

The Numerical Simulation of a Steel Converter
Using Smoothed Particle Hydronamics Coupled
with the Discrete Element Method

Travis Young

December 5, 2016

Abstract

In the steel making process, high carbon content liquid pig iron is poured into a basic oxygen converter. Oxygen is top blown onto the liquid steel surface, from a lance, at high velocities. The oxygen reduces the carbon content in the liquid metal by forming carbon dioxide and carbon monoxide gas which rises out of the converter. Due to the extremely high temperatures, the behaviour within the converter cannot be observed directly. To gain insight into the activity inside the converter, numerical simulation methods are required. Two numerical methods called Smoothed Particle Hydronamics (SPH) and the Discrete Element Method (DEM) have been coupled to model the multi-phase fluid behaviour of this system. SPH is used to model the boundaries, the incompressibility, the surface tension, and the viscosity of each fluid while DEM is used to model the interaction between the molten steel and the oxygen at the interface. Originally a pure SPH code was implemented, however due to the large density gradient at the steel-oxygen interface, spurious and unnatural behaviour was observed. To address these issues, the steel-oxygen interaction was instead modelled with DEM type contact forces. DEM's normal, damping and tangential forces that are calculated between the steel and the oxygen particles are not related to the fluid density, and thus avoid pressure based issues that arise from large density variations. The code was written in C++ in the Codeblocks IDE and compiled on Linux. The contents of this paper will describe the SPH and the DEM methodologies in detail as well as discuss their application in the simulation of the basic oxygen converter.

Contents

1 Computational Fluid Dynamics (CFD)	3
1.1 The Navier-Stokes (NS) Equations	4
2 Smoothed Particle Hydronamics (SPH)	5
2.1 The Smoothing Kernel	5
2.1.1 Positivity	6
2.1.2 Compact Support	6
2.1.3 Normalization	6
2.2 The SPH Equations	7
2.3 Pressure	8
2.4 Viscosity	8
2.5 Gravity	9
2.6 Surface Tension	9
2.7 Boundary Particles	11
2.8 Pros	12
2.9 Cons	12
3 The Discrete Element Method (DEM)	13
3.1 The Contact Model	13
4 Coupling SPH and DEM	15
4.1 The Large Density Gradient	15
4.2 The Large Velocity Gradient	16
4.3 Using DEM at the Interface	16
5 The Steel Converter	18
5.1 Basic Oxygen Steelmaking	18
5.2 Implementation	18
5.2.1 Pseudo code	19
5.2.2 Initialize Parameters	19
5.2.3 Neighbour Search	19
5.2.4 Calculate Density and Pressures	20
5.2.5 Calculate SPH and DEM Forces	20
5.2.6 Accelerate Particles	20

5.2.7	Output	20
5.3	Particle Types	21
5.4	Particle Deletion	21
5.5	Geometry	21
5.6	The Lance	21
5.7	Visualization	22
5.8	Stability	22
6	Conclusions	23

Chapter 1

Computational Fluid Dynamics (CFD)

Computational Fluid Dynamics (CFD) is the utilization of computers to solve the Navier-Stokes (NS) Equations for fluid dynamics problems. Since the 1960s, computers have been powerful enough to model high turbulence fluid systems. With recent technological advances in computing, the field of CFD continues to grow and gain more and more importance in technical projects while being able to run simulations more quickly and affordably than ever before.

CFD methods are broken up into two general categories; Eulerian (mesh-based) CFD and Lagrangian (particle-based) CFD. Both the SPH and DEM methodologies are of the latter variety. The difference between Lagrangian CFD and Eulerian CFD boils down to the relative location of the observer. A Eulerian perspective can be described as focusing on stationary locations in space and observing the field properties such as velocity, density etc as the fluid body passes through these points. Naturally the tighter knit the mesh, the more accurate the results that a simulation can return. The weakness of grid based methods lies in the definition of complex geometries and calculating properties at fluid interfaces. Fluid interfaces can be especially challenging when using Eulerian techniques, as they are extremely complex and can be hard to predict.

A Lagrangian perspective on the other hand is not fixed in space, rather the properties of a small discrete volume of the fluid are observed as it is displaced in time. Lagrangian methods are of particularly use in free surface simulations, which can be difficult to model with Eulerian methods. If we consider the locations where properties are determined as mesh points, then a Eulerian mesh would be a static non changing mesh , whereas a Lagrangian mesh would be dynamic, constantly deforming with the shape of the fluid body. Figure 1.1 provides a comparison between these two perspectives. Software packages such as ANSYS FLUENT and STAR CCM, are commonly used Eulerian CFD programs, whereas the open source SPHysics seems to be the most popular SPH program.

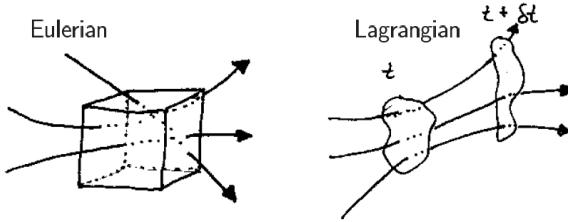


Figure 1.1: Eu7lerian vs. Lagrangian

1.1 The Navier-Stokes (NS) Equations

The Navier-Stokes (NS) Equations, a set of partial differential equations that describe free surface fluid flow, were derived by Claude-Louis Navier and George Gabriel Stokes in the 1840s. The set of equations was originally discovered through fundamental conservation laws (momentum, mass, continuity and energy) and first order approximations, however later it was also shown that alternatively the NS Equations could be derived from molecular dynamics in the early 1900s as well as through cellular automata in the 1980s. Despite the fact that the equations are considered to be among the most important equations ever formulated, a unique three dimensional solution (or anti- solution) has yet to be found. The search for a solution to the Navier-Stokes Equations has led to them being included in the Clay Mathematics seven Millennium Prize Problems which offers a million dollars for the first valid solution. Even without a unique solution the equations can effectively be utilized through discretization and approximation techniques.

