

New boundary conditions for simulating the filling stage of the injection molding process

Injection
molding
process

Wagner de Campos Galuppo

Institute for Polymers and Composites, University of Minho, Guimarães, Portugal

Ana Magalhães

FJN – Moldes, Leiria, Portugal

Luís Lima Ferrás

Department of Mathematics, Center for Mathematics, Universidade do Minho, Guimarães, Portugal, and

João Miguel Nóbrega and Célio Fernandes

Institute for Polymers and Composites, University of Minho, Guimarães, Portugal

Received 6 April 2020
Revised 12 June 2020
15 June 2020
Accepted 17 June 2020

Abstract

Purpose – The purpose of this paper is to develop new boundary conditions for simulating the injection molding process of polymer melts.

Design/methodology/approach – The boundary conditions are derived and implemented to simulate real-life air vents (used to allow the air escape from the mold). The simulations are performed in the computational library *OpenFOAM*® by considering two different fluid models, namely, Newtonian and generalized Newtonian (Bird–Carreau model).

Findings – A detailed study on the accuracy of the solver *interFoam* for simulating the filling stage is presented, by considering simple geometries and adaptive mesh refinement. The verified code is then used to study the three-dimensional filling of a more complex geometry.

Originality/value – The results obtained showed that the numerical method is stable and allows one to model the filling process, simulating the real injection molding process.

Keywords Injection molding, Simulation, Boundary conditions, *OpenFOAM*®

Paper type Research paper

1. Introduction

Nowadays, any project activities should be supported by suitable simulation tools, aiming the optimization of the process and the minimization of the resources spent on the design process. In the market, there are several modeling codes able to simulate extremely complex

This work is funded by FEDER funds through the COMPETE 2020 Programme and National Funds through FCT (Portuguese Foundation for Science and Technology) under the projects UID-B/05256/2020, UID-P/05256/2020 and MOLDPRO-Aproximações multi-escala para moldação por injeção de materiais plásticos (POCI-01-0145-FEDER-016665).

The research of L.L. Ferrás was partially financed by the Portuguese Funds through FCT within the Projects UID-B/00013/2020, UID-P/00013/2020 and the scholarship SFRH/BPD/100353/2014.

The authors would like to acknowledge the Minho University Cluster (NORTE-07-0162-FEDER-000086) for providing the HPC resources that contributed to the research results reported within this paper.



processes, but the dissemination of these codes in industry faces two major difficulties, the cost of proprietary software licenses and/or the nonexistence of human resources able to use these tools adequately. For the former, a possible solution is to resort to open source modeling software. For the latter, specialized human resources should be progressively educated to use modeling tools, to support the product development in an efficient manner. Notice the easy access to the software inherent to open-source alternatives, and it also facilitates the personnel education.

The consumption of plastics has increased significantly in the past few decades, overcoming the consumption of any other raw material. Because of their versatility, polymers can be used in several sectors, such as automotive, packaging, agriculture and electrical. To obtain the final plastic product with the desired shape, one has to transform raw material, granules or powders of plastic/polymer, into a polymer melt that can easily be shaped, and one of the most used techniques to achieve this goal is the injection molding process. Therefore, in this work, we are interested in the simulation of the polymer injection process, more precisely, the filling stage, and accordingly to what was stated above, the open-source *OpenFOAM*® (2004) software seems to be the adequate tool to perform such studies.

The first developments on the mathematical modeling of the injection molding filling stage were limited to one-dimensional cases and then evolved to the Hele–Shaw approximations that neglect pressure variation along the thickness direction and assumes a fully developed velocity profile at every location (Kennedy, 2009). Nowadays, parts are becoming increasingly complex, meaning that thickness plays an important role that cannot be overlooked. Therefore, the validity of the Hele-Shaw approximation became limited to simple geometries/conditions. To overcome this problem, the use of three-dimensional modeling to simulate injection molding is a demand. More recent studies use three-dimensional numerical models. Héту *et al.* (1998) presented a three-dimensional finite element method capable of predicting velocity, pressure, temperature and the position of the flow fronts. The polymer melt behavior was modeled using Carreau and Arrhenius constitutive models. In the same year, Pichelin and Coupez (1998) simulated the filling process considering a viscous incompressible flow under isothermal conditions. To solve the polymer melt motion equation, they introduced a Taylor-Galerkin scheme. Three years later, Chang and Yang (2001) presented a finite volume approach to simulate the injection molding filling stage. They simulated the isothermal flow of an incompressible Newtonian fluid. They also compared their results with the results obtained by assuming the Hele-Shaw simplification. The two approaches produced identical results for the filling of thin cavities, but the three-dimensional model showed a better accuracy in the filling of thicker cavities. More recently, Yan *et al.* (2007) solved the Navier-Stokes equations using streamline-upwind/Petrov-Galerkin (SUPG) and pressure-stabilizing/Petrov-Galerkin (PSPG) formulations. Their results were compared with the proprietary software Moldflow (Autodesk, 2020), revealing that the applied model achieved identical results for the filling process. A large number of works are currently being published on the subject of numerical simulation of the injection molding process, considering new and different phenomena that may occur along the process, and also taking into account the rheology of the different fluids. For instance, the study of Azaman *et al.* (2013) on residual stress distribution in the injection molding process using wood polymer composites and the study of Kim and Isayev (2015) in birefringence using co-injection molding.

A more recent work on this topic is given by La *et al.* (2018), where a numerical analysis of the injection molding process was carried out using the interpolated domain decomposition method. The information on the filling behavior of the polymer melt was first calculated from a 2.5D injection molding analysis. The 3D injection molding analysis was then conducted with

the calculated pressure distribution. This technique is suitable for micro-injection, but when used at macro scales it loses information and cannot capture all the flow features.

The main problem with 3D simulations is the computational cost. This computational cost increases with mesh refinement, and therefore, highly accurate transient numerical simulations require additional computational power (clusters and supercomputers) that may not be accessible to everyone. A possible solution is the use of local adaptive mesh refinement, i.e. performing a mesh refinement only in the regions of higher gradients (for example, at the interface). This technique will be explored in this work.

Another possible solution to reduce the computational costs is the use of a different type of discretization of the governing equations, where the traditional mesh is replaced by new methods that can more easily deal with the interface topology, discretization and implementation. This is the case of the smoothed-particle hydrodynamics (SPH) method, a meshfree Lagrangian method. The computational cost of SPH simulations per number of particles is significantly less than the cost of grid-based simulations per number of cells (when the metric of interest is related to fluid density). The major drawback of this method is the accurate implementation of boundary conditions. The works of [Xu and Yu \(2019, 2017\)](#) illustrate (for the first time) the feasibility of this method to deal with the non-isothermal injection molding processes. They use the corrected kernel gradient to increase the computational accuracy, and the Rusanov flux is introduced into the continuity equation to alleviate large and random pressure oscillations. They provide an enhanced treatment of the wall boundary, which can model arbitrary-shaped mold walls. It is shown that the SPH method is a powerful computational tool for the simulation of non-isothermal free surface flows during the injection molding process.

