT) development was the idea to introduce an evolving domain technique where new mesh blocks and boundaries can be adapted into mainstream FE tools for casting applications. The method could be linked into a CFD mould filling simulation for initial thermal boundary definitions. Although some Euler-based CFD simulations including moving mesh boundaries and splitting/collapsing mesh techniques have already been employed for dynamic systems, the link to Lagrangian thermal-mechanical simulations was not established. The proposed DMT approach would treat the evolving/progressing cast billet during the casting simulation as appended mesh blocks. A mesh block can be appended in predefined or calculated manner which would be attached to the main domain through overlapping boundary concept [18].

For the casting application of the technique (vertical or horizontal), a directional boundary evolution scheme (i.e., mesh block along the cast billet direction) has been implemented using an in-house Python script. The required steps for DMT can be summarized as:

- The initial thermal boundaries and cooling rates are defined using the results of either CFD filling simulations or analytical-experimental thermal boundary assumptions.
- The initial geometry of the main domain (i.e., starting head, mold) and its discretization are carried out where the boundaries are defined and initial system matrices are assembled.
- The FE solver’s initiation time steps/iterations are carried out (using both thermal and mechanical solvers) and nodal coordinates for the billet domain are updated.
- New mesh block is appended based on the evolution of the billet domain (with casting speed) and the domain discretized grid is modified and adapted for shrinkage and thermal deformation changes.
- The solver global system matrices are updated and the material/thermal input energies are distributed amongst the newly generated mesh block.
- The mapping process is carried out at new boundaries by adapting time history results (from previous time steps).
- The thermal energy balance is reached through a manual/automated energy sink/source scheme and the previous converged solution is used as a first step for a newly updated domain.
- The FE simulation continues with the new mesh/boundaries till the next evolution step is triggered.

To assess the performance of the DMT technique for full-scale industrial casting applications, some pilot studies have been conducted to determine the speed and agility of the technique for semi-continuous casting applications. As the numerical solution is performed on a full parallel-processing machine, a secondary partitioning scheme (decomposition) is used to distribute
the discretized domain over parallel computational nodes. The generated mesh zones would become part of the main domain as both mechanical and thermal solvers are initializing at restart points. The time history mapping and interface procedures between the existing domain and newly generated mesh-zone (with similar or different resolutions) can be represented using mathematical overlapping techniques [18]. Figure 3 shows two numerical domains with their discretized meshes for conventional “Block deactivation and activation Technique” (BT) and new DMT where both domains are meshed with equal sized elements (for double-symmetric quarter models).

![Figure 3](image)

**Figure 3.** (a) Conventional BT model with its mesh-blocks for stepwise activation; (b) DMT mesh model with its continuous appended mesh layers at 1.4 m billet length; and (c) graphical algorithm representation for DMT implementation using Python script.

For the conventional BT, the same geometry and mesh size are adapted and stationary thermal boundaries are defined using the same cooling rate and HTC data. However, in contrast to the DMT technique, the whole length of the simulated cast billet is initially meshed and the thermal contacts are setup for the full-length billet. The common strategy here is to deactivate the billet elements at the start of simulations (all elements except initial mold filling elements) and reactivate them (block-wise) as the simulation of the dynamic process is proceeding. Hence, there is no need of restarting solvers for this technique, as the system matrices are already assembled using all elements in the numerical domain. It is well understood that conventional deactivation processes would not remove elements from assembled matrices, rather it reduces their numerical impacts to very small values (e.g., using very small multipliers). For the proposed DMT, mesh blocks are appended into the main domain at solver restarts where effects of new material and its thermal energy content (at melt temperature) are mapped and distributed using the balancing technique into the entire numerical domain [18,24].

4. Industrial Case Study

To study the application of the proposed dynamic method for real-size industrial cases, a semi-continuous aluminum casting process has been adapted for a numerical simulation. Since the main interest in the simulation is to estimate the thermal field and mechanical behaviour of cast billets (e.g., stress/strain, cracking, etc.), the multi-physical thermal-mechanical solver was a viable choice. The thermal evolution and the influence of temperature on the mechanical properties could
carefully be investigated, while the evolution of the domain could be handled using the dynamic mesh technique. The preferred FE platform for the work herein is LS-DYNA® which can handle heavy parallel computing along with multi-physical solutions.

