

A free surface finite element model for low Froude number mould filling problems on fixed meshes

H. Coppola-Owen and R. Codina^{*,†}

Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

SUMMARY

The simulation of low Froude number mould filling problems on fixed meshes presents significant difficulties. As the Froude number decreases, the coupling between the position of the interface and the resulting flow field increases. The usual two-phase flow model provides poor results for such flow. In order to overcome the difficulties, a free surface model that applies boundary conditions at the interface accurately is used. Moreover, the use of wall laws on curved boundaries also fails in the case of low Froude number flows. To solve this second problem, we combine wall laws with ‘do nothing’ boundary conditions. A special feature of our approach is that ‘do nothing’ boundary conditions are only applied in the normal direction. These two key ingredients together with the Level Set method allow us to simulate three-dimensional mould filling problems borrowed directly from the foundry. Copyright © 2010 John Wiley & Sons, Ltd.

Received 28 July 2009; Revised 11 December 2009; Accepted 13 December 2009

KEY WORDS: incompressible free surface flows; mould filling; fixed mesh; level set; stabilized finite elements

1. INTRODUCTION

The numerical simulation of mould filling processes has become a widespread tool for improving casting technology. Regions with high velocities that can lead to premature wear of the mould can be predicted. The quality of the resulting piece can also be improved, for example, by determining regions of possible air entrapment. An overview of computational methods for free surface flows in casting and Industry-Standard Mould-Filling codes can be found in [1].

Contrary to what one might naively think, we have observed that in mould filling problems, lower filling velocities typically lead to more complex simulations. That is to say, low Froude number flows pose special difficulties for two-phase flows. The lower the Froude number, the higher the importance of the gravitational forces. Since the spatial distribution of the gravitational forces is determined by the position of the interface, the coupling between the position of the interface and the resulting flow increases as the Froude number decreases. An accurate representation of the pressure in the elements cut by the interface is needed for such flows. By enriching the pressure finite element shape functions [2] or by using a free surface model [3], we have obtained very

^{*}Correspondence to: R. Codina, Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.

[†]E-mail: ramon.codina@upc.edu, URL: <http://codina.rmee.upc.edu/>

Contract/grant sponsor: Departament d'Universitats, Recerca i Societat de la Informació of the Generalitat de Catalunya (Catalan Government)

Contract/grant sponsor: European Social Fund

satisfactory results in simple examples. In this work, we extend the application of the free surface model [3] to real mould filling problems. By free surface model we understand that the influence of air on the molten metal is neglected.

In [3], a free surface model on a fixed mesh that only simulates the region occupied by the fluid and neglects the influence of air has been presented. This method provides very good results for low Froude number flows, thanks to a careful treatment of the elements cut by the front that allows to accurately impose the boundary conditions at the interface. In molten metal mould filling applications, surface tension is usually negligible and therefore a zero traction boundary condition must be applied at the free surface. By enhancing the integration rules at the elements cut by the front and integrating only in the filled part, this natural boundary condition can be correctly applied at the interface, even if the interface does not coincide with element faces. This is the key ingredient of the formulation we propose.

CFD approaches for moving interfaces problems are typically categorized into two main groups: Eulerian, fixed mesh or interface capturing techniques [4–11] and Lagrangian, and in the more general case arbitrary Lagrangian Eulerian (ALE), moving mesh or interface tracking techniques [12–14]. The model we use is clearly a fixed mesh, interface capturing technique but it is not so obvious whether to classify it as an Eulerian or as a Lagrangian formulation. Actually it has two versions: a simplified one that solves the momentum equations in an Eulerian manner and another one that uses an ALE formulation but on a fixed mesh [3].

In interface tracking techniques, the mesh is updated in order to track the interface. The simplest approach is to deform the mesh without changing its topology, but this is possible only for very simple flows. As the flow becomes more complex and unsteady, remeshing is required, and consequently the projection of the results from the old to the new mesh is needed. In 3D calculations, these operations can introduce costs that can render moving mesh techniques unfeasible. This is the main reason why fixed mesh methods are preferred.

Interface capturing techniques use a fixed computational domain with an interface function to capture the position of the interface. The interface is captured within the resolution of the fixed mesh and the boundary conditions at the interface are somehow approximated. The most common approach is to treat free surface flows as a particular case of two-phase flows where the second fluid is air or some pseudo fluid with much smaller values of the physical properties than the first fluid. Instead, the approach used in this work neglects the influence of the air and only one fluid is simulated. In this sense it is a pure free surface model.

The use of the typical two-phase flow interface capturing approach for low Froude number flows leads to poor results. The extreme case is that of two different density fluids at rest (the denser one in the bottom) under gravity force when the interface is not aligned with element faces. As explained in [2], the main problem is that the discontinuous pressure gradient cannot be accurately represented by the usual finite element shape functions in cut elements. When the two most important terms in the momentum equation are the pressure gradient and the gravitational forces, the errors in the pressure introduce huge errors in the velocity field that spoil the simulation. In [2], we proposed a solution based on enriching the pressure shape functions in the elements cut by the interface that is valid for two-phase flows. In [3], an alternative solution that is valid only for free surface flows was proposed. The advantage of using a free surface model is that as only one fluid is simulated, the problem of dealing with a discontinuous pressure gradient disappears. As we have already mentioned, the key point is to apply boundary conditions at the interface accurately. We have found that the free surface model provides better results than the enriched pressure two-phase flow model and therefore it will be used in this paper.

Fixed mesh methods generally share two basic steps: one where the motion in both phases is found as the solution of the two-fluid Navier–Stokes equations and one where an equation for an interface function that allows to determine the position of the interface, and thus the properties to be assigned in the previous step, is solved. As we have already mentioned, the model we will present has two versions, a simplified one that solves the Navier–Stokes equations in an Eulerian manner and the other one that uses an ALE formulation. In both versions, a level set function is used to determine the position of the interface.

Mould filling problems involve relatively high Reynolds number flows. In order to avoid excessively refined meshes close to the mould walls, we follow the typical approach of using wall laws [15] instead of prescribing the velocity to zero at the wall. For low Froude number flows, we have found that the use of non-zero boundary conditions on curved walls generates spurious velocities. The solution we adopt is a modification to the approach suggested in [16].

The next section reviews the free surface model. Starting from the typical Eulerian two-phase flow model, the simplified Eulerian free surface model is derived. Then the main steps in free surface FM-ALE model are briefly revised. Section 3 presents the solution we have adopted for dealing with wall laws on curved boundaries. Section 4 solves three mould filling examples where the advantages of the free surface model we propose show up clearly. The final section presents the conclusions of our work.

2. FREE SURFACE MODEL

We have already mentioned that two versions of the free surface model have been developed: a FM-ALE model and a simplified Eulerian version. For the problems we are dealing with, both of them provide very similar results despite the latter involves some further approximations. For the examples presented in this work, the simplified Eulerian version shall be used because it has proved computationally more efficient.

In order to motivate the simplified Eulerian version, we first briefly introduce the typical Eulerian two-phase flow model. The Eulerian simplified free surface model can be thought as the case when the properties of the second fluid and its effect on the first fluid become negligible. The small traction that acts on the fluid can be approximated as a zero traction. A review of how the simplified Eulerian model is obtained is presented in the second subsection. A brief review of the FM-ALE free surface model that provides nearly identical results is presented in the third subsection.

2.1. Eulerian two-phase flow model

As it has been mentioned, fixed mesh methods for two-phase flows generally share two basic steps, one where the motion in both phases is found as the solution of the two-fluid Navier–Stokes equations and the other one where an equation for an interface function that allows to determine the position of the interface is solved.