Chapter 2

Smoothed Particle Hydronamics (SPH)

SPH is a Lagrangian CFD method that was initially developed for astrophysical applications by Gingold, Monaghan and Lucy in 1997. Since its original formulation, SPH has been adapted and modified to fit a variety of physical scenarios including the modelling of multiphase fluid flow and compressible solid mechanics. SPH is an interpolation method which determines field values such as density, pressure and viscosity based on the weighted summations of neighbouring particles' properties. By using interpolated values for ρ etc the Navier-Stokes equations can be discretized and solved without exhaustive computation. Like all Lagrangian particle-based methods, SPH particles can carry information such as, but not limited to; density ρ , pressure p , mass m colour field etc.

The purpose of SPH in this simulation is to model the incompressibility and the motion of the oxygen and molten steel fluid bodies. SPH forces are only calculated with particles of the same type and boundary particles . The interaction between the steel and oxygen at the interface is entirely modelled through a DEM contact model. DEM's handling of the interface is discussed in chapter 4.

2.1 The Smoothing Kernel

The key feature of SPH is its interpolation of field values, which is based on the weighted sum of neighbouring particles. The contribution of each neighbouring particle is determined by the "smoothing kernel" (sometimes also referred to as the smoothing function). Several smoothing kernels have been designed since the birth of SPH, with different applications in mindThe stability of the fluid system is heavily reliant on the choice of smoothing kernel, and naturally different fluid systems will require different properties in a kernel. Below is outlined the kernels used in this program.

To determine a general scalar field value A , at a particle's location, the particle will take in weighted contributions from surrounding particles that fall within its domain of influence. This domain is defined by the "smoothing radius", which is extended radially outwards from the particle. The smoothing kernel takes in two parameters; the smoothing radius h as well as the distance to the neighbouring particle, Δr . The following are some of the kernels used in this code. The poly6 kernel is a general kernel.

$$W(r) = \frac{315}{64\pi h^9} (h^2 - r^2)^3 \quad (0 \text{ for } r > h) \quad (2.1)$$

The gradient of the spiky kernel is used for calculating pressure forces.

$$W(r) = \frac{15}{\pi h^6} (h - r)^3 \quad (0 \text{ for } r > h) \quad (2.2)$$

The laplacian of this viscocity kernel is used to determine the viscocity forces.

$$W(r) = \frac{15}{2\pi h^3} \left(\frac{-3r}{2h^3} + \frac{2}{h^2} - \frac{h}{2r^3} \right) \quad (0 \text{ for } r > h) \quad (2.3)$$

Smoothing kernels must satisfy the following properties:

2.1.1 Positivity

$$W(r - r_j, h) \quad \text{if } (r - r_j) < h \quad (2.4)$$

Within the domain of influence all neighbouring particle neighbours add a positive contribution to the interpolation summation.

2.1.2 Compact Support

$$W(r_i - r_j, h) = 0 \quad \text{if } (r - r_j) > h \quad (2.5)$$

Outside of the domain of influence, particles will not add contributions to the interpolation summation.

2.1.3 Normalization

$$\int W(r_i - r_j, h) dr = 1 \quad (2.6)$$

The smoothing function is normalized to unity as described above. The combination of these three requirements shows that the smoothing kernel's behaviour is similar to that of Dirac's delta function.

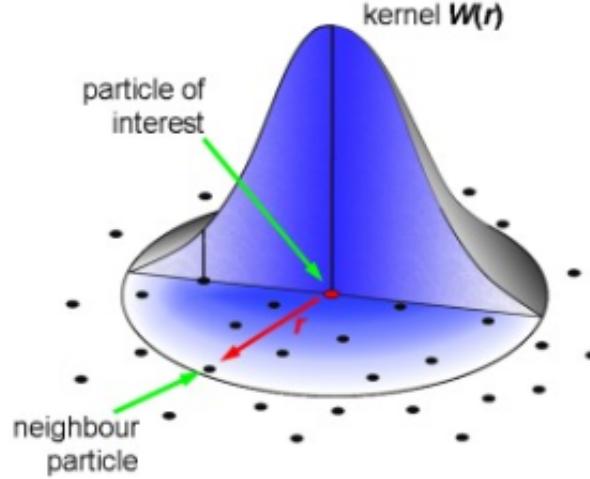


Figure 2.1: The smoothing kernel

2.2 The SPH Equations

Below is a general scalar field value A , plugged into the SPH equation.

$$A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} W(r - r_j, h) \quad (2.7)$$

Where $A_s(r)$ is the field value, m_j is a neighbour particle's mass, A_j is the neighbour particle's field value, ρ is the neighbour particle's density and $W(r - r_j, h)$ is the smoothing function that takes in the distance between particles $r - r_j$ and the smoothing radius h in as parameters. To determine the value of a specific field value $A_s(r)$ (eg. density, colour field etc.) at a particle location x, y, z first take the sum of each neighbouring particle's field value as j iterates through all the neighbours that lie within the domain of influence. Then multiply this by the smoothing function which determines the neighbour's contribution based on the distance of separation. The smoothing radius has compact support, meaning that if the distance between the particles is greater than the smoothing radius, the contribution of the neighbour particles will be zero. The closer the neighbour to the centred particle the larger the contributions to the field value summation.

$$\nabla A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(r - r_j, h) \quad (2.8)$$

$$\nabla^2 A_s(r) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(r - r_j, h) \quad (2.9)$$

Similarly the gradient and the Laplacian can be expressed as above. A nice property of the SPH formulation is that when taking the field derivatives, only the derivative of the smoothing function needs to be determined.

