

Lagrangian finite element model for the 3D simulation of glass forming processes

P. B. Ryzhakov^{a,b}, J. García^{a,b}, E. Oñate^{a,b}

^a*Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE)
Gran Capitán s/n, 08034 Barcelona, Spain*

^b*Universitat Politècnica de Catalunya (UPC), Barcelona, Spain*

Abstract

We propose here a numerical model for a three-dimensional simulation of glass forming processes. Using the basic philosophy of the Particle Finite Element method (PFEM), we introduce several new features adapting the strategy to suit the problem of interest. A modified fractional step method for the solution of the flow equations is applied. This approach, on the one hand, inherits the computational efficiency of the original fractional step approach, and on the other hand shows better mass conservation features. These features are particularly attractive taking into account the importance of the correct prediction of the glass product's wall thickness. A smart mesh update strategy and a simple mechanical contact scheme are introduced. In order to account for temperature-dependent viscosity, the heat equation is coupled to the mechanical model. Viscosity is obtained from the temperature field via an empirical law. The model is validated and an example modeling the processes in the final blow mold of the bottle manufacturing process is proposed.

Keywords: bottle manufacturing, numerical simulation, benchmark, PFEM, counter blow, thermo-mechanical

1. Introduction

In spite of its very long history, bottle manufacturing remains a challenging process and requires further improvements. The continuously growing competition calls for the optimization of the existing processes, diminishing the risks of producing deficient bottles. Thus, such optimization must be based upon a detailed knowledge of the process variables (such as the final topology, wall thickness, stress and temperature distribution) and their dependence upon the input parameters (inlet pressure, cooling conditions, etc). The typical questions that need to be answered are: How can the container weight be reduced without compromising on its strength? What are the optimal operation conditions

Email address: pryzhakov@cimne.upc.edu (P.B. Ryzhakov)

(air pressure, mould temperature) and duration of the different forming stages? Up-to-date the answers to these and similar questions are predominantly based upon the experience and craftsmanship rather than scientific knowledge.

Numerical modeling and simulation can serve as an efficient tool for answering many questions arising when facing unexpected effects in the real products. Apart from being considerably cheaper than conducting expensive and time-consuming trial-and-error procedures common to factories, only numerical simulation can provide such (otherwise impossible or difficult to obtain) results as: stress distributions within the solidifying melt and temperature distribution.

Up-to-date, there exist several computational tools used by industries for bottle manufacturing simulation. Usually these software model the glass manufacturing process using axis-symmetric formulations. This approximation greatly reduces the associated computational costs. However, it over-simplifies the process: even though many bottle molds are purely axis-symmetric, nearly all containers produced do have non-symmetrical thickness distributions. Additionally, axisymmetric formulations cannot be applied to modeling bottles with non-circular cross-sections, such as e.g. fragrance containers. Thus, 3D simulations appear to be obligatory for obtaining reliable predictions. However, 3D simulations are typically characterized by excessive computational times.

Generally, two classes of methods can be applied to the problem at hand: the fixed mesh (Eulerian) and the mesh-moving (Lagrangian or Arbitrary Lagrangian Eulerian (ALE)) ones. Eulerian formulations require excessively fine meshes for the correct representation of the domain evolution and typically introduce errors in mass conservation (whenever the Level Set method is used for representing the glass-air interface evolution). On the other hand, Lagrangian approaches lead to strongly non-linear systems of equations and large mesh deformations. Thus, a robust and computationally efficient 3D model still presents a major challenge.

In the present work a 3D viscous incompressible fluid formulation using an updated Lagrangian framework is proposed, where the current configuration is the reference one. It adopts the basic features of the Particle Finite Element Method (PFEM) [1]. The key idea of the PFEM is that the variables of interest are stored at the nodes instead of the Gauss points. This results in a hybrid between a standard FE and a mesh-free method. A finite element mesh is created at every time step of the transient problem and the solution is then stored at the nodes. At every time step the governing equations are solved in the standard Finite Element (FE) fashion. The discrete operators are updated at every non-linear iteration step according to the newly obtained domain configuration, ensuring excellent convergence of the iterative procedure. The nodes obtain their new positions and the mesh is re-generated using an unconstrained Delaunay technique. The approach is adapted to the problem at hand introducing a simple but efficient contact algorithm, boundary tracking and a re-meshing strategy. In terms of the method for solving the governing equations, we use a modified fractional step approach, combining the classical technique with a quasi-incompressible prediction [2], [3]. The approach on the one hand allows for a highly computationally efficient solution and, on the other hand, leads to