The velocity and pressure fields of two incompressible fluids moving in the domain $\Omega = \Omega_1 \cup \Omega_2$ with $\Omega_1 \cup \Omega_2 \neq \emptyset$ during the time interval (t_0, t_f) can be described by the incompressible two-fluid Navier–Stokes equations [4]:

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right] - \nabla \cdot [2\mu \boldsymbol{\varepsilon}(\mathbf{u})] + \nabla p = \mathbf{f}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where ρ is the density, $\mathbf{u}$ the velocity field, μ the dynamic viscosity, p the pressure, $\boldsymbol{\varepsilon}(\cdot)$ the symmetric gradient operator and $\mathbf{f}$ the vector external body forces, which includes the gravity force $\rho \mathbf{g}$ and buoyancy forces, if required. The density, velocity, dynamic viscosity and pressure are defined as

$$\mathbf{u}, p, \rho, \mu = \begin{cases} \mathbf{u}_1, p_1, \rho_1, \mu_1 & \mathbf{x} \in \Omega_1, \\ \mathbf{u}_2, p_2, \rho_2, \mu_2 & \mathbf{x} \in \Omega_2, \end{cases}$$

where Ω_1 indicates the part of Ω occupied by fluid number 1 and Ω_2 indicates the part of Ω occupied by fluid number 2. The extent of Ω_1 and Ω_2 is given by the level set function ψ introduced below.

Let $\boldsymbol{\sigma}$ be the stress tensor and $\mathbf{n}$ the unit outward normal to the boundary $\partial\Omega$. Denoting by an over-bar prescribed values, the boundary conditions to be considered are

$$\begin{aligned} \mathbf{u} &= \bar{\mathbf{u}} \quad \text{on } \Gamma_{\text{du}} \times (t_0, t_f), \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \bar{\mathbf{t}} \quad \text{on } \Gamma_{\text{nu}} \times (t_0, t_f), \\ \mathbf{u} \cdot \mathbf{n} &= 0, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_1 = t_1, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_2 = t_2 \quad \text{on } \Gamma_{\text{mu}} \times (t_0, t_f), \end{aligned} \quad (3)$$

where $\mathbf{n}$ is the unit outward normal to the boundary $\partial\Omega$ and vectors $\mathbf{g}_1$ and $\mathbf{g}_2$ (for the three-dimensional case) span the space tangent to Γ_{mu} . Observe that Γ_{du} is the part of the boundary with Dirichlet velocity conditions, Γ_{nu} the part with Neumann conditions (prescribed stress) and Γ_{mu} the part with mixed conditions. These three parts do not intersect and are a partition of the whole boundary $\partial\Omega$. Initial conditions have to be appended to the problem.

Regarding the second step, the evolution of the fluid interface is updated using the so-called *level set* method (see [4–6] for an overview), also called *pseudo-concentration technique* [7], which is very similar to the volume of fluid (VOF) technique [8]. This formulation has been widely used to track free surfaces in mould filling (see for example [9–11, 17], among other references) and other metal forming processes.

The basic idea of the level set method is to define a smooth scalar function, say $\psi(\mathbf{x}, t)$, over the computational domain Ω that determines the extent of subdomains Ω_1 and Ω_2 . For instance, we may assign positive values to the points belonging to Ω_1 and negative values to the points belonging to Ω_2 . The position of the fluid front will be defined by the iso-value contour $\psi(\mathbf{x}, t) = 0$. The evolution of the front $\psi = 0$ in any control volume $V_i \subset \Omega$ that is moving with a divergence-free velocity field $\mathbf{u}$ leads to

$$\frac{\partial \psi}{\partial t} + (\mathbf{u} \cdot \nabla) \psi = 0 \quad \text{in } \Omega \times (t_0, t_f). \quad (4)$$

This equation is hyperbolic and therefore boundary conditions for ψ have to be specified at the inflow boundary, defined as

$$\Gamma_{\text{inf}} := \{\mathbf{x} \in \partial\Omega \mid \mathbf{u} \cdot \mathbf{n} < 0\}.$$

Function ψ is the solution of the hyperbolic equation (4) with the boundary conditions:

$$\begin{aligned} \psi &= \bar{\psi} \quad \text{on } \Gamma_{\text{inf}} \times (t_0, t_f), \\ \psi(\mathbf{x}, 0) &= \psi_0(\mathbf{x}) \quad \text{in } \Omega. \end{aligned}$$

The initial condition ψ_0 is chosen in order to define the initial position of the fluid front to be analyzed. The boundary condition $\bar{\psi}$ determines which fluid enters through a certain point of the inflow boundary.

For the numerical solution of the level set equation, it is preferable to have a function without large gradients. Since the only requirement such a function must meet is $\psi = 0$ at the interface, a signed distance function ($|\nabla \psi| = 1$) is used. Under the evolution of the level set equation, ψ will not remain a signed distance function and thus needs to be reinitialized. This can be achieved by redefining ψ for each node of the finite element mesh according to the following expression:

$$\psi = \text{sgn}(\psi^0) d, \quad (5)$$

where ψ^0 stands for the calculated value of ψ , d is the distance from the node under consideration to the front and $\text{sgn}(\cdot)$ is the signum of the value enclosed in the parenthesis.

Since the objective of this paper is to analyze the improvements that can be obtained in the solution of the Navier–Stokes equations and not to optimize the solution of the level set equations, a very simple algorithm has been used to calculate the distance d . Using linear elements, the free surface is approximated by triangular planes p (lines in 2D). Then the perpendicular distance $d_i p$ of each grid point i to each plane p can be computed. The minimum distance from each nodal

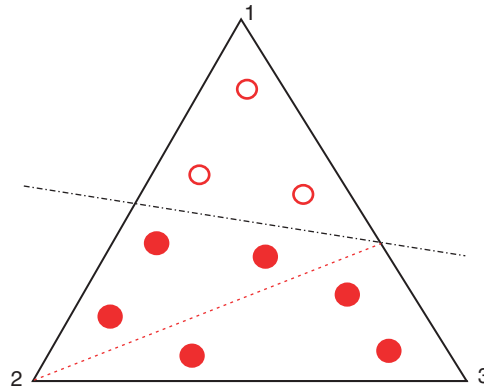


Figure 1. Splitting of elements for enhanced integration.

point to the planes is the required distance between the point and the front ($d_i = \min_p \{d_i p\}$). Actually, this approach is computationally very expensive for big meshes. In order to reduce the computational cost, we have reinitialized the level set function only in a small layer of elements close to the interface. For other reinitialization schemes, see [6, 11, 18].

2.2. Simplified Eulerian free surface model

In order to introduce the simplified Eulerian free surface model, we start from the Eulerian two-phase flow model. First, we suppose we wish to solve the two-fluid Navier–Stokes equations using a Dirichlet Neuman domain decomposition approach. Contrary to what is done in [19] in our model, we suppose no overlapping between the subdomains. For the fluid Neuman boundary conditions are used and for the air Dirichlet conditions are imposed. A special feature of this Domain Decomposition approach is that the interface between subdomains does not coincide with element faces. That is, cut elements have one part in the fluid subdomain and the other one in the air subdomain.

The traction exerted by the air on the fluid is very small due to its much lower values of physical properties. The first approximation is to replace the small traction exerted by the air on the fluid by a zero traction. This uncouples the behavior in the fluid from the behavior in air and thus avoids the need of iterating between both subdomains. This small approximation leads to the free surface model.

Since the fluid domain is formed by fully filled elements and the filled part of cut elements, special integration rules need to be used in cut elements. The idea is to subdivide the elements cut by the front into ‘subelements’ only for integration purposes. In Figure 1, we show the enhanced integration rules for a 2D linear triangular element cut by the front. Nodes 2 and 3 belong to the fluid. The element is subdivided into three triangles where the usual integration rules are used. Integration is performed only in the filled part of the element. The use of the enhanced integration allows to impose the zero traction boundary condition correctly at the exact position of the interface. The advantage over the usual two-phase flow model is that as only the fluid is being solved, no discontinuous pressure gradient needs to be captured. As we have already mentioned in the introduction, the problem with the usual two-phase flow model is that the discontinuous pressure gradient cannot be represented correctly by the usual finite element shape functions if the interface is not aligned with element faces.

