DOI 10.36146/2026_03_36

A comparative cfd study in tundish flow with particle tracking

P. Brunier

Continuous casting machine (CCO) is fundamental to produce steel at reasonable costs. The tundish represents the main source of steel for this kind of facility, acting as a buffer between the ladle and the nozzles feeding casting profiles. This component significantly influences the quality and yield of continuous casting operations. Our study focuses on leveraging Computational Fluid Dynamics (CFD) to track particle behaviour within the tundish. By simulating the flow patterns, we aim to optimize tundish design in terms of steel cleanliness. We defined two main scenarios that have different impact on molten steel flow and inclusion transport, with the intention to minimize defects and enhance overall efficiency.

Our approach involves tracking individual particles—both in terms of trajectory and residence time—allowing us to understand their impact on product quality. Intense postprocessing has been done with python to perform analysis on a structured framework, making comparison between the two cases much easier.

Finally, the head reason is to reveal by simulations how inclusions move and could accumulate within the steel flow. This kind of knowledge prompt strategies to reduce inclusions in the final product.

KEYWORDS: TUNDISH; PARTICLE TRACKING; FLUID DYNAMICS; STEELMAKING; THERCAST;

INTRODUCTION

This study aims to investigate the behavior of inclusions within the tundish by leveraging Computational Fluid Dynamics (CFD) simulations. Two different configurations are examined: one featuring a wall with angled holes, called dam, and the other employing a box designed to dampen the momentum of the incoming flow. These configurations are evaluated in terms of how they influence the molten steel flow and the transport of inclusions by tracking individual particles and analyzing their release time and trajectories to assess how each design performs in terms of inclusion removal at a steady state. The analysis leans on a post-processing algorithm developed in Python, which enables a detailed and structured comparison of the two setups, introducing innovative tools such as Kernel Density Estimation (KDE) to better understand inclusion distribution and dynamics. The goal of this work is to deepen the understanding of inclusion transport phenomena in the tundish and to identify effective design strategies that minimize defects and improve overall casting quality.

Philippe Brunier
Cogne Acciai Speciali, Aosta (Italy)

GEOMETRIES

The two configurations of interest involve a different approach in fluid transport behavior: in the first configuration there is a refractory wall with five holes that allow the steel to feed the tundish pool. These holes possess a certain inclination that force the flux to the top region of the liquid bath: this promote a better inclusion removal since it will be more likely for an inclusion to interact with and subsequently to be trapped by the slag. The second case involves a so-called box in which the flow is dumped, in terms of momentum, before being released. This is particularly useful in the initial transient, preventing unwanted splashing of liquid steel, potentially dangerous. The details of these two components rely on industrial secret and cannot be shared.

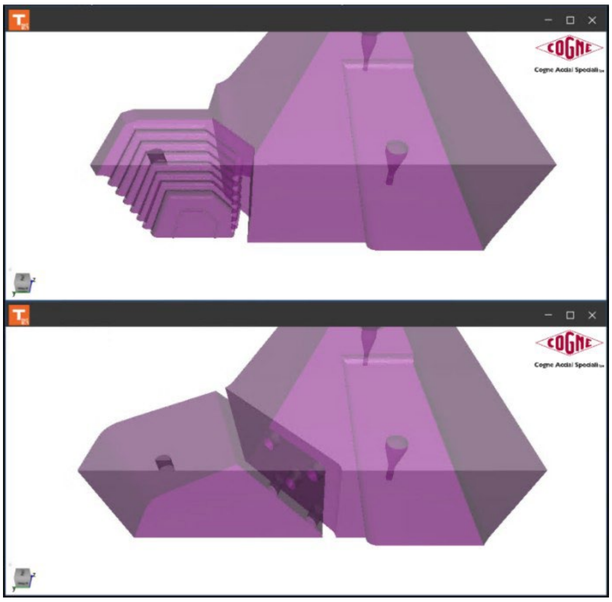

Fig.1 - The two geometries compared in this study: box (top) and dam (bottom).

The continuous casting inlet pipe has been modelled as well considering its slide valve, regulating the flow into the strain casting mold. During steady-state operations, its position is assumed to be quasi-static, so the liquid steel channel can be assumed as a fixed geometry.

