TECHNISCHE UNIVERSITÄT WIEN

Institute of Production Engineering and Photonic Technologies

Master Thesis

Improvement of Lagrangian approach for the multi-physical simulation of powder based additive manufacturing laser processes

executed to obtain the academic degree of

Master of Science under the direction of

Univ.-Prof. Dipl.-Phys. Dr.-Ing Andreas Otto

(Institute of Production Engineering and Photonic Technologies)

and under the supervision of

MSc. Rodrigo Gómez Vázquez

(Institute of Production Engineering and Photonic Technologies)

submitted to the Vienna University of Technology

Faculty of Mechanical and Industrial Engineering

by

Francisco Javier Codina Álvarez

11721396

Heinestraße 11/25

1020 Wien

Vienna, September 2019

Declaration in Lieu of Oath

I hereby declare to be the sole author of this thesis and that no part of my work has been previously published or submitted for publication. I certify, to the best of my knowledge, that this thesis does not infringe upon anyone's copyright nor violate any proprietary rights concerning ideas, techniques, quotations, or any other material from the work of others. All knowledge arising from external sources has been fully acknowledged below in accordance with standard referencing practices.

Francisco Javier Codina Álvarez

Vienna, September 2019

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor Rodrigo Gómez Vázquez for all the support given during my work in the Institute of Production Engineering and Photonic Technologies. I would also like to thank Andreas Otto and the whole Institute of Production Engineering and Photonic Technologies for giving me the opportunity to work in this project.

I am extremely gratefull to my parents for all their support along the way, for their caring and sacrifices and for educating and preparing me for my future and to my sister Marta, whose work capacity and positive attitud I highly admire.

I would like to thank all my friends back in Madrid who I really miss and who despite the distance do not allow friendship to decay. Thanks also to my friends here in Vienna for making this city feel like home despite the distance.

Finally, special thanks to my flatmates Pati and Félix for making this such a special year and whose company I will highly miss but never forget.

Abstract

Among all the additive manufacturing technologies, metal powder based techniques become very suitable when coming to produce custom functional components of complex geometries within an unbeatable time. This technology is still developing and trying to improve, to achieve this goal a great understanding of the process is needed. Multiphysical simulations allow us to analyze the process without invloving the huge cost of systematic experimental studies.

OpenFOAM is an open-source simulation environment aimed for the solution of continuum mechanics problems, especially computational fluid dynamics (CFD), based on C++. It has been used by the Institute of production Engineering and Photonic Technologies of the TU Wien to develop during the last ten years a simulation software designed to acquire better knowledge of additive manufacturing technologies. The Lagragian library is a very useful tool for the simulation of metal powder based techniques and thanks to its combination with the Finite-Volume approach some promisisng results have been obtained. Nevertheless, particles interaction implementation require a closer look, for this reason the coupling of discrete element methods (DEM) and volume of fluid (VOF) requires further work to improve precision of the simulation results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

Additive manufacturing technologies are gaining more and more importance in an industrial environment already immersed in a new revolution of the production methods, the so-called transition to "Industry 4.0". Among them, powder metal based techniques become very suitable when coming to produce custom functional components of complex geometries within an unbeatable time. Unfortunately these techniques are not exempt of quality defects such as thermal distortion [11], porosity [1], etc. in special when alloys of complex weldability (e.g. high-degree superalloys) need to be processed. A thorough understanding of the process evolution as well as of the driving mechanisms for these defects results crucial in order to develop the right (predictive) correcting actions.

In this context, multiphysical simulations can allow a complete barrier-free process analysis without involving otherwise inassumable costs of systematic experimental studies.

Therefore the Research Unit for Photonic Technologies of the TU Wien has been developing over the last decade a sophisticated simulation software based on OpenFOAM® [7] open-source simulation environment aimed to the study of many kinds of laser assisted manufacturing processes. Recent works combining the use of Lagrangian particles with the traditional Finite-Volume approach showed promising potential for the study of powder-based AM processes. However, the current implementation still requires special attention, in special to the mechanics of the particles interaction, since they are crucial for avoiding unphysical predictions.

1.2 Objectives

The goal of this thesis will be therefore the **extension the current capabilities of the simulation model** in order to also **enable the simulation of metal powder-based AM processes.**

More precisely, this involves following tasks:

- Porting a previously developed implementation of a numerical coupling between a Lagrangian Discrete Element Method (DEM) approach used to

model the powder material and the Eulerian Finite-Volume (FV) Volume-of-Fluid (VOF) approach that describes the workpiece and surroundings.

- Validation of the implemented DEM-to-VOF coupling and estimation of the error for conservation of properties being transferred from the powder to the VOF-approach when melting, namely mass, momentum and energy.
- Setup of a mechanistic DEM-approach for particle collisions.
- Implementation of an efficient method to generate powder beds.
- Prototype simulation of exemplary powder-based AM-processes showing the main macroscopic characteristics.

Requirements for this purpose will be:

- Proficient knowledge of C/C++ programming language.
- Basic knowledge of the underlying physics of the rigid body dynamics as well as of the fluid dynamics.

Chapter 2

DEM-to-VOF coupling

When simulating additive manufacturing processes a popular approach, is using finite volume based solvers. This is, apply a Eulerian (VOF) approach to fluids simulation. One of the main characteristics of it is the use of a fixed mesh, through which the fluids move. This works good until discrete particles are to be added to the simulation, to predict and track each particle movement the previous approach is far from optimal. On the other hand, a Lagrangian (DEM) approach is optimized for the simulation of moving nodes, making easier the tracking of each discrete element. The problem comes when simulating AM processes which need an efficient simulation of both: discrete elements and volume of fluid. Although there are a few applications in OpenFoam that couple these bith approaches (e.g. reactingParcelFOAM solver), none of them is suitable for the simulation of laser additive manufacturing without having to make substantial changes to the core of these models.

For that reason, extending the capabilities of an already existing VOF-code for the simulation of laser assisted processes with an additional DEM-approach for the powder was seen more convenient.

Of course, this implied carefully coupling of both approaches, as it will be explained in the further lines:

2.1 FV continuum

2.1.1 Mass

The conservation of mass fraction for each phase in FV is:

$$\frac{d}{dt}(\rho\alpha) + \nabla \cdot (\vec{U}\rho\alpha) = \sum_{phases} (S_{pc}) \tag{2.1}$$

Where α is the volume fraction of a phase in a cell $\alpha = \frac{V_{phase}}{V_{cell}}$, ρ the density of the phase and S_{pc} source term accounting for the mass exchanges among the continuum phases (can be possitive or negative).

For the coupling with the DEM a new source term needs to be added, so the molten mass from a particle is included in the FV continuum. This mass is calculated as a sum of the molten mass of all the particles: $\frac{\sum_{part} S_{m_{p\to c}}}{V_{cell}}$ leaving the new conservation equation as:

$$\frac{d}{dt}(\rho\alpha) + \nabla \cdot (\vec{U}\rho\alpha) = \sum_{phases}(S_{pc}) + \boxed{\frac{\sum_{part} S_{m_{p\to c}}}{V_{cell}}} \tag{2.2}$$

This source term, once the condition $h_p >= c_p T_{melting}$ is reached, is calculated as:

$$S_{m_{p\to c}} = \frac{m_{p\to c}}{\Delta t} \tag{2.3}$$

$$m_{p\to c} = min\left(m_p,\ \frac{Q_{c\to p}}{H_{melting}}\Delta t\right)$$

Being $m_{p\to c}$ the molten mass from the particle that is being transferred to the continuum, m_p the mass of the whole particle, $Q_{p\to c}$ the heat transferred from the continuum to the particle, $H_{melting}$ the latent heat for melting and Δt the time interval. The minimum is taken so the mass transferred to the continuum is never larger than the mass of the particle.

2.1.2 Energy

The energy conservation equation for each phase of the FV continuum is:

$$\frac{d}{dt}(\rho h\alpha) + \nabla \cdot (\vec{U}\rho h\alpha) = \sum_{phases}(S_{pc}) + \frac{\alpha Q_{laser}}{V_{cell}} \tag{2.4}$$

Local instantaneous thermal equilibrium among all phases is assumed by averaging the properties:

$$\overline{\rho h} = \sum_{phases} \rho h\alpha,\quad \overline{\rho c_p} = \sum_{phases} \rho c_p \alpha \tag{2.5}$$

$$\bar{T}' = \frac{\overline{\rho h}}{\overline{\rho c_p}}$$

And with such assumption the heat conduction equation is applied:

$$\frac{\overline{\rho c_p}(\bar{T} - \bar{T}')}{\Delta t} = \nabla[\bar{k}\nabla(\bar{T})] \tag{2.6}$$

$\bar{k}$ is the averaged heat conduction coeficient, calculated as the harmonic average of all the phases:

$$\bar{k}^{-1} = \sum_{phases} \frac{\alpha_i}{k_i} \tag{2.7}$$

The coupling with DEM is again made by adding a source term to the main equation (2.4):

$$\frac{d}{dt}(\rho h\alpha) + \nabla \cdot (\vec{U}\rho h\alpha) = \sum_{phases}(S_{pc}) + \frac{\alpha Q_{laser}}{V_{cell}} + \boxed{\sum_{part} \frac{S_{h_{p\to c}}}{V_{cell}} - \sum_{part} \frac{Q_{c\to p}}{V_{celli}}} \tag{2.8}$$

The first source term is adjusted with the molten mass from the particle that is being transferred to the continuum as:

$$S_{h_{p\to c}} = \frac{m_{p\to c}\,(c_p T_{melting} + H_{melting})}{\Delta t} \tag{2.9}$$

As before, this source term is included once the condition $h_p >= c_p T_{melting}$ is fulfilled.

The second source term is the heat given to the particle and is calculated by the DEM, so it will be later explained.

2.1.3 Momentum

The momentum conservation is made for the mixture of phases by using averaged values as in (2.5):

$$\begin{aligned} &\frac{d}{dt}(\bar{\rho}\vec{U}) + \nabla\cdot(\vec{U}\bar{\rho}\vec{U}) = \\ &\nabla\left[\bar{\mu}\nabla(\vec{U})\right] - \nabla(p) + \bar{S}_{gravity} + \bar{S}_{surf.tension} - \bar{D}_{solid}\vec{U} \end{aligned} \tag{2.10}$$

Being $\bar{S}$ source terms from forces and $\bar{D}_{solid}$ a porosity term to avoid movement of the solid phases. Is a big value if there is any solid phase within a cell.

In this case there are two source terms, one concerning the momentum given to the continuum through the particle molten phase and another concerning the momentum lost because of the drag forces applied to the particle.

$$S_{u_{p\to c}} = \frac{m_{p\to c}}{\Delta t}\vec{U}_p, \quad S_{drag} = \frac{3}{4} m_p \frac{C_d Re}{\rho_p d_p^2}(\vec{U} - \vec{U}_p) \tag{2.11}$$

So the coupled equation for the mixture of phases is:

$$\begin{aligned} &\frac{d}{dt}(\bar{\rho}\vec{U}) + \nabla\cdot(\vec{U}\bar{\rho}\vec{U}) = \\ &\nabla\left[\bar{\mu}\nabla(\vec{U})\right] - \nabla(p) + \bar{S}_{gravity} + \bar{S}_{surf.tension} - \bar{D}_{solid}\vec{U} \\ &+ \boxed{\frac{\sum_{part} S_{u_{p\to c}} - S_{drag}}{V_{cell}}} \end{aligned} \tag{2.12}$$

2.2 DEM

Energy and momentum are two-way coupled, this means, the equations controlling the FV continuum are affected by the discrete elements (as has been seen) and vice versa. For this reasonthe equations of the DEM have been also expanded with new source terms. The mass is not two-way coupled, since there is no transference of it from the continuum to the discrete elements

2.2.1 Energy

The energy transfer is dominated by convective transfer. The equation that controls the heat transfer is:

$$Q_{c\to p} = h_{convection} A_p (T_c - T_p) \tag{2.13}$$
$$h_{convection} = \frac{k_c Nu}{d_p}$$

$Q_{c\to p}$ is the heat transferred from the continuum to the particle (negative, if its from particle to continuum), A_p the area of the particle, T_i the temperature, k_c is the locally averaged thermal conductivity of the continuum, d_p the diameter of the particle and Nu is the Nussel number calculated by the Ranz-Marshall correlation [14].

2.2.2 Momentum

The kinematic equation controlling the particles motion is:

$$m_p \frac{d\vec{U}_p}{\Delta t} = m_p \vec{g}\left(1 - \frac{\vec{\rho}_c}{\rho_p}\right) + \frac{3}{4} m_p \frac{C_d Re}{\rho_p d_p^2}(\vec{U}_c - \vec{U}_p) \tag{2.14}$$

$$C_d = \begin{cases} 0.424, & Re > 1000 \\ \dfrac{24}{Re} + 4(Re)^{1/3}, & Re \leq 1000 \end{cases}$$

Having in the right part of the equation two terms: the first is the gravity (buoyancy) force term and the second the drag force term (solid spheres [6]). Where m_p is the mass of the particle, ρ_i density of particle or continuum, d_p diameter of the particle and $\vec{U}_i$ velocity of particle.

2.3 Validation of DEM-VOF coupling

To finish the coupling of the two models a number of evaluations have been made. In particular for the mass, energy and momentum transference and conservation.

Setup The three validations have been made in the same case. The domain dimensions are 2.1x18.2x2.1 mm, divided with a mesh of 0.7 mm cell side (Fig. 2.1). With this geometry the transversal section has a 3x3 cells grid, having one in the centre to be occupied by the melting particle. The upper and lower boundaries are opened and used as input and output respectively. A flow (U= -2m/s) is imposed through the domain, so a certain initial momentum in the whole domain, different from zero, can be calculated. To simplify the flow in the domain a "Slip" boundary condition is used on the walls for the velocity, a fixed value on the inlet and a zero gradient condition on the outlet.

Table 2.1: Domain parameters

Domain dimensions	**Cell dimensions**	**Velocity boundary conditions**		
		Walls	**Inlet**	**Outlet**
2.1x18.2x2.1 mm	0.7x0.7x0.7 mm	Slip	(0, -2, 0) m/s	Zero gradient

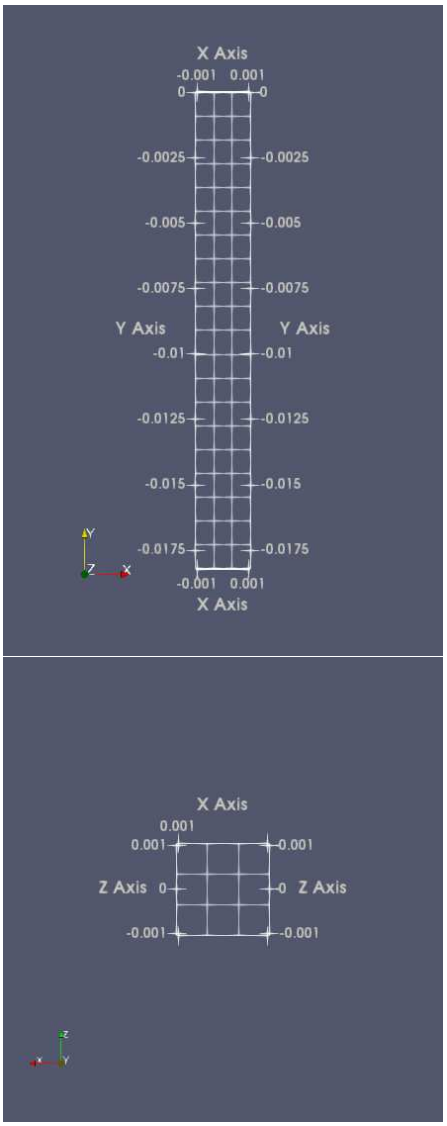


Figure 2.1: Meshed domain

The simulated fluid is based on argon, although several changes have been applied in order to simplify the simulation and the validation. To reduce the cooling effect of the injected particle, the specific heat capacity of the argon has been increased, as well as the thermal conductivity coefficient, in order to accelerate the melting. The initial temperature is 1500K to asure that the phase change temperature is reached (1000K). To simplify the sources the compressible terms in the momentum equation have been removed and the the argon phase defined as liquid instead of gas, so it does not compress when its temperature is reduced.