It should be remarked that although 3D simulations are the most accurate and reliable numerical techniques for modeling the injection molding process (when compared to any other approximate technique such as: 1D, 2.5D or Hele-Shaw), in some cases this simplified analysis is of importance ([Mollaabbasi et al., 2019](#)). It all depends on the filling process being studied and the *need* for accurate information. For example, in the analysis performed in [Mollaabbasi et al. \(2019\)](#), one cannot predict the interface propagation, but the results are qualitatively sound and compare with the experimental counterpart.

The objective of this work is then the modeling of the filling stage by simulating the real-life air vents (used to extract the air from the mold). For that, new boundary conditions are proposed and the equations governing the two-phase flows are solved using the Volume of Fluid (VoF) method. The simulations are performed in the open-source computational library *OpenFOAM®* considering two different fluids, namely, Newtonian and generalized Newtonian (Bird-Carreau model). A detailed study on the accuracy of the *interFoam* solver for simulating the filling stage is presented, and, the verified code is then used to study the filling of a specimen. Notice that the originality of the current manuscript lays on the idea to simulate the venting during the injection molding, different from the previous proposed approaches, and provides an additional way for the injection molding community to optimize the flow uniformity. Additionally, the novelty of this work is on the detailed 3D numerical study using both static and dynamic meshes.

Regarding real-life air vents, we have that the dimensions may go from $0.036 \times 12 \text{ mm}^2$ for Nylon, Acetal and PET to $0.077 \times 12 \text{ mm}^2$ for most thermoplastics except Nylon and Acetal ([Rosato and Rosato, 2012](#)). For the simulation of such air vents, the most common method is to model only the molten phase without considering the air venting system (the molten polymer fluid front evolves along the mold without any influence from the trapped air) ([Zhou, 2013](#)). The second most common method is to numerically consider that the *end-wall* of the mold is an outlet boundary condition (both polymer and air can exit the mold).

More recent boundary conditions were proposed in the literature, that can better model real-life air-vents. In the works of Hétu *et al.* (1998) and Chang and Yang (2001), no polymer melt is allowed to cross the mold walls, but the air is assumed to be free to leave the mold as the polymer melt advances (this is done for all solid walls). Another way to model the air-venting system is to consider a small patch with the size of a real air vent and to allow the material (air and molten polymer) to exit through that boundary. This later boundary conditions are used by proprietary software, and one cannot access their real computer implementation. More recent papers on air-vents for injection molding make use of such software (Özdemir and Pahlavani, 2018); therefore, the *black-box* boundary conditions do not allow a clear understanding of the process being modeled. In this work, we develop new ways to model the air vents and also provide the boundary conditions used. In our approach, we took the percentage of molten polymer which fills the cell touching the venting gap has the criteria to release the air from the mold cavity.

This work is organized as follows. In Section 2, the governing equations are presented and the numerical method is briefly described. Section 3 is devoted to the implementation of the new boundary conditions. Sections 4 and 5 present a detailed study on mesh refinement and location of the air vents boundary conditions. The work ends with the main conclusions.

2. Governing equations and numerical method

2.1 Governing equations

During the filling stage of the injection molding process the melt is assumed to be incompressible. The transient flow of non-Newtonian inelastic, incompressible fluids is then governed by continuity,

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

and the momentum conservation equations,

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} + \sigma k \nabla \alpha \quad (2)$$

where:

- \mathbf{u} = velocity vector;
- \mathbf{g} = gravity;
- p = the pressure;
- ρ = fluid density;
- σ = surface tension;
- k = mean curvature of the free surface; and
- $\boldsymbol{\tau}$ = Newtonian extra stress tensor.

which is given by:

$$\boldsymbol{\tau} = \eta \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) = 2\eta \mathbf{D} \quad (3)$$

where \mathbf{D} is the symmetric rate of strain tensor and the viscosity η is either a constant (for Newtonian fluids), or a function of the second invariant of the rate of the strain tensor, \mathbf{D} . For the latter, in this work we have considered the Carreau model, given by:

$$\eta(\dot{\gamma}) = \mu_{\infty} + (\mu_0 - \mu_{\infty}) \left[1 + (\lambda \dot{\gamma})^2 \right]^{(n-1)/2} \quad (4)$$

where μ_∞ , μ_0 , λ and n are constant parameters which are obtained from a fit to the rheological data obtained experimentally for the material being injected, and $\dot{\gamma} = \sqrt{2(\mathbf{D} : \mathbf{D})} = \sqrt{2\text{tr}(\mathbf{D}^2)}$.

In this work, two phases are considered, air and liquid (the polymer melt), and this is modeled using the VoF method. The VoF method is a numerical procedure used for tracking the location of the interfaces between fluids. It assigns a scalar α to all computational cells. Being $\alpha = 1$ in one of the phases and $\alpha = 0$ in the other phase, and it takes intermediate values for cells where the interface is located (Hirt and Nichols, 1981). Fluid properties such as density and viscosity are calculated as the weighted average of the different phases properties according to each volume fraction. Therefore, for a case with only two fluids, 1 and 2, where α is the volume fraction of 1, the density ρ and kinematic viscosity η are given by Ubbink (1997):

$$\rho = \alpha\rho_1 + (1 - \alpha)\rho_2, \quad (5)$$

$$\eta = \alpha\eta_1 + (1 - \alpha)\eta_2. \quad (6)$$

The method used in this work (based on the *OpenFOAM*[®] *interFoam* solver) decomposes the velocity vector field \mathbf{u} in the same way as density and viscosity are decomposed, and creates an auxiliary vector field \mathbf{u}_r (relative velocity) that is used in the calculation of the α field (Berberović *et al.*, 2009). The scalar components of velocity are then given by:

$$\mathbf{u} = \alpha\mathbf{u}_1 + (1 - \alpha)\mathbf{u}_2, \quad (7)$$

$$\mathbf{u}_r = \mathbf{u}_1 - \mathbf{u}_2. \quad (8)$$

The governing equation for α is:

$$\frac{\partial\alpha}{\partial t} + \nabla \cdot (\alpha\mathbf{u}) + \nabla \cdot [\alpha(1 - \alpha)\mathbf{u}_r] = 0. \quad (9)$$

This allows to use the one fluid approach to solve the continuity equation (1) and momentum balance equation (2). For more details, please refer to Ubbink (1997).

2.2 Numerical method

The numerical simulations were performed by solving the system of equations given by equations (1)-(4) together with equation (9). To do that, a straightforward finite volume discretization procedure was used and the PIMPLE (the PIMPLE algorithm is a combination of PISO (Pressure Implicit with Splitting of Operator) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations)) method was used to calculate the pressure and velocity fields. For the solution of the polymer melt volume fraction (the α equation) discretized system of equations, the smoothSolver with a symmetric GaussSeidel smoother was used. For the pressure correction equation the Preconditioned (DIC) Conjugate Gradient method (Hestenes and Stiefel, 1952) was used, and for the velocity, the smoothSolver with a symmetric GaussSeidel smoother was used once again.