Let's look at an example at how the field value ρ density can be obtained through the SPH formulation.

$$\rho(r) = \sum_j m_j \frac{\rho_j}{\rho_j} W(r - r_j, h) \quad (2.10)$$

Plugging ρ density into the SPH equation gives us the following, which can later be used to calculate the pressure on the centred particle.

$$\rho(r) = \sum_j m_j W(r - r_j, h) \quad (2.11)$$

2.3 Pressure

The pressure is linked to the density through the ideal gas equation. The pressure term prevents the fluid particles from collapsing in on themselves and preserves the fluid volume. Often the Tait equation is used instead of the Ideal Gas Equation as it has been seen to provide good incompressibility, albeit at the cost of computational time and power.

$$p = k(\rho - \rho_0) \quad \text{IdealGasLaw} \quad (2.12)$$

Where p is the pressure, k is the gas stiffness, ρ is the calculated density and ρ_0 is the rest density. The stability of the fluid system is dependent on the pressure and viscosity term. The pressure force is a spring-type force, which works to restore the fluid body to its rest density, while the viscosity acts as a damping force, removing energy from the system. The difference between ρ , the calculated density and ρ_0 , the rest density determines the magnitude of the pressure force. Realistically liquids are nearly incompressible, however maintaining incompressibility with SPH can be particularly challenging. Ideally k would convey a real world gas stiffness coefficient, however in simulations large k values can result in extreme numerical instability. The limiting factor is the size of the time-step, the smaller the time step the better the compressibility. The best that can be done is to assign a value for k that is as large as possible given a particular time step size. Once the pressures are known the pressure force term from the Navier Stokes Equations can be determined as follows:

$$F_i^p = \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(r_i - r_j, h) \quad (2.13)$$

2.4 Viscosity

The viscosity term is extremely important for numerical stability. Without a damping force like viscosity, the particles will continue to gain energy until eventually the whole system explodes. Some SPH methods externally damp the system, for example, methods that do not use boundary particles often explicitly reflect and dampen particle velocities at boundaries. Such methods

are generally not physically accurate. For the implementation described in this paper, boundary particles without velocity damping were used.

$$F_i^v = \mu \nabla^2 v(x_i) = \sum \frac{\mu_i + \mu_j}{2} \frac{v_j - v_i}{\rho_j} \nabla^2 W(x_i - x_j, h) \quad (2.14)$$

2.5 Gravity

Gravity is simply implemented as an acceleration field that acts on all particles within the domain. Sometimes it is useful to perform simulations with gravity off to gain insight into the behaviour of the fluid. For example creating a cube out of particles in zero-gravity, shows the effect of surface tension as the cube slowly becomes a sphere. The surface tension acts inward to minimize the curvature.

2.6 Surface Tension

Surface tension is modelled through the usage of a Continuum Surface Force (CSF), a volume force which acts on a region near the interface of two fluids, also known as the fluid surface. The interface locations at which the CS forces act, are determined through the use of "colour fields". Neighbouring particles that are of the same fluid type as the centred particle have a colour field value explicitly equal to 1, while different particle types or void space implicitly has a colour field value of 0.

To get the smoothed colour field , the weighted summation of neighbouring colour fields, 1 is simply plugged into the general SPH equation at fluid particle locations for fluids of the same type and 0 is (implicitly) input in at void space or for particles of a different type. Intuitively the smoothed colour field is simply a weighted summation of how many same type neighbouring particles are located within a centred particle's smoothing radius. At interface locations the smoothed colour field will be significantly lower than that within the body of the fluid. By determining the gradient of the smoothed colour field, the spatial derivative of the colour field, the surface can be determined. The surface is a location where there are sharp changes in the centred particle's smoothed colour field value.

Now that the locations of the interface (surface) particles have been determined, the next step is to calculate the force that acts on them.

$$F_s = \sigma \kappa \hat{n} \delta_s \quad (2.15)$$

Where σ is a surface tension coefficient, κ is the curvature at the surface location, \hat{n} is the direction, perpendicular to the surface that acts towards the center of the fluid, and δ is a Dirac delta function. First, to find \hat{n} , the normalized normal vector, we need to find the surface normal.

$$n = \nabla c_s \quad (2.16)$$

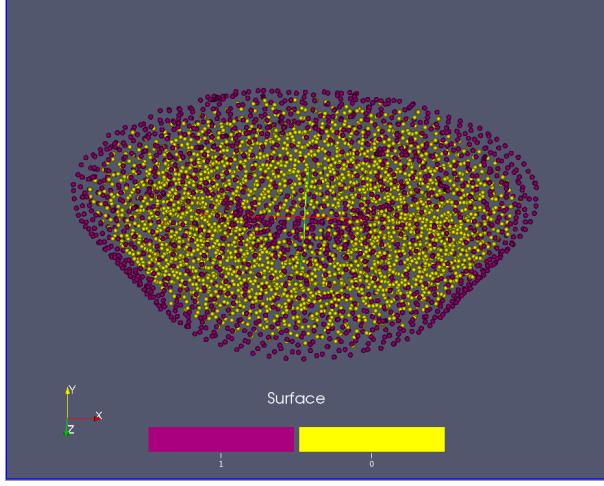


Figure 2.2: The purple particles represent the surface of the fluid, while the inner particles are yellow.