Table 2.2: Artificial gas parameters

To	Velocity	Density	Specific heat capacity (Cp)	Thermal conductivity (K)
1500 K	(0, -2, 0) $m\ s^{-1}$	1.6115 $kg\ m^{-3}$	5210.928 $JK^{-1}kg^{-1}$	100 $Wm^{-1}K^{-1}$

The particle's material is based on stainless steel, although some parameters have also been changed to assist the melting. The phase change temperature and the specific heat capacity have been reduced, in order to have a faster process. The initial temperature of the particle is 0K so there is no netto energy contribution, expecting the total energy in domain to remain the same after the melting of the particle.

Table 2.3: Particle material parameters

To	Velocity	Density	Specific heat capacity (Cp)	Melting temperature (Tmelt)	Diameter
0 K	(0, -2, 0) m/s	8.02 $\cdot 10^2$ kgm^{-3}	48.35 $JK^{-1}kg^{-1}$	1000 K	600 μm

A display of the simulation can be seen in Fig. 2.2. Six snapshots of the melting process have been taken, the material transference can be observed through the α_{liquid} (alpha.liquid) parameter, it indicates the volume fraction of liquid material in each cell:

$$\alpha_{liquid} = \frac{VolumeMeltedMaterial_{cell_i}}{TotalVolume_{cell_i}} \tag{2.15}$$

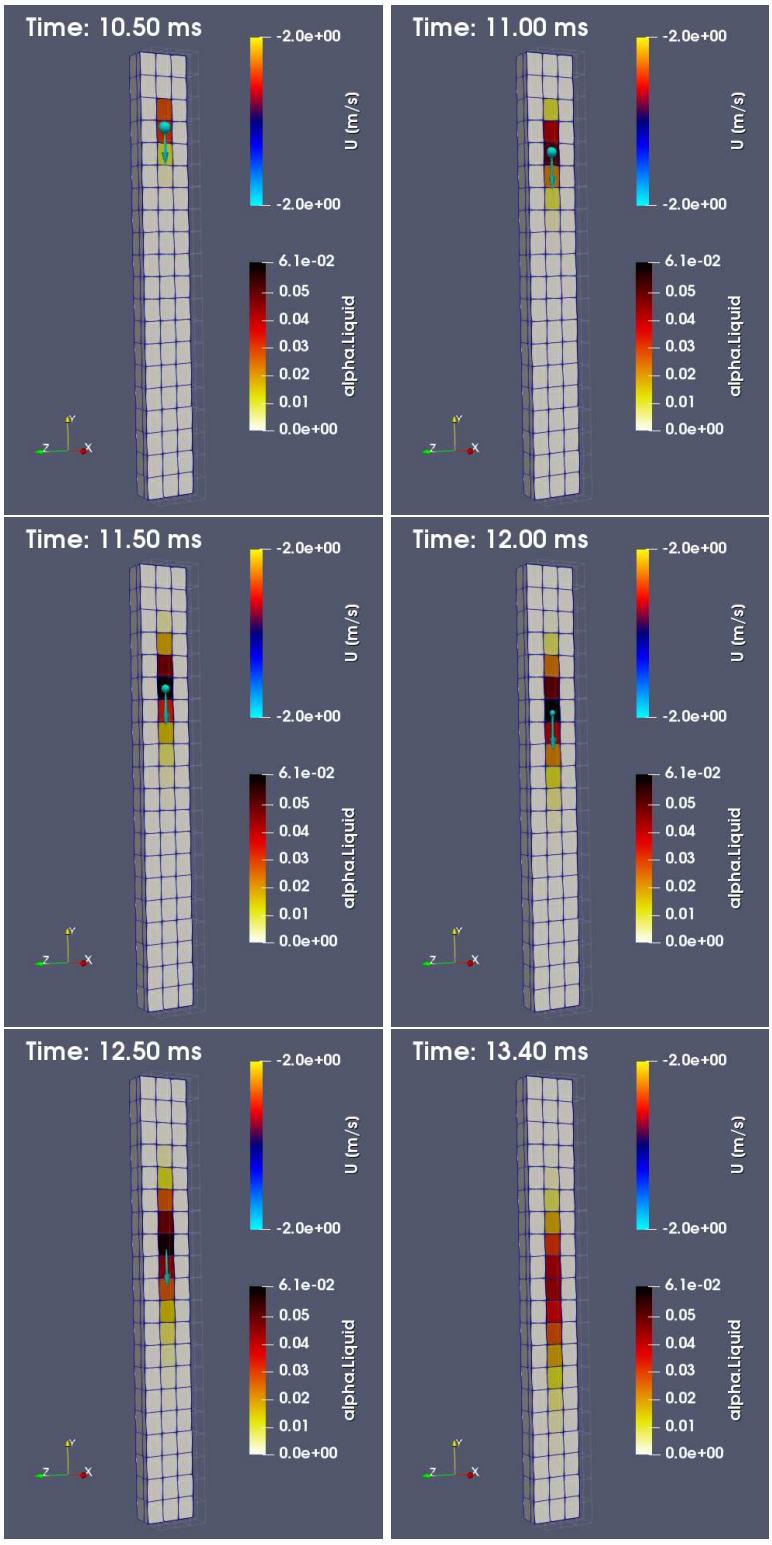


Figure 2.2: Simulation of described setup for validation of mass, energy and momentum conservation

2.3.1 Mass validation

Balance The evaluation is made at the end of each time step. Comparing one magnitude in the system of the previous time step (old) and the actual one (new). In this case the mass is calculated as:

$$mass_t = \sum_i^{cells} \left(\int_{cell_i} \rho_i \; dv \right) \tag{2.16}$$

The difference between them is caused by the mass sources in the system, which have been narrowed to mass from phase changes and mass flow through the boundaries.

$$mass_{phaseChange} = \sum_i^{cells} \left(\int_{cell_i} \frac{moltenMass_i}{vol_i} dv \right)$$

$$mass_{boundaries} = \sum_i^{boundaryCells} (\phi_i \cdot \rho_i)$$

Once these terms are calculated they are used in the overall balance:

$$mass_{new} - mass_{old} = mass_{phaseChange} + mass_{boundaries} + error \tag{2.17}$$

The two sides of the equation can be grouped in the following way (excluding the error):

$$mass_{difference} = mass_{new} - mass_{old} \tag{2.18}$$

$$mass_{sources} = mass_{phaseChange} + mass_{boundaries} \tag{2.19}$$

And afterwards the relative error between these two magnitudes is calculated as:

$$relativeError = \frac{mass_{sources} - mass_{difference}}{mass_{new}} \tag{2.20}$$

Results All plots start at the particle injection time and end when its influence is over. There are two clearly differentiated parts: the particle melting and giving its mass to the domain (0.01-0.013 s) and the output of such part of the domain with the molten mass (0.014-0.024 s). Given the particle parameters in Table 2.3 its mass is:

$$mass_{particle} = V \; \rho = 9.069 \cdot 10^{-7} \; kg \tag{2.21}$$

This value is the exact rise of mass in domain while the particle melts, and which afterwards escapes through the output boundary (Fig. 2.3).

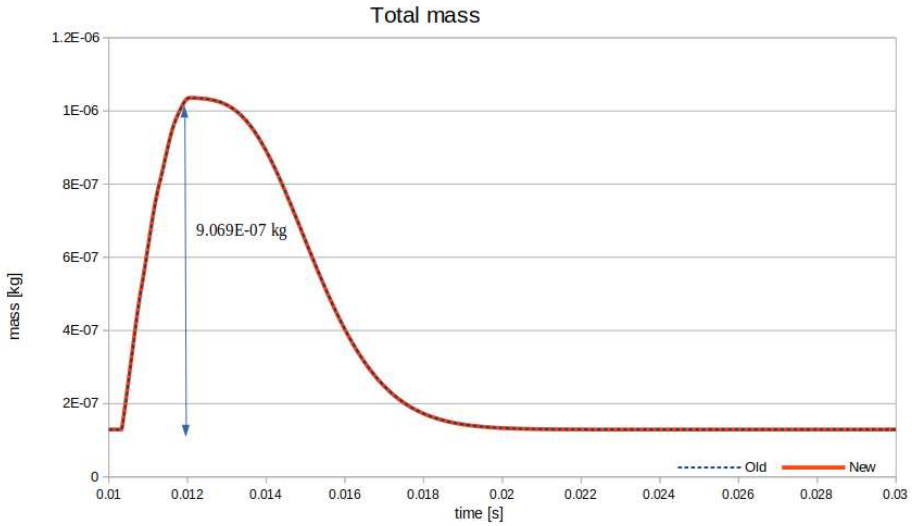


Figure 2.3: Total mass in domain. The old and new mass overlap since the difference between both curves is much smaler that the total magnitude, so no visual difference can be seen

The particle melts aproximately in t=0.012s. The domain mass rise when the particle material is transferred to the volume of fluid and afterwards decreases when it escapes through the lower boundary until the domain mass returns to its initial value (Fig. 2.3).

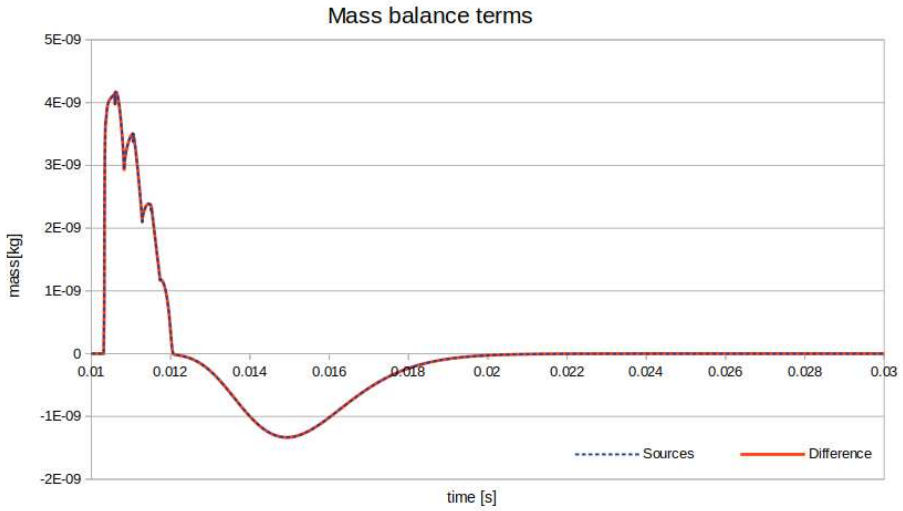


Figure 2.4: Mass balance terms from Equation 2.17. The difference between old and new mass and mass sources overlap since the difference between both curves is much smaler that the total magnitude, so no visual difference can be seen

In Figures 2.4 and 2.5 the two parts can be distinguished. At the beginning

of the melting process the exposed area of the particle is higher, so there is a higher volume of the particle affected by the heat transference. Fot this reason the molten mass source term is bigger at the start and it decreases from this poit forward (Fig.2.5). The influence of the boundary source term starts when the molten phase reaches the output boundary, this mass leaving the domain means a negative source term which decreases the overall total mass leaving it with the initial value.

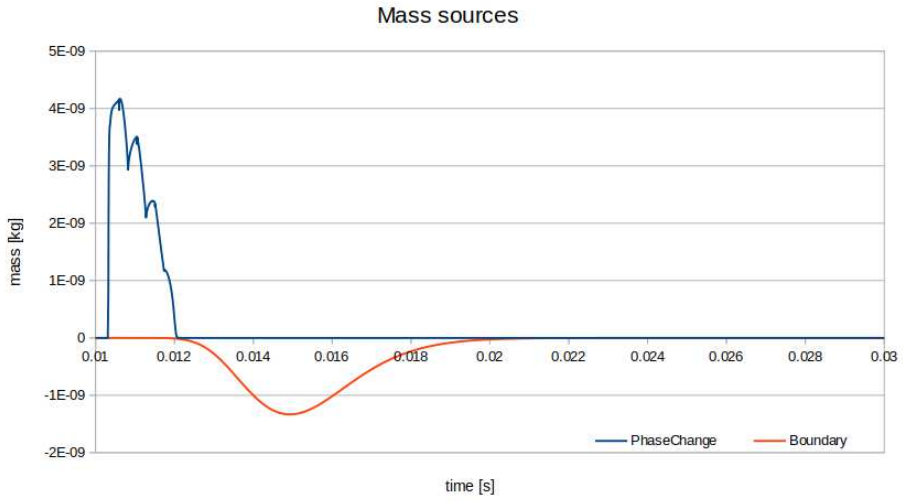


Figure 2.5: Mass sources terms from Equation 2.19

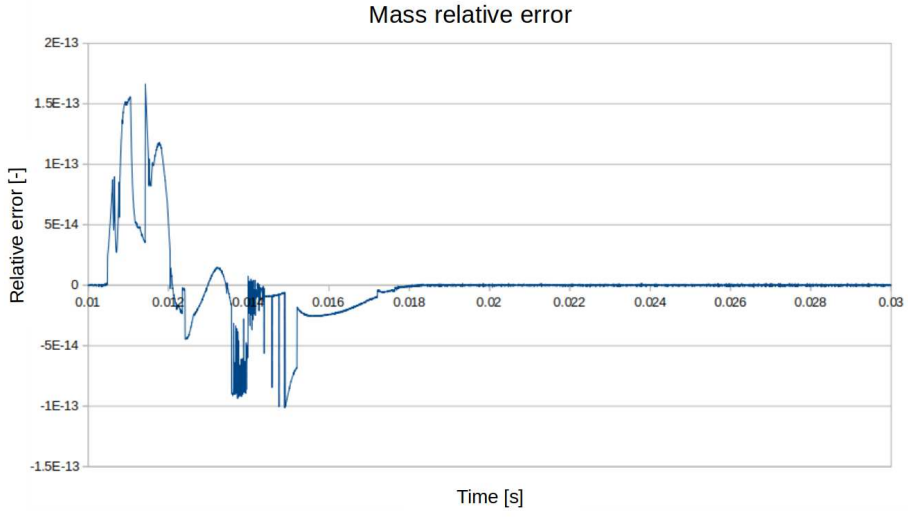


Figure 2.6: Mass relative error calculated with Equation 2.20

The error (Fig. 2.6) is kept in the order of magnitude of resolution of the solver, so the mass transference from LPT to VOF is conservative.