Once the fluid has been solved, the air should be dealt with. In [3], we actually solve the Navier–Stokes equations in the air domain. In the examples presented in this work we have found that the results show no significant difference if instead of solving for the air, the velocities obtained in the fluid are extrapolated into the air region. In order to do so, the air nodes are classified into levels. The first level $i = 1$ corresponds to the air nodes of cut elements. The level $i + 1$ is formed by the nodes connected to level i . The velocities in each of the air nodes starting from the second

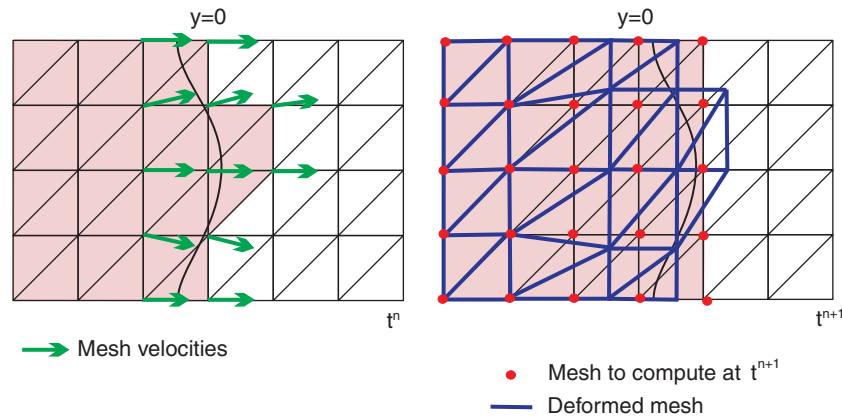


Figure 2. Two-dimensional FM-ALE schematic.

level are calculated as the average of the velocities in the nodes from the lower level connected to it. A similar approach is used in [20].

The velocities in the air are needed to transport the level set function that determines the position of the interface. Since the fluid domain is moving and the mesh is fixed, there will be nodes that in step $n+1$ will belong to the fluid but in the previous step belonged to the empty region. In such nodes, we will use the velocities calculated for the empty region when modeling the transient term. The validity of such approach is justified pointing out that it is what happens in the case of a two-phase flow when the properties of the second fluid tend to zero. A more consistent approach can be obtained by using an ALE formulation. We briefly describe the FM-ALE free surface model in the next subsection. Nevertheless for the examples presented in this paper, no significant difference has been observed between the results obtained with both models.

2.3. FM-ALE free surface model

As an alternative to the simplified Eulerian free surface model in [3], we extended the FM-ALE approach [21] to free surface problems. In this case, the air is totally neglected and therefore the domain formed only by the fluid is moving and must be treated with a ALE approach. The special feature of the method is that fixed background mesh is used and at each time step the actual mesh corresponds to a portion of this mesh. A complete description of the FM-ALE approach with extensions to other problems can be found in [22].

Figure 2 presents an schematic of the FM-ALE approach. The left figure shows the mesh M^n that corresponds to the domain Ω^n . It is a portion of the mesh M^0 that extends over the whole domain that may be occupied by the fluid. It also shows the domain velocities that are used to deform the mesh to time step $n+1$. The virtually deformed mesh is shown in thick lines in the right figure. The term virtual refers to the fact that it is not used to calculate the results at time step $n+1$. Instead, the domain velocity and the velocity at time step n are projected from the deformed mesh onto the new mesh M^{n+1} where $\mathbf{u}^{n+1}$ and p^{n+1} are finally computed. The substeps needed to evolve from time step n to time step $n+1$ can be summarized as follows:

1. Find the interface position at step $n+1$ by solving the level set equation with $\mathbf{u}^n$.
2. Compute the mesh velocity $\mathbf{u}_{\text{dom}}^{n+1}$ (see [3]).
3. Virtually deform the mesh M^n to M_{virt}^{n+1} .
4. Define the mesh on Ω^{n+1} , M^{n+1} .
5. Project $\mathbf{u}_{\text{dom}}^{n+1}$ and $\mathbf{u}^n$ from M_{virt}^{n+1} to M^{n+1} .
6. Solve on M^{n+1} to compute $\mathbf{u}^{n+1}$ and p^{n+1} .

The main differences with the classical ALE approach are

- The front is represented by a level set function, and not by the position of the material points.
- The front does not coincide with element faces. Cut elements are split into subelements only for integration purposes. As in the simplified Eulerian free surface model, this allows to apply the zero traction boundary condition correctly at the exact position of the interface.
- After deforming the mesh from one time step to the other using classical ALE procedures, the results are projected back to the original mesh.

3. SPACE AND TIME DISCRETIZATION

Both the Navier–Stokes and the level set equations are solved using a finite element model based on a stabilized finite element method. Linear P1 elements are used for all the unknowns, although the formulation can be applied with any interpolation.

Let V_h^* and Q_h^* be the finite element spaces to interpolate vector and scalar functions, respectively, constructed in the usual manner and using the same interpolation from a finite element partition $\Omega = \bigcup \Omega^e$, $e=1, \dots, n_{el}$, where n_{el} is the number of elements. In the case of the free surface formulation, we are proposing in this paper that Ω is not formed by all the elements but only by those full of liquid and by the liquid part of elements cut by the interface, as determined by the level set function ψ . From these spaces, one can construct the subspaces $V_{h,u}$ and Q_h for the velocity and the pressure, respectively. The former incorporates the Dirichlet conditions for the velocity components and the latter has one pressure fixed to zero if the normal component of the velocity is prescribed on the whole boundary. The space of velocity test functions, denoted by V_h , is constructed as $V_{h,u}$ but with functions vanishing on the Dirichlet boundary. A BDF time discretization with time step δt is used to discretize the problem in time.

Orthogonal subgrid scale (OSS) stabilization [23, 24] is used to deal with convection-dominated flows and to circumvent the well-known div-stability restriction for the velocity and pressure finite element spaces [25], allowing in particular for equal interpolation of both unknowns. The OSS monolithic discrete problem associated with the Navier–Stokes equations (1)–(2), discretizing in time using BDF2, and linearizing the convective term using a Picard scheme, can be written as follows: Given a velocity $\mathbf{u}_h^n$ at time t^n and a guess for the unknowns at an iteration $i-1$ at time t^{n+1} , find $\mathbf{u}_h^{n+1,i} \in V_{h,u}$ and $p_h^{n+1,i} \in Q_h$, by solving the discrete variational problem:

$$\begin{aligned} \int_{\Omega} \left[\frac{\rho}{\delta t} \left(\frac{3}{2} \mathbf{u}_h^{n+1,i} - 2\mathbf{u}_h^n + \frac{1}{2} \mathbf{u}_h^{n-1} \right) \cdot \mathbf{v}_h + 2\mu \varepsilon(\mathbf{u}_h^{n+1,i}) : \varepsilon(\mathbf{v}_h) + \rho(\mathbf{u}_h^{n+1,i-1} \cdot \nabla \mathbf{u}_h^{n+1,i}) \cdot \mathbf{v}_h \right. \\ \left. - p_h^{n+1,i} \nabla \cdot \mathbf{v}_h - \mathbf{f}^{n+1} \cdot \mathbf{v}_h \right] d\Omega + \sum_e \int_{\Omega^e} \tau_1^{n+1,i-1} \rho(\mathbf{u}_h^{n+1,i-1} \cdot \nabla \mathbf{v}_h) \\ \cdot [(\rho \mathbf{u}_h^{n+1,i-1} \cdot \nabla \mathbf{u}_h^{n+1,i}) - P_h(\rho \mathbf{u}_h^{n+1,i-1} \cdot \nabla \mathbf{u}_h^{n+1,i-1})] d\Omega = 0, \\ \int_{\Omega} [q_h \nabla \cdot \mathbf{u}_h^{n+1,i}] d\Omega + \sum_e \int_{\Omega^e} \tau_1^{n+1,i-1} \nabla q_h \cdot [(\nabla p_h^{n+1,i} - \mathbf{f}^{n+1}) - P_h(\nabla p_h^{n+1,i-1} - \mathbf{f}^{n+1})] d\Omega = 0, \end{aligned}$$

for $i=1, 2, \dots$ until convergence, that is to say, until $\mathbf{u}_h^{n+1,i-1} \approx \mathbf{u}_h^{n+1,i}$ and $p_h^{n+1,i} \approx p_h^{n+1,i-1}$ in the norm defined by the user. P_h is the projection onto the finite element space [23] obtained in our case with a lumped mass matrix.