To approximate the transient, advection, diffusion and source terms the following numerical schemes were considered: the Euler method (Atkinson, 1989) for the evolution in

time, the Gauss linear for gradients (Atkinson, 1989), the Gauss linearUpwind grad(U) (Courant *et al.*, 1952) was used for the convection of velocity, the advection terms are approximated by the Gauss vanLeer and Gauss interface compression methods, the Gauss linear corrected was considered for the Laplacian terms, and finally, for the remaining terms, a Gauss linear method was used.

Note that for the different simulations, different solvers and different numerical schemes can be used. This will be explained along the text. Additionally, the readers are referred to Pedro *et al.* (2020) for the numerical validation of the current algorithm.

3. Air vents boundary condition

As explained before, the aim of this work is the implementation of new boundary conditions that can simulate the real behavior of air vents. As shown in Figure 1, the air vents are used to allow the air escape from the mold. These pathways need to be thick enough to let air out easily, but thin enough to avoid the molten plastic to flow through it. Examples of a typically used depth for the vents are: ABS and Polycarbonate, $5.08 \mu\text{m}$; Polyethylene, Polystyrene and Polypropylene, $25.4 \mu\text{m}$; PVC flexible, $38.1 \mu\text{m}$. The vent width W can be anywhere from $1/3 \text{ cm}$ wide or more, and a common width is $2/3 \text{ cm}$. L should be a minimum of 0.076 cm and a maximum of 0.32 cm (Rosato and Rosato, 2012). Although Figure 1 is only showing one air vent, it is common to have at least 30% of the perimeter of the cavity with equally spaced air vents (Rosato and Rosato, 2012).

Basically, as illustrated in Figure 2, in this new boundary condition the following conditions are imposed: when the value α is higher than a certain threshold value ε ($0 \leq \varepsilon < 1$) in the cell touching the venting gap, the velocity of the material at that boundary is set to zero (it behaves as a wall) and a zero-gradient boundary condition is used for the pressure. If instead $\alpha \leq \varepsilon$, the pressure at that boundary is set to zero (atmospheric pressure) and a zero-gradient boundary condition is considered for the velocity, allowing the air to exit the mold (as it is done for an outlet boundary condition). This boundary condition was tested in the MSc thesis of Magalhães (2016),

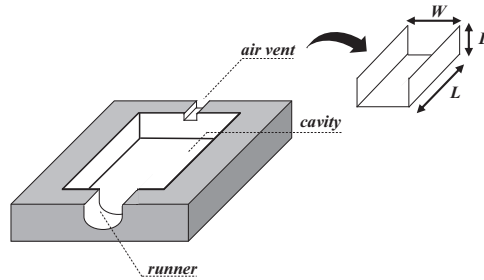


Figure 1.
Schematic geometry of a mold with air vents

```

valuePhaseFractionPressure.C
if (alphaCellp[facei] > referenceAlpha_)
{
    refGrad()[facei] = Zero;
    valueFraction()[facei] = Zero;
}
else
{
    refValue()[facei] = Zero;
    valueFraction()[facei] = 1.0;
}

valuePhaseFractionVelocity.C
if (alphaCellp[facei] > referenceAlpha_ || phip[facei] < 0.0)
{
    refValue()[facei] = vector::zero;
    valueFraction()[facei] = 1.0;
}
else
{
    refGrad()[facei] = vector::zero;
    valueFraction()[facei] = 0.0;
}

```

Figure 2.
Snippet of the boundary condition code

and some stability problems were reported because the air was allowed to enter the mold from the air vents. Therefore, in this work, we also add a new feature to the boundary condition, which do not allow the air to enter the mold through the air vent.

The boundary condition should respect the real dimensions of the air vents. For really small air vents, it was considered only one cell face as an air vent. The way to do that is to first generate the mesh and edit the mesh definition to change the boundary by adding directly an extra patch with only one cell (or more cells, depending on the mesh size). An example is provided in [Figure 3](#), where the methodology proposed to create a boundary condition per cell is illustrated. It should be remarked that the supplementary material provided comprises both the new boundary conditions (new libraries) and a case study.

4. Two-dimensional case study

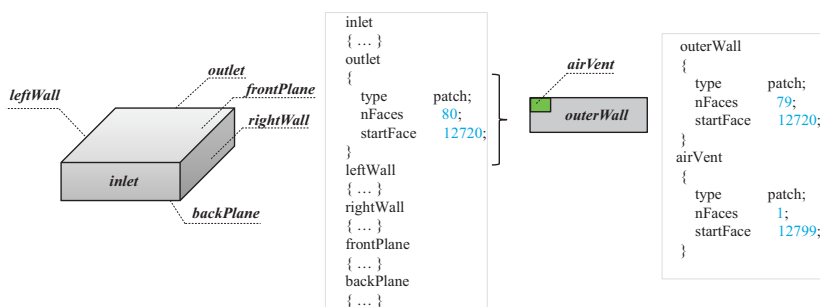
In the 2D case study, several tests are performed to both the original – *primary air vent* (the air can exit and enter the mold) and the improved air vents – *pure air vent* boundary conditions (the air can only exit the mold). The outlet works as an air vents boundary condition.

4.1 Primary and pure air vent boundary conditions: a mesh refinement study

To test the boundary conditions, a simple block with dimensions 50×50 mm and three different mesh refinements were considered, 20×20 , 40×40 and 80×80 cells. The case of a fixed Courant and a fixed time-step was also taken into account.

Newtonian model parameters: kinematic viscosity of the polymer melt $\nu_{polymer} = 0.2 \text{ m}^2/\text{s}$, kinematic viscosity of the air $\nu_{air} = 0.0812 \text{ m}^2/\text{s}$, $\rho_{polymer} = 1000 \text{ kg/m}^3$, $\rho_{air} = 1.23 \text{ kg/m}^3$, inlet velocity = 6.33 mm/s . Similar parameters were used by [Chang and Yang \(2001\)](#).