2.3.2 Energy validation

Balance For the energy, the total magnitude is calculated as follows:

$$E_t = \sum_i^{cells} \left(\int_{cell_i} \rho_i \; h_i \; dv \right) \tag{2.22}$$

Again an old and a new magnitude are compared in each time step, and the difference is caused by two main source terms: phase change and boundaries.

$$E_{phaseChange} = \sum_i^{cells} \left(\int_{cell_i} \frac{moltenMass_i \; h_i}{vol_i} dv \right)$$

$$E_{boundaries} = \sum_i^{boundaryCells} (\phi_i \cdot \rho_i \cdot h_i)$$

Included directly as sources in the total energy balance:

$$E_{new} - E_{old} = E_{phaseChange} + E_{boundaries} + error \tag{2.23}$$

In a similar way to the mass balance, the two sides are grouped (excluding the error):

$$E_{difference} = E_{new} - E_{old} \tag{2.24}$$

$$E_{sources} = E_{phaseChange} + E_{boundaries} \tag{2.25}$$

And the following error is:

$$relativeError = \frac{E_{sources} - E_{difference}}{E_{new}} \tag{2.26}$$

For the same reason as in equation (2.17), lists should be integrated through the whole domain. The total theoretical balance would be:

$$E_{t+1} - E_t = energySourceTerm + error \tag{2.27}$$

The difference of energies after and before has been extracted and compared with the $energySourceTerm$ (Fig. 2.9), calculating the relative error afterwards (Fig. 2.10):

$$relativeEnergyError = \frac{energySourceTerm - (E_{t+1} - E_t)}{energySourceTerm} \tag{2.28}$$

This error must be kept to a minimum in order to consider the energy transference conservative.

Results The energy plots start again with the particle injection. This time the two differentiated parts are the pre-melting ($E < 0J$) and the melting ($E > 0J$) and can be seen in Figure 2.9. First the particle absorbs energy until it reaches the melting temperature (sensible energy), then the energy is used to overcome the melting heat (latent energy) and finally the absorbed energy is given back to the system when the particle mass is transferred to VOF. The total energy that the particle needs to melt is the sum of latent and sensible heat:

$$E_{sensible} = mass_{particle}\ C_p\ (T_{melt} - T_{ref}) = 0.043714\ J \tag{2.29}$$

$$E_{latent} = mass_{particle}\ h_{melt} = 0.248611\ J \tag{2.30}$$

$$E_{particle} = E_{sensible} + E_{latent} = 0.29146\ J \tag{2.31}$$

In Figure 2.7 the energy difference matches the reduction of the system energy due to the sensible heat. The latent heat cannot be seen because when it melts the absorbed energy to overcome the melting heat is given back to the system in the same time step and since the balance is made from one timestep to the next, it is not reflected.

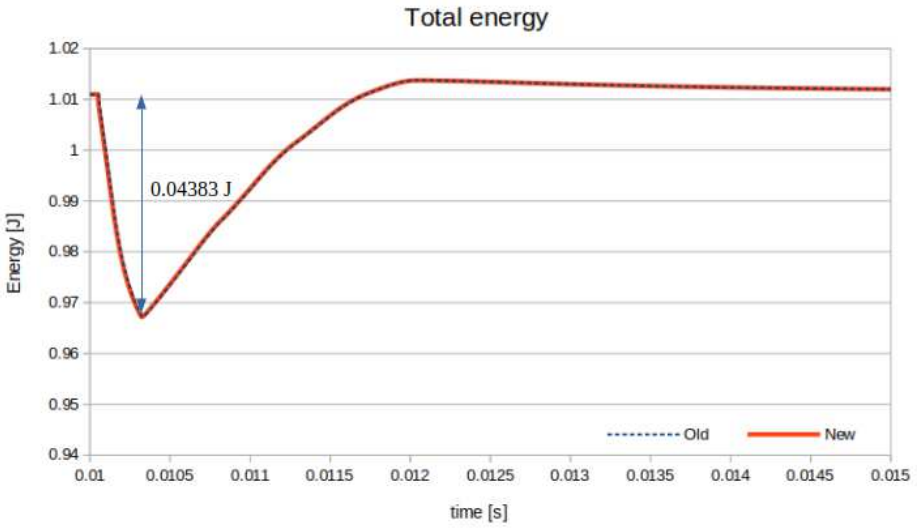


Figure 2.7: Total energy in domain. Both curves overlap visually due to the small difference between them

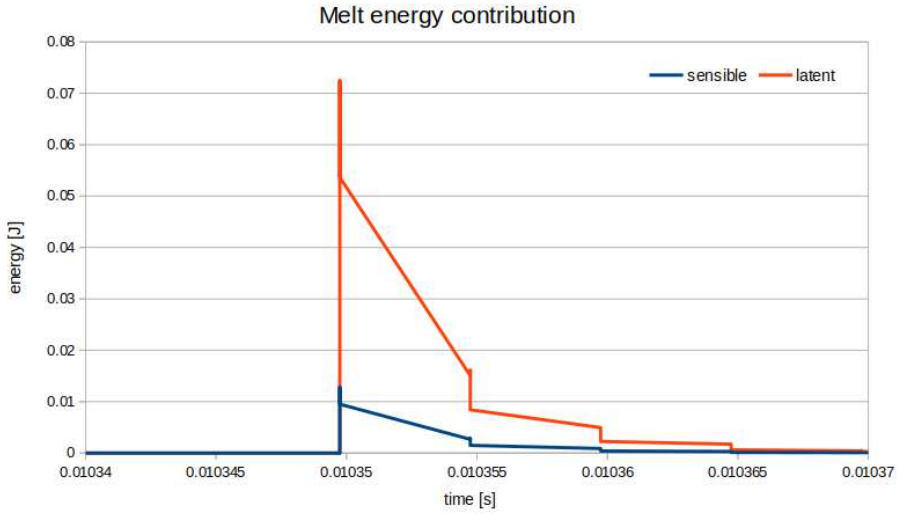


Figure 2.8: Energy source terms from the melting paticle, the latent heat curve seems to have no effect in Figure 2.7 since compensated by the exact same heat that the particle is absorbing in the same timestep. The sum (area) from both curves are: $E_{sensible} = 0.04383\ J$ and $E_{latent} = 0.24748\ J$

In Figure 2.8 the two contributions to the particle melt energy (Eqn. 2.31) can be seen. The sum (area) from both curves are: $E_{sensible} = 0.04383\ J$ and $E_{latent} = 0.24748\ J$, which compared to the theoretical ones (Eqn. 2.29 & 2.30) we get a relative error of:

$$error_{sensibleHeat} = 3.63 \cdot 10^{-4}; \quad error_{latentHeat} = 5.37 \cdot 10^{-4} \tag{2.32}$$

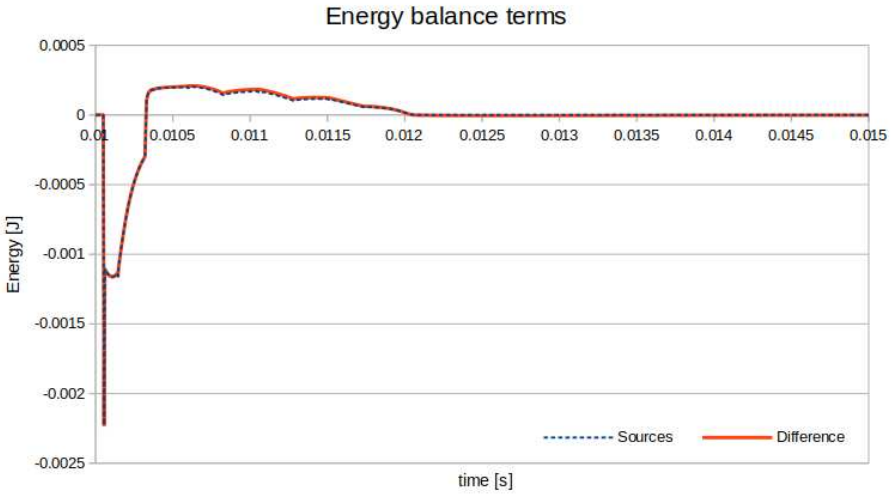


Figure 2.9: Energy balance terms from Equation 2.23. The difference between both curves is much smaller then the total magnitude, so no visual difference can be seen

In the beginning the source term reaches its minimum (particle absorbing energy), which means maximal energy transference from the domain to the particle. When the particle is injected the exposed area and the temperature difference between the particle and the system are higher (Fig. 2.9), the particle keeps absorbing energy until it reaches the melting temperature, then its mass is transferred to the VOF with its energy. Therefore, the energy contribution starts being positive (energy transference from the particle to the domain) until the particle is completely molten and it gets back to zero.

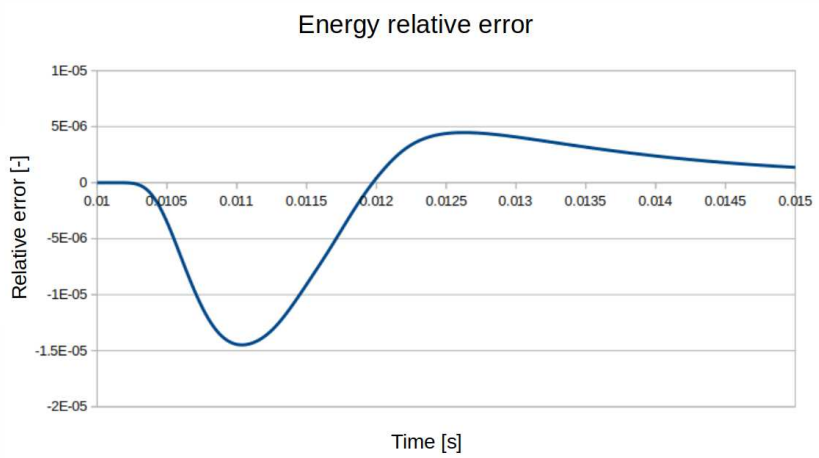


Figure 2.10: Energy relative error calculated with Equation 2.26

Figure 2.10 shows the relative error calculated as described in equation 2.26. Although is not as low as in the mass balance, the energy relative error is still kept under 1.5 10^{-5}, which is low enough to consider the energy transference conservative.

2.3.3 Momentum validation

Balance Given the case, where the motion is all in the vertical direction, from all the vectorial magnitudes (momentum and velocity) only the Y component is taken into account adn showed. All given values are to be read as a scalar magnitude that corresponds to the second component of the vector. [1]

The total momentum in the system at the end of each time step is calculated as:

$$M_t = \sum_{i}^{cells} \left(\int_{cell_i} u_i \cdot \rho_i \, dv \right)$$

[1] When calculating the velocity through the boundaries, other velocity components could have some influence, but since the flow is imposed to go in the vertical direction there is no influence in the higher and lower boundaries. The transversal boundaries are set as walls, so no flow can come through them in X or Z direction.

Again, the difference from one time step to the next is caused by the momentum sources in the system, which are two: the momentum source due to lagrangian particles phase changes and momentum flow through the boundaries.

$$M_{phaseChange} = \frac{meltingMass}{totalMass} \sum_{i}^{particles} (mass_i \cdot u_i)$$

$$M_{boundaries} = \sum_{i}^{boundaryCells} (\phi_i \cdot \rho_i \cdot u_i)$$

This calculation could give problems whith simultaneous particles melting with different velocities. Given the case where only one particle is involved in the process there is no posible conflict.

Once these terms have been calculated the overall balance is:

$$M_{new} - M_{old} = M_{phaseChange} + M_{boundaries} + error \tag{2.33}$$

The two sides of the equation have been grouped in two terms (excluding the error)

$$M_{difference} = M_{new} - M_{old} \tag{2.34}$$

$$M_{sources} = M_{phaseChange} + M_{boundaries} \tag{2.35}$$

And the relative error between these two magnitudes has been calculated as:

$$relativeError = \frac{M_{sources} - M_{difference}}{M_{new}} \tag{2.36}$$

Results All plots start at the particle injection time and end when its influence is over. There are two clearly differentiated parts: the particle melting and giving its momentum to the domain (0.01-0.013 s) and the output of such part of the domain with higher momentum (0.014-0.024 s). Given the particle parameters in Table 2.3 its momentum is:

$$M_{particle} = V\ \rho\ U_y = 1.81408 \cdot 10^{-6}\ kg\ m\ s^{-1} \tag{2.37}$$

Since the velocity is negative and the momentum is a vectorial magnitude the overall momentum decreases when the particle melts. This reduction of the momentum is exactly the momentum of the particle (Fig 2.11), which is afterwards recovered when the molten phase escapes the domain through the output boundary.

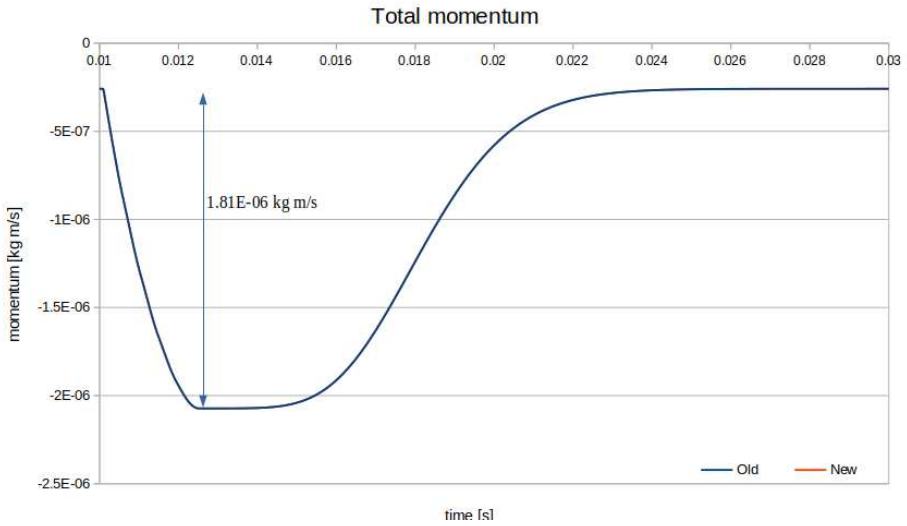


Figure 2.11: Total momentum in domain. The old and new momentum overlap because the difference is much smaller than the total magnitude, so no visual difference can be seen.

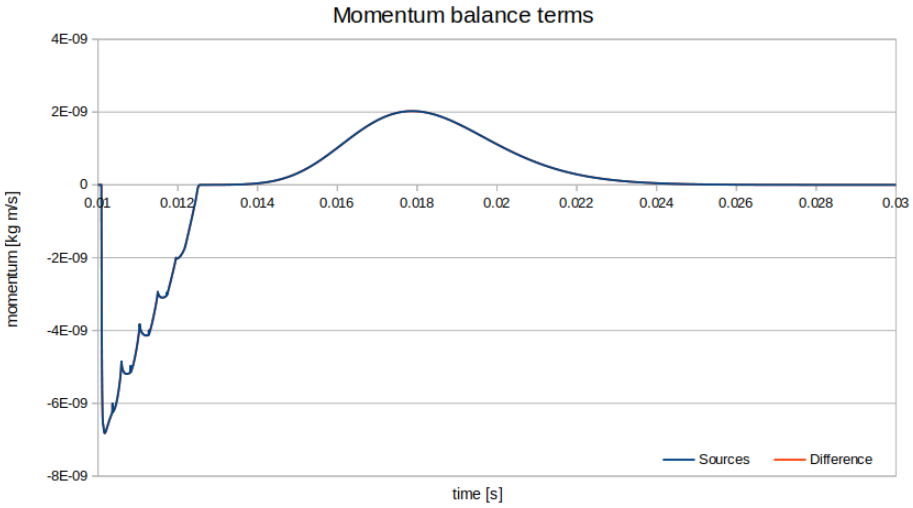


Figure 2.12: Momentum balance terms from Equation 2.33. The sources term and the difference between old and new momentum overlap because the difference is much smaller than the total magnitude, so no visual difference can be seen.