As stated earlier, the conventional approach for dynamic systems on these FE platforms is to pre-mesh the whole cast domain (since the final shape of the domain is known) and to activate elements during the simulation. The CPU computational time and memory storage issues are the main drawbacks of such an approach which restricted the application of the technique for practical industrial casting simulations (especially for limited in-core computer storage space). Hence, for continuous casting applications, gradual extension of the numerical domain during the calculation could avoid solving large matrices for all time steps and save computer resources during the solution. Accordingly, for these large industrial applications, the efficiency of the new technique has to be broadly investigated before it can be employed in the simulation of foundry processes.

As a starting point for this case study, an aluminium semi-continuous casting simulation using conventional activation technique has been carried out (a double-symmetric quarter model). However, instead of activating single elements during the process (which is computationally not viable), elements are placed in a block formation for step-wise activation. This so-called block activation scheme attempts to mimic element block insertion in order to compare the performance with the new DMT. The main difference between conventional BT and block activation scheme, used here, is that these blocks of elements are connected to each other by thermal-mechanical contact elements instead of node sharing. The mesh blocks were “activated” gradually based on the casting speed by initializing a predefined melt temperature on nodes and increasing the heat conductance between the existing mesh domain and the newly activated block. This means that a steady thermal contact between all element layers of the billet domain are triggered and balanced.

The block mesh activation methodology described above was used for a rigorous comparative study with the proposed DMT. For the DMT, element blocks (or mesh layers) are directly appended to the billet domain during discrete solvers restarts. The nodal temperatures for all new elements are initialized to the temperature of the melt pool. The remaining billet mesh is initialized by mapping the results from the last restart onto the new mesh. After the definition of thermal and mechanical boundary conditions and the adjustment of the top nodes’ row to the defined melt pool level, a new simulation step can be started. The whole simulation procedure is controlled by an in-house developed Python™ code which handles the manipulation of LS-DYNA® input files, the adding of element blocks and all relevant tasks prior to restart of the solver. Hence, some aspects of the block activation technique are similar to the new DMT, with some major differences:

- For the conventional technique, all element blocks (and their contact elements) already exist in the domain from the start of the simulation.
- New mesh blocks are individual parts for the conventional BT which are attached to rest of the domain using contact elements while for DMT elements are added to the already existing billet domain.
- Since new blocks are individual parts in BT, the contact elements need to be setup at the start of simulation to enable heat conductance and mechanical contact during simulation.
- Activated element blocks in conventional BT consist of 3 layers of elements (to limit the CPU time) compared to higher-resolution single layer of element insertions for the DMT.

The starting point for transient simulations in both techniques is an initial temperature distribution within the pre-filled starting head. The melt filling has been simulated in a similar way by gradually adding element layers with predefined mesh size until the starting head is full. After the setup of thermal boundary conditions (based on HTCs), the transient temperature distribution of the billet elements as well as the starting head could be calculated. The extension of the numerical domain for casting application would gradually increase the thermal mass of the system, which needs to be taken into account by heating the elements to the specified temperature of the melt before establishing
the contact with the existing billet elements. Figure 4 shows a comparison of strategies for both conventional block activation and DMT mesh techniques from the start to the end of process. It should be noted that for the conventional BT, all mesh blocks should be created on the top of the starting head before deactivation. These mesh blocks would move down with the starting head as deactivated body meshes (with minimal physical effects). The activation process of these mesh blocks is triggered by their vertical position with respect to the melt level. This is contrary to DMT mesh generation strategy, where layers of meshes are appended to the domain always at the melt level during solver restarts (no mesh blocks are present on top of the starting head at the start of simulation).

In Figure 4a, the de-activated parts (blocks of elements) are clearly visible in dark blue (cold) color, while there are no such additional parts in Figure 4b for DMT. During the progression of the casting process, the blocks are activated and move downwards until they reach the final position at the end of the process, seen in Figure 4a using transparent colors. In contrast to the conventional BT, the element rows are directly added to the billet top part in the case of the DMT in Figure 4b.