In fact, the formulation described corresponds to one of those analyzed in [24] for the linearized stationary problem, in which stabilization of the convective term and the pressure gradient are

treated independently. In fact, for the latter we have found more efficient to include the body forces in the projection onto the finite element space.

The parameter τ_1 is chosen in order to obtain a stable numerical scheme with optimal convergence rates (see [23] and the references therein for details). It is computed within each element domain Ω^e as

$$\tau_1 = \left[\frac{4\mu}{(h_{\min}^e)^2} + \frac{2\rho|\mathbf{u}^e|}{h_{\text{dir_flow}}^e} \right]^{-1},$$

where $h_{\min}^e$ is minimum element length and $h_{\text{dir_flow}}^e$ is the element length in the direction of the flow.

For the level set we use SUPG stabilization due to the pure convective type of equation. The temporal evolution is treated with a BDF1 time discretization.

4. MIXED BOUNDARY CONDITIONS ON CURVED WALLS

In Equation (3), we have defined mixed boundary conditions for the Navier–Stokes equations as the conditions that prescribe a zero velocity in the normal direction to the boundary and a traction in the tangential direction. Two typical cases are slip boundary conditions, where the tangential traction is zero, and wall boundary conditions, where the tangential traction depends on the velocity. For the numerical simulation of turbulent flows on complex geometries, such as the mould filling examples presented in this work, the use of wall laws is mandatory since the simulation of the boundary layer is computationally too expensive.

When domains with curved boundaries are discretized using the finite element method, the normal to the different element faces belonging to the boundary that meet at a node do not coincide. In order to apply the zero normal velocity condition to the discretized problem, once the system that describes the problem without boundary conditions has been obtained in Cartesian coordinates, the degrees of freedom corresponding to nodes on the curved boundary must be rotated into a local system such that the normal component can be prescribed to zero. The problem with curved boundaries is that the normal defined at the node does not coincide with the normal to each of the element faces on the boundary that meet at the node. Therefore, the normal velocity to the faces will not be exactly zero and there will be some flow through the faces (the condition $\mathbf{n} \cdot \mathbf{u} = 0$ will not be exactly satisfied at the discrete level). The normal at the node is usually defined in such a way that the flow through the element faces on the boundary associated with the velocity at the node is zero [26]. Such nodal normals are called consistent normals because they are the best solution to satisfy mass conservation.

Even when consistent normals are used, problems that have not been discussed until quite recently [16] can be observed. If one tries to simulate a fluid at rest under gravity forces in a domain with curved boundaries, using mixed boundary conditions spurious velocities will appear. Figure 3 shows the spurious velocities obtained in a circular domain of radius $r = 1.0$ with slip boundary conditions for $\rho = 1$, $\mu = 1$ and $g = 10$ in the downward direction. For real flows, these spurious velocities can be hidden by the real velocities and are not noticed. As the Froude number decreases they become more noticeable, specially on coarse meshes.

In [16], a solution for this problem in free surface flows has been proposed that consists in applying a ‘do nothing’ boundary condition [27] on mixed boundaries. Actually, the normal component to the node is subtracted from the usual ‘do nothing’ boundary conditions but this has no effect on the flow but only on the resulting reactions. Therefore, unless one is interested on the reactions on the boundary, the usual ‘do nothing’ boundary condition can provide exactly the same solution. In [16], only slip boundary conditions are discussed but the method could easily be extended to other mixed boundaries.

‘Do nothing’ boundary conditions consist in sending to the matrix side (instead of right-hand side) the traction boundary conditions when tractions are not known. Despite at the continuous level this leads to an ill-posed problem; for the discrete finite element problem, it typically yields

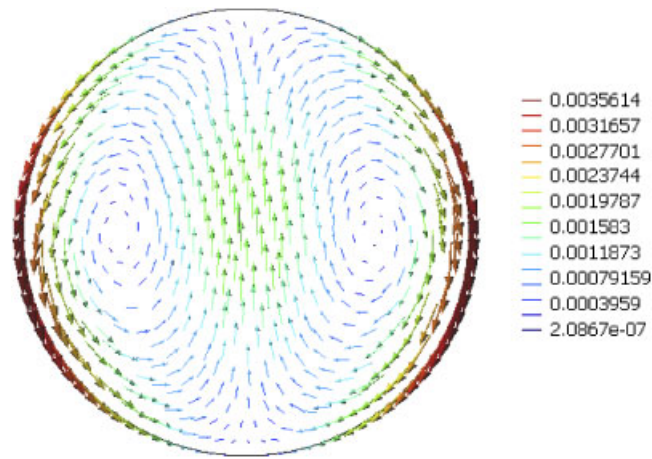


Figure 3. Spurious velocities obtained for a downward gravity force.

very good results. It is commonly used at outflow boundaries where both normal and tangent components of the traction vector are unknown. For the mixed boundary cases we assume that the tangent components of the traction vector are known and therefore we believe that the ‘do nothing’ boundary condition should be applied only in the normal direction and not in all the directions as proposed in [16]. That is to say, instead of integrating (on the matrix side)

$$-\int_{\Gamma_{\text{mu}}} \mathbf{v} \cdot [-p\mathbf{n} + 2\mu\mathbf{n} \cdot \boldsymbol{\varepsilon}(\mathbf{u})] d\Gamma = -\int_{\Gamma_{\text{mu}}} \mathbf{v} \cdot [\mathbf{n} \cdot \boldsymbol{\sigma}] d\Gamma$$

as proposed in [16], one should only integrate

$$-\int_{\Gamma_{\text{mu}}} \mathbf{v} \cdot \{\mathbf{n} \cdot [-p\mathbf{n} + 2\mu\mathbf{n} \cdot \boldsymbol{\varepsilon}(\mathbf{u})]\} \mathbf{n} d\Gamma = -\int_{\Gamma_{\text{mu}}} \mathbf{v} \cdot [(\mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{n})\mathbf{n}] d\Gamma$$

For a flow at rest, we have obtained the exact solution with both formulations. For a problem with non-zero velocities, the first condition we must expect from the proposed formulations is that in the case with planar boundaries the solution obtained without any modification should remain unaltered when the modification for curved boundaries is applied. This is what happens when the ‘do nothing’ boundary condition in the normal direction we propose is applied. When the modification proposed in [16] is used, the boundary conditions in the tangential direction are altered and so is the solution.

In order to show the difference between using the ‘do nothing’ boundary condition only in the normal direction or in all directions, the test case we propose is the Stokes flow in a channel. Taking into account symmetry, only half of the channel is simulated. On the symmetry face a slip boundary condition is applied. Despite the test case does not have curved boundaries, the proposed correction should also work on planar boundaries. Moreover, the absence of curved boundaries allows us to solve the problem with no special correction. In Figure 4, we compare the solutions obtained with both corrections against the solution obtained without any correction. The results obtained without any boundary correction are exactly the same as those obtained with the ‘do nothing’ boundary condition applied only in the normal direction. On the other hand, the use of the ‘do nothing’ boundary condition in all directions [16] leads to an incorrect solution because the correct boundary condition on the symmetry face has been altered. The vertical gradient of the horizontal velocity is not forced to be zero at the bottom wall. Therefore, any horizontal constant gradient velocity that is zero at the upper face can be added to the solution. This is what makes the system matrix singular and leads to the meaningless solution obtained when the ‘do nothing’ boundary condition is applied in all directions.