To get a normalized unit vector we divide by the magnitude.

$$\hat{n} = \frac{n}{|n|} \quad (2.17)$$

Once the normalized unit vector is known, the curvature κ can easily be determined. κ is the gradient of the normalized unit vector, in the direction at which the surface tension force points in. However it may be more intuitive to think of it as the laplacian (the second field derivative) of the colour field, the rate at which the change in the smoothed colour field increases or decreases. The gradient of the colour field provides insight into where the colour field values change abruptly. Locations where the change is sudden and large usually indicate a surface particle. The laplacian, the second spatial derivative therefore provides some insight into the shape of surface, or how it changes in space. A larger curvature value results in larger surface tension force because more force is needed to smooth out a shape with a larger curvature.

$$\kappa = \nabla \hat{n} = \frac{-\nabla^2 c_s}{|n|} \quad (2.18)$$

Finally put it all together:

$$F_s = \sigma \kappa \hat{n} \delta_s = -\sigma \nabla^2 c_s \frac{n}{|n|} \delta_s \quad (2.19)$$

SPH's compact support requirement, that the neighbouring particles' influences fall to zero as they exit the smoothing radius, naturally fulfils the CSF δ function requirement. As such the surface tension force can be simply described as

$$F_s = -\sigma \nabla^2 c_s \frac{n}{|n|} \quad (2.20)$$

where there is a larger contribution when particles are closer and no contribution from particles that lie outside of the smoothing radius.

2.7 Boundary Particles

The boundary conditions are handled through the usage of "boundary particles". Boundary particles are fundamentally the same as any other fluid-type particles, they interact with all other particle types and take weighted summations of neighbouring particles' field values. They carry the same properties such as mass, pressure, density and so on. The one difference with boundary particles is that at the end of the time step the boundary particles are not accelerated like other particles. Instead they remain stationary unless moved by an external force. Boundary particles are a convenient way to maintain boundary conditions as the calculations needed for the boundary particles are included in the main SPH loop. For this reason, boundary particles can be faster than other boundary methods. Boundary particles however, are not without downsides. Boundary particles do not work well at short time steps. At short time step intervals, if a fluid particle is travelling fast enough, its density may not spike quickly enough when it approaches a boundary particle. If the velocity is too high and/or the time step is short, the fluid particles may pass through the boundary particles. Boundaries are modelled as a two layers of boundary particles slightly offset to discourage leakage. Because of the need for two layers, complex geometry can often be hard to model with boundary particles. Moving boundaries such as the spinning of an industrial mill for example can also be hard to model without leakage. As a result simple or simplified geometry works best with SPH.

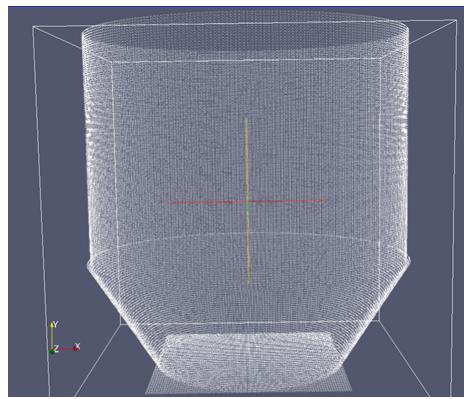


Figure 2.3: The converter boundaries

2.8 Pros

- Pressure terms are calculated based on the difference between a particle's calculated density and its rest density. Unlike mesh-based methods, that require the computational solving of linear systems of equations, SPH relies on the neighbouring particles to determine the pressure term as well as other field values.
- Mass conservation is guaranteed since each particle carries a set mass .
- Fluid boundaries do not need to be expressed explicitly. Multi-fluid systems will naturally separate and create free surfaces, eliminating the need for complex meshes.
- Computationally cheaper compared to eulerian methods. SPH can be used for real time and even interactive simulation (albeit sometimes at the cost of accuracy).
- Can handle irregular geometry in a fluid body well, unlike a eulerian implementation where the mesh is fixed in space, limiting the range of analysis for problems with complex and changing geometry.
- Can obtain time-history data such as velocity, momentum etc at any fluid particle location unlike a eulerian implementation which can only obtain information at stationary grid nodes.

2.9 Cons

- Systems with large density gradients can produce spurious unnatural pressure forces.
- High velocities require a much smaller time step which leads to much more computation time and power.
- Complex boundary geometry can make it difficult to prevent leakage when using boundary particles.
- Hard to ensure a high level of incompressibility and numerical stability.

Chapter 3

The Discrete Element Method (DEM)

The Discrete Element Method is another particle-based method. Traditionally it is used to simulate solid particles in granular flow, powder, and rock mechanics. However, when it is coupled with fluid simulation methods it can prove to be quite useful in the handling of extreme density gradients. In its coupled form, it is commonly used in suspended sand in water simulations. However, it can also be applied to gas-liquid systems such as that of a steel converter, where the density of the liquid steel is over seven thousand times denser than the oxygen that exits the lance. The reason that large density gradient systems can be more easily modelled when coupled with DEM is because its field values do not rely on density. The effect of one neighbouring DEM particle's force on another is purely based on the proximity of the two. Because of this property, spurious forces related to high density gradients can be completely avoided when using DEM. For the converter simulation, DEM contact forces are calculated between the steel and oxygen particles that collide at the interface. Both the fluid motion of the oxygen and the steel are modelled entirely using SPH. The magnitude of the interaction between the steel and the oxygen particles must be appropriately set to model the desired system.