SIMULATION SETUP

The Finite Element Method (FEM) is a numerical technique for solving partial differential equations (PDEs) over complex geometries, discretizing the domain into smaller elements which takes the name of a mesh. In computational fluid dynamics (CFD), FEM is employed to approximate solutions to the Navier–Stokes equations and related transport phenomena. This method is particularly advantageous in problems involving multi-physics coupling or flow-structure interactions. For this work, THERCAST [1] is used to perform CFD simulations while Python drives the postprocessing operations. The solver affords on Large Eddy Simulation (Variational multiscale method) as approach to model turbulence in the liquid steel, considering elasto-viscoplastic behavior for the mushy zone as well as solidification shrinkage for solidified domain.

Since this is a steady state (no change of volume over time), a mass balance equation over the control volume imposes the following condition:

$$\dot{Q}_{in} = \sum_i^N \dot{Q}_{out,i} \qquad [1]$$

Thus, inlet and outlet have imposed velocity, according to the [1] and both are adiabatic in terms of heat flux. Free surface has been modeled imposing a surface tension coefficient and no augmented viscosity for turbulence. Anyway, the criterion applied on particle tracking is based on geometrical coordinates meaning that these boundary conditions have no impact on this work.

In fluid reactors engineering, the time constant expresses how quickly the system responds to input changes (such as concentration, flow or temperature). It reflects the average residence time of fluid and indicates how fast the tundish reacts to a given input. It is defined as:

$$\tau = V/\dot{Q} \qquad [2]$$

Where:

V is the control volume, $[m^3]$
$Q^{\cdot}$ is the volumetric flow rate, $[m^3/s]$

For this study, $\tau \approx 9$ minutes.

Considering that typically a transient is extinguished in a time between 3- and 5-time constants, this means a time between 30 minutes and 1 hour of CFD simulated time. Another aspect concerns the control volume itself since it has a direct impact on the number of nodes of the mesh: we cannot make a coarse mesh because it will impact on the resulting velocity field, and so on the particle trajectory prediction. In order to make reasonable comparisons, we must set a proper mesh size for both cases. Convergence studies showed that the number of nodes has been set around 100k per m^3 of volume.

FEM optimization

As a standard practice, is exploited the symmetry of the system which means in this case that we can consider only ½ of the global geometry. Further efforts have been made to optimize the simulation as much as possible: this is the main reason behind the choice of a virtual wall so only the liquid volume is directly involved in the computation while the wall is modelled in terms of equivalent thermal resistance. We used a mesh with fixed elements (non-adaptive remeshing) because we want to be robust in terms of particle tracking: a remeshing based on velocity, for instance, could locally make the mesh coarser invalidating or at least penalizing the particle trajectory estimation.

Particle tracking

Particle tracking is a technique used to study inclusion trajectory in a fluid domain over time. The model assumes no interactions between inclusions. Therefore, there is no agglomeration or cohesion between particles, which establishes a decoupled approach. Despite this aspect, drag forces as well as buoyancy and fluid pressure and momentum balance have been considered turbulence.

In this work 200 particles per size are released near to their source (ladle jet), totaling in almost 500 files per size (1 and 10 microns). All particles are injected simultaneously (pulse release) to better observe transport dynamics. They are placed near the main inlet jet to enhance transport toward the casting strands, as global trapping was not the focus here. Each file will result into a unique outcome: trapped by slag, entering one of the strands, or remaining in the pool. Particles still in the pool reflect dead zones, where low velocity or long residence time limits their removal. The asymptotic pool value may represent the dead volume fraction (k-pool).

THEORETICAL ASPECTS

Dimensionless philosophy

In order to compare different volumes, which imply different time constants, a classical approach is to normalize these quantities. Being *C* a concentration and *t* a time, then we can make it dimensionless simply by doing:

$$C^*(t^*) = C(t/\tau)/C_0 \qquad [3]$$

This approach allows us to compare different geometries.