In Figures 2.12 and 2.13 the two parts can be distinguished. When the particle starts melting the momentum source term is bigger since the molten mass is the maximal in the process, from this point forward the particle only reduces its size and consequently does the momentum source term associated to its mass, until it disappears, leaving the phase change source term in 0 (Fig.2.13).

When the molten mass reaches the output boundary its negative momentum leaves the domain, which leads to a positive momentum source. The area under both curves (Fig.2.13) correspond to the momentum of the particle (Eqn.2.37), with a relative error of $2.9664 \cdot 10^{-6}$ for the particle phase change source term and of $4.465 \cdot 10^{-6}$ for the boundaries one.

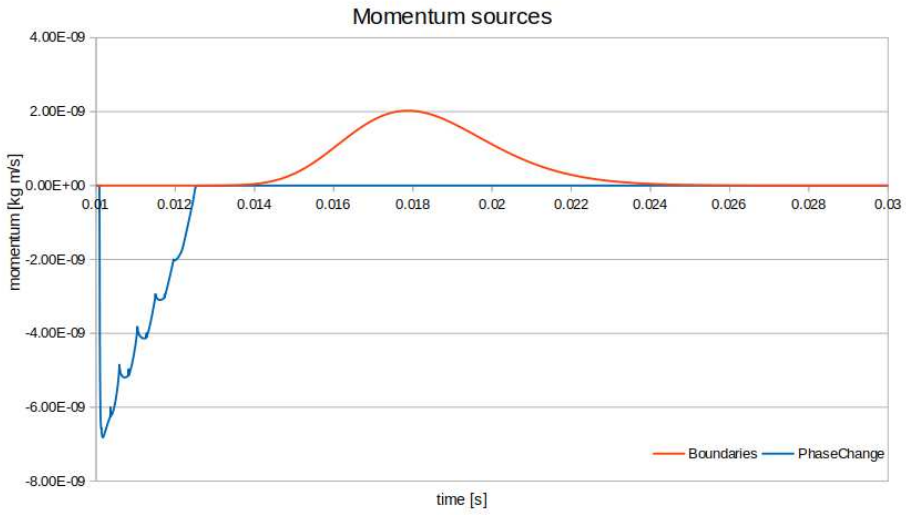


Figure 2.13: Momentum sources terms from Equation 2.35

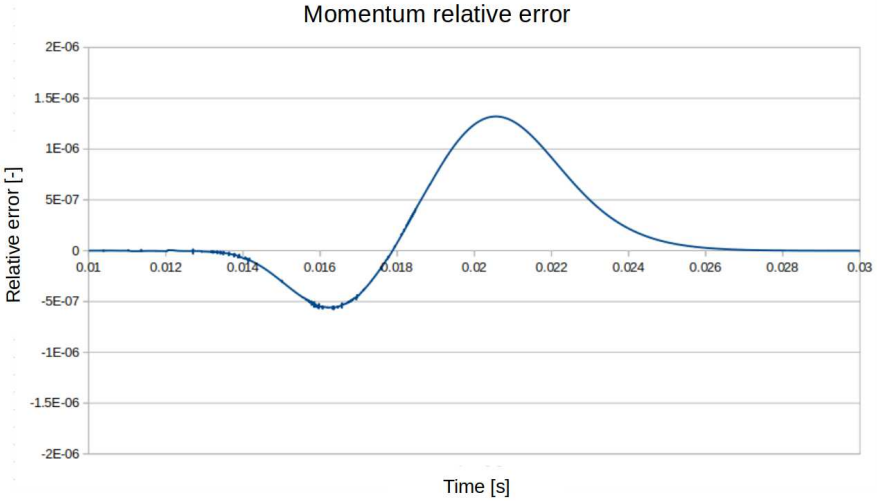


Figure 2.14: Momentum relative error calcualted with Equation 2.36

Figure 2.14 shows the relative error calculated as described in equation 2.36. The maximal error found is lower than 1.5 10^{-6}, for this reason the tranference of momentum from the particle to the system is also considered conservative.

2.3.4 Summary

Proving the correct transference of mass, energy and momentum from the particles to the VOF is key to start simulating additive manufacturing processes, for this reason a specific setup has been prepared, keeping it as simple as possible. In it one particle is injected, with a certain velocity and a certain temperature (lower than the melting temperature, see table 2.3), in a system that would rise its temperature until it reaches the melting point (Tables 2.1 and 2.2), taking place then the phase change which could be analysed (simulation snapshots in Fig. 2.2).

The analysis is made through three balances: mass (Eqn. 2.17), energy (Eqn. 2.23) and momentum (Eqn. 2.33). Comparing in each one of them the initial properties of the particle with its contribution to the system. Finally a relative error has been calculated for each balance, proving that the tranference of all three, mass, energy and momentum, is conservative, since the maximal absolute values of the relatives errors (Figs. 2.6, 2.10 and 2.14) are: $2\ 10^{-13}$ in the mass balance, $1.5\ 10^{-5}$ in the energy balance and $1.5\ 10^{-6}$ in the momentum balance.

From this point forward the coupling between the DEM to the VOF approach is taken as succesful, moving on now to issues faced regarding the interactions between particles.

Chapter 3

DEM mechanics

3.1 Lagrangian method

Although already implemented as has been explained in Section 2.2, some changes and improvements have been made in the Discrete Elements Method (DEM) mechanics. Due to the specific case treated in this project some points of the interactions between particles are especially sensitive. For this reason some bugs that could be minor in some other setups were unacceptable and had to be solved.

OpenFOAM's lagrangian library *pair collision* model solves each interaction as a physical collision, unlike the *stochastic collision* model, which is used to solve the motion of huge clouds with a statistical perspective.

The motion algorithm of the cloud can be divided in four parts: **sub-cycles** calculation, **motion**, **forces** calculation and second **motion**.

Sub-cycles: The collision forces are calculated when an overlap is found, which is checked only at the end of each step. For this reason, if the timestep is too big two particles could cross themselves without noticing each other, ending up at the other side of the collision point ignoring they should have overlaped. To avoid this situation the timestep must be reduced, which is made by adjusting it as a fraction of the Hertzian contact [10] between two particles:

$$t_{contact} = 2.87 \left(\frac{m^{*^2}}{R^* E^{*^2} U_{rel}} \right)^{1/5} \tag{3.1}$$

$$m^* = \left(\frac{1}{m_1} + \frac{1}{m_2} \right)^{-1}, R^* = \left(\frac{1}{R_1} + \frac{1}{R_2} \right)^{-1}, E^* = \left(\frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2} \right)^{-1} \tag{3.2}$$

Where m_i is the mass of one particle, r_i its radius, U_{rel} is the relative velocity between the two of them, E_i the Young's modulus and v_i the Poisson's ratio.

Given the large number of particles involved, the contact is calculated as a theoretical one, taking the dimension and velocity values that minimize the contact time as posible, guaranteeing that all potential collisions would be considered. The minimum value is reached by taking the higher velocity and the

minimum dimension between all particles and are applied as follows:

$$R_1 = R_2 = R_{min} \ ; \ U_{rel} = U_{max}$$

Then replacing the mass by: $m_i = \frac{4}{3}\Pi R_i^3$ the equation (3.1) results in:

$$\Delta t_{min} = 5.429675 \ R_{min} \left(\frac{\rho}{E^* \sqrt{U_{max}}} \right)^{0.4} \tag{3.3}$$

Using this Δt_{min} might cause instabilities, as some of the generated overlaps could be larger than expected, generating bigger repulsive forces without physical sense. To prevent this, a fraction of it is used, multiplying by a factor

$$\frac{1}{collisionResolutionSteps}$$

Where *collisionResolutionSteps* is user given. In order to obtain good results usually values between 5 and 10 have been used. Finally, to get the number of sub-cycles per time step the Δt_{step} is divided by the multiplication of the two previous factors:

$$\boxed{n_{sub-cycles} = \frac{\Delta t_{step}}{\Delta t_{min} \frac{1}{collisionResolutionSteps}}} \tag{3.4}$$

Motion: The particle motion is simulated through a Leapfrog integration algorithm in two steps, before and after the calculation of the forces [4].

Forces: When an overlap is found between two particles a normal force is calculated in the following way:

$$\boxed{\bar{F}_n = \hat{r}_{AB} \ k_N \ d_{overlap}^b - \eta_N \ (\vec{U}_{AB} \cdot \hat{r}_{AB})} \tag{3.5}$$

$$\hat{r}_{AB} = \frac{\vec{r}_{AB}}{|\vec{r}_{AB}|}, \ k_N = \frac{4}{3}\sqrt{R_{eff}} E^*, \ \eta_N = \alpha \sqrt{m_{eff} k_N} d_{overlap}^{0.25}$$

$$R_{eff} = \left(\frac{1}{R_A} + \frac{1}{R_B} \right)^{-1}, \ m_{eff} = \left(\frac{1}{m_A} + \frac{1}{m_B} \right)^{-1}$$

Where $\vec{r}_{AB}$ is the distance between centres, $d_{overlap}$ the normal overlap distance, $\vec{U}_{AB}$ the relative velocity R_i is the radius, m_i the mass, E^* definition is in equation (3.2), the coefficient b is used to define the Hertzian contact with the value 1.5 [10],the coefficient α is related with the energy restitution during the collision, it is set as 0.01 for an elastic behaviour without energy disipation [15].

Motion: Finally the second part of the Leapfrog integration algorithm is executed and the second motion part with the new forces is calculated.

3.2 Faced issues

3.2.1 Overlaps in motion

While simulating particle collisions some instabilities could be found, usually showing itself as a sudden increase in the velocity of some particles as if a very fast one had collided with them, which was not happening. These kinds of instabilities could be found during the evolution of the cloud, and the injection of the particles. Because of the resemblance to the "pop-corn" explosion these instabilities will be called that way.

The force product from a collision depends directly on the overlapped area between the two particles (equation 3.5). When this overlap is higher than it should be, the calculated force is also higher, giving the particles an unexpectedly high velocity which is the result that is seen as the "pop-corn" explosion. The undesirable high overlaps take place when the time interval in the particle motion is too big, and thus the collision is calculated too late, when the two particles motion is already larger than it should. Increasing the sub-cycles reduces the Δt_{step}, making possible the detection of collisions earlier, averting larger and unphysical overlaps, and thus avoiding "pop-corn". When a big number of particles are stacked up this kind of instability is much more probable, because the number of interactions between particles is higher. The affected particle collides with all the surrounding ones creating an expansion effect. Figure 3.1 shows the event in the right part of the box, although all the particles are nearly settled, suddenly a velocity is given to certain particles, causing the instability and making some of them leave the domain through the upper boundary.

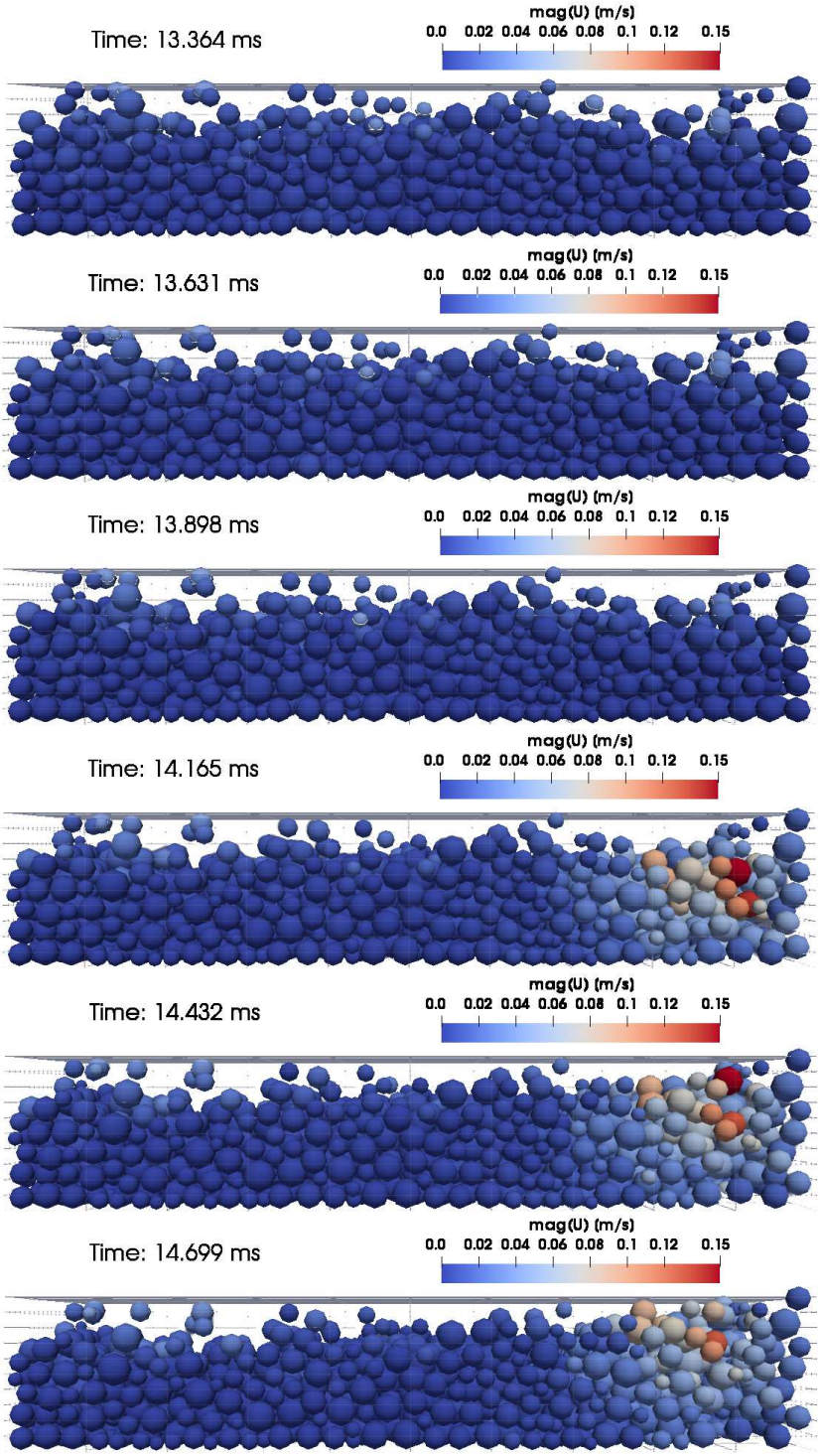


Figure 3.1: Velocity of particles in powder bed generation with "pop-corn"

For the control of the instability the *collisionResolutionSteps* can be modified, but the computational cost of increasing it should be taken into account. As said in section 3.1, the number of sub-cycles is increased with the *collisionResolutionSteps* directly, increasing the number of operations. The selected value is set between 5 and 10, depending on the number of particles, trying to reduce the computational cost when it is too high.