3.1 The Contact Model

DEM models contact force interactions between DEM particles. Although DEM is seen as a family of methods that follow the same proximity based force summation ideology, the contact models can vary greatly depending on the nature of the system. Typically the contact model contains a normal force between the DEM particles, a damping force for stability as well as a tangential force to describe rolling and frictional behaviour. The contact model that was used in the steel converter simulation is described below.

The Hertzian Dashpot Force is the force that acts between DEM particles

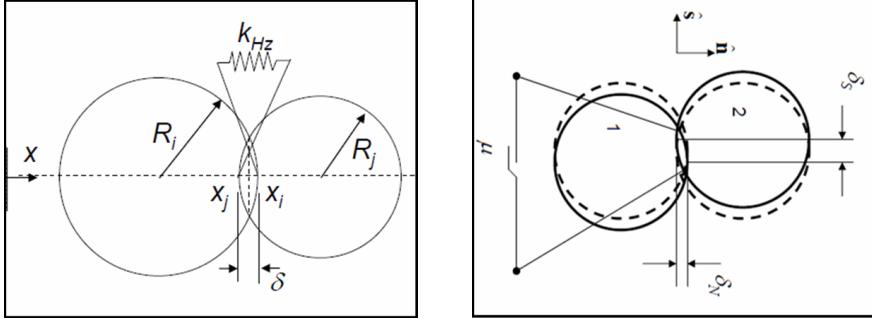


Figure 3.1: Normal Force

Figure 3.2: Tangential Force

in the normal direction. The force is similar to the SPH pressure force, as they both act like damped springs.

$$F_N = K_h \delta_n^{\frac{3}{2}} \quad (3.1)$$

Where K_h Is the Hertzian stiffness and δ_n is the particle overlap in the normal direction.

$$\delta = \left| d - \frac{H}{2} \right| \quad (3.2)$$

Generally δ , the overlap of the DEM spheres, can be described as $\delta = |d - (r_1 + r_2)|$ where $r_1 + r_2$ are the sums of the radii of both particles and d is the distance between their centres. In this DEM implementation, the radius of each particle is $H/4$ where H is the same smoothing radius defined for our SPH methodology.

The tangential force, which helps simulate frictional and rolling behavior, can easily be calculated once the normal force has been determined.

$$F_T = \mu |F_N| \quad (3.3)$$

Where F_N is the previously calculated Hertzian force which acts along the normal, and μ is a friction coefficient.

A damping force that acts along the normal is necessary to ensure numerical stability.

$$\nabla F_D = \eta \Delta v \delta_n^{\frac{1}{4}} \quad (3.4)$$

Where η is a damping constant and δ_n is the overlap in the normal direction.

Chapter 4

Coupling SPH and DEM

The final simulation of the steel converter was implemented with a DEM coupled SPH code. SPH is used to model all interactions between particles except those at the interface between the steel and oxygen, which is handled by DEM. Prior to this final code, a pure SPH code was used for the simulations. SPH forces were calculated between all particle types (molten steel, oxygen and boundary) at all locations within the domain. The pure SPH code had instabilities and showed unnatural behaviour at the steel-oxygen interface. Oxygen particles would pass into the steel body at high velocities, hardly affecting the shape of the steel surface. At other times oxygen particles would stick to the surface of the steel rather than being repelled off as one would expect. There are two underlying reasons as to why a pure SPH formulation cannot be effectively used to model the interface of this system.

4.1 The Large Density Gradient

The first factor is the large density variation between the oxygen and steel particles at the interface. The density ratio between oxygen and molten steel is above 7000. High density variations which fall within the smoothing radius can cause unnatural smoothed density fluctuations which result in falsified pressure values. The SPH pressure force works to maintain incompressibility, by trying to return the smoothed density back to the fluid's rest density (the density at which the fluid is incompressible). When using pure SPH, low density oxygen particles add high density steel particle densities to their SPH summation which results in smoothed density values that differ greatly from the rest density. Pressure values resulting from such density fluctuations are enormous and are the cause of extreme numerical instability and unphysical behaviour.

It should be noted that as described by Muller et al (2005), miscible fluids with density ratios larger than 10 cannot be realistically simulated using a standard SPH density summation. In that paper, Mueller describes a method to model rising bubbles in water. He achieves a visually realistic simulation by

reducing the density difference between air and water by a magnitude of 10, and by adding an artificial buoyancy force to the air particles. Such a method could not be used in this paper as it cannot model the complex collisions that occur at the steel-oxygen interface.

4.2 The Large Velocity Gradient

The second factor that makes the pure SPH code inappropriate for this simulation is the large velocity gradient. The oxygen particles travel at a very high speed relative to the initially stationary steel particles. High velocities are not handled well at the interface, as oxygen particles have a tendency to pass through the surface before a pressure forces are exerted on them. This kind of issue can theoretically be resolved by reducing the size of the time step to an appropriately small value. However, given the scope of the magnitude of this simulation ($\approx 400,000$ particles) and the available computing power, further reducing the timestep is not an option. Even if further reduction of the timestep were possible, it would still leave the high density issue unattended.

4.3 Using DEM at the Interface

To address the unnatural behaviour at the interface between the oxygen and the steel particles, a DEM type contact model was implemented in the place of SPH. Simply put DEM handles the interactions at the interface while SPH is left to handle everything else. During the SPH neighbour search, a centred steel particle will add densities from other steel particles and boundary particles to its SPH density summation while ignoring oxygen particles altogether. Conversely oxygen particles will neglect steel particles and will only add other oxygen particles and boundary particles to the density summation. By neglecting the opposite fluid-types during the density summation, the spurious pressure forces resulting from large density spikes are eliminated. After the density and pressure values are determined, a separate neighbour search is performed to sum the DEM forces between the oxygen and the molten steel. It is important to note that unlike the SPH neighbour search, the DEM force summations do not take boundary particles into account, as DEM strictly only handles the steel-oxygen interaction.