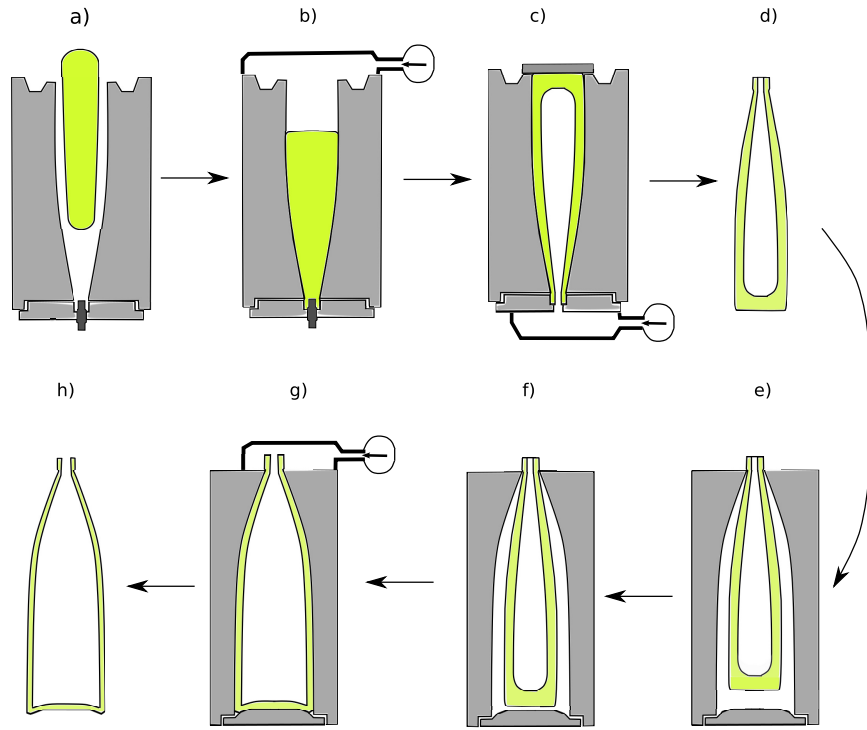


Figure 1: Different stages of bottle manufacturing process

an accurate mass conservation, which is essential for the problems of interest.

The paper concludes with two numerical examples. The first one is used for the validation of the model. The second one shows the potential of the method. Moreover, it can be used as a reference for the comparison and validation of the future models for the bottle forming simulation. Even though several glass forming simulation results are published in literature ([4], [5], [6], [7]), there exist no well-established benchmark up-to-date. The example we propose focuses on the modeling of the *final blow* stage of a glass manufacturing process. The material and geometrical data as well as all the boundary conditions necessary for reproducing the example are specified.

2. Glass forming

2.1. Industrial process

Prior to presenting the model for the glass (in particular, bottle) forming, let us review the industrial process and introduce the corresponding terminology. The standard bottle manufacturing process is sketched on Fig. 1.

Typically, high speed machines are fed a stream of molten glass¹ that is cut with a shearing blade into “gobs” of predetermined weight. These gobs fall into the first blank mold as shown in Fig. 1 a), where the temperature drops to the so-called “working temperature range” (some 1150°C for soda-lime glasses). At the base of the mold a cylindrical plunger for shaping the bottle neck is inserted. Air pressure or a plunger is applied in order to push the gob to the bottom of the mold (Fig. 1 b)). Afterwards, air compressed to $\approx 0,15$ MPa is blown from the bottom of the mold forcing the gob to rise and take the shape of the mold (Fig. 1 c)). This stage is known as a *counter-blow* process, suggesting that the “blow” is performed against gravity. The intermediate product of the counter-blow is known as “parison”.

Afterwards, the parison is removed from the first mold, turned upside down (Fig. 1 d)) and transferred into the second mold, where it is hung in order to stretch due to gravity (Fig. 1 e)), usually until the contact with the mold bottom is established. Finally, air pressure (slightly lower than the one used in the counter blow mold) is applied leading to the final shape of the bottle (Fig. 1 f)). This stage is known as *final blow* process. The bottle is then removed from the mold and is transferred to the annealing oven where it is reheated to remove the residual stresses produced during forming and finally it is cooled to the ambient temperature. The forming process, from the time when the gob is dropped into the first mold until the final product is removed from the second mold takes around 6 seconds.

2.2. Material properties

Glass is a visco-elastic material. At low temperatures (roughly, below 400 °C) elastic effects dominate, while above 550 °C elastic effects are negligible. One can also consider the transition zone where both effects are important (see (Fig. 2(a))). In the following we shall consider the temperature dependent properties of a typical soda-lime glass.

Mechanical properties: viscosity and density. The viscosity is the most important property in the glass forming process. For example, in a typical temperature range encountered in glass forming processes (between approximately 700 and 1200°C) glass viscosity varies from $140 \cdot 10^6$ Pa s at 700°C to some 400 Pa s. The dependence of glass viscosity upon temperature is typically given by Fulcher expression [8]:

$$\log_{10}\mu = A + \frac{B}{T - T_0} \quad (1)$$

where T_0 , A and B are constants from experiments. In the present work, the following parameter values are considered (except for example 2): $T_0=220^\circ\text{C}$, $B=4700$, $A=-2.8$. The glass viscosity as a function of temperature is shown in a logarithmic scale on Fig. 2(b). Due to very large variations of viscosity during

¹The most prevalent glass used for glass containers is soda-lime $Na_2O - CaO - SiO_2$ glass.