[Figure 4](#) shows the evolution of the polymer melt front in the simple channel described before [[Figure 4\(f\)](#)]. A fixed Courant number is assumed. The results obtained with the *primary air vents* present some instabilities [[Figure 4\(a\)-\(c\)](#)], with unstable air flux, entering and leaving the mold, as evidenced by the velocity vectors. As expected, the quality of the results improve with the decrease of the Courant number. Note that for the mesh 80×80 and $Co = 0.1$, the results obtained are not physically sound, and they could be improved by simply considering a $Co = 0.01$. As the mesh refinement increases the results become more stable. The results obtained with the new boundary condition *pure air vent* are stable, even



Notes: In this case, the outlet patch was divided into two new patches, the outer Wall and the air Vent

Figure 3.
Example of how to transform the boundary patch to obtain an independent patch for the air vent

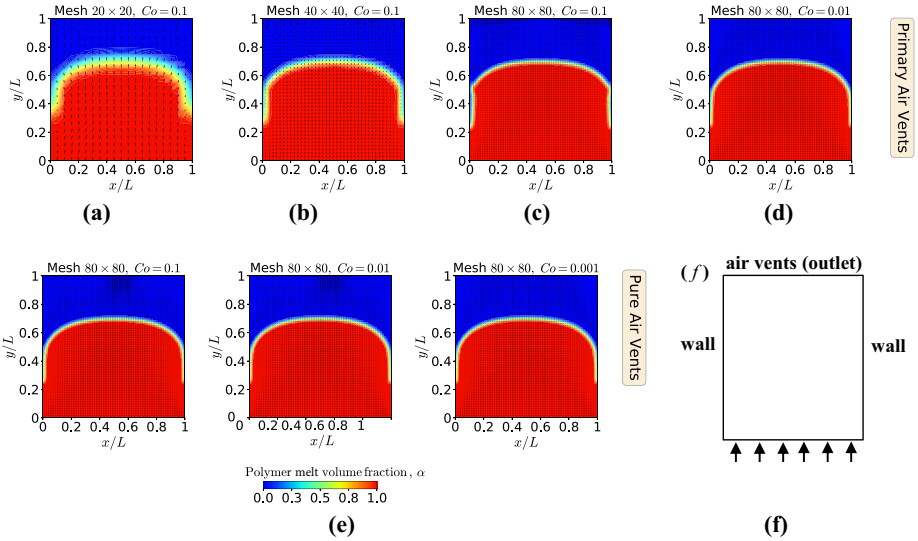


Figure 4. Two-phase flow in a 2D square channel of dimensions 50×50 mm, distribution of polymer melt volume fraction and cell velocity vectors

Notes: (a)-(d) Results obtained for the primary air vents with different mesh refinements; (e) Results obtained for the pure air vent boundary condition with mesh 80×80 ; (f) Schematic of the geometry and boundary conditions

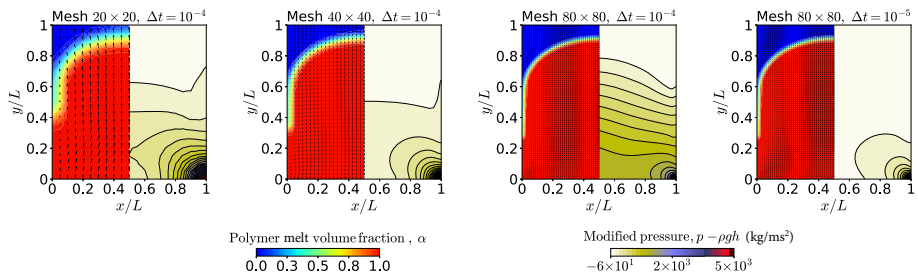
for a $Co = 0.1$, as shown in [Figure 4\(e\)](#). These results were obtained for a mesh 80×80 , but a stable flow was also achieved for the coarser meshes.

[Figure 5](#) shows the polymer melt volume fraction and pressure fields for two different values of time-step (Δt) using the *pure air vent* boundary condition. On the left: the results obtained for three different meshes and $\Delta t = 10^{-4}$ s. Note that the α evolution is smooth with no stability problems. Regarding the pressure field, it is not smooth near the right wall with coarser meshes or with high time-steps.

It should be remarked that in some works found in the literature, to improve the stability of the method, the viscosity of air is artificially increased, as happens for example in [Chang and Yang \(2001\)](#). In this work, stable results were obtained with the real material parameters.

The solver was also tested by considering generalized Newtonian fluids modeled by the Bird-Carreau model given in [equation \(4\)](#). A fit to the shear viscosity rheological data of a polystyrene melt ([Munstedt, 2011](#)) was performed leading to the following parameters:

Figure 5. Distribution of polymer melt volume fraction and cell velocity vectors, and pressure fields using two different values of Δt (in seconds) for the *pure air vents* boundary condition



$v_{0polymer} = 53 \text{ m}^2/\text{s}$, $v_{\infty polymer} = 0 \text{ m}^2/\text{s}$, $\lambda = 1.1 \text{ s}$, $n = 0.15$ and $\rho_{polymer} = 1000 \text{ kg/m}^3$. The simulations were stable, similar to what happened for the Newtonian fluids. Note that this fluid is shear thinning and, therefore, a lower viscosity is expected in regions of higher shear rates. This fact should improve the numerical stability, because the difference between the air and polymer melt viscosities becomes smaller leading to a smoother transition. The results obtained for three different Courant numbers are shown in Figure 6 (the results are stable for all the Courant numbers considered). It should be remarked that Magalhães (2016) report oscillations of the interface near the wall, and, the use of the new boundary condition allowed to suppress them.

4.2 Location of the air vents

Until now, the location of the air vents boundary conditions is restricted to the outlet region. In this case study, the air vents boundary condition was also considered at the side walls. Note that the cases where the air vents are restricted to specific regions (or even specific cells) will be studied in Section 5.

Figure 7 shows the mold filling of a Newtonian and a Bird-Carreau fluid using three different Courant numbers. Note that the existence of non-null velocity vectors near the side walls due to the exit of the air. This boundary condition allows improvements in the results because a higher percentage of filling of the mold was obtained and also the instantaneous removal of air reduces the oscillations at the interface (especially for the Bird–Carreau fluid).

4.3 Influence of α threshold value (ε) on the mass conservation

As explained before, the new boundary condition assume that when the polymer melt volume fraction (α) is higher than a certain value ε ($0 \leq \varepsilon < 1$) in the cells that contain the venting face, the velocity of the material at that boundary is set to zero (it behaves as a wall), and a zero-gradient boundary condition is used for the pressure. If instead $\alpha \leq \varepsilon$, the pressure at that boundary is set to zero and a zero-gradient boundary condition is considered for the velocity, allowing the air to escape the mold (similar to an outlet boundary condition). The influence of ε on the mass conservation: as expected, without a sharp interface, a small portion of fluid may be *lost*, passing through the air vents.