The contact based forces' independence from density, relieves all the pressure issues which occur between the steel and the oxygen. The magnitude of the contact forces is only based on adjustable constants and the proximity of the particles. By adjusting the magnitude of the contact interaction appropriately, the velocity based issues can also be avoided. With an SPH interface interaction model, forces are unpredictable, depending on calculated pressures whose accuracy are limited by the size of the time step. With DEM, oxygen particles will have an instantaneous effect on the molten steel surface once a certain proximity to the steel is reached. Modelling in this way is more appropriate for collision

type systems, as few oxygen particles will pass into the molten steel. The inspiration for this implementation is drawn from "Node to Node Contacts for SPH Applied to Multiple Fluids With Large Density Ratios" by Jingxiao and Jason Wang. In this paper it is suggested that a DEM normal contact force and a damping force are used between the interactions of the two fluids instead of SPH. Their ideas were initially added to the pure SPH formulation and showed some promise. A cavity was formed in the steel as the oxygen collided with it, however the cavity was static showing no mixing or rolling phenomenon that one would expect. To combat this weakness, a tangential contact force was also added into the DEM model. The tangential force, successfully modelled the mixing phenomenon that was lacking in Jingxiao and Wang's methodology.

Chapter 5

The Steel Converter

5.1 Basic Oxygen Steelmaking

The basic oxygen converter is charged with liquid pig iron from a ladle straight after it has left the blast furnace. Typically the molten iron that is charged into the converter has about a 4% carbon content along with other traces of metals including but not limited to silicon, phosphorous, and sulphur. Pure oxygen is blown onto the surface of the molten steel/iron. As the oxygen comes into contact with the liquid metal, the carbon within is ignited and leaves the converter as carbon dioxide and carbon monoxide gas. In this way the carbon content of the liquid metal is reduced to controllable levels. After a twenty minute blow time, typical carbon contents will be under 1% carbon. Fluxes are also fed into the converter to absorb other undesirables and to group them in the form of slag. The liquid steel is tapped out of the converter leaving the behind the slag which will be cooled and later sold as a component of concrete.

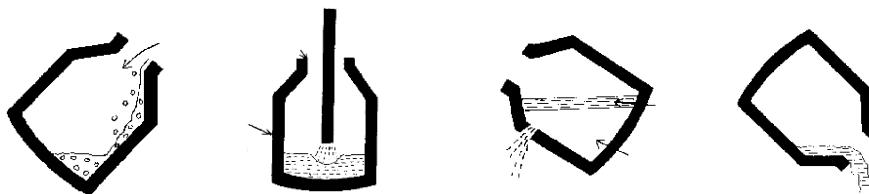


Figure 5.1:

Figure 5.2:

Figure 5.3:

Figure 5.4:

5.2 Implementation

The code was written in C++ and compiled for use on a Linux RedHat server with g++ . The SPH code draws inspiration from a method described by Matthias Mueller in Particle- Particle Fluid Interactions. However to make the

code appropriate to the problem, SPH was coupled with DEM and the boundary conditions and visualization were handled in a different manner.

5.2.1 Pseudo code

- Initialize Parameters
- For(number of time steps)
 - Neighbour Search
 - Compute Density and Pressures
 - Calculate SPH and DEM Forces
 - Accelerate Particles
 - Output Files
 - Adjust Time Step
- Close Files and End Program

5.2.2 Initialize Parameters

Parameters such as particle mass, density, boundary geometry, lance speed, lance locations, time steps, runtime etc. can all be specified through the user interface at runtime. Attention must be made when initializing parameters, as even one order of magnitude off can result in a numerical explosion.

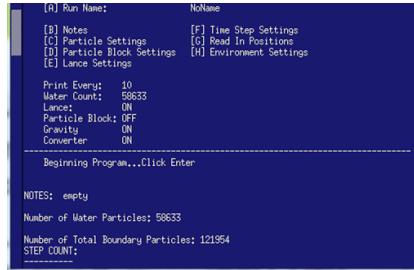


Figure 5.5:

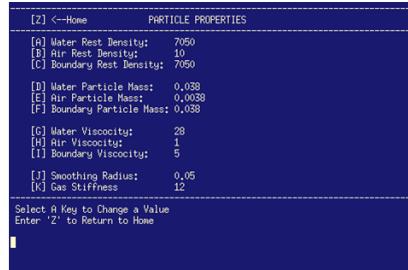


Figure 5.6:

5.2.3 Neighbour Search

The converter space is divided into a neighbour sorting grid, to organize and store particle locations. The length, width and depth of the 3D grid blocks are of the same size as the smoothing radius. By sizing the grid blocks in such a way, centred particles will only have to search for neighbouring particles that lie within the same grid block and the twenty six neighbouring grid blocks that surround it. At the start of each time step all three particle types (steel, oxygen,

and boundary) within all grid blocks will search for neighbouring particles in the nine neighbouring grid blocks. The distances between the centred particle of interest and neighbouring particles will be calculated, as the relative effect of each neighbour density will depend on this distance.

5.2.4 Calculate Density and Pressures

The centred particle will add neighbouring boundary particles and other particles of the same fluid-type to its smoothed density summation. Centred steel particles will ignore oxygen particles during the density summation and centred oxygen particles will similarly ignore steel particles. Once the smoothed densities have been determined the pressure of the centred particle can be calculated according to the ideal gas law.