From RTD to KDE: an innovative approach

Building a model with concentrations and residence time distributions (RTDs) involves, by its nature, continuous quantities, which in turn involve integrals. This simulation deals instead with discrete events because particle tracking is discretized through both space and time. RTD is basically a plot showing how a substance distributes into the liquid domain. Its measure is concentration; therefore, we can express it as:

$$C \rightarrow \frac{number\ of\ inclusions}{volume\ of\ the\ system} = \frac{N}{V_0} \qquad [4]$$

With this definition of concentration, we can compute an RTD as [2][4]:

$$RTD(t) = \frac{C(t)}{\int_0^\infty C(t)dt} = \frac{N(t)/V_0}{\int_0^\infty N(t)/V_0 dt} = \frac{N(t)}{\int_0^\infty N(t)dt} = \frac{N(t)}{N_0} \qquad [5]$$

By definition, RTD is normalized. In other terms:

$$\int_0^\infty RTD(t)dt = 1 \qquad [6]$$

Kernel Density Estimation (KDE) curve is a nonparametric method for estimating the probability density function of a random variable from a sample of data [3]. It means that, given an event *E*, a KDE shows the probability *p* that the event *E* occur:

$$KDE \ \approx p_E(t, t+dt) \qquad [7]$$

Since it is a probability, its probability density function must be normalized, as an RTD curve:

$$\int_0^\infty R(t)dt = \int_0^\infty p(t)dt \qquad [8]$$

We can then make the hypothesis that using KDE plot to estimate RTD curve is acceptable because both quantities belong to the domain [0, ∞) and are normalized (the integral is equal to 1). This allows us to use kernel density estimation techniques which work well on a set of discrete events. Being plausible that the two integrals are equal, it also means that the shape of the two curves could probably be the same, even if the metrics are different. This aspect has not been deepened since we are interested in the corresponding peak time, which should not depend on the probability density by itself.

Anyway, to give a reliable statistical model we must consider all possible outcomes, which are:

- A particle never reaches the liquid pool because it has been blocked by the dumping device (dam/impact box).
- A particle is trapped by the slag.
- A particle passes through a casting strand and becomes an inclusion for the cast billet.
- A particle is still floating into the liquid pool, remaining in the liquid steel. This means that the lasts billets may have more inclusions than the standard produced.

Due to this analogy with the definition of probability the decision made is to afford on counting principle in order to define a fraction (or percentage), depending on time, of particle trapped by the slag as:

$$k_{SLAG} = \frac{particles\ that\ reach\ the\ surface}{number\ of\ particle\ released} = \frac{N_{slag}}{N_r} \qquad [9]$$

This approach has been used also to make an estimation of the 'holding' efficiency of the two components (dam, box). For example:

$$\varepsilon_{BOX} = \frac{nb\ of\ particles\ trapped\ into\ the\ device}{total\ number\ of\ particles} \qquad [10]$$

Therefore, we can define the fraction of particles released as:

$$1 - \varepsilon_{box} = \varepsilon_r \qquad [11]$$

So, the total number of particles released is simply:

$$N_r = N_0\varepsilon_r = N_0(1 - \varepsilon_{device}) \qquad [12]$$

Rewriting equation 9 we get:

$$k_{SLAG} = \frac{N_{slag}}{N_0(1-\varepsilon_{device})} \qquad [13]$$

Similarly, for the particles that flow through a continuous casting strand, from here called 'cc line':

$$k_{LINE_i} = \frac{particles\ that\ flow\ into\ i{-}th\ cc\ line}{number\ of\ particle\ released} \qquad [14]$$

Having defined the end of a particle in only 3 possible ways we can also say that:

$$k_{SLAG} + \sum_i^{N_{lines}} k_{LINE,i} + k_{POOL} = 1 \qquad [15]$$

From which we can estimate k_{POOL}. Notice that while k_{SLAG} and k_{LINE},i increase over time, k_{POOL} decrease. A good design should have:

- Low k_{POOL}: its regime value is very low if not even zero. The bigger the slope, the better is the design because it means low dead volumes into the tundish.
- High k_{SLAG}: it means that a big portion of inclusions have floated and reached the top of the control volume. For this model, this condition is sufficient to consider them as trapped by the slag.
- Low k_{LINE},I since we want to cast the cleanest billets possible.

With this approach we can now compare the two designs to figure out if there is a better configuration in terms of inclusion transport.