Figure 8 shows the mold filling with a Newtonian fluid by considering $\varepsilon = 0.0001, 0.01, 0.5$ and 0.99 . In the first case, $\varepsilon = 0.0001$, if the cell adjacent to the air vents is filled with 0.00001% of polymer melt, the air vent closes and both the polymer melt and air become trapped. The use of a coarser mesh would lead to a partially filled mold cavity. As ε

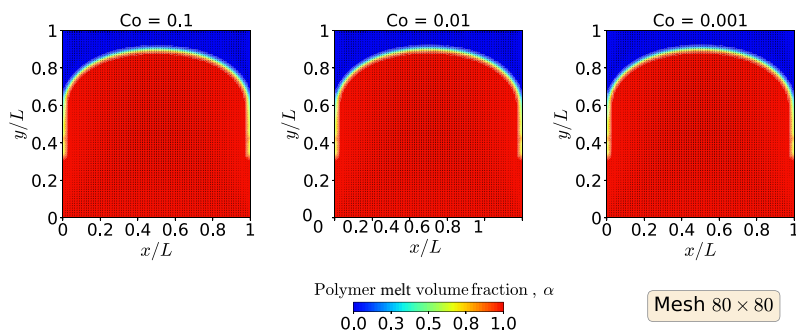


Figure 6.
Distribution of polymer melt volume fraction and cell velocity vectors for a Carreau fluid using three different Courant numbers (mesh 80×80)

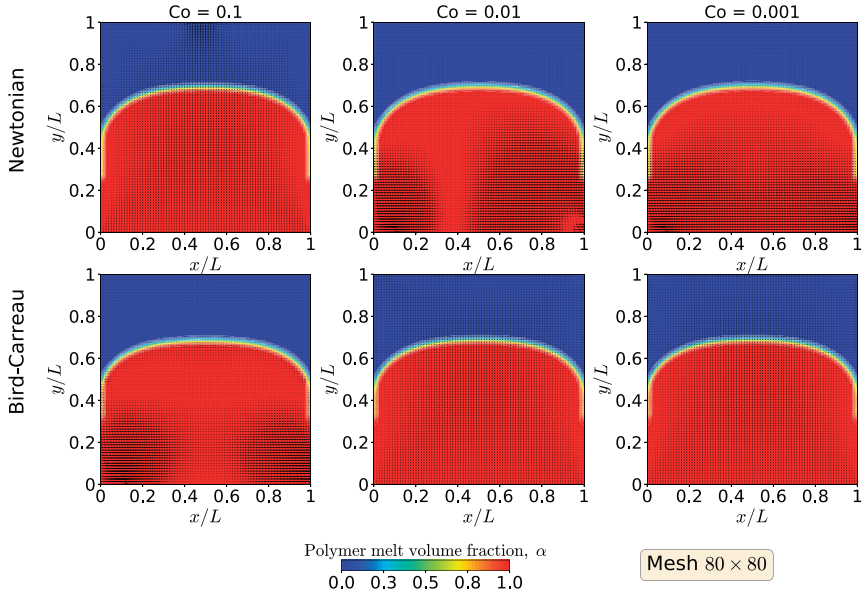


Figure 7. Distribution of polymer melt volume fraction and cell velocity vectors for a Newtonian and a Bird-Carreau fluid using three different Courant numbers (mesh 80×80)

Notes: The air vents are located at the side walls and also at the outlet

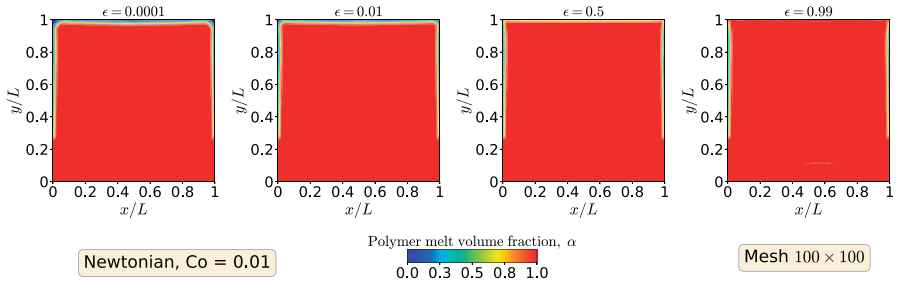


Figure 8. Distribution of polymer melt volume fraction with a Newtonian fluid for four different values of ϵ (0.0001, 0.01, 0.5 and 0.99)

Notes: The mesh used is 100×100 . The air vents are located at the outlet

increases, the mold cavity becomes totally filled, but there is loss of polymer melt (a quantification of that loss will be provided in the next section for the 3D cases).

5. Three-dimensional case study

This 3D case study is performed on a simple geometry (hexahedron) with the dimensions of the tip of a tensile test specimen, as show in Figure 9. This type of specimen is often used to characterize the mechanical behavior of different polymeric materials. The fluid considered in this work was obtained from a typical polymer used

in the injection molding industry, and the parameters for the Bird–Carreau model are $v_{0,polymer} = 5.51 \text{ m}^2/\text{s}$, $v_{\infty,polymer} = 0 \text{ m}^2/\text{s}$, $\lambda = 0.49 \text{ s}$, $n = 0.38$ and $\rho_{polymer} = 762.3 \text{ kg}/\text{m}^3$. The polymer melt inlet velocity was defined equal to $6.33 \text{ mm}/\text{s}$.

5.1 Mesh refinement study

A mesh refinement study was performed considering three different meshes, $22 \times 45 \times 4$, $44 \times 90 \times 8$, $88 \times 180 \times 16$ and four different Co numbers. To avoid the cluttering of figures only the results obtained for $Co = 0.01$ are shown. Also, the results obtained were similar for other Co numbers used. The only case that presented some oscillations in the pressure field was for the less refined mesh and $Co = 0.1$.

Figure 10 shows the polymer melt volume fraction and pressure fields distributions along the channel. Note that as shown in Figure 9, two air vents are present near the corners. The reason for choosing the corners is because the probability of having trapped air in this region is higher, which is a criterium used in practice when designing molds. With a third dimension the interface is more difficult to capture, and, it is more prone to instabilities. However, both fields are stable, and no unphysical oscillations are observed. Note that for the more refined meshes, small oscillations at the interface near the wall due to trapped air, are observed. This is also expected in practice, a phenomenon usually designated by fountain flow (Rosato and Rosato, 2012). Regarding the pressure field, it can be seen that a smooth distribution is obtained and mesh refinement leads to a slightly different pressure distribution along the mold. It should be remarked that some air is trapped at the center region of the outlet, as the air vents are located at the corners.

In Figure 11, the polymer melt volume fraction in percentage and integrated pressure along time are shown (note that compression is not considered in this work). As expected an almost linear filling of the mold is obtained due to the simplicity of the geometry and due to the fact that the flow is incompressible. It is not possible to obtain a filling of 100%, only 99%. The differences between the two meshes are small, but, for the most refined mesh, the integrated pressure field shows an exponential and smooth increase when the air vents are closing. This is due to the small size of the air vents and to the fact that for a less refined mesh the mold entraps air close to the wall, allowing a higher number of iterations without diverging.