3.2.2 Overlaps in injection

Some of the injection models generate random positions for particles within a selected region. Although the probability of two particles being injected in the same point is low enough, the chances of two being generated close enough to have an overlap, or just one particle overlapping with the wall, are high. When this happens, the particles are injected in a position that starts solved as a collision and consequently a force is calculated and applied to them. Since this overlap is randomly generated the created force can be a big one, taking place a "pop-corn". This kind of instability is more probable to be seen when the particles number increases and its velocity decreases, since they accumulate in the injector and the chances to randomly generate an overlap with a new injected particle are higher. Since one new injected particle can overlap with a previously injected particle (Fig. 3.2) or with the walls (Fig. 3.3) each of the cases is approached separately:

Particle-particle To solve it, part of the collision routine is executed while injecting, until normal overlap is checked, but in this case, when afirmative, the injected particle is removed and the injection algorithm started again generating a new random position. To prevent getting into an infinite loop, when there is no available position to inject a limitation of attempts is given, and when it is reached, a warning message is given telling the user that no injection was possible. Although effective, these changes add more computational costs, for the overlap needs to be calculated for each injection and sometimes more than once.

Particle-wall To avoid the injection in a position where the particle overlaps with a wall, the applied algorithm is the same as in the previous case, first the collision with a wall rutine is executed to check if there is any overlap, if there is, the particle is removed and a new one injected, by restarting the algorythm, with a limitation of attempts.

The main limitation with this implementation is that when the maximal number of attempts is reached the particle is not injected. If this is a recurrent situation then the injected particle flow would not be the desired and especified in the case definition. An implementation that could solve this would be to count the number of particles that had not been injected, and make it whenever it is possible.

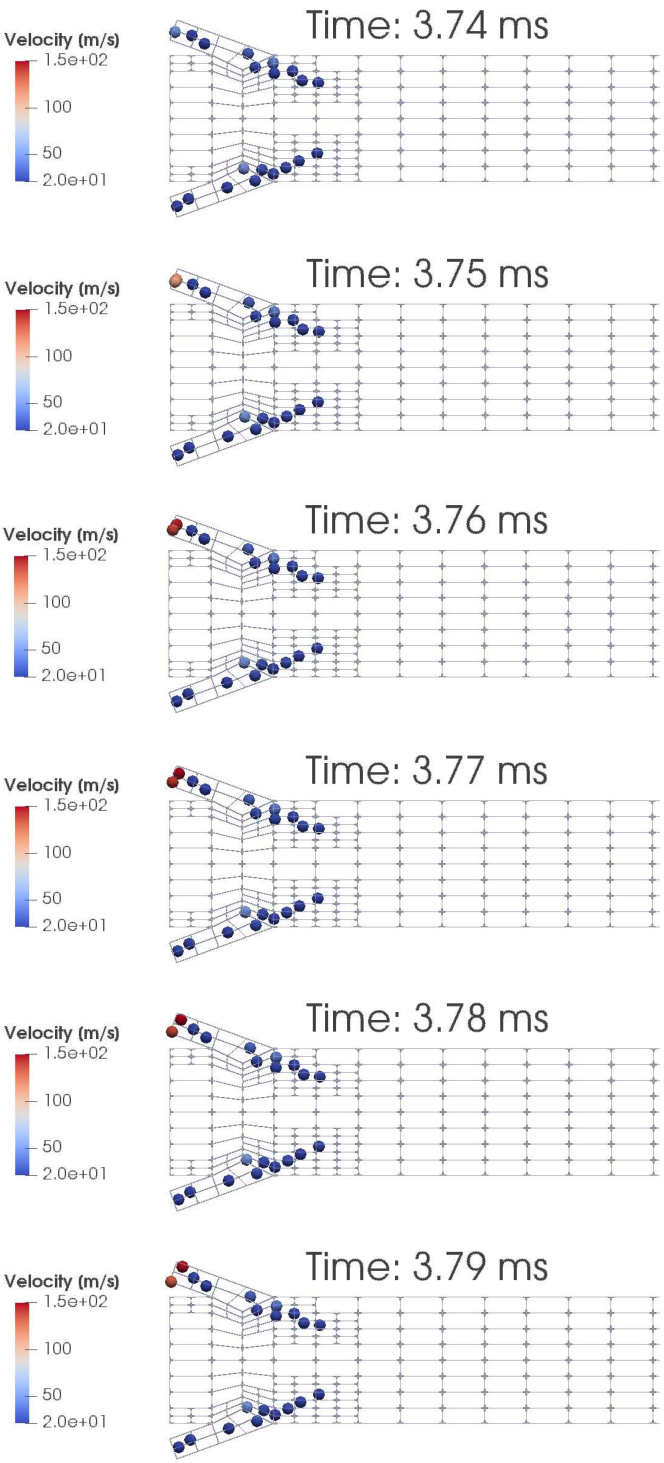


Figure 3.2: "Pop-corn" in the top injector due to an overlap between two injected particles. It can be seen comparing its velocity with the rest of the injected particles, they are accelerated when they are injected, because one overlaps with the other when it is injected, and then a collision is calculated

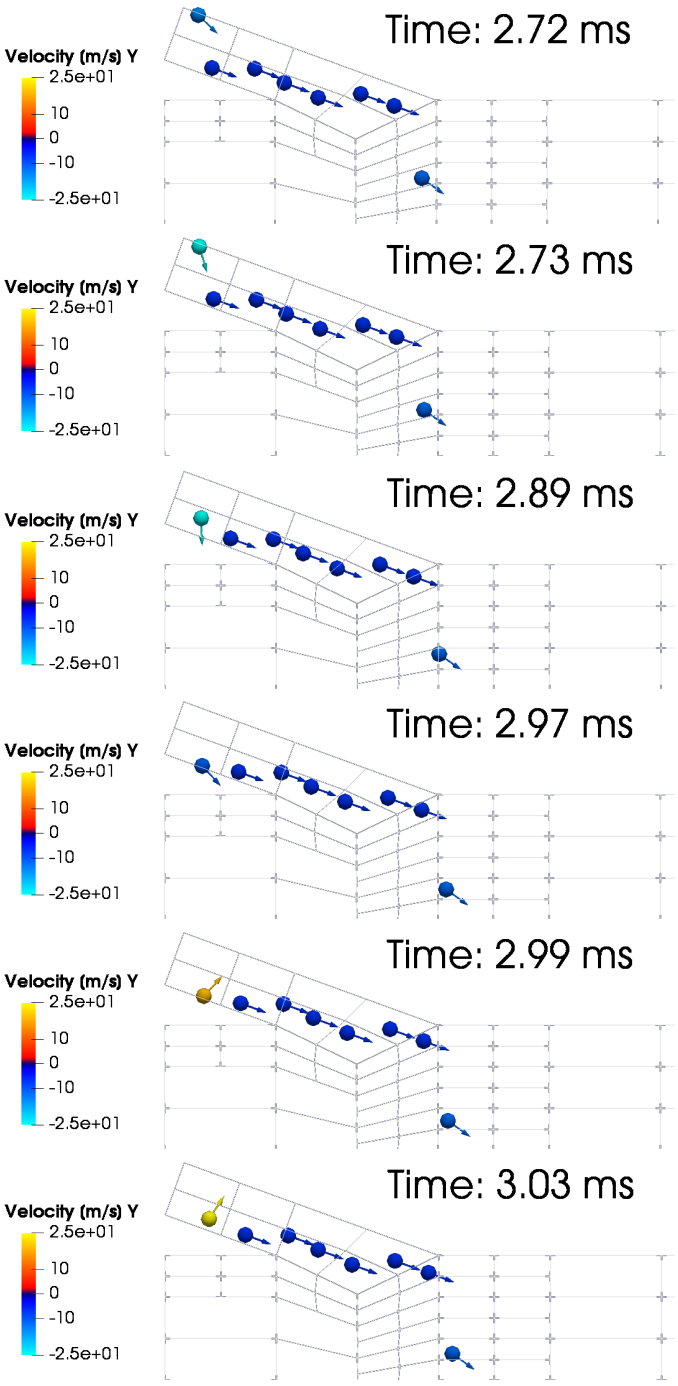
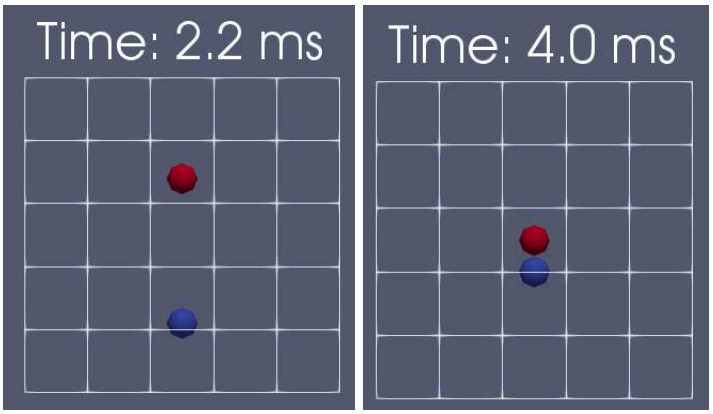


Figure 3.3: Overlap with wall in injection. Instead of creating a "pop-corn" this overlap changes the flow direction, which was originally towards the centre of the domain. Overlapped particle moves from one wall to another as a repulsive force is calculated due to its injecting position.

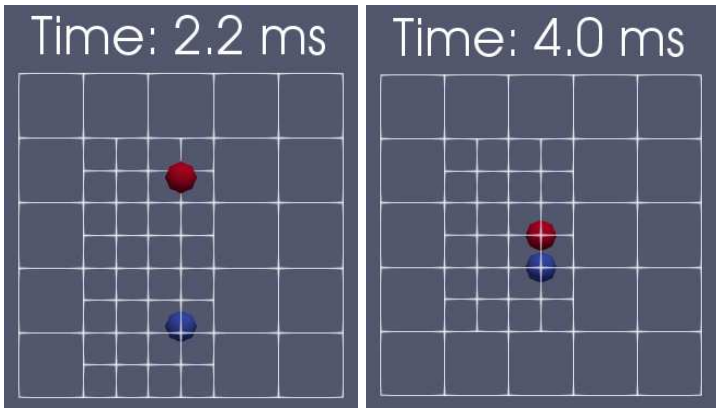
3.2.3 Dynamic AMR

Compatibility One of the main limitations of the FV-DEM coupling is that FV simulations commonly use adaptative mesh refinement (AMR), which was not supported by the lagrangian library. Such update was not foreseen in the standard OpenFOAM code so we had to implement this change in order to allow a simultaneous use of the particle collisions and AMR.

Collisions between lagrangian particles are programmed using an interaction lists based algorithm. This is, each particle has the information of the cell where its centre is, and for each of these cells a list of the neighbours is built as potential cells where a collision could happen if a colliding element (particle or wall for instance) was there. With a static mesh once these interaction lists are set there is no need to re-define them, but when using a dynamic mesh the cells and their neighbours are constantly changing and thus a reconstruction of the interaction lists is needed in each time loop.



(a) Static mesh

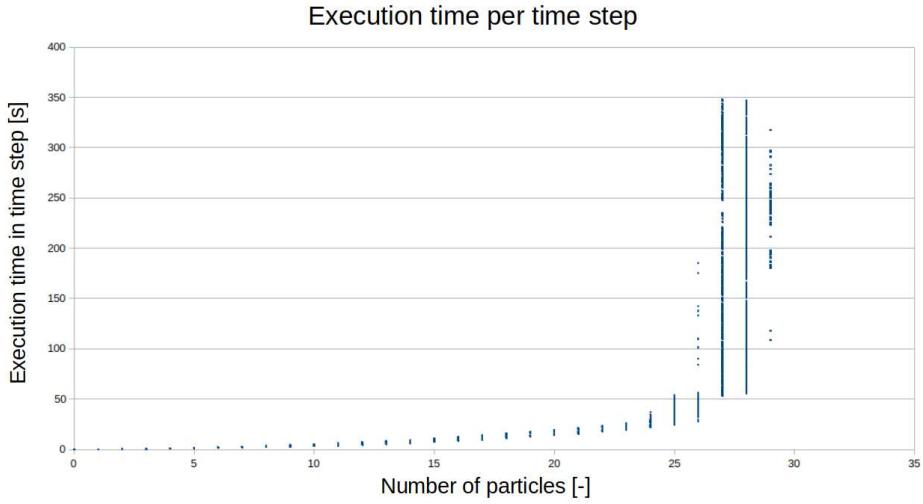


(b) Dynamic mesh

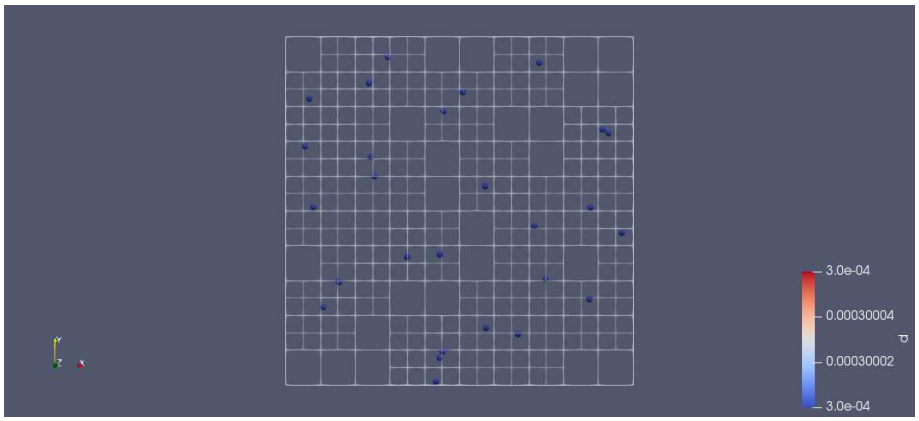
Figure 3.4: Collision of two particles

Although succesfully implemented, collisions with dynamic mesh do not increase the contact resolution. The collision resolution is controlled by the collision sub-cycles algorithm, for this reason using the mesh refinement to optimize

the collision algorithm is redundant. Since the collision between two particles is calculated using their geometrical parameters, this is, centres distance and diameters, the surrounding mesh has no influence in the procedure. On the other hand, the execution time increases greatly since the interaction lists are to be reconstructed in each sub-cycle. In a simulation with a two particle collision (Fig. 3.4) the execution times are increased from $t = 11.85s$ with a static mesh to $t = 13.82s$ with a dynamic one. Although in this example it may seem a minor problem, this difference increases exponentialy with the number of particles, as the number of cells refinement escales and so the interaction lists reconstructions.