Description of the algorithm

The computational algorithm used for this kind of analysis is based on the following principle: the life of a single particle could end in a deterministic way:

- Trapped by the slag, if its z coordinate reaches the top of the volume.
- Fallen into one continuous casting strand.
- Still floating into the liquid pool, potentially polluting the steel during the production of the last billets (while the tundish is emptying).

The algorithm collects the 'end time' and automatically assigns a label for each particle. Once this procedure is completed, we can have access to more accurate statistics, including also average time of the event, having distributions over time, even predicting some behavior before the simulation runs up to 5-tau, which should allow in fact to a shorter computation. This has been done assuming a classical transient equation of the kind:

$$k(t^*) = k_\infty(1 - \exp[-r(t^* - t_0^*)]) \qquad [16]$$

Where:

- k_∞ is the regime value.
- t_0^* is the initial dimensionless release time (10s of physical time for both simulations).
- r is the rate, indicating how fast we extinguish the transient.

The post processor, written in python, must find where each particle goes before. To perform this, the main regions of interest have been defined in preprocessing, once the regions of interest has been defined, it is just a matter of coordinates: if the particle falls into one of these regions it is automatically labelled for the further statistics. These regions of interest are identified by .stl volumes read also by the postprocessor before entering in the screening loop. A sensor file is a collection of results at each computed increments, and all these data are collected into a python pandas dataframe object. Once read, the algorithm scans the z coordinate at first (slag / strands), then x and y to eventually determine the cc line (central, peripheric). It automatically stops once one of these conditions are reached, moving to the next file.

Considering the simulation time (1.75 - 2τ) and the low timestep imposed by the CFD solver, every sensor file results having several thousands of lines, with a dimension around 4.5 MB per file. Once data are collected and properly labelled it is possible to go ahead with the KDE calculation, as well as the fitting over time to retrieve constants of the systems for the considered events.

RESULTS

Flux & temperature

The first results analyzed are the velocity field and the temperature of the liquid bath. Top surface, shown in figure 2, is between 5°C and 10°C hotter with the box rather than the dam. Anyway, looking at the symmetry plane (figure 3), we can deduce that this difference disappears while reaching the tundish bottom wall. This suggests that for low velocity (<0.1 m/s) the transport mechanism does not more rely on advection (high Pe number), but we should consider diffusion term as well. A deeper analysis suggested that advection/transport is still the major contributor since the velocity field leads the temperature distribution.

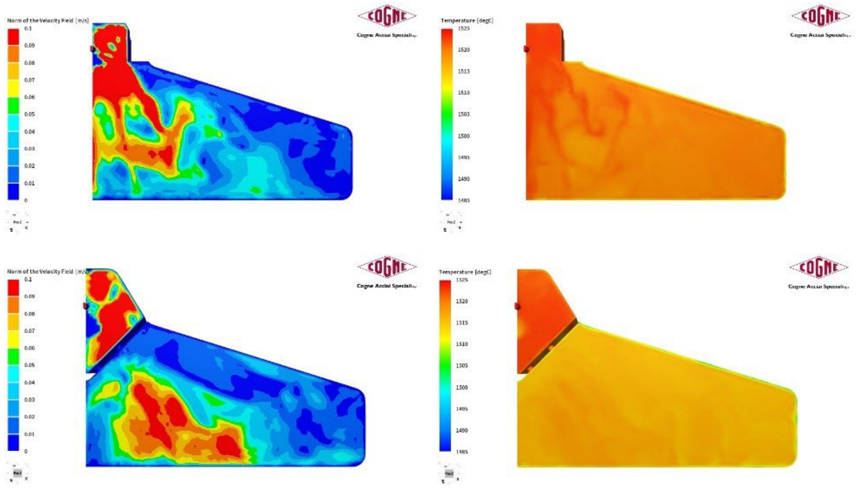

Fig.2 - Surface plane: velocity (left) and temperatures (right), for box (top) and dam (bottom).