5.2 Adaptive mesh refinement vs static meshes

Due to the excessive execution time of the simulations, the ability to use dynamic meshes was tested. These meshes allow a local adaptive mesh refinement in regions of high gradients of α . Two cases were considered:

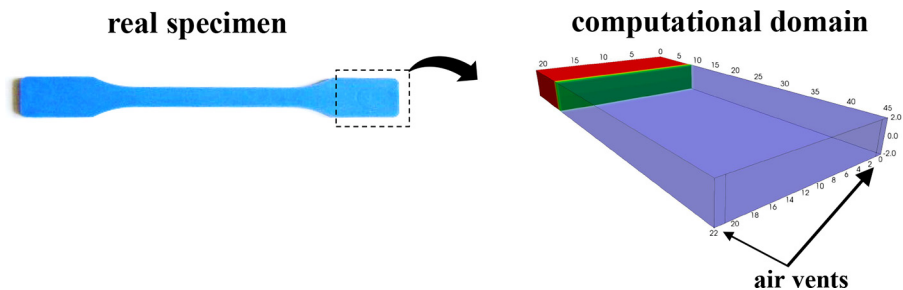


Figure 9.
Geometry used in the filling of a 3D rectangular hexaedron (dimensions: $22 \times 45 \times 4 \text{ mm}$)

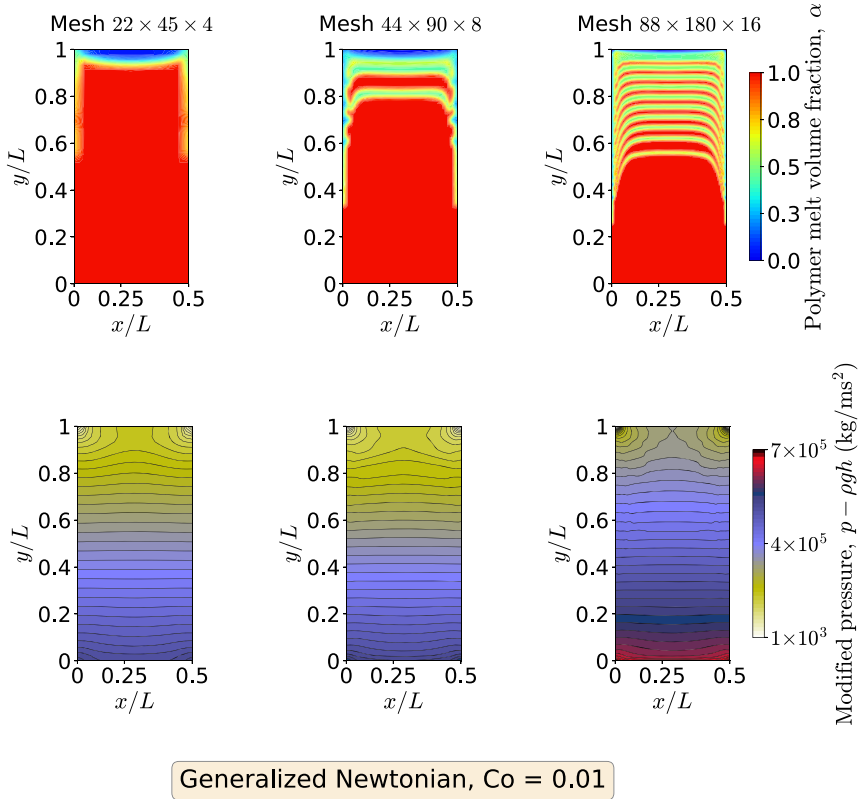


Figure 10. Polymer melt volume fraction and pressure distributions obtained for $Co = 0.01$ and three different mesh refinements: $22 \times 45 \times 4$, $44 \times 90 \times 8$, $88 \times 180 \times 16$

- (1) Compare the results obtained with mesh $44 \times 90 \times 16$, and the results obtained with mesh $22 \times 45 \times 4$ (with one level of local adaptive mesh refinement); and
- (2) Compare the results obtained with mesh $88 \times 180 \times 16$, and the results obtained with mesh $22 \times 45 \times 4$ (two levels of local adaptive mesh refinement).

These results are shown in [Figure 12](#).

The results obtained for case (1) are similar, with the only difference being the representation of the interface near the exit. Since the objective is to obtain good qualitative results, the use of dynamic meshes proved to be a valuable option to achieve it in a shorter period of time, or, for complex geometries. For case (2) larger differences were expected, but, as shown in the top and lateral views (with and without a translucent α), the contours are practically the same.

6. The filling of a tensile test specimen mold

6.1 Geometry and mesh

The previous test on mesh refinement and location of air vents allowed us to understand the behavior of these new *pure air vent* boundary condition, and therefore the next logical step is

to use this boundary condition in more complex geometries. The tensile test specimen was presented in Figure 9, and the respective mold cavity is shown in Figure 13. The specimen mold cavity is formed by a sprue where the material is injected, the runner, the gate and the part (the specimen itself). The volume (V) of the 3D tensile test specimen is of 20 cm^3 .

6.2 Results and discussion

To have an idea of the simulation times and flow behavior, the simulations were performed by first considering the PIMPLE algorithm and a really high value of the maximum Courant number ($Co = 25$). As expected, the results obtained showed unphysical oscillations (especially for the pressure field). The maximum cell size is 0.00045 mm for the sprue, runner and specimen, and the maximum cell size is 0.000015 mm for the air vents and gate.

With an inlet velocity (U_{in}) of 5 cm/s , it takes approximately 32 s ($V/U_{in} \cdot \pi \cdot (\varphi II)^2$) to fill the mold. Notice that the inlet velocity used was chosen only for academic purposes and is not compared to the one verified in real-life. The aim is to assess the implementation of the

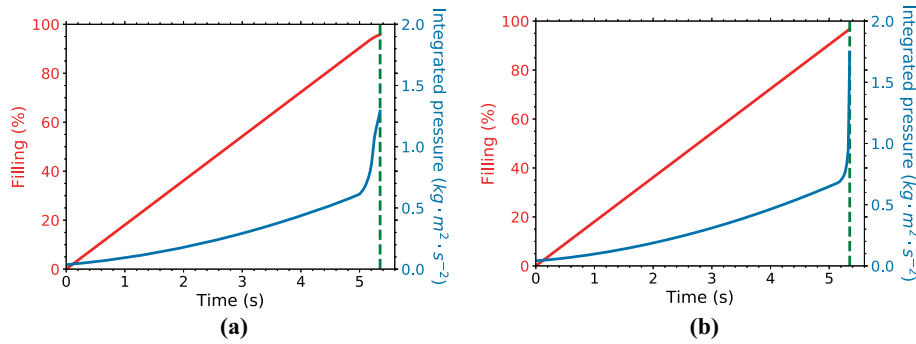
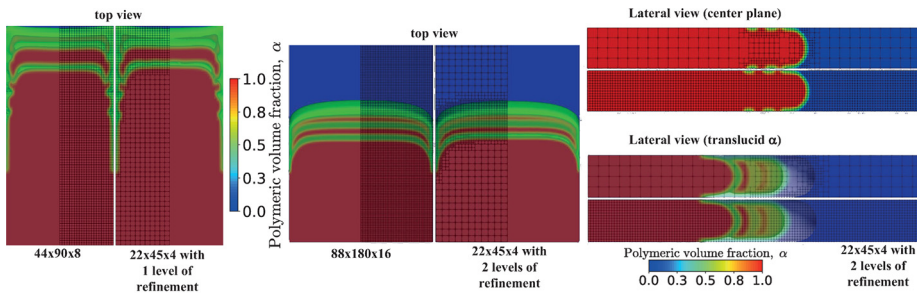