(a) Execution time per timestep - number of particles



(b) Simulated box with particles

Figure 3.5: Collision of two particles

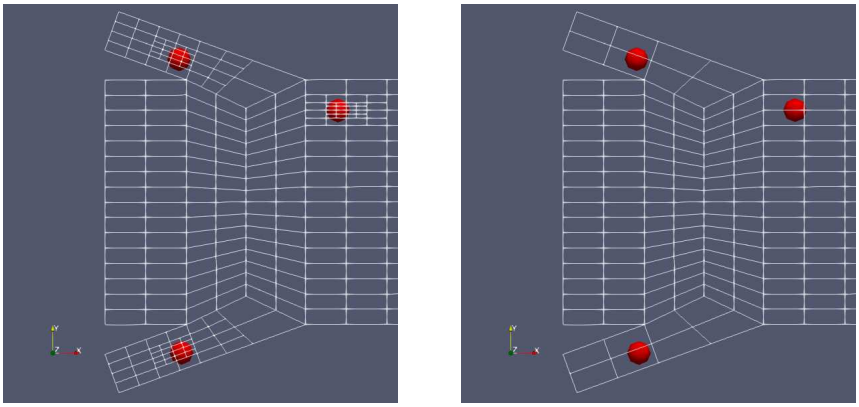
A simulation where particles are introduced gradually shows this increase in the execution time per timestep (Fig. 3.5a), a $5mm$ side box with an input/ouput in the upper boundary, a particle input rate of $1390 particles/s$ with a velocity of $0.5m/s$ simulated during $0.1s$ which needed a total execution time

of 674.84s (Fig. 3.5b), this same simulation with static mesh took 43.7s.

Although making compatible the lagrangian approach with dynamic mesh refinement is a big advantage when simulating together with VOF, it is not efficient when working with a big number of particles, which is a great number of the cases. Thus, it is advisable to work with static mesh when the case allows it to, applying the dynamic mesh refinement only if it is absolutely necessary.

Efficency To solve the bottle neck that simulating particles with dynamic mesh means, a second mesh to solve the physical interaction between particles has been implemented. Building the interaction lists is the most computationally expensive process while simulating particles with AMR, therefore building them with a secondary "invariable mesh" saves a lot of computation time for they are needed to be build once. When the simulation starts a copy of the initial mesh is made, this *invariable mesh* is passed on to the particles cloud. From this point all the movement and interaction between particles is solved with the interaction lists that were made in the first time step and are never updated. It is important to emphasize that the *invariable mesh* is only used for the physical interaction between particles, that is, for collisions, all other aspects are solved in the main mesh.

With this dual mesh method a new possibility to simulate DEM with dynamic mesh is given. Its main limitation is that the mesh never adapts to the particles, so that if for any reason a dynamic mesh is needed to solve the particles motion the dual mesh approach is not effective. For this scenario the previous approach in 3.2.3 should be used.



(a) Main dynamic mesh (b) Secondary static mesh

Figure 3.6: Dual mesh

Stability When a lagrangian particle melts there are several fields (e.g. energy, volume fraction, mass), that are passed on to the fluid volume. Until now this properties transmission is made only in the cell where the particle centre is. When a particle occupies more than a cell inconsistencies can appear, such as a sudden increase of the density, for the program is fitting the whole mass in a smaller volume; velocity peaks due to momentum conservation or energy instabilities because of the enthalpy that occupied a wider volume is now being

limited in a smaller cell. Therefore instabilities such as a drop of the Δt per timestep occur, interrupting the simulation (Fig. 3.7)

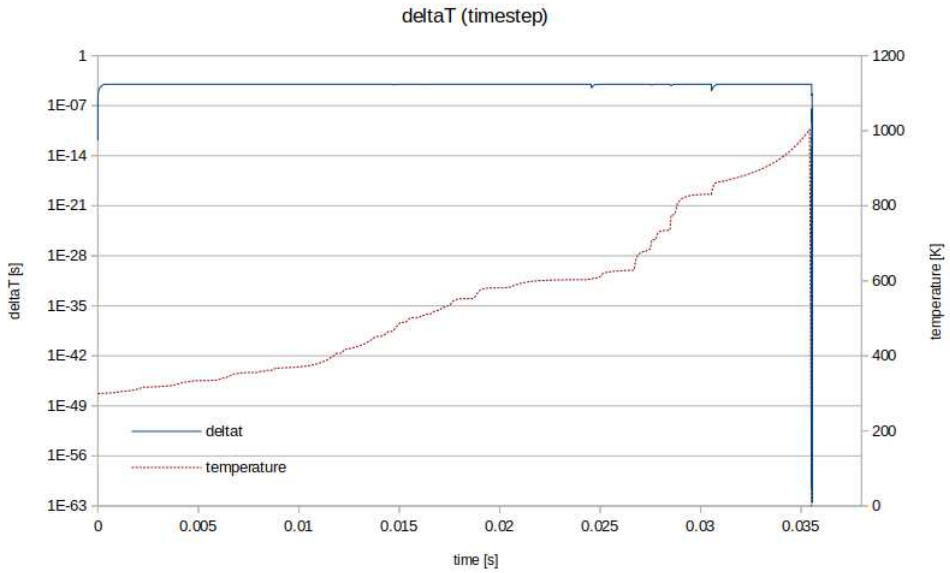
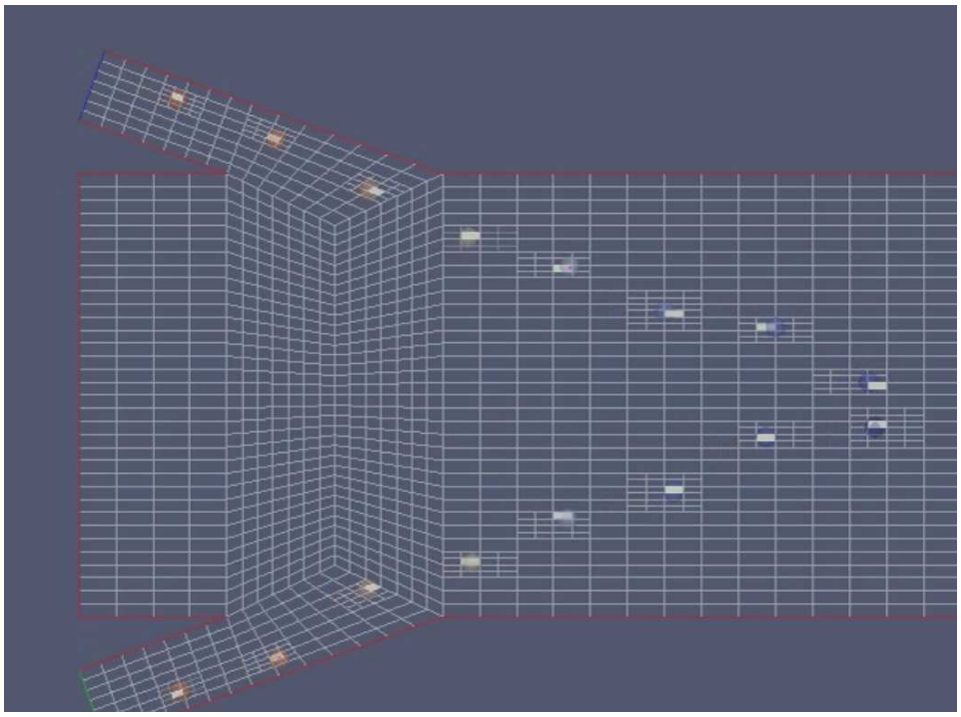
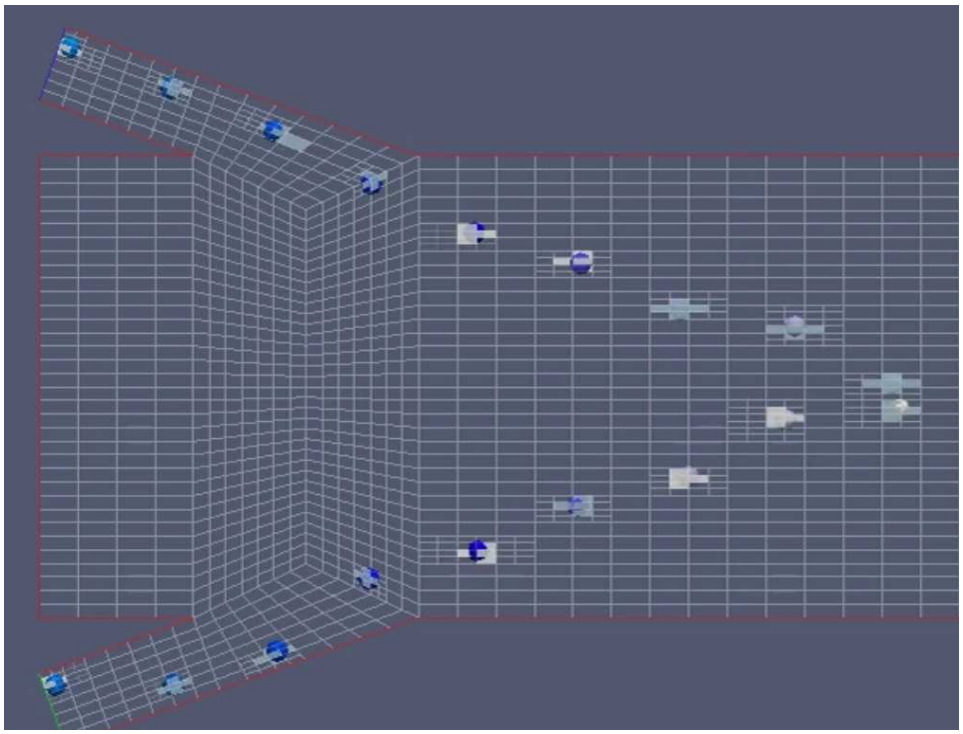


Figure 3.7: Evolution of Δt when the melting temperature (T=1000K) is reached. The instability comes when the time interval drops

Furthermore, in VOF simulations to obtain accurate results it is often necessary to achieve a fine spatial discretization (i.e. a fine mesh). In the meanwhile, the size of the lagrangian particles is imposed by the process itself (i.e. powder size), so it can turn out that, for a given case, the size of the particles becomes comparable (or even bigger) to the actual size of the cells where they reside.



(a) No multi-cell feature activated



(b) Multi-cell feature activated

Figure 3.8: When multi-cell is activated neighbour cells from the centre one are activated using the node search algorithm

In figure 3.8 the cell occupation expansion can be seen. The search for occupied cells was previously developed by J. Peternel [13]. The algorithm searchs for occupied cells in the neighbour cells of the centre one, and applies that same algorithm recursively to the ones that are occupied. This search system increase the number of operation exponentialy when it is applied to all the neighbours of al the occupied cells, for this reason is now limited to direct neighbours of the centre cell (the algorithm is only used once per particle). The search is made by comparing the normal vector of the boundary between the centre cell and the neighbour one pointing outwards, with other vector that goes from the middle of the bounday to the farthest point of the particle in the perpendicular direction of the boundary. If the normal component of both vectors points in the same direction then that cell is occupied (Fig. 3.9).

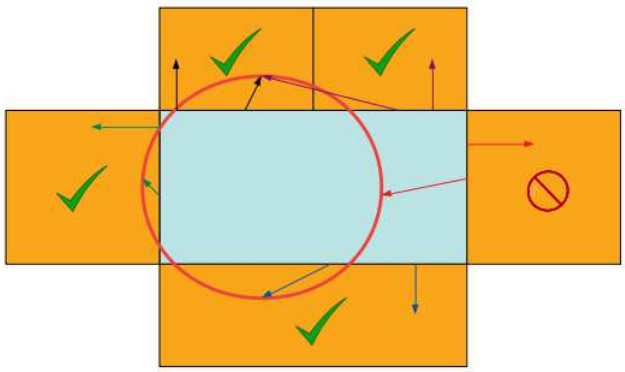
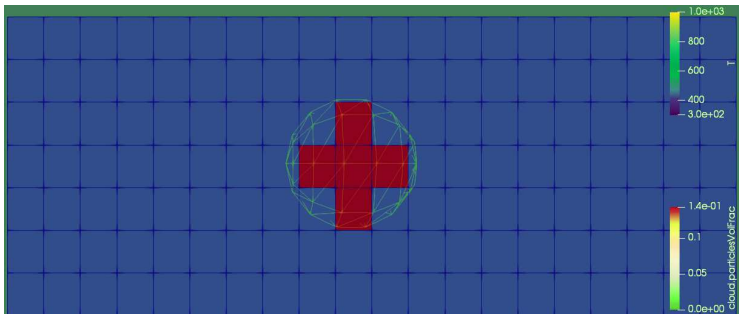
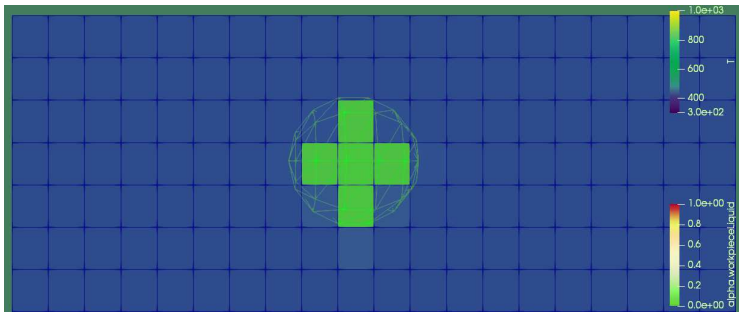


Figure 3.9: Occupied cells search algorithm [9]



(a) Volume fraction



(b) Workpiece liquid volume fraction

Figure 3.10: One particle at melting temperature simulation. The volume is divided into seven cells, having each a volume fraction of 0.1429. The liquid volume fraction (alpha.workpieceLiquid = volLiquidCell/volCell) shows liquid is generated in the occupied cells

Once the occupied cells are identified the volume fraction for each cell is determined, the calculation has been improved from

$$volFractionCell_i = \frac{1}{nOccupiedCells} \tag{3.6}$$

to:

$$volFractionCell_i = \frac{volCell_i}{\sum_{j=0}^{n} vollCell_j} \tag{3.7}$$

Having now a weighted distribution among the cells (Fig. 3.11). Then all the extensive fields whose value is to be shared by the cells is multiplied by said volume fraction.

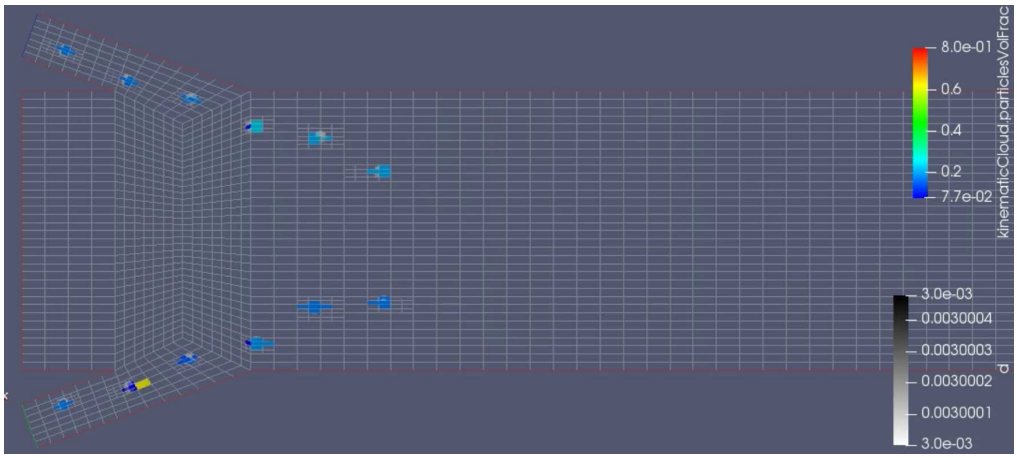


Figure 3.11: Simulation with weighted volume fraction

As a last improvement to the multi-cell feature, mass transmition has been added to the model as a new field to be given from the lagrangian particle to the VOF fraction.