A comparison of temperature results for the semi-continuous aluminum casting case is shown in Figure 5 where results of quarter-model cast billet (due to double-symmetric condition) are shown at the end of simulation. It should be noted that due to differences in mesh insertion/activation strategies (mesh block with multi-layer elements for conventional BT versus higher-resolution single layer mesh for DMT) and thermal contact activation strategies, the results are not identical for both techniques. However, the results from both techniques mostly look similar except for a difference of how the temperature is transferred into the billet center. In the case of the conventional BT in Figure 5a, the elevated temperatures only reach half of the billet length, while in the case of DMT, elevated temperatures are extended to the billet lower half as well. For the conventional BT model, heat conductance inside of one block seems to transfer more thermal energy compared to contacts (with a
pre-defined HTC of the same value) used for thermal conductivity within DMT model. Hence the depth of the melt pool for conventional BT is shallower than DMT. However, despite differences on how the temperature is transferred into the billet center, the results from both techniques largely look similar.

Figure 5. Comparison of temperature results between; (a) BT; and (b) DMT for 1.4 m billet length.

5. Computational Performance

The challenge of simulating industrial-scale continuous and semi-continuous casting processes with their long billet sizes (and also time-dependent generation scheme based on pre-defined casting speed) has eagerly prompted the creation of new ideas and techniques. As stated earlier, one of the main burdens of these new techniques is the computational time and resources. Although, the recent progresses in computer science and engineering would provide a vast amount of computing power and resources, the task of achieving simulation results with accuracy and enough resolution is still daunting. Many different schemes, including improving solver technologies, better parallelization, and higher order element formulation are used to address these deficiencies for the simulation of modern foundry processes, with limited success. The fact is that for dynamic systems, it is essential to have a strategy to evolve the numerical domain according to real pace of manufacturing process which ultimately delivers a dynamic and growing computational arena. The proposed DMT technique herein, can deliver such a method where faster and more accurate results can be achieved using growing and evolving domain concepts. The computational resources for this technique are commissioned gradually as the real billet is cast and the size of the domain is growing. Although, this comes at the cost of restarting solvers at discrete time steps where new assembly of matrices and extra Input/Output (IO) activities are inevitable. To assess the computational efficiency of the DMT technique against the conventional BT, a through and comprehensive numerical investigation has been carried out to evaluate the aptness of the technique for real foundry semi-continuous aluminum cast processes. Table 1 shows the casting process parameters for both simulation techniques while Table 2 shows a scenario table designed to assess the computational time and resources for both techniques, along with CPU computational times for all scenarios on a single computing node.
Table 1. Casting process parameters.

| Parameter                  | Value          |
|----------------------------|----------------|
| Melt Temperature [°C]      | 630            |
| Billet Width [m]           | 1.24           |
| Billet Thickness [m]       | 0.3            |
| Casting Speed [m s\(^{-1}\)] | 0.01         |
| Cooling Water Temperature [°C] | 20            |
| HTC- Air Cooling [kW m\(^{-2}\) K\(^{-1}\)] | 1.5 (average) |
| HTC- Water Cooling [kW m\(^{-2}\) K\(^{-1}\)] | 11 (average) |

Table 2. Computational scenarios for DMT and BT techniques using a single compute node with 16 cores.

| Scenario No. | Billet Length [m] | No. of Elements | CPU Time DMT [s] | CPU Time BT [s] | IO Time DMT [s] | IO Time BT [s] | CPU Ratio DMT/ BT |
|--------------|-------------------|-----------------|------------------|----------------|----------------|----------------|------------------|
| S1           | 0.5               | 27,189          | 16,345           | 20,581         | 78.62          | 6.90           | 79%              |
| S2           | 1                 | 41,357          | 45,577           | 77,387         | 179.18         | 7.02           | 59%              |
| S3           | 1.4               | 59,573          | 84,545           | 163,063        | 271.84         | 13.77          | 52%              |

Different meshing and contact strategies were used to carry out casting simulations for DMT and BT. While both techniques have used contact elements to simulate the cooling boundaries (air and water cooling), for BT, the contacts had to be setup for the full length of the billet. Alternatively, for DMT, as new mesh-layers are generated and attached to the main domain at solver restarts, contact setups needed to be updated. Figure 6 shows the schematic setup for the thermal contact boundaries along the billet length during the vertical casting process. The HTC values for the cooling process can be calculated from a parallel numerical-analytical-experimental framework presented earlier in the manuscript.

Figure 6. Thermal boundary setup for vertical casting process, including its air and water cooling strategies.