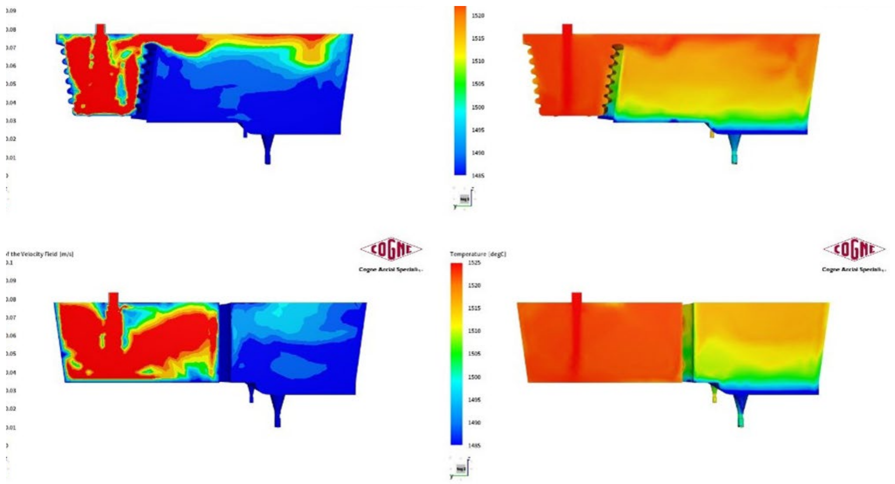

Fig.3 - Symmetry plane: velocity (left) and temperatures (right), for box (top) and dam (bottom).

Postprocessed particle tracking

We proceed now with the KDE plot of dimensionless time. We noticed that most probable time in the KDE curve does not coincide with the mean value calculated with standard techniques, but it is always lower. This implies that the distribution is not symmetric. An effort has been put also to find a distribution where the mode and the mean are the same number. The main distribution with this property is the normal distribution, which is defined between (-∞,∞) while out domain belongs to (0, ∞). Since the aim of this work is to compare the two different setups we choose not to go deeper with this topic for the moment, evaluating just the peak dimensionless time:

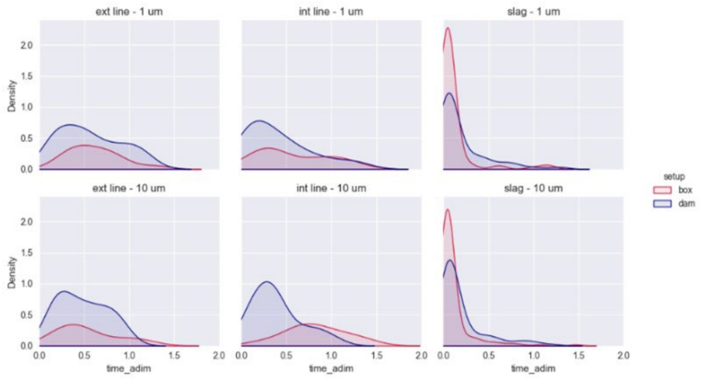

Fig.4 - KDE plot for the box (red) and the dam (blue) for the considered statistics: external line (left), internal (center) and slag (right).

By fitting these values with equation (16) we can get the k values over time, describing the overall behavior of the two setups:

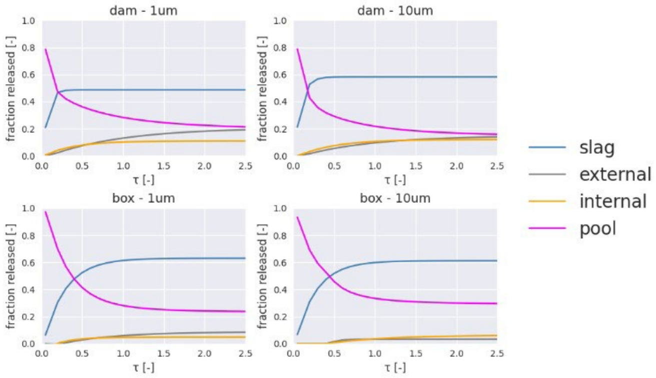

Fig.5 - k values over tau, describing the overall release fitted behavior.

We can deduce that the box setup leads to lower k_{line} values, with a bigger contribute of the slag. This is particularly evident for the external line, where the dam and 1 micron particles: the flux is conveyed to the external lines because of its design holes that convoy the liquid steel on that particular region.