The test case that has been presented is an extreme case to show the improvement introduced by our modification to the method proposed in [16]. Since the implementation and the computational

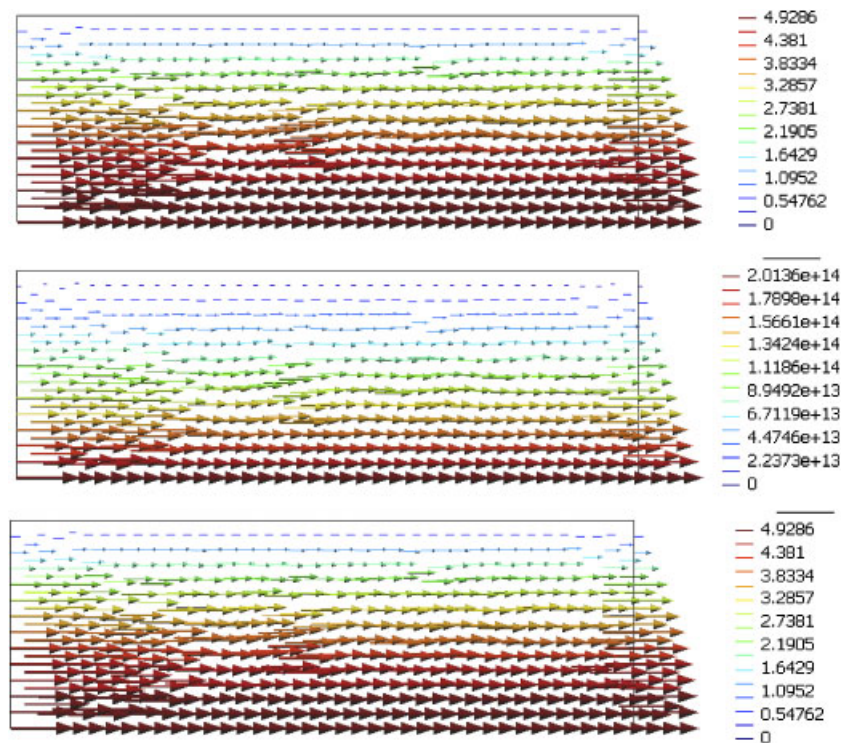


Figure 4. Velocities in a channel: (top) no modification, (middle) modification proposed in [16] and (bottom) our modification.

cost of both methods is nearly the same, we believe that the ‘do nothing’ BC only in the normal direction should be used.

5. MOULD FILLING EXAMPLES

In this section, three examples borrowed directly from the foundry are used to test the improvements introduced by the proposed method.

Since the flow in mould filling problems is turbulent, the viscosity in the Navier–Stokes equations has been calculated using the Smagorinsky model as $\mu = \mu_L + \mu_T$, where μ_L is the molecular, constant, viscosity and $\mu_T = \mu_T(\mathbf{u})$ is the additional turbulent viscosity defined by $\mu_T = \rho C^2 h^2 \sqrt{2 \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{u})}$, where h is the size of the element where it is computed and C^2 is the Smagorinsky constant. $C^2 = 0.05$ is used in the examples. The objective of this work is not related to the analysis of the influence of the turbulence model and therefore a simple model has been chosen. The Smagorinsky model is also used for mould filling simulations in [28, 29].

Owing to the high Reynolds number of the problems we are dealing with, no slip boundary conditions would require extremely fine meshes along the boundary that would make them computationally unfeasible. The solution we have adopted is to use wall functions [15] that describe the behavior of the flow near a solid wall. The normal component of the velocity is set to zero. In the tangential direction, a traction that depends on the velocity at the boundary and is opposed to the direction of the flow is applied:

$$\tau_w = -\rho \frac{u_*^2}{|\mathbf{u}|} \mathbf{u},$$

where u_* can be determined from the following set of equations:

$$\begin{aligned} u^+ &= \frac{1}{\kappa} \ln(1 + \kappa y^+) + 7.8 \left[1 - e^{-y^+/11.0} - \frac{y^+}{11.0} e^{-0.33y^+} \right], \\ y^+ &= \frac{\rho \delta u_*}{\mu}, \\ u^+ &= \frac{\rho |u| u_*}{\tau_w}. \end{aligned}$$

δ is the distance between the computational boundary and the wall, $\kappa=0.41$ is the Von Karman constant and y^+ and u^+ are non-dimensional distances and velocities, respectively. The correction presented in Section 4 has been vital for avoiding spurious velocities when wall laws are used for low Froude number examples.

OSS stabilization has been used for the Navier–Stokes equations, but some cases have also been run with ASGS [23] and no significant difference has been observed. For the convective nonlinearity, the Picard iteration is used. The tolerance is set to one percent variation in the L2 norm of the velocity and a maximum of 7 iterations are allowed. Typically the nonlinearity converges in less than three iterations. For the solution of the monolithic system, a preconditioned GMRES iterative solver [30] is used. The stopping criteria for the solver is that the residual is smaller than 10^{-6} times the right-hand side. A maximum of 500 iterations are allowed, but the solver usually converges in less than 30 iterations. The Krylov dimension is set to 50. An ILUT preconditioner with threshold 0.001 and filling 20 is used [30]. For the Level Set equation, the convergence of the GMRES solver is very easy even without preconditioner.

In Section 2 two alternatives have been proposed for solving the free surface problem, a FM-ALE model and a simplified Eulerian model. As we have already mentioned, the results obtained with both models are very similar. The results obtained with the simplified Eulerian model are presented because they are computationally cheaper. At the end of this section, the results obtained with the FM-ALE model are briefly discussed.

The runs were performed on a PC with AMD Athlon(tm) 64 X2 Dual Core Processor 4400+ running at 2.2 GHz with 3 GB of RAM using the Intel Fortran compiler under Ubuntu.

5.1. Hollow mechanical piece

The first example is a hollow mechanical piece made of steel with physical properties: $\rho=7266.0$ and $\mu=6.7 \times 10^{-3}$ (SI units). This piece is interesting because it has relatively thin walls that make the mesh quite complex. The code is forced to obtain acceptable results with few elements in the thickness. The arrangement we simulate consists of two pieces together with the filling channel used during the actual filling process. The inlet velocity is 0.113 m/s and the size of each piece is approximately $0.16 \times 0.16 \times 0.13 \text{ m}^3$. The whole filling process takes 11 s.

Two unstructured triangular meshes have been used. The coarse one has 72 032 elements and 16 149 nodes and the fine one has 575 803 elements and 116 214 nodes. They are shown in Figure 5. For both meshes, the time steps' size is 0.02 s. The Reynolds number based on the inlet velocity and the length of the filling channel is $Re=2.45 \times 10^4$, and the Froude number is $Fr=0.0065$.

Figure 6 shows the evolution of the interface for four time steps during the filling process using the coarse mesh. In the first step, the interface is still inside the filling channel. For the second one, it has entered both pieces. In the third one, the interface reaches the bottom of each piece. As we will comment later, this is one of the most complicated moments in the simulation. In the final figure, more than half of each piece has been filled. The evolution of the front is very similar in both pieces. Despite a coarse grid has been used, the evolution of the interface is captured quite satisfactorily as one can observe by comparing with the results shown for the fine mesh in Figure 7.

Knowing how the interface evolves is important during the mould design as it can be used to change the position of the inlets or alter the filling velocity to improve the quality of the resulting

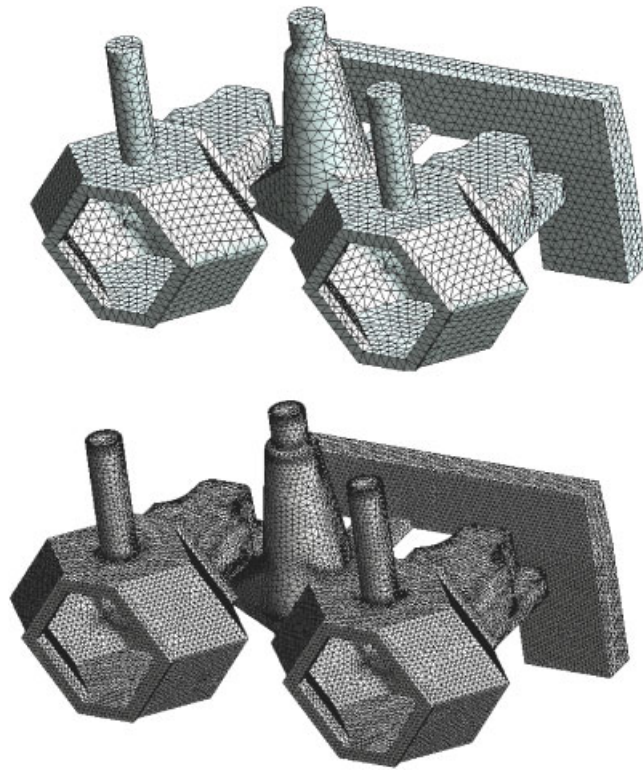


Figure 5. Coarse and fine meshes for the hollow mechanical piece.