5.2.5 Calculate SPH and DEM Forces

SPH forces such as pressure, viscosity, buoyancy and surface tension forces are calculated separately for steel and oxygen. External forces such as gravity are also calculated and added to both particle types. Afterwards, the DEM forces are calculated exclusively between steel and oxygen type particles at the fluid-fluid interface.

5.2.6 Accelerate Particles

Each particle is accelerated according to the integration scheme below. If there is an external acceleration on the boundaries, then the boundary particles are accelerated and then scaled to maintain a constant shape. If the moving boundaries are not scaled after each acceleration sequence the boundaries will distort with each time step. Given enough steps, the boundaries will eventually explode and be deleted as they leave the system domain.

$$Position = \Sigma(v + 0.75a\Delta t)\Delta t \quad (5.1)$$

$$Velocity = \Sigma a\Delta t \quad (5.2)$$

5.2.7 Output

Data is output every set number of time steps. The desired data to be printed out can be chosen in the initialization menu at runtime. The CSV data format includes; boundary positional data, steel positional data, oxygen positional data and the grid-block indexes. For the VTK format , density, pressure, surface, acceleration, velocity and positional data can be printed out for all three particle types; steel, oxygen and the boundary. Settings information files are also printed out.

5.3 Particle Types

There are three particle types used in this simulation. However, the number of particle types that can be used is only limited by the computational power and time limitations of the computer running the simulations. The first particle type is the steel-type particle, the second is the oxygen-type particle, and the third is the boundary-type particle. The converter simulation was first tested as a water-air system , as it is easier to model water-air than a steel-oxygen because of the smaller density gradient .

5.4 Particle Deletion

Particles will only be deleted when they exit the converter domain. Particles leave the domain because of leakage, splashing or the rise of oxygen after its impact with the steel surface. Some particle leakage outside of the converter is inevitable and unavoidable, as at high speeds there are numerical instabilities which can lead to particles passing through the boundaries. Once a particle leaves the converter, it is deleted to prevent the unnecessary waste of computing power needed to calculate its neighbours.

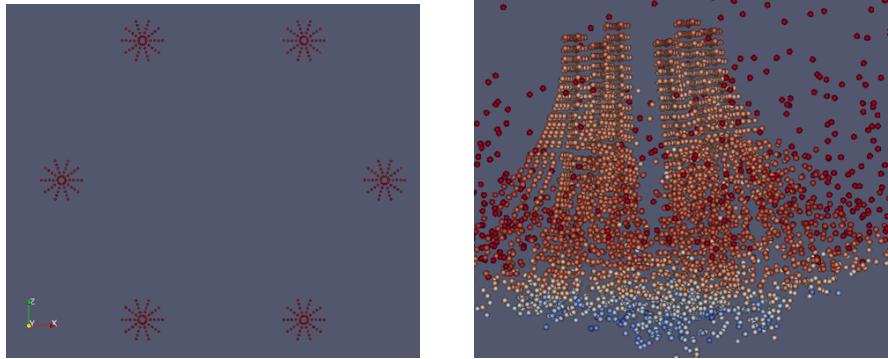
5.5 Geometry

The geometry of the converter is approximately modelled on a 4:1 scale. The height of the converter is actually shorter than the real life model for two reasons. The first being that the area of interest is only the surface of the fluid, the full height of the converter is not needed. The second reason is that keeping the unneeded full height of the converter slows down the program severely. The boundary particles are so closely placed together that any extra height could result in hundreds of thousands of extra particles to calculate values at. After the oxygen particles collide with the steel surface they rise out of the converter and soon leave the converter domain to be deleted. The taller the converter domain, the larger the number of oxygen particles at any one time, which greatly affects the computational workload.

5.6 The Lance

The lance is not explicitly formed out of boundary particles; rather oxygen particles are seeded at the locations where the nozzles would be located in a real world system. The number of nozzles, the height , the velocity and the angle of the oxygen coming out of the lance can all be easily set on the menu screen upon run time. The oxygen particles are seeded in a star shape from the nozzle locations. The velocities of each oxygen particle are given as a range so that there is a consistent average magnitude . Since the velocities are given as average once the oxygen particles leaves the seed point they will be sufficiently mixed

so they do not hit the surface of the fluid in the original star formation. The angles at which oxygen comes out of the nozzles can be adjusted by adjusting the initial velocity ranges.



5.7 Visualization

Particle data is printed out in two formats, CSV , comma separated values which carry only positional information and the legacy VTK format that carries additional information such as density, pressures, colour field etc. The VTK format is the native Paraview format that can be used to view any scalar value in time by assigning colour gradients. The benefit of the csv format is its simplicity. Since the csv format only contains particle locations it is perfect for initializing particles at set locations. It has come of particular use in filling the converter to a certain height. Because the converter is not a simple shape, often the best way to fill it is to insert a large block of fluid into the converter and let the simulation run until the fluid has settled. Obviously, doing this to fill the converter every single time would be a tremendous waste of time.

5.8 Stability

Stability is an issue that is dependent largely on two things; the size of the time step interval and the damping of the system (usually in the form of viscosity). If the time step is too large then velocities will not be damped by the other particles and will easily explode. Reducing the time step interval makes the system much more stable, it also helps prevent leakage out of the converter. Leakage occurs when a particle approaches boundary particles, however because of high speeds, it does not interact with the boundary until it has surpassed it, at which point it is accelerated out of the converter.