5.1. Computational Efficiency and CPU Timing

To investigate the computational efficiency of the DMT on a parallel computing platform, all proposed scenarios in Table 2 were simulated for both DMT and BT models on different numbers of
computing nodes. Three geometries and their corresponding structured meshes were generated for the billet lengths 0.5 m, 1 m, and 1.4 m, accordingly. The technical specifications of the cluster nodes are given in Table 3.

**Table 3.** Specifications of computational nodes and software used for all simulation scenarios.

| CPU Name          | No. of Sockets | Cores per Socket | Total Memory [MB] | Communication Between Nodes | Parallelization Scheme | LS-DYNA Release | Accuracy       |
|-------------------|----------------|------------------|-------------------|----------------------------|------------------------|-----------------|----------------|
| Intel Xeon E5-2687W v4 | 2              | 8                | 65,536            | InfiniBand                 | Platform MPI           | MPP R8.1.0      | Double precision |

The time-history results were post-processed to calculate the CPU computational times and IO commitments. Figure 7 shows the three-dimensional plot of computational efficiency against billet lengths and number of computational cores. The efficiency for this 3D graph is defined as the ratio of the total simulation time for each scenario over the base scenario simulation’s time where the base scenario is the DMT simulation of 1.4 m billet length using two parallel computing nodes (with 32 cores). Accordingly, the higher efficiency values shown in Figure 7 correspond to the lower total simulation times for casting scenarios.

**Figure 7.** Comparison of numerical efficiency for three billet lengths and different number of cores between DMT and BT.

It can be clearly seen how the billet length influences the computational efficiency and the impact of the parallelization on both BT and DMT. DMT can benefit much more than BT from heavy parallelization. Only in the case of 64 cores and only for the short billet length of 0.5 m, BT can beat DMT slightly. The biggest difference between BT and DMT, however, is in storage use. Figure 8 shows how the matrix size increases during a DMT run while a constant large matrix size is maintained for BT. The slightly larger matrix size for DMT at 1.4 m billet length is mainly due to differences in boundary and contact conditions.
Figure 7. Comparison of numerical efficiency for three billet lengths and different number of cores between DMT and BT. It can be clearly seen how the billet length influences the computational efficiency and the impact of the parallelization on both BT and DMT. DMT can benefit much more than BT from heavy parallelization. Only in the case of 64 cores and only for the short billet length of 0.5 m, BT can beat DMT slightly. The biggest difference between BT and DMT, however, is in storage use. Figure 8 shows how the matrix size increases during a DMT run while a constant large matrix size is maintained for BT. The slightly larger matrix size for DMT at 1.4 m billet length is mainly due to differences in boundary and contact conditions.

Figure 8. Comparison of matrix size during simulation over billet length between DMT and BT for 1.4 m billet length.

5.2. Computational Resources and Memory

One of the important computational aspects of parallel processing is the committed memory and communications between computational cores. There are different techniques for secondary partitioning of numerical domain to distribute and balance computational loads across the designated cores. Since the proposed DMT has the capability to add new blocks of mesh into the numerical domain using solver restarts, the solver has to carry out secondary partitioning regularly at insertion times. Figure 9 shows a comparison of element processing times for the mechanical and thermal solvers between DMT and BT for three billet lengths and three different parallelization settings.

Figure 9. Comparison of element processing time for; (a) mechanical; and (b) thermal solvers over billet length between DMT and BT for all three billet length scenarios.

Onl y in the case of 64 cores (3 computing nodes), BT is faster in element processing (in mechanical solver) than DMT. In the case of thermal element processing (Figure 9b), DMT is only faster for 16 cores. The benefit of having less elements does not affect the thermal case, because the elements are not activated anyway. IO time is the main drawback of the restarts’ processes, which have to be repeated during DMT. Figure 10 shows how the IO times increase with billet length and number of cores for the DMT. In BT, since IO is only necessary once, this issue is negligible.

Figure 10. Comparison of IO processing time over billet length between DMT and BT for all three billet length scenarios.
Only in the case of 64 cores (3 computing nodes), BT is faster in element processing (in mechanical solver) than DMT. In the case of thermal element processing (Figure 9b), DMT is only faster for 16 cores. The benefit of having less elements does not affect the thermal case, because the elements are not activated anyway. IO time is the main drawback of the restarts’ processes, which have to be repeated during DMT. Figure 10 shows how the IO times increase with billet length and number of cores for the DMT. In BT, since IO is only necessary once, this issue is negligible.