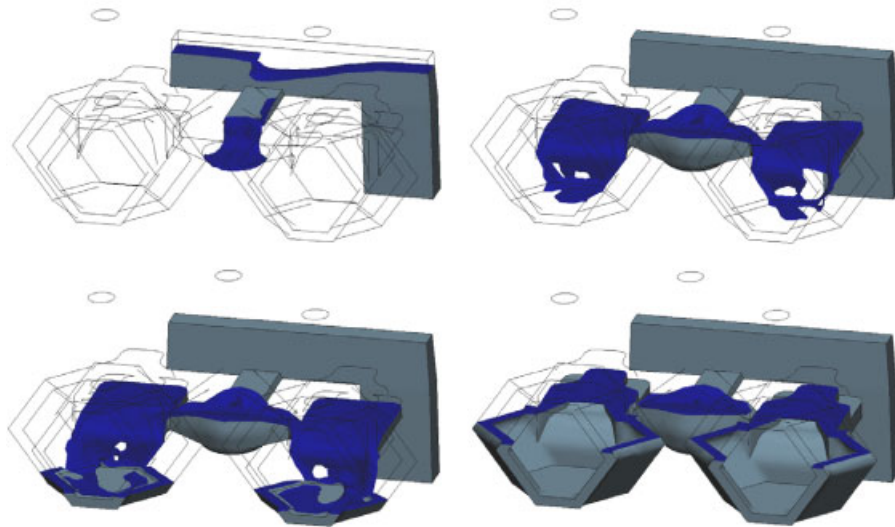


Figure 6. Interface position at $t = 1.6, 3.2, 4.8$ and 7.6 s using the coarse mesh.

piece. When defects appear, having some insight on the way the flow evolves is of great help to the foundry person because it is very difficult to actually see what is happening inside the mould.

The evolution of the interface using the fine mesh is shown in Figure 7. The shape of the interface is smoother than the one obtained with the coarse mesh, but there is no major difference in the way the flow evolves. The most noticeable change is that for each time step, the results obtained with the fine mesh show a bigger percentage of filled volume. This is related to numerical mass

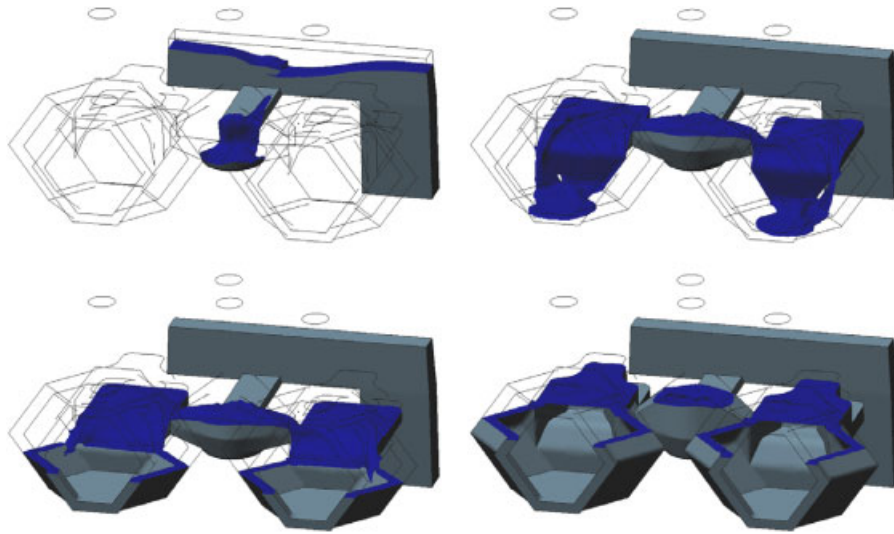


Figure 7. Interface position at $t = 1.6, 3.2, 4.8$ and 7.6 s using the fine mesh.

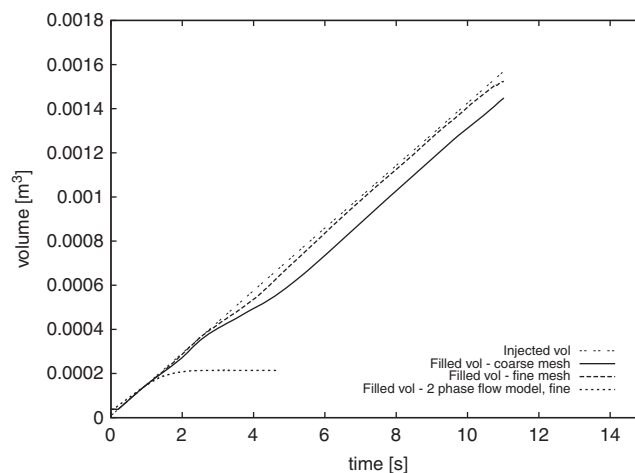


Figure 8. Filled volume vs injected volume for both meshes.

losses and is analyzed in more detail in Figure 8. Since foundry pieces are usually complex and it is common to fill several pieces at the same time (not only two as in the example), it is important to have a code that can provide the user with acceptable results even with coarse meshes.

In Figure 8, we compare temporal evolution of the injected and filled volumes using both meshes. The injected volume is the same for both meshes. The difference between the filled and the injected volumes is the numerical mass loss. It is reduced as the mesh is refined as one would expect. The amount of mass loss can give us some idea on the quality of our results and indicates the most complex moments during the simulation. In our example, we can see that the most important mass loss occurs when the filled volume is between 0.0004 and 0.0006 m^3 . It corresponds to the moment when the bottom of each piece is being filled. This suggests that a mesh refinement close to that area might improve the solution. When the fine mesh is used, the mass loss is very small. We have also included results obtained with the Eulerian two-phase flow model and the fine mesh to show that after approximately 1.5 s, the mass loss becomes so important that all of the injected volume has been lost due to numerical errors. In [3], we have observed the same behavior in simpler examples.

Table I. Cpu time.

	Total (s)	Matrix N. Stokes (%)	Solver N. Stokes (%)
Coarse mesh	12 593	69.6	20.2
Fine mesh	90 038	45.2	45.6

In Table I, the computational times for the previous simulations are presented. It can be observed that the solution of the Navier–Stokes equations requires most of the time. For the coarse mesh, the assembly of the matrix takes approximately three times more time than the solution of the linear system. With the fine mesh, both tasks take approximately the same time.

5.2. Alloy wheel

The second example is an automotive alloy wheel. The flow is created by applying a pressure on the fluid as is done in the actual filling process for this piece. The flow rate is then determined by the resistance exerted on the fluid. We have observed that the friction may be high in the vertical tube through which the molten metal is injected. Therefore, for this case, we will simulate the whole filling channel.

The pressure at the inlet varies linearly from $2.21 \times 10^4 \text{ N/m}^2$ at the beginning of the simulation to $1.17 \times 10^5 \text{ N/m}^2$ after 4.4 s. The physical properties we have used are those of aluminum, $\rho = 2700.0$ and $\mu = 1.3 \times 10^{-3}$ (SI units). The Reynolds number based on a typical velocity inside the wheel (0.5 m/s) and the wheel radius (0.5 m) is $Re = 5.19 \times 10^5$. The Froude number is $Fr = 0.05$. The mesh is formed by 489 313 tetrahedral elements and 109 318 nodes. The time steps size is 0.01 s.

Figure 9 shows the evolution of the interface for different time steps. For the first time step, the whole domain is shown and for the remaining steps only the details at the wheel are shown. Once the molten metal reaches the top of the filling tube, it slides through the bottom of the spokes until it reaches their end. Then it turns and fills the lower part of the wheel. Finally it raises through the vertical walls of the wheel. Simultaneously, the filling of the spokes is completed. Since we are using a free surface model, air is not taken into account; hence, there is no possibility for the formation of air bubbles. The free surface model can be modified so that it can take into account the formation of bubbles as suggested in [20, 28]. Moulds can be classified into two groups depending on whether they allow air to escape through their walls or not. When the air is allowed to escape, such as in sand moulds, the importance of taking into account the effect of air bubbles is less important.