Chapter 4

Application to additive manufacturing (AM)

4.1 Laser Metal Deposition - LMD

Laser Metal Deposition is a laser assisted AM-technique frequently used for maintaining or repairing high-end industrial parts such as turbine blades. LMD process presents an efficient alternative against the traditional replacement of damaged components.

The main characteristics of this process are visually summarized in Figure 4.1 A laser beam is used to melt down the surface of the part being processed. The addition material is continuously projected onto the meltpool in powder form by using a nozzle coaxial with the laser. The nozzle also injects a protective gas in order to avoid oxidation of the metallic alloys involved. Practical use of LMD involves subsequent scanning passes, which can be either on top of each other (i.e. reconstructing a wall), laterally overlapped (i.e. reconstructing a surface), or their combination.

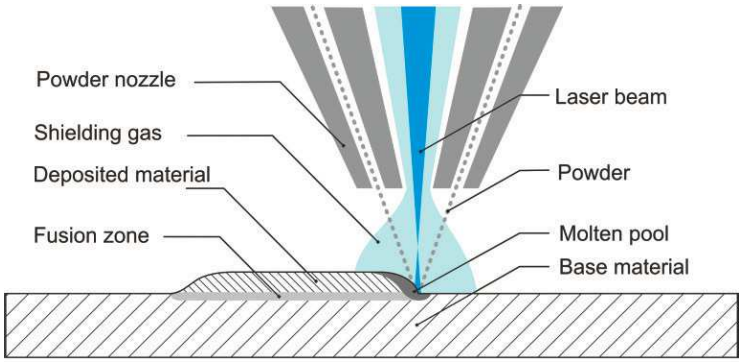


Figure 4.1: Schematic cross-section of LMD technique [8]

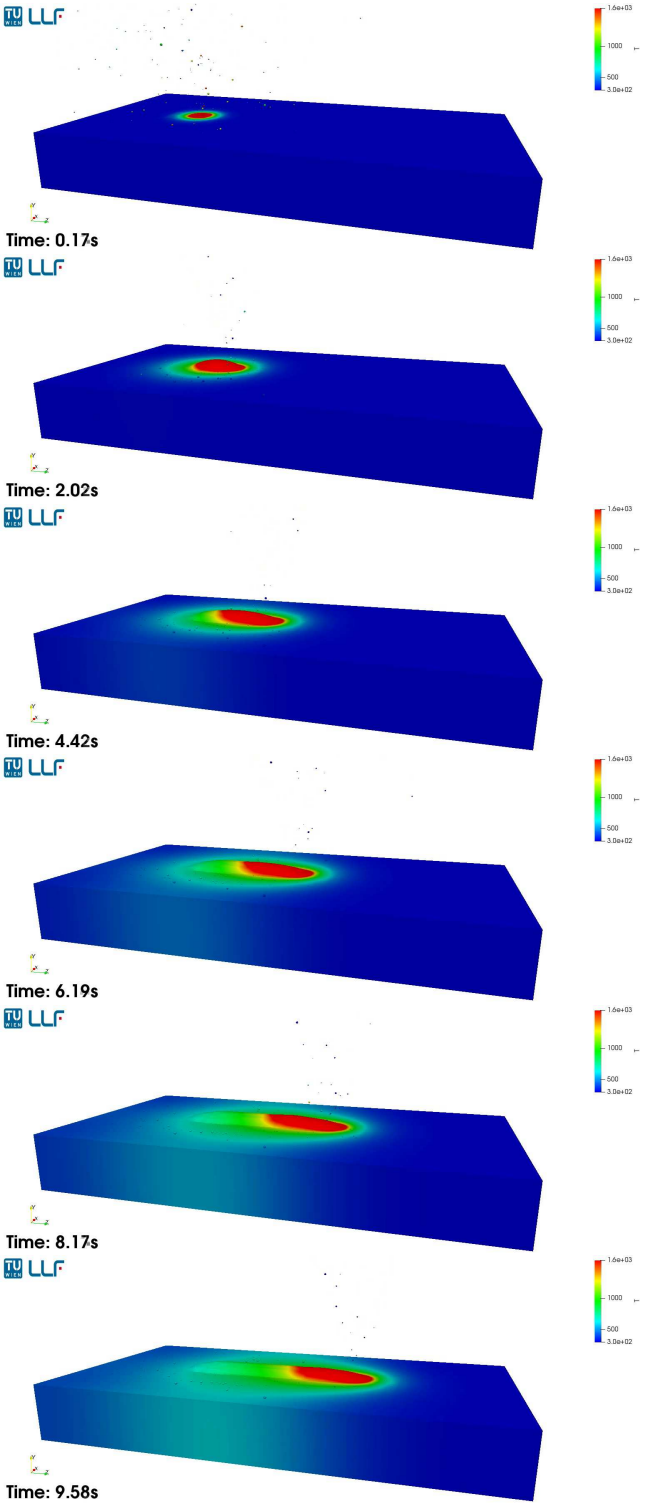


Figure 4.2: Single-pass LMD simulation performed using hereby presented advances (conditions in Table 4.1)

The overall dimensions of the final weld bead, and therefore the thermal cycles too, are determined by the powder deposition efficiency, which is mainly a question of geometrical matching of the powder stream and the meltpool surface. Particles falling outside the meltpool will bounce away and eventually leave the process region without having any effect on the mass addition. For the proper prediction of the deposition efficiency, an accurate two-way kinematic coupling (i.e. drag force model) is necessary. On the other hand, the low spatial density of particles being injected removes the importance of collisions between particles –their use is mainly for avoiding collisions inside the meltpool, where they melt down –. Similarly, the direct light absorption by the metal powder in LMD is negligible compared to the amount being coupled onto the meltpool surface.

Table 4.1: Summary of process parameter used for the benchmark test

Laser power	**Spot size**	**Scanning speed**	**Surrounding gas**	**Substrate**	**Powder composition**	**Powder feed rate**
1 kW	2.5mm (Gaussian)	100 mm/min	Ar	AISI304 (50x100x5 mm)	Inconel 718 (40-200 μm)	1.6 g/min

An exemplary simulation on LMD was carried out in order to check the prediction capabilities of the laser simulation model after improvements. Figure 4.1 shows a snapshot of the simulation results. The conditions used for this benchmark, are presented in Table 4.1 and the comparison of the simulation results against experimental is provided by Figure 4.3 along with a summary in Table 4.2.

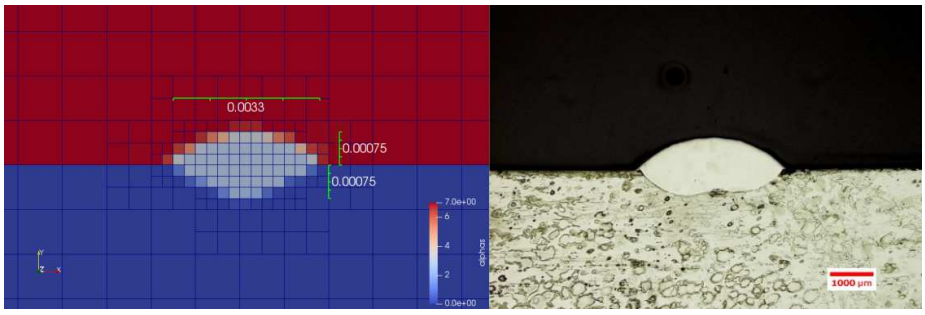


Figure 4.3: Side-by-side comparison of results between simulation and experiment [5] for the benchmark condition after finishing a single pass (see conditions in 4.1).

Table 4.2: Simulation and experiment bead comparison. Note that the accuracy is limited by the mesh size. The size of the smallest elements in this case is 0.25mm

	Bead width [mm]	**Bead heigth [mm]**	**Bead penetration [mm]**	**Powder deposition efficiency** [%]
Simulation	3.3	0.75	0.75	86.5
Experiment	3.46	0.82	0.61	85.8

$$Powder\ deposition\ efficiency(\%) = \frac{Final\ weight\ of\ plate - Initial\ weight\ of\ plate\ (before\ LMD)}{Total\ powder\ feed} X100$$

4.2 Selective Laser Melting - SLM

A SLM powder-bed fabrication process consists in the generation of a tridimensinal volume applying a bidimensinal layer at a time. This is, a metalic powder-bed layer is spread on the so-called build platform, where the laser scans a two-dimensional slice of the volume. While the powder melts it generates a melting pool that after solidification becomes part of the manufactured piece. Afterwards the build platform is lowered by the thickness of a single slice, new powder is applied and the process is repeated until the whole piece is complete.

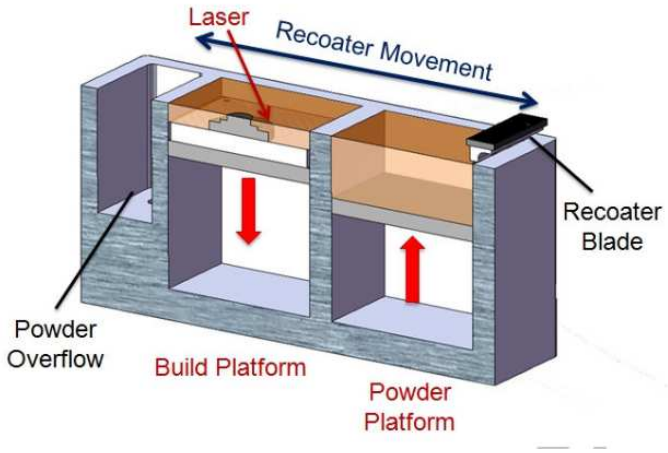


Figure 4.4: Schematic cross-section of SLM powder-bed system [2]

In Figure 4.8 a standard SLM powder-bed process is shown schematicaly. The "new" powder is moved from the powder platform by the recoater blade, which applies a layer, as thick as one slice of volume, on the build platform and throws the overflow into the powder overflow pit. Then comes the laser scan, the building platform is lowered and the process is repeated.

4.2.1 Powder bed

As explained in Chapter 1, the manufactured volume is fabricated layer by layer. To make each one of them first a powder layer is applied and then the laser scanning takes place.

One goal of this thesis is to simulate a single pass laser scan, to achieve this the simulation has two clearly differentiated parts: the powder application and the laser scanning.

In the physical process, the powder bed is achieved by spreading over a film of powder using mechanical actuators (so-called "spreaders") (Section 4.2). Since the introduction of moving meshes would considerably increase the complexity

of the simulations, a simple alternative has been slected to obtain similar results with minimum effort.

The powder bed has been generated with a solver optimized to simulate particle dynamics and then the results are used as input of the new simulation for the laser scan, this time with the laser solver. This way the complete solver, more time expensive, is only used when strictly indispensable.

The powder application is made by injecting particles from above in a box shaped domain. A big cell resolution in the incidence zone is desirable, but unfortunately the application of a fine mesh through the whole domain is not feasible as, in terms of computing resources, it would be too expensive. The handicap of using a dynamic mesh when simulating particles is that it requires the reconstruction of the interaction lists in each sub-cycle, needing much more time to simulate (Section 3.2.3). For this reason a static mesh was used, which would be afterwards inherited by the SLM simulation as the secondary static mesh in a dual mesh approach (Fig. 4.5).

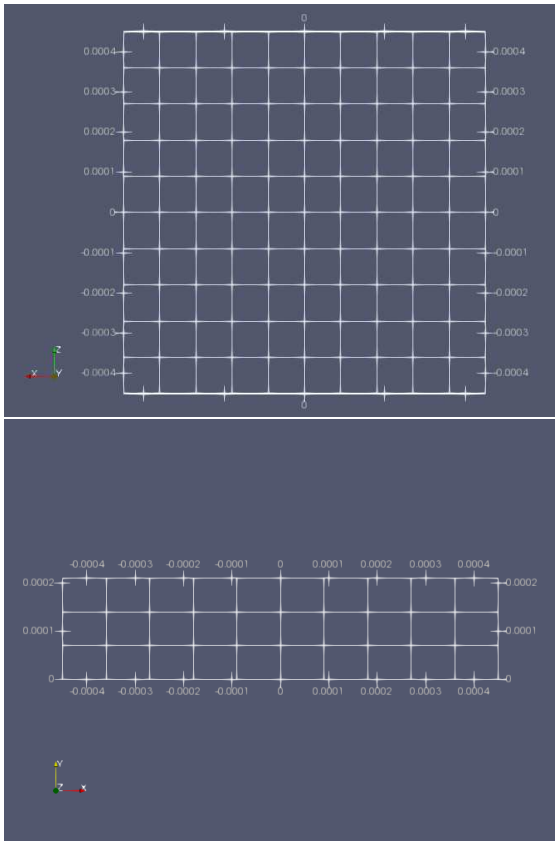


Figure 4.5: Meshed domain

The simulation results are to be compared with experimental data in literature. For this reason experiments by P.Bidare et al. [3] and M.J. Matthews et al.[12] have been selected as reference. The parameters used for the simulation can be seen in the table 4.3. A certain initial velocity ($-2m/s$) is given to the particles to accelerate the simulation, which are also dragged by gravitational

forces. Great rebounds of the particles when they collide with the boundaries and between them is undesirable as it would increase the bed settling time and the chance of the particles leaving the domain. For this reason the reconstitution coefficient has been modified, this coefficient is adjusted with the parameter α (Eqn. 3.5) and it controls the amount of energy that is lost in the collision [15]. While it should be kept as 0.01 for a elastic behaviour it has been artificially increased (more energy loss) to allow a faster settlement of the bed to 1 in collisions with walls and to 1.5 in collisions between particles. In addition, the friction of the particles when they interact with each other or with the boundaries has been minimized in order to promote the self-packing. Moreover the gravity force has been increased by a factor of ten to accelerate the simulation and reduce rebound.

Table 4.3: Powder bed generation simulation parameters

Particle mean diameter [μm]	Particle diameter range [μm]	Parameter α_{wall}	Parameter $\alpha_{particle}$	Domain dimensions [mm]	Cell dimensions [μm]
30	15 - 45	1	1.5	0.90x0.21x0.90	90x70x90

In the figure 4.6 six snapshots of the powder bed generation simulation can be seen. Particles are injected with an initial velocity, then are accelerated due to gravitacional drag (which is enhanced) and finally they collide with the lower boundary where powder bed settles. It can be seen how the interactions do not produce great rebounds, alowing the simulation to finish earlier. Once the particles reach the lower surface they keep moving for a while until they reach they reach the minimum energy state, which coincides with the state with higher compaction factor. At the end of the simulation the powder bed thickness was around $60\mu m$. Since the compaction factor was aproximately 0.5 (Table 4.4) the expected thickness of the resolidified bead after the laser scan would be half the thickness of the powder bed, that is $60 * 0.5 = 30\mu m$. A snapshot of the final state of the powder bed deposition simulation can be seen in figure 4.6, this state is the one that was afterwards used as the initial one for the SLM simulation.