Figure 11. Filling in % and integrated pressure ($\sum_i p_i \cdot \alpha_i \cdot V_i$) for different mesh refinements

Notes: (a) $22 \times 45 \times 4$; (b) $88 \times 180 \times 16$. Note that i represents a computational cell and V_i is the respective volume



Notes: (left) Comparison of the results obtained with mesh $44 \times 90 \times 8$ and the results obtained with mesh $22 \times 45 \times 4$ (1 level of local adaptive mesh refinement); (middle and right) Comparison of the results obtained with mesh $88 \times 180 \times 16$ and the results obtained with mesh $22 \times 45 \times 4$ (2 levels of local adaptive mesh refinement)

Figure 12. Dynamic meshes

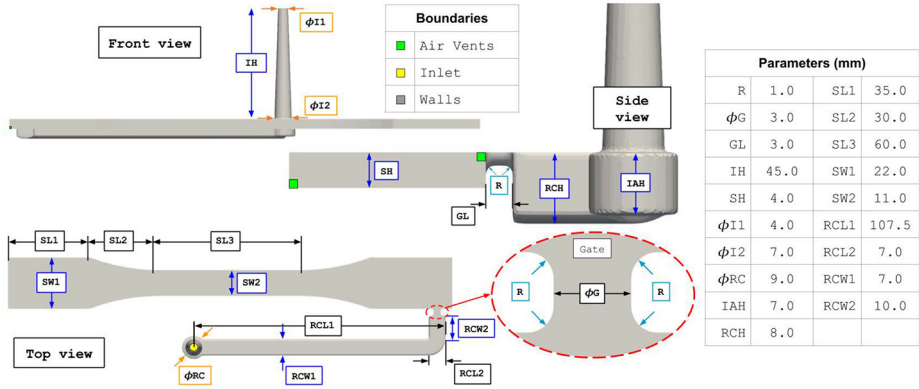


Figure 13. Geometry of the specimen mold cavity (dimensions in mm)

new boundary condition in a 3D case study. The integrated pressure for the case $Co = 25$ is shown in Figure 14(a). It can be seen that in the interval 24.3 to 27.0 s really high oscillations in pressure are obtained. In this interval the fluid is going through the specimen expansion region near the end. Oscillations are present in the integrated pressure in the convergent region of the specimen. To smooth this oscillations it was considered the PISO algorithm with maximum Courant number of 0.8 and 0.1. The results are shown in Figure 14(b). It can be seen that for the case $Co = 0.8$ only small perturbations are obtained, but these oscillations are eliminated with $Co = 0.1$ [see inset in Figure 14(b)].

Note that for the three cases studied the evolution of the interface was practically the same and the amount of trapped air was also similar. Only the quantitative integrated pressure results were different. It should be remarked that each simulation was performed in parallel using four cores, and, it took 2 and 10 days to simulate the cases $Co = 25$ and 0.8, respectively, and it took 10 days to simulate the interval 24.3 to 27.0 s for $Co = 0.1$.

Figures 15 and 16 show the results of the mold filling for the case $Co = 0.8$ and using different perspectives (the videos are given as supplementary material). Figure 15 shows the existence of entrapped air near the corners. This air is partially dragged along the mold filling. Near the gate (Figure 16), the code captures well the evolution of the interface in a region where high velocities are obtained. These tests were performed for different meshes,

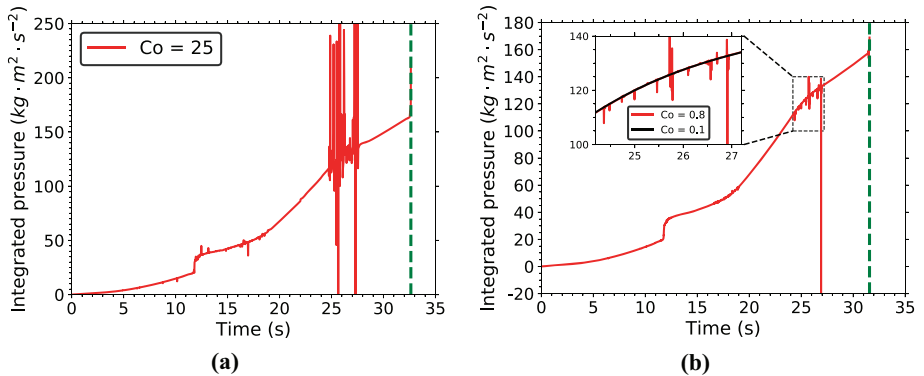


Figure 14. Integrated pressure versus time for $Co = 25, 0.8, 0.1$

and the final conclusion was a maximum Courant number of 0.8 and 0.1 could provide accurate results for the evolution of pressure along time (considering a mesh of 251666 cells with 97% of quadrilateral cells refined near the regions of expected high shear rates – the region involving the gate and near the air vents).

Finally, it can be concluded that the new boundary conditions are stable and allow the removal of air from mold, simulating real air vents.

7. Conclusions and future work

New boundary conditions to simulate the air vents used in the injection molding process were proposed. The boundary conditions were implemented in the *OpenFOAM*® computational library (solver *interFoam*). The numerical code was tested using 2D and 3D case studies with static and dynamic local adaptive mesh refinement. The results obtained showed that the numerical method is stable and allows one to model the filling process, simulating the real injection process. The new implementation allowed to suppress oscillations in both the velocity and pressure fields. These oscillations were present in our past implementation of the boundary conditions, where air was allowed to enter the mold.

This new modeling tool allows to study in more detail the injection molding process, and, consequently, the improvement of the experimental procedure, with the support of the detailed results obtained numerically. With the numerical simulations engineers and technicians will be able to follow the evolution of the polymer melt front inside the mold and,

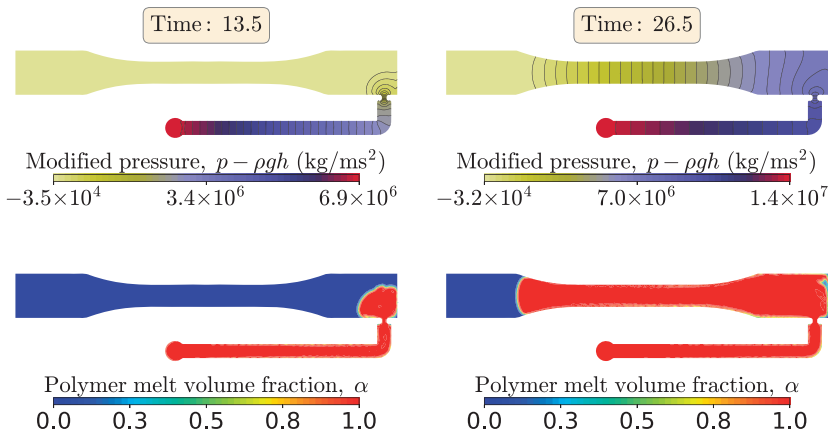


Figure 15. Polymer melt volume fraction and pressure distributions for $Co = 0.8$ and $\epsilon = 0.5$ (time in seconds)