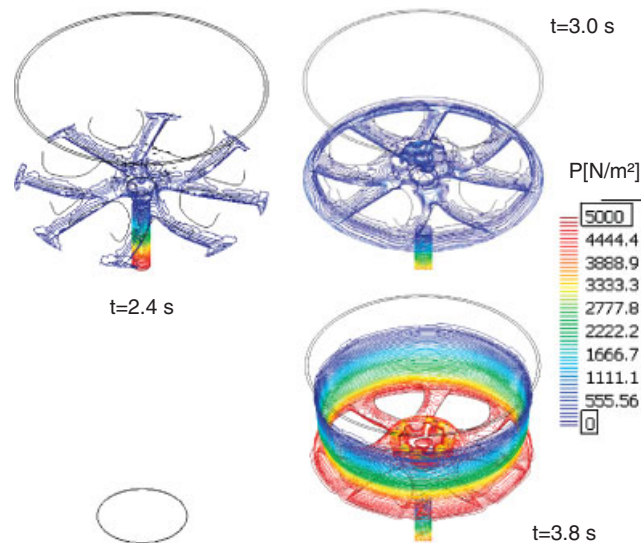
The pressure contour lines are presented in Figure 10. A fixed scale with a maximum of 5000 N/m^2 has been used to focus on the pressures inside the wheel. In the filling tube the pressure is nearly hydrostatic, but due to the scale we have used it is not shown.

The total CPU time for the simulation has been 38 579 s. As in the previous example, the resolution of the Navier–Stokes equations took most of the time, with 18 220 s for the matrix assembly and 15 470 s for the linear solver.

5.3. Shovel

The third piece was presented to us as a really demanding case. It is the shovel for a power shovel. The filling process takes approximately half a minute and the shovel is nearly 1 m long. The inlet velocity we have used during the simulation is 0.5 m/s. The Reynolds number based on the previous velocity and length is $Re = 4.44 \times 10^5$, and the Froude number is $Fr = 0.031$. As in the first example, the material used is steel. The mesh is formed by 1 619 428 tetrahedral elements and 319 052 nodes. The time steps size is 0.01 s.

Figure 11 shows the evolution of the interface for selected time steps. The filling channel used for this piece splits into two branches. One of the branches is closer to the inlet than the other one. As the interface reaches the first branch, the molten metal starts falling through it. Approximately 1 s is the time before the flow starts falling through the second branch. Therefore, the side of the

Figure 9. Interface evolution at $t = 2.4$, 3.0 and 3.4 s.Figure 10. Pressures at $t = 2.4$, 3.0 and 3.4 s.

shovel closer to the first branch is filled earlier than the part connected to the second branch. When the molten metal exits each of the two branches, it slides into two circular parts with a hole in the middle located beneath the end of each branch. Once when each of these two parts are full, the flow spreads through the base of the shovel and finally raises along the lateral walls to complete the filling process.

In Figure 12, we compare the temporal evolution of the injected and filled volumes. There is some mass loss that concentrates mainly during the filling of the two circular parts located at the bottom of the shovel. We have observed that this mass loss can be reduced by increasing the Smagorinsky constant. The mass loss could also be corrected by slightly displacing the interface at

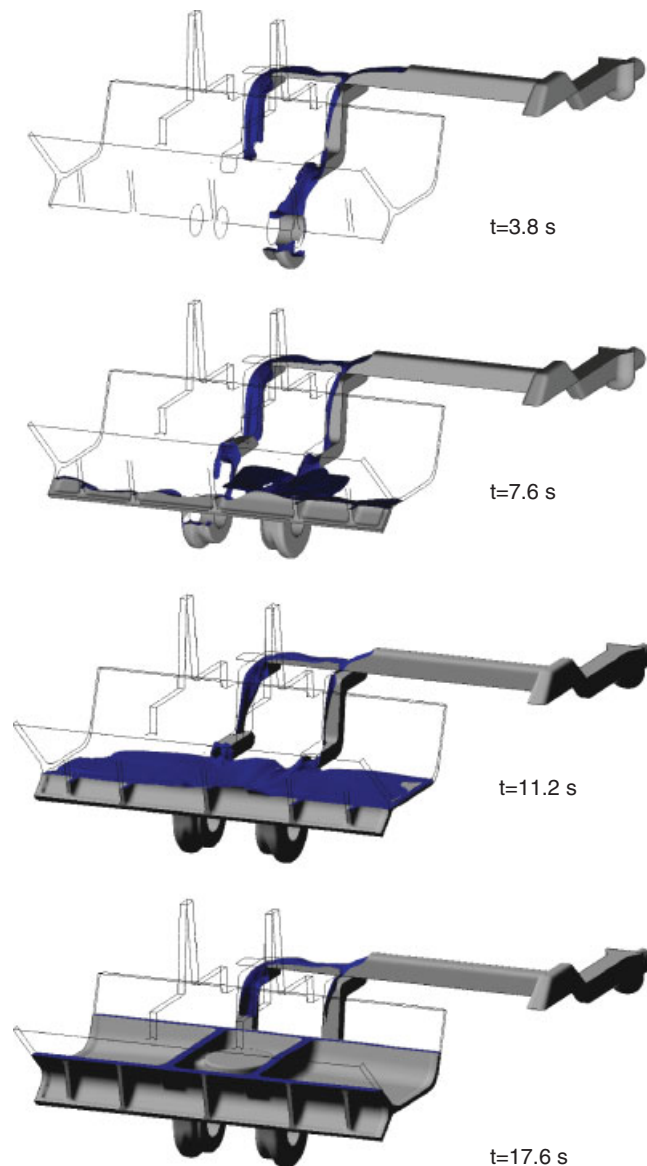


Figure 11. Interface position.

each time step as suggested in [20, 31]. As in the hollow mechanical piece, we have also included results obtained with the Eulerian two-phase flow model to see that after approximately 4 s the mass loss becomes so important that the results lose any sense.

Despite, we only show results for one mesh. This example has also been run with a coarser mesh. The conclusions from the comparison between the results on both meshes agree with those obtained for the hollow mechanical piece. That is, the evolution of the front is very similar on both meshes and the main difference is that the mass loss is reduced as the mesh is refined. Therefore, we believe that although the results shown in Figure 12 show a mass loss of approximately 13%, they can be considered representative of the actual flow of molten metal. If a smaller mass loss is required, a finer mesh should be used.

The total CPU time for the simulation has been 915 204 s. Similar to what happens in the previous examples, 40.1 and 39.8% of this time correspond to the matrix assembly and linear solver for the Navier–Stokes equations, respectively.

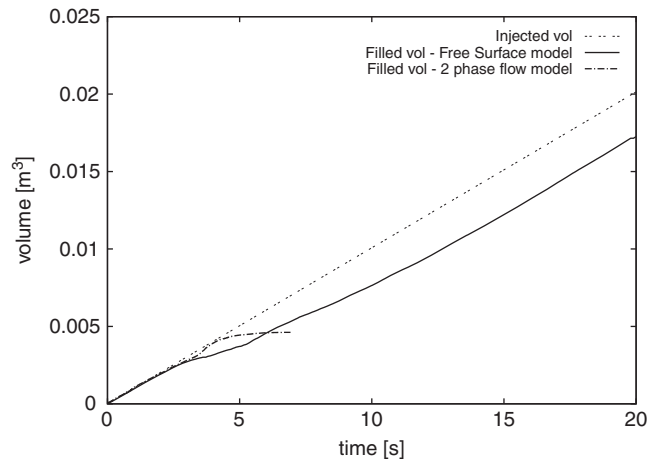


Figure 12. Filled volume vs injected volume.

Table II. Cpu time.