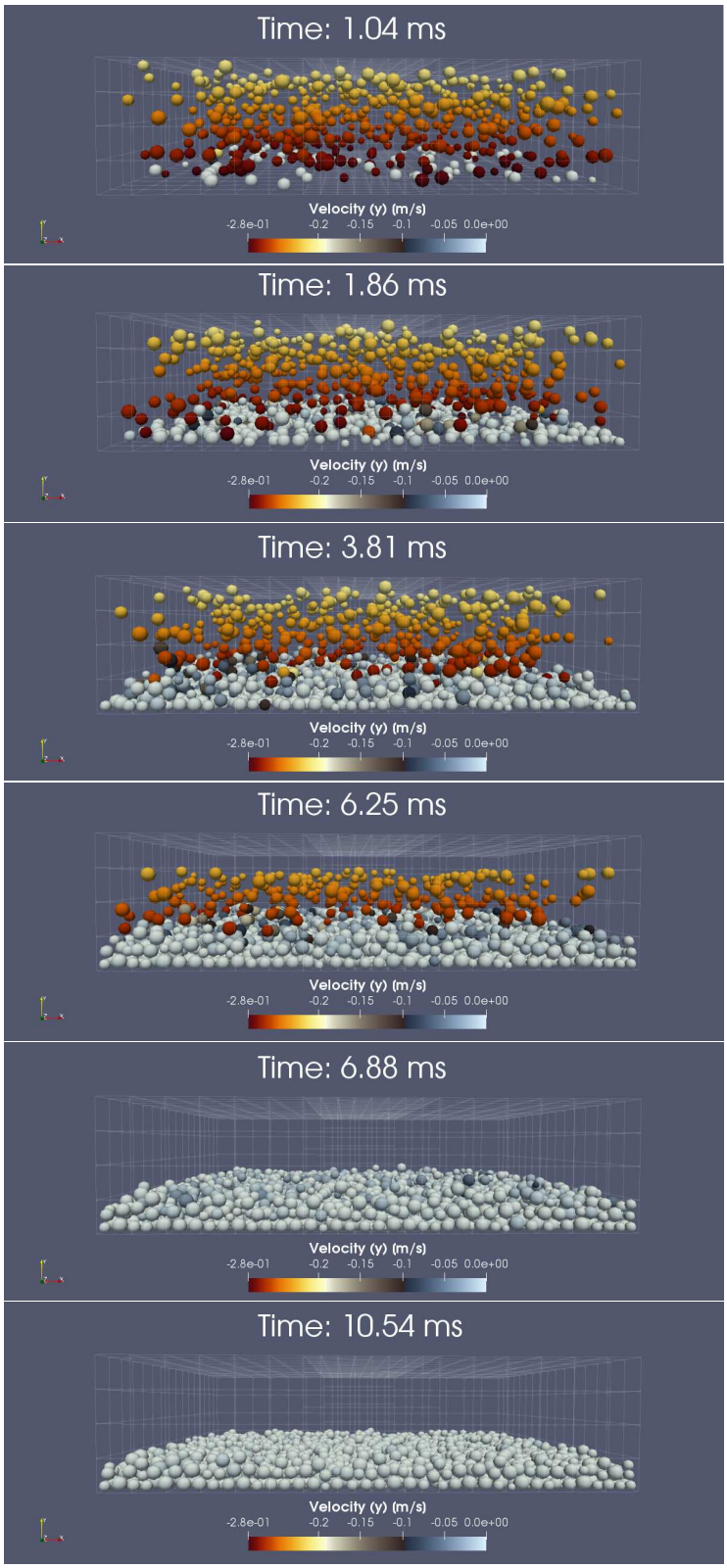


Figure 4.6: Powder bed generation process detail

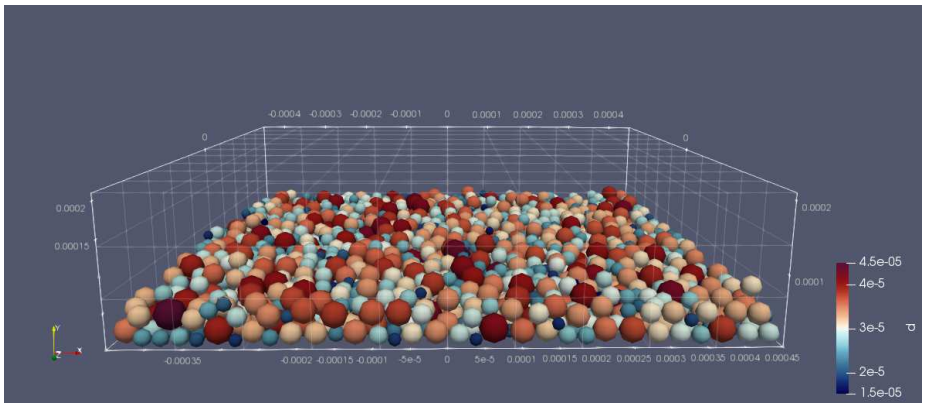


Figure 4.7: Powder bed final state

Table 4.4: Powder bed generation simulation results

Powder bed thickness [μm]	Compaction factor
~ 60	0.5

4.2.2 SLM simulation

With the generated powder bed an exemplary simulation on SLM has been made, this time to check the interaction between settled particles. As seen in [3], while scanning a powder bed the ejection of some particles is expected. In the simulation this takes place after the pass of the laser beam, due to drag forces and interaction between particles, reaching a velocity of 1 m/s. The initial parameters for the simulation are summarized in the table 4.5.

Table 4.5: SLM simulation parameters

Laser power	**Spot size**	**Scanning speed**	**Powder composition**
100 W	0.1 mm (Gaussian)	0.5 m/s	SS 304 (15-45 μm)

The meshed domain and the particles are taken from the powder bed simulation, which means both simulations share the same parameters (Table 4.3), although this time an AMR with dual mesh approach (Section 3.2.3) is used, which inherits the mesh from the powder bed simulation mesh as the secondary static mesh (Fig. 4.5). The maximum refinement level is 2, which means that the minimal dimension that cells can reach is 11.25x8.75x11.25 μm, escalated in two time steps, so one cell cannot be refined twice in the same step. The elements that trigger mesh refinement are: the presence of liquid or solid state in VOF (not as a dicrete element), interfaces between VOF and laser energy. Although for an elastic behaviour the coefficient of restitution should have been 1, it has been set to 0.5 by rising the parameter α to 0.5 in both collisions with walls and between particles [15].

Table 4.6: SLM simulation, particles parameters

Particle mean diameter [μm]	Particle diameter range [μm]	Parameter α_{wall}	Parameter $\alpha_{particle}$	Domain dimensions [mm]	Cell dimensions [μm]
30	15 - 45	0.5	0.5	0.90x0.21x0.90	90x70x90

The final simulation on SLM (Fig. 4.8) was made to check the prediction capabilities of the model, regarding particle dynamics, after improvements. The green region in Figure 4.8 indicates where the volume fraction of workpiece material (either liquid or resolidified) is higher tan 0.1, this means the weld bead is much smaller that the region, and also much smaller than expected, but nevertheless the main goal of this simulation was not to correctly predict the correct generation of a weld bead, which was explored in Section 4.1, but to check if the improvements applied to the particle dynamics and the stability of the simulations with particles worked.

The particles motion after the laser scan was the main concern in this simulation, expecting some of them to fly away due to the laser heat influence [3]. This behaviour was succesfully predicted, having in the simulation some expelled particles at the approximate velocity of 1 m/s, moreover the laser beam showed influence in the dynamics of the whole powder bed. Regarding the simulation stability, the dual mesh approach worked succesfully, enabling the possibility of simulating the dynamics a great number of particles with a secundary static mesh, while the fluid dynamics were solved with a main dynamic mesh. Since there is a huge number of interactions between particles, the chances to get instablities due to wrong overlaps were higher, nevertheless none were found. Although in the SLM simulation there was no injection of particles to check the proper application of the improvements in it, during the powder bed generation (Section 4.2.1) all the particles were injected, again with no instabilities in the injection or in the motion and settlement of the particles.

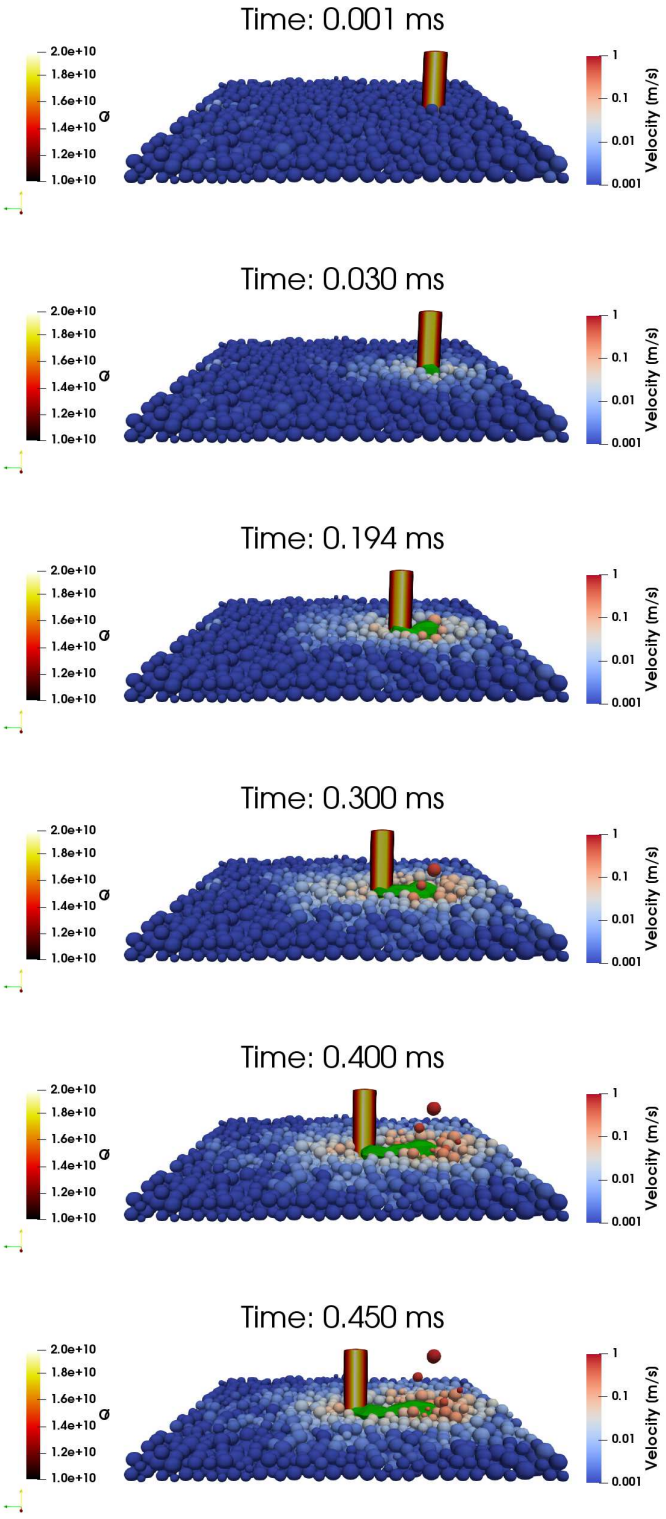


Figure 4.8: AM simulation from generated powder bed

Chapter 5

Outlook

The capabilities of a previously existing multiphysical model for laser assisted processes have been extended in order to allow the simulation of laser assisted additive manufacturing processes.

The base multiphysical model has been succesfully used over the last years to simulate laser-beam welding, cutting, remote cutting, as well as more sofisticated ultra-short pulse ablation processes.

The biggest novelty of this new approach is that it covers simultaneously phenomena from mesoscale (i.e. powder dynamics) to macroscale (i.e. welding bead generation) within the same simulation. In contrast with found publications, by means of a coupled DEM-VOF approach it has been possible to simulate the powder particles on a DEM-framework, coupled with a FV continuum approach that allows the simulation of both, particle and VOF dynamics, as well as the matter transition from one to the other.

With further work not only the improvement the accuracy of current results shall be possible, but also, the achievement of better predictions regarding the properties of resolidified material and, as a final goal, reliable results concerning defects on the production process such as residual stresses or cracking. Some specific aspects to improve are:

Load balancing Needed for a more efficient computation. Currently when using more than one processor the division of work does not take into account the amount of particles that each one is holding. For a more efficient calculation, the load each core has should be balanced taking into consideration the amount of particles in it.

Particles distribution Although the particles normal size distribution has given good results the particles do not follow an exact normal distribution. It would be beneficial to work with custom distributions, such as a Rosin-Rammler, to find more reliable results.

Physical accuracy Although the multi-cell feature 3.2.3 has been implemented to resolve stability problems, it can be used to improve some physical behaviours such as energy absorption, mass deposition during melting or momentum tranference to VOF.

Non-rigid collisions Currently the collision model only works properly whith rigid particles. When the young modulus is too low, then the calculated force is not enough to change the direction of the particle and there is no collision. Therefore a model that takes into account also the collisions of non so rigid particles is yet to be implemented.

Numerical diffusion When melting of discrete elements takes place, thus transferring the matter to VOF-system, some numerical diffusion happens. Avoiding it can lead to better results concerning molten mass, deposited energy or transferred momentum.

Pre-heated powder In several production techniques the pre-heating phase is of high significance, especially in SLM. In these steps some bonding between particles develop, this phenomena is not implemented and can reasonably improve the results.

Residual thermally induced stresses As a final goal needed to predict correctly the properties of the AM-processed parts, these residual stresses are to be simulated properly. Although this issue can not yet be adressed, it is important to keep in mind that it should be accomplished to finish developing the simulation model.

Bibliography

[1] C. Panwisawas et al. "On the role of thermal fluid dynamics into the evolution of porosity during selective laser melting". In: *Scripta Materialia, 105* (2015), pp. 14–17.

[2] L.N.Carter et al. "The Influence of the Laser Scan Strategy on Grain Structure and Cracking Behaviour in SLM Powder-Bed Fabricated Nickel Superalloy". In: *Journal of Alloys and Compounds, 615* (2014).

[3] P. Bidare et al. "Fluid and particle dynamics in laser powder bed fusion". In: *Acta Materialia* 142 (2018), pp. 107–120.

[4] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation*. Reading, Massachusetts: Adam Hilger, 1991.

[5] *Courtesy of Mitsubishi Heavy Industries, Ltd.*

[6] Nedderman E.I. "Stokes' Law For Solid Spheres and Spherical Bubbles". In: *Thermopedia* ().

[7] The OpenFOAM Foundation. *openFOAM*. URL: `https://openfoam.org/`.

[8] Benjamin Graf et al. "Design of Experiments for Laser Metal Deposition in Maintenance, Repair and Overhaul Applications". In: *Procedia CIRP* 11 (2013). 2nd International Through-life Engineering Services Conference, pp. 245–248.

[9] Courtesy of J.Peternel.

[10] K. L. Johnson. *Contact mechanics*. Cambridge University Press, 1985.

[11] J.C. Gebelin L.L. Parimi M.M. Attallah and R.C. Reed. "Direct laser fabrication of Inconel-718: effects on microstructure and distortion". In: *Powder Technology, 71* (2012), pp. 511–519.

[12] Manyalibo J. Matthews et al. "Denudation of metal powder layers in laser powder bed fusion processes". In: *Acta Materialia* 114 (2016), pp. 33–42.

[13] J. Peternel. "Model of Powder Deposition and Melting on the Melt Pool Surface". University of Ljubljana, Sept. 2017.

[14] R.M.Young and E.Pfender. "Nusselt number correlations for heat transfer to small spheres in thermal plasma flows". In: *Plasma Chem Plasma Process* (1987).

[15] T. Tanaka Y. Tsuji and T. Ishida. "Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe". In: *Powder Technology, 71* (1992), pp. 239–250.