The role of the dumping device

Different geometries lead to different flux, so different particle behavior. It could be of interest to make an in-depth analysis on the 'dumping' components in order to see if there is a difference already close to the pouring region. In fact, the two items have been built with two different control volumes. Since time constant directly depends on the volume because the flow rate can be considered constant, we can easily scale τ according to the change of volume. Fitted values show a faster release (bigger r from equation 16) and a lower fraction of particle released (k_∞ values).

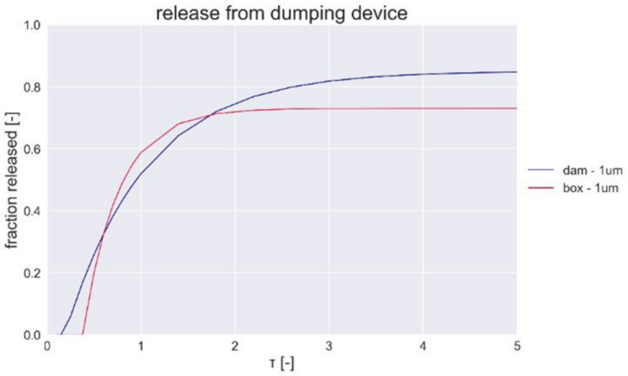

Fig.6 - Impact of the dumping device on the overall release of particles (1 micron) for the dam setup (blue) and the box (red).

CONCLUSIONS

We are finally able to assess if one configuration is better than another in terms of both fluid dynamics and particle tracking. A first analysis of the regime value shows that the box device releases less inclusions, and it is kinetically more stable (faster transient). The main impact of inclusion transport is localized into the external casting strand, especially for lighter particles (1 micron) which showed a dump in particle flow of approximately 50% into the casting profile. Internal strands (1 micron) also follow this trend, with a resulting gain of 10-15% in slag contribution. For heavier particles, this trend is less emphasized but still present

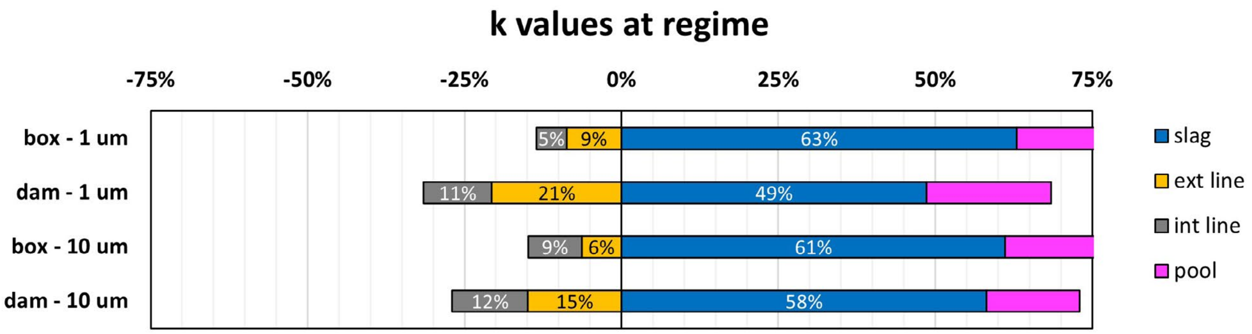

Fig.7 - Resumed fitted values for the two setup and two set of particles

Next steps will be to study particle transport from the gate into the continuous casting production line in order to see if these bigger particles generate inclusions inside the solid billet or if they float anyway, meaning that the main mechanism of bigger particles relies on the continuous casting mold.

REFERENCES

[1] https://www.transvalor.com/en/thercast

[2] Y. Sahai, T. Emi, "Melt flow characterization in continuous casting tundishes". ISIJ International, 1996, Volume 36, Issue 6, p. 667-672. https://doi.org/10.2355/isijinternational.36.667

[3] A. W. Bowman, A. Azzalini, "Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations", Oxford Statistical Science Series, vol. 18, Oxford University Press, 1997. https://doi.org/10.1093/oso/9780198523963.001.0001

[4] H. Lei, "New insight into combined model and revised model for RTD curves in a multi-strand tundish", Metall. Mater. Trans. B. 2015, 46(6):2408–13. doi:10.1007/s11663-015-0435-6

TORNA ALL'INDICE >