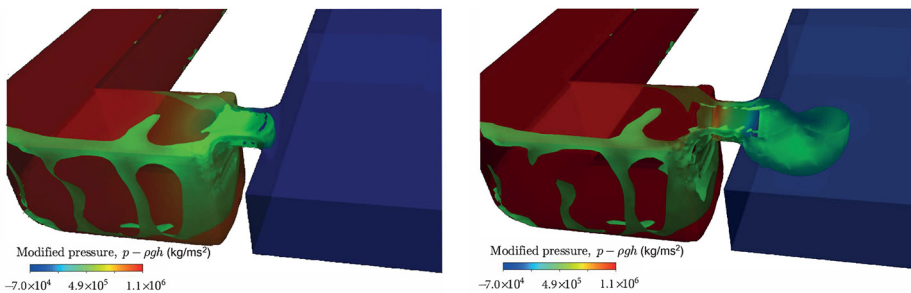


Figure 16. Polymer melt volume fraction (contour of $\alpha = 0.5$ in all plots represented with green color) and pressure distributions at gate location for $Co = 0.8$ and $\epsilon = 0.5$

therefore, anticipate and correct potential problems in the final product, such as the presence of weld lines.

In the future, the research group will perform the implementation of these new boundary conditions in solvers that can deal with more complex and realistic constitutive equations for polymer melts, which take into account viscoelasticity.

References

- Atkinson, K.A. (1989), *An Introduction to Numerical Analysis*, 2nd ed., John Wiley and Sons, New York, NY ISBN 978-0-471-50023-0.
- Autodesk (2020), “Moldflow”, available at: www.autodesk.com/products/moldflow/overview (accessed 12 June 2020).
- Azaman, M.D., Sapuan, S.M., Sulaiman, S., Zainudin, E.S. and Khalina, A. (2013), “Numerical simulation analysis of the in-cavity residual stress distribution of lignocellulosic (wood) polymer composites used in shallow thin-walled parts formed by the injection moulding process”, *Materials and Design*, Vol. 55, pp. 381-386.
- Berberović, E., van Hinsberg, N.P., Jakirlić, S., Roisman, I. and Tropea, C. (2009), “Drop impact onto a liquid layer of finite thickness: dynamics of the cavity evolution”, *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, Vol. 79, p. 036306.
- Chang, R.-Y. and Yang, W.-H. (2001), “Numerical simulation of mold filling in injection molding using a three-dimensional finite volume approach”, *International Journal for Numerical Methods in Fluids*, Vol. 37 No. 2, pp. 125-148.
- Courant, R., Isaacson, E. and Rees, M. (1952), “On the solution of nonlinear hyperbolic differential equations by finite differences”, *Communications on Pure and Applied Mathematics*, Vol. 5 No. 3, pp. 243-255.
- Hestenes, M.R. and Stiefel, E. (1952), “Methods of conjugate gradients for solving linear systems”, *Journal of Research of the National Bureau of Standards*, Vol. 49 No. 6.
- Hétu, J.F., Gao, D.M., Garcia-Rejon, A. and Salloum, G. (1998), “3D finite element method for the simulation of the filling stage in injection molding”, *Polymer Engineering and Science*, Vol. 38 No. 2, pp. 223-236.
- Hirt, C.W. and Nichols, B.D. (1981), “Volume of fluid (VOF) method for the dynamics of free boundaries”, *Journal of Computational Physics*, Vol. 39 No. 1, pp. 201-225.
- Kennedy, P. (2009), “Development of injection molding simulation”, *Injection Molding Technology and Fundamentals*, Hanser Publishers, Munich, pp. 553-558.
- Kim, N.H. and Isayev, A.I. (2015), “Birefringence and interface in sequential co-injection molding of amorphous polymers: simulation and experiment”, *Polymer Engineering and Science*, Vol. 55 No. 1, pp. 88-106.
- La, M., Lee, J.G. and Park, S.J. (2018), “Numerical and experimental investigation of plastic injection molding of micro-engineered surfaces”, *Polymer Engineering and Science*, Vol. 58, pp. E73-E81.
- Magalhães, A. (2016), “OpenFOAM simulation of the injection moulding filling stage”, Master Thesis, University of Minho, Portugal.
- Mollaabbasi, R., Behzadfar, E. and Taghavi, S.M. (2019), “A simplified semi-analytical model for the filling and cooling process in plastic molding”, *Physics of Fluids*, Vol. 31 No. 6, p. 063105.
- Munstedt, H. (2011), “Rheological properties and molecular structure of polymer melts”, *Soft Matter*, Vol. 7, pp. 2273-2283.
- OpenFOAM (2004), “The open source CFD toolbox”, available at: www.openfoam.com/
- Özdemir, I.B. and Pahlavani, H. (2018), “Effects of air vents on the flow of reacting polyurethane foam in a refrigerator cavity”, *Advances in Polymer Technology*, Vol. 37 No. 7, pp. 2420-2428.

-
- Pedro, J., Ramôa, B., Nóbrega, J.M. and Fernandes, C. (2020), "Verification and validation of openInjMoldSim, an open-Source solver to model the filling stage of thermoplastic injection molding", *Fluids*, Vol. 5 No. 2, p. 84.
- Pichelin, E. and Coupez, T. (1998), "Finite element solution of the 3D mold filling problem for viscous incompressible fluid", *Computer Methods in Applied Mechanics and Engineering*, Vol. 163 Nos 1.4, pp. 359-371.
- Rosato, D.V. and Rosato, M.G. (2012), *Injection Molding Handbook*, Springer Science and Business Media New York, NY.
- Ubbink, O. (1997), "Numerical prediction of two fluid systems with sharp interfaces", PhD thesis, Imperial College, University of London, UK.
- Wang, W., Li, X. and Han, X. (2012), "Numerical simulation and experimental verification of the filling stage in injection molding", *Polymer Engineering and Science*, Vol. 52 No. 1, pp. 42-51.
- Xu, X. and Yu, P. (2017), "Modeling and simulation of injection molding process of polymer melt by a robust SPH method", *Applied Mathematical Modelling*, Vol. 48, pp. 384-409.
- Xu, X. and Yu, P. (2019), "Extension of SPH to simulate non-isothermal free surface flows during the injection molding process", *Applied Mathematical Modelling*, Vol. 73, pp. 715-731.
- Yan, B., Zhou, H. and Li, D. (2007), "Numerical simulation of the filling stage for plastic injection moulding based on the Petrov-Galerkin methods", *Journal of Engineering Manufacture*, Vol. 221 No. 10, pp. 1573-1577.
- Zhou, H. (2013), *Computer Modeling for Injection Molding: simulation, Optimization, and Control*, John Wiley and Sons, Beijing.

Corresponding author

Célio Fernandes can be contacted at: cbpf@dep.uminho.pt

For instructions on how to order reprints of this article, please visit our website:

www.emeraldgrouppublishing.com/licensing/reprints.htm

Or contact us for further details: permissions@emeraldinsight.com