Chapter 6

Conclusions

For the simulation of the basic oxygen steel converter, smoothed particle hydrodynamics was successfully coupled with the discrete element method to provide a reasonably visually realistic simulation. The multiphase system included high density liquid steel, oxygen and stationary boundary particles. The code was written in C++ in the CodeBlocks IDE to be compiled on Linux. A pure SPH formulation was not found to be very effective in this situation because of its limitations in handling high velocities and large density gradients. To address these issues, at the steel-oxygen interface, DEM forces were calculated to determine the interaction instead of SPH forces.

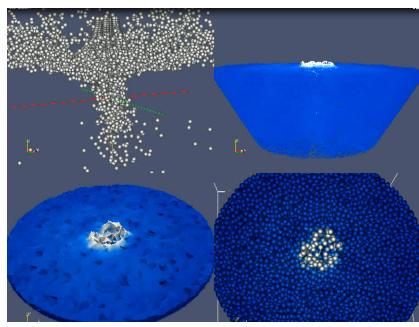


Figure 6.1: the final converter simulation

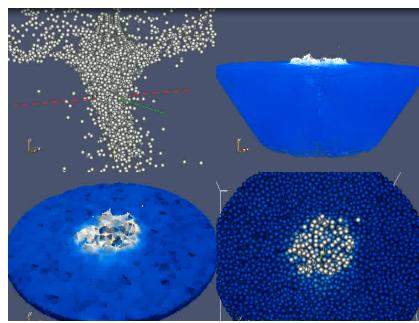


Figure 6.2:

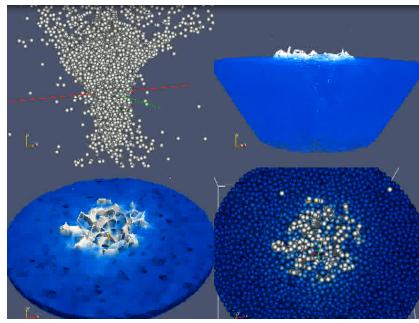


Figure 6.3:

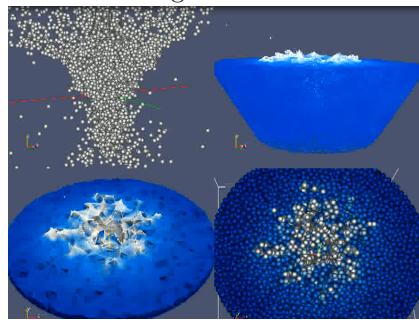


Figure 6.4:

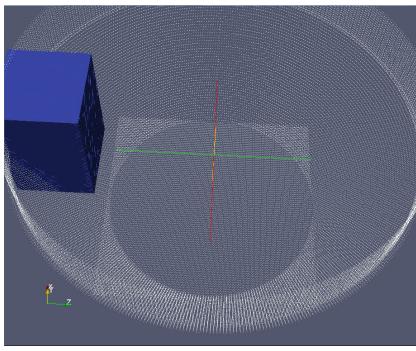


Figure 6.5: A block break simulation

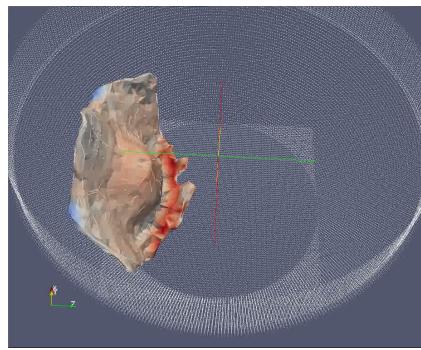


Figure 6.6:

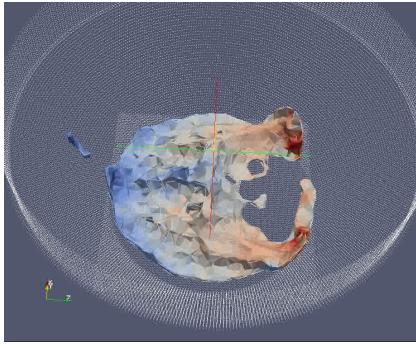


Figure 6.7:

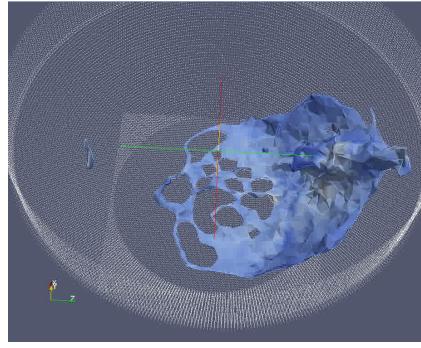


Figure 6.8:

Bibliography

- [1] J.J. Monaghan *Smoothed Particle Hydronamics*, *Annu. Rev. Astronom. Astrophys* 30:543-574, 1992.
- [2] J.J. Monaghan *Simulating free surface flows with SPH*, *Ann. Rev. Astronom. Astrophys* 1994.
- [3] M. Mueller ,D. Charypar , M. Gross *Particle-based fluid simulation for interactive applications* In SCA '03:Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (Aire-La-Ville, Switzerland, 2003), Eurographics Associations, pp.154-159
- [4] M.Mueller, B.Solenthaler,R.Keiser,M.Gross, *Particle Based Fluid-Fluid Interaction* In Proceedings for SIGGRAPH'05 Symposium on Computer Animation (SCA 2005) pp 237-244, Los Angeles, USA, July 31- August 4, 2005
- .
- [5] J. Xu., J. Wang. *Node to node contacts for SPH applied to multiple fluids with large density ratio*. 9th European LS-DYNA Conference, 2013