	Total (s)	Matrix N. Stokes (%)	Solver N. Stokes (%)
Hollow coarse	25 568	44.0	11.8
Hollow fine	186 951	26.2	28.9
Wheel	80 199	24.5	21.3

5.4. Results with the FM-ALE model

As we have anticipated at the beginning of this section, for the mould filling examples we have not observed any significant difference between the results obtained with the simplified Eulerian model and those obtained with the FM-ALE model whose derivation could be considered more sound since less approximations are assumed. The variations introduced in the evolution of the free surface or the filled volume by the use of the FM-ALE model instead of the simplified Eulerian model are hardly noticeable and therefore the figures are not to be repeated.

In Table II, we present the CPU times obtained with the FM-ALE so that they can be compared against the ones obtained with the simplified Eulerian model. The computational times for the Navier–Stokes matrix assembly and the solution of the linear system are similar to the ones obtained with the simplified Eulerian model. The total CPU time increases due to the additional steps required by the FM-ALE model. For the moment, little effort has been put into optimizing those steps because as both models produce similar results it has been cheaper to use the simplified Eulerian model.

Despite, we have not found significant differences between the results obtained with the FM-ALE model and the simplified Eulerian model, we believe that in more demanding examples the FM-ALE model may prove advantageous. In [22], we have extended the application of the FM-ALE model to a wider range of problems, including fluid–structure interaction. An example of a cylinder moving in a rectangular domain is presented where the advantage of using the FM-ALE model is easily observed. It is mentioned that a large time step is used in order to observe the improvements introduced by the FM-ALE model.

6. CONCLUSIONS

In this paper, we have extended our free surface flow model to 3D mould filling problems borrowed directly from the foundry. The advantages over the usual two-phase flow model already observed

in [3] are also verified in the low Froude number mould filling examples presented in this work. The correct imposition of boundary conditions at the interface is the key ingredient of the method we propose. For low Froude number flows, where the coupling between the position of the front and the resulting flow field is high, the region close to the interface must be simulated as accurately as possible.

Two versions of the free surface model have been proposed in [3], one that uses an ALE formulation on fixed meshes (FM-ALE) and the other one that is more closely related to the Eulerian formulations. As in the simpler examples presented in [3], no significant difference has been observed between the numerical results obtained with both versions.

Compared with the two-phase model, not only does the free surface model we propose avoid spurious velocities and enhance mass conservation, thanks to the correct imposition of boundary conditions at the interface, but also the computational cost is reduced because only the fluid is solved. If the influence of the air must be taken into account, modifications similar to those proposed in [20, 28] could be included.

The low Froude number regime has also introduced difficulties for the use of wall laws in domains with curved boundaries. We have developed an improvement to the approach proposed in [16] that avoids the creation of spurious velocities close to the walls.

Three mould fillings have been used to test the validity of the approach we propose. They involve low Froude Number flows that, as we have already mentioned, introduce significant difficulties. On the other hand, the presence of thin walls leads to big but nevertheless coarse meshes that expose the model to the difficulties encountered in real problems. The results we have obtained can be considered very satisfactory. The small mass losses that we have preserved as an indicator of accuracy of the model could easily be corrected by slightly moving the interface to compensate the mass loss as suggested in [20, 31].

A simple reinitialization technique explained in Section 2.1 has been used. More advanced reinitialization techniques could have been used [6, 11, 18]. This could be the subject of a future publication. In any case, for the low Froude number examples that have been presented, we expect that the improvements will be much smaller than those obtained with a free surface model that imposes boundary conditions at the interface accurately. Possibly, the advantage of an improved reinitialization would be an improved computational efficiency and not so much an improved accuracy.

ACKNOWLEDGEMENTS

H. Coppola-Owen would like to acknowledge the support received from the Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya (Catalan Government) and the European Social Fund through a doctoral grant.

REFERENCES

1. Cross M, Pericleous K, Croft TN, McBride D, Lawrence JA, Williams AJ. Computational modeling of mold filling and related free-surface flows in shape casting: an overview of the challenges involved. *Metallurgical and Materials Transactions B* 2006; **37B**:879–885.
2. Coppola-Owen AH, Codina R. Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions. *International Journal for Numerical Methods in Fluids* 2005; **49**:1278–1304.
3. Coppola-Owen AH, Codina R. A finite element model for free surface flows on fixed meshes. *International Journal for Numerical Methods in Fluids* 2007; **54**:1151–1171.
4. Chang Y, Hou T, Merriman B, Osher S. A level set formulation of Eulerian interface capturing methods. *Journal of Computational Physics* 1996; **124**:449–464.
5. Sussman M, Almgren AS, Colella JB, Howell LH, Welcome ML. An adaptive level set approach for incompressible two phase flows. *Journal of Computational Physics* 1999; **148**:81–124.
6. Osher S, Fedkiw RP. *Level Set Methods and Dynamic Implicit Surfaces*. Springer: Berlin, 2003.
7. Thompson E. Use of the pseudo-concentration to follow creeping viscous during transient analysis. *International Journal for Numerical Methods in Engineering* 1986; **6**:749–761.
8. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.

9. Codina R, Schäfer U, Oñate E. Mould filling simulation using finite elements. *International Journal of Numerical Methods for Heat and Fluid Flow* 1994; **4**:291–310.
10. Lewis RW, Usmani AS, Cross JT. Efficient mould filling simulation in metal castings by an explicit finite element method. *International Journal for Numerical Methods in Engineering* 1995; **20**:493–506.
11. Codina R, Soto O. A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique. *International Journal for Numerical Methods in Fluids* 2002; **40**:293–301.
12. Ramaswamy B. Numerical simulation of unsteady viscous free surface flow. *Journal of Computational Physics* 1990; **90**:396–430.
13. Ramaswamy B, Kawahara M, Nakayama T. Lagrangian finite element method for the analysis of two-dimensional sloshing problems. *International Journal for Numerical Methods in Fluids* 1985; **6**:659–670.
14. Radovitzky R, Ortiz M. Lagrangian finite element analysis of Newtonian fluid flows. *Journal of Computational Physics* 1998; **43**:607–619.
15. Launder BE, Spalding DB. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering* 1974; **3**:269–289.
16. Behr MA. On the application of slip boundary conditions on curved boundaries. *International Journal for Numerical Methods in Fluids* 2004; **45**:43–51.
17. Pichelin E, Coupez T. Finite element solution of the 3D mold filling problem for viscous incompressible fluid. *Computer Methods in Applied Mechanics and Engineering* 1998; **163**:359–371.
18. Hartmann D, Meinke M, Schröder W. Differential equation based constrained reinitialization for level set methods. *Journal of Computational Physics* 2007; **227**:6821–6845.
19. Garcia-Espinosa J, Valls A, Oñate E. ODDL: a new unstructured mesh finite element method for the analysis of free surface flow problems. *International Journal for Numerical Methods in Engineering* 2008; **76**:1297–1327.
20. Löhner R, Yang C, Oñate E. On the simulation of flows with violent free surface motion. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:5597–5620.
21. Houzeaux G, Codina R. A finite element model for the simulation of lost foam casting. *International Journal for Numerical Methods in Fluids* 2004; **46**:203–226.
22. Codina R, Houzeaux G, Coppola-Owen H, Baiges J. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics* 2009; **228**:1591–1611.
23. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:4295–4321.
24. Codina R. Analysis of a stabilized finite element approximation of the Oseen equations using orthogonal subscales. *Applied Numerical Mathematics* 2008; **58**:264–283.
25. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: Berlin, 1991.
26. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method*. Wiley: New York, 2000.
27. Papanastasiou TC, Malamataris N, Ellwood K. A new outflow boundary condition. *International Journal for Numerical Methods in Fluids* 1992; **14**:587–608.
28. Caboussat A. Numerical simulation of two-phase free surface flows. *Archives of Computational Methods in Engineering* 2005; **12**:165–224.
29. Gao DM. A three dimensional finite element-volume tracking model for mould filling in casting processes. *International Journal for Numerical Methods in Fluids* 1999; **29**:877–895.
30. Saad Y. *Iterative Methods for Sparse Linear Systems*. PWS: Boston, MA, 1996.
31. Elias RN, Coutinho ALGA. Stabilized edge-based finite element simulation of free-surface flows. *International Journal for Numerical Methods in Fluids* 2007; **54**:965–993.