![Figure 10. Comparison of IO processing time over billet length between DMT and BT for all three billet length scenarios.](image)

The investigation of the IO times for parallel processing of dynamic systems becomes vital when it is considered that large numerical domains with huge assembled matrices are required for the simulation of industrial scale continuous and semi-continuous processes. The variation of IO times during these processes are also of interest since it can slow down the progression of computational tasks during the transient simulation. Figure 11 shows the variation of IO times during the simulation of the casting process for three billet length scenarios using DMT. Three different IO times are recorded and shown during the solver restarts; namely, key file reading time, decomposition time and initialization time. The summations of these three IO times are also shown in the Figure 11, where it does not show significant time variations between three billet length scenarios. The decomposition times (secondary partitioning for parallel processing) show steady values during the insertion of new element layers for all scenarios (at solver restarts), while there are variations on file reading and initialization IO times.
The investigation of the IO times for parallel processing of dynamic systems becomes vital when it is considered that large numerical domains with huge assembled matrices are required for the simulation of industrial scale continuous and semi-continuous processes. The variation of IO times during these processes are also of interest since it can slow down the progression of computational tasks during the transient simulation. Figure 11 shows the variation of IO times during the simulation of the casting process for three billet length scenarios using DMT. Three different IO times are recorded and shown during the solver restarts; namely, key file reading time, decomposition time and initialization time. The summations of these three IO times are also shown in the Figure 11, where it does not show significant time variations between three billet length scenarios. The decomposition times (secondary partitioning for parallel processing) show steady values during the insertion of new element layers for all scenarios (at solver restarts), while there are variations on file reading and initialization IO times.

Figure 11. Comparison of variation of IO processing time for DMT using four computing nodes (with 16 cores each) for billet lengths (a) 0.5 m; (b) 1.0 m; and (c) 1.4 m.
6. Discussion

The introduction of a dynamic approach using element insertion, mapping and restarting technique has been presented herein where a framework including a controlling module (using in-house Python code) has been developed for continuous and semi-continuous casting processes. The dynamic technique can be employed for the simulation of foundry processes with transient growing domains. One of the advantages of the new approach lies in its overall reduced computational efforts/resources and agility of implementation on commercial solvers. Detailed investigations of computational time and resources were carried out using a pre-defined scenario table to establish the computational efficiency of the technique against a popular conventional technique. The findings can be summarised as:

- Overall computation time (including CPU and IO times) are meaningfully lower for the proposed DMT compared to the conventional activation technique.
- The overall IO time for DMT with many solver restarts and repeated matrices’ assemblies is higher than the conventional technique (with single start of the solver). However, the total simulation wall clock time for DMT is significantly lower than the conventional technique due to the fact that the CPU computational time is generally more than two orders of magnitude larger than IO times for real-size casting applications.
- Considering today’s efficient existing partitioning and parallel processing routines implemented in commercial software, the IO times for decomposition of the numerical domain shows a steady trend over the different billet lengths. This means that with a carefully chosen number of computational nodes (e.g., based on the size of the final domain), the computing resources can be optimised for a manageable overall simulation time.
- The implementation of the technique and its flexibility towards enhancing commercial software capabilities for dynamic systems would inspire the use of the technique in simulation of modern foundry processes.
- Although the thermal-mechanical simulation of casting processes will provide a valuable source of information for foundry processes, the link to melt delivery and filling simulation would enhance the optimisation capabilities of these simulations. Hence, the next development step for the proposed technique is to establish a sound numerical link to pre-solidified melt flow and filling simulations.

Moreover, novel computer technology routines can be employed to shorten required IO time for the DMT solver’s initialization using embedded pre-mesh data which can be stored in the active computer memory for the FEM solver (instead of reading from disk) in the future.

7. Conclusions

One of the main challenges for today’s simulations of foundry processes is how to handle the dynamic nature of these processes using conventional multi-physical/scale solver technologies. The computational time’s affordability and accuracy of the simulation results also play a vital role for real-size industrial applications of these techniques. The variations of parameters and the transient nature of thermal boundaries (with dynamic HTCs) during these processes will also cause a huge challenge for industrial-based simulation tools. The dynamic technique developed herein is a first step towards the development and implementation of more ambitious numerical frameworks for faster and more practical industrial simulation tools. The proposed DMT for casting applications which has been presented in this paper is paving a way for easier implementation of dynamic generative systems into the mainstream software tools (LS-DYNA® for this research work).

Some technical aspects of the new DMT and its dynamic mesh strategy have shortly been presented and the industrial application of the method for the case of semi-continuous casting processes were discussed. In the final part of the paper, an in-depth comparative study on the computational performance of the technique against a popular conventional BT is presented. The issue of CPU time and committed computer resources for the simulation of casting applications have briefly been
scrutinized, and the potential saving of the computational time and memory for industrial-scale applications were highlighted. Although the numerical framework can easily be adapted to various geometries, its application for full continuous casting processes has not yet been examined. The link between melt delivery/filling simulations and thermal-mechanical DMT has also not been fully established yet, and these are potential next steps which are planned for the continuation of the research work.

**Author Contributions:** The simulation of test scenarios along with partial writing of the manuscript has been carried out by J.K. where number of simulations on different size numerical models have been performed. He has also completed the post-processing of the numerical simulation results for presentation within the manuscript. The conceptual and fundamental technical arguments of the manuscript have been contributed by A.M.H., where basic theory and design of numerical framework have been elaborated. The draft preparation and editing of the manuscript have been shared by both authors. Both authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Land of Federal State of Upper Austria in the FD Framework (within P3HeRo Project OÖ Fin-010104/187).

**Acknowledgments:** Authors would like to thank the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), Land of Federal State of Upper Austria in the FD Framework (within P3HeRo project OÖ Fin-010104/187) and also the Austrian Institute of Technology (AIT) for the technical/financial support in this research work. Authors would also like to thank DI Stefan Scheiblhofer, a former employee of the LKR Light Metals Technologies, for his technical input and contributions into the manuscript. The final review and proofing of the manuscript have been carried out internally by the LKR.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Lee, J.-E.; Yeo, T.-J.; Oh, K.H.; Yoon, J.-K.; Yoon, U.-S. Prediction of cracks in continuously cast steel beam blank through fully coupled analysis of fluid flow, heat transfer, and deformation behavior of a solidifying shell. *Met. Mater. Trans. A* 2000, 31, 225–237. [CrossRef]

2. Koric, S.; Hibbeler, L.; Liu, R.; Thomas, B.G. Multiphysics Model of Metal Solidification on the Continuum Level. *Numer. Heat Transf. Part B Fundam.* 2010, 58, 371–392. [CrossRef]

3. Zappulla, M.L.; Cho, S.-M.; Koric, S.; Lee, H.-J.; Kim, S.-H.; Thomas, B.G. Multiphysics modeling of continuous casting of stainless steel. *J. Mater. Process. Technol.* 2020, 278, 116469. [CrossRef]

4. Brötz, S.; Horr, A. Framework for progressive adaption of FE mesh to simulate generative manufacturing processes. *Manuf. Lett.* 2020, 24, 52–55. [CrossRef]

5. Montevecchi, F.; Venturini, G.; Scippa, A.; Campatelli, G. Finite Element Modelling of Wire-arc-additive-manufacturing Process. *Procedia Cirp* 2016, 55, 109–114. [CrossRef]

6. Keller, N.; Neugebauer, F.; Xu, H.; Ploshikhin, V. Thermo-mechanical Simulation of Additive Layer Manufacturing of Titanium Aerospace structures. In Proceedings of the EURO LIGHTMAT, Bremen, Germany, 3–5 September 2013.

7. Keller, N.; Ploshikhin, V. New Method for fast predictions of residual stress and distortion of AM parts. In Proceedings of the Solid Freeform Fabrication Symposium, Austin, TX, USA, 4–6 August 2014.

8. Lindwall, A.L.J.; Lindgren, L.E. Thermal FE-simulation of PBF using adaptive meshing and time stepping. In Proceedings of the Simulation for Additive Manufacturing, Munich, Germany, 11–13 October 2017.

9. Lundback, A.; Lindgren, L.E. Modelling of metal deposition. *Finite Elem. Anal. Des.* 2011, 47, 1169–1177. [CrossRef]

10. Simunovic, S.; Nyicz, A.; Noakes, M.W.; Chin, C.; Oancea, V. Metal Big Area Additive Manufacturing: Process Modeling and Validation. In Proceedings of the Science in the Age of Experience, Chicago, IL, USA, 15–18 May 2017.

11. Tripathy, S.; Chin, C.; Oancea, V.; London, T.; Ankalkhope, U. Process Modeling and Validation of Powder Bed Metal Additive Manufacturing. In Proceedings of the NAFEMS World Congress, Stockholm, Sweden, 11–14 June 2017.

12. Brötz, S.; Scheiblhofer, S. *Prozesssimulation Drahtbasierter Additiver Fertigungsverfahren*; NAFEMS Online-Magazin: Munich, Germany, 2019; Volume 50, pp. 81–88.
13. Ludwig, O.; Marco, A. State of the Art in Modelling of Continuous Casting. In Proceedings of the 6th European Conference on Continuous Casting, Riccione, Italy, 4–6 June 2008.

14. Panagiotis, M. Modeling metal deposition in heat transfer analyses of additive manufacturing processes. *Finite Elem. Anal. Des.* 2014, 86, 51–60.

15. Claus, S.; Bigot, S.; Kerfriden, P. CutFEM Method for Stefan—Signorini Problems with Application in Pulsed Laser Ablation. *SIAM J. Sci. Comput.* 2018, 40, B1444–B1469. [CrossRef]

16. Bastian, P.; Engwer, C. An unfitted finite element method using discontinuous Galerkin. *Int. J. Numer. Methods Eng.* 2009, 79, 1557–1576. [CrossRef]

17. Fausty, J.; Bozzolo, N.; Muñoz, D.P.; Bernacki, M. A novel level-set finite element formulation for grain growth with heterogeneous grain boundary energies. *Mater. Des.* 2018, 160, 578–590. [CrossRef]

18. Horr, A. Computational Evolving Technique for Casting Process of Alloys. *Math. Probl. Eng.* 2019, 2019. [CrossRef]

19. Horr, A.M. Integrated material process simulation for lightweight metal products. *CMEM 2015* 2015, 1, 201–212. [CrossRef]

20. Jäger, S.; Horr, A.M.; Scheiblhofer, S. Thermal evolution effects on solidification process during continuous casting. In Proceedings of the 6th Decennial International Conference on Solidification Processing, Old Windsor, UK; 2017; Volume 2017, pp. 675–678.

21. Horr, A.M.; Kronsteiner, J.; Muhlstatter, C. Recent Advances in Aluminium Casting Simulation: Evolving Domains & Dynamic Meshing. In Proceedings of the Aluminium 2000 Conference, Verona, Italy, 20–24 June 2017.

22. Horr, A. Thermal Energy Approach for Aluminium Process Simulations: Source-Capacity Technique. *Mater. Today Proc.* 2019, 10, 263–270. [CrossRef]

23. Horr, A. On Analytical Concepts of Novel Multi-Resolution Casting Simulations. In Proceedings of the IOP Conference Series: Materials Science and Engineering; 5th Inter. Symposium on Cutting Edge of Computer Simulation of Solidification, Casting and Refining, Salzburg, Austria, 17–21 June 2019; Volume 529, p. 012055. [CrossRef]

24. Horr, A.M. Notes on New Physical & Hybrid Modelling Trends for Material Process Simulations. In Proceedings of the IOP conference Series, 4th International Conference on Applied Physics, Simulation and Computing, Rome, Italy, 23–25 May 2020.

25. Chen, J.C. Correlation for Boiling Heat Transfer to Saturated Fluids in Convective Flow. *Ind. Eng. Chem. Process. Des. Dev.* 1966, 5, 322–329. [CrossRef]

26. Mühlstätter, C.; Jäger, S. CFD zur Abbildung von Effekten in der Leichtmetallherstellung. In Proceedings of the 13. Cadfem ANSYS Simulation Conference Austria, Linz, Austria, 24–25 April 2018.

27. Wagner, S.; Affenzeller, M. HeuristicLab: A Generic and Extensible Optimization Environment. *Adapt. Nat. Comput. Algorithms* 2005, 538–541. [CrossRef]