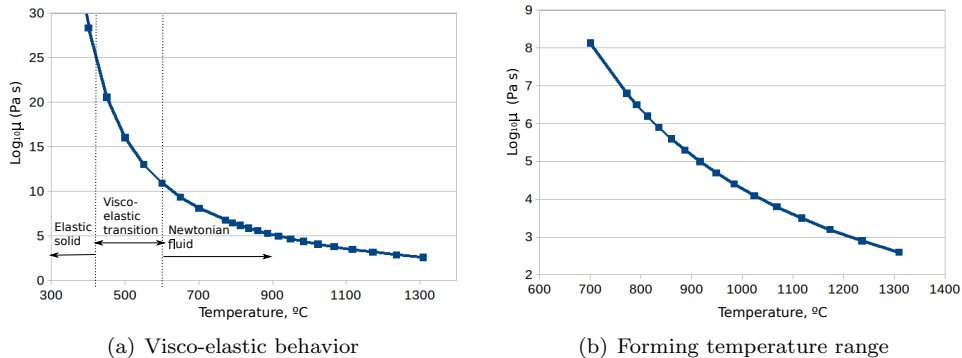


Figure 2: Viscosity-temperature curve of soda-lime glass used for beverage containers [8].

the forming process it is mandatory to include thermo-mechanical coupling in the model in order to obtain realistic predictions.

The temperature distribution in the glass is non-uniform at any stage of the glass manufacturing process. Thus, the viscosity is also non-homogeneous both in space and time. In the context of numerical modeling, Lagrangian formulations are very advantageous when dealing with non-constant properties: the property (for example, viscosity) is automatically transported being “attached” to the moving nodes. In the Eulerian framework, the representation of each non-constant property evolution would require solving the corresponding transport equation.

Glass density does not undergo considerable changes (2438 kg/m³ at 700°C and 2367 kg/m³ at 1100 °C), thus constant density is an acceptable approximation.

Thermal properties. Heat transfer in the glass is governed not exclusively by conduction, but also by the radiation, which may even be predominant. For strongly absorbing semi-transparent materials this radiation can be modeled as a diffusion process, thus an effective conductivity taking into account both processes is often defined [4]. Real radiation models are complex and computationally expensive. In the scope of this work they are not discussed.

Variation of the specific heat in the temperature range of interest is negligible (1400-1420 J/kg*K between 700 and 1200°C). The value of the diffusivity D changes from 0.0000015 m²/s at 700°C to 0.0000065 m²/s at 1100 °C.

3. Numerical model

The numerical model for the complex phenomenon of glass forming developed here consists of a mechanical and thermal modules. These models are coupled in order to take into account the temperature-dependent material properties’ evolution.

Kinematic framework. Lagrangian description for modeling the glass forming processes appears attractive as it allows to track the evolution of the deforming domain naturally. The position of the evolving domain coincides with the position of the mesh nodes, defined by the solution of the flow problem. An accurate interface capturing using Eulerian approaches would require much higher mesh resolution to achieve similar precision. The additional cost associated to the Lagrangian description is due to the necessity of re-meshing the computational domain in order to avoid excessive element distortion.

The Particle Finite Element Method (PFEM) applied in the present work for modeling the glass forming process is based on the updated Lagrangian description of the governing equations. The key idea of the PFEM is that the variables of interest are stored at the nodes instead of the Gauss points. A finite element mesh is created at every time step of the transient problem and the solution is then stored at the nodes [1]. The nodes are generally maintained (unless adaptive refinement or erasal is performed), thus the mesh re-generation consists in reconnecting the existing nodes. The nodes move to their new position according to their velocity and then the finite element mesh is re-generated using an unconstrained Delaunay triangulation/tesselation [9].

In the following the governing equations for the glass are specified. Air is neglected, thus the glass-air interface becomes simply the glass domain boundary. The fully sticking contact with the mold is considered. In Section 3.2 the issues related to the boundary identification and contact treatment are explained in detail.

3.1. Mechanical model

At forming temperatures, elastic effects in the glass are negligible and the behavior is nearly isochoric. Thus, hot glass can be modeled as a viscous incompressible fluid. The total stress tensor can be decomposed into hydrostatic and viscous parts and, therefore, the motion of the hot glass can be described by the Navier-Stokes equations for viscous incompressible fluid. These can be written for the glass domain Ω in Cartesian coordiantes as

$$\rho \frac{D\mathbf{v}}{Dt} + \nabla p - \nabla \cdot (2\mu\epsilon(\mathbf{v})) = \rho \mathbf{g} \quad (2)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (3)$$

where \mathbf{v} is the velocity vector, p the pressure, t - time, $\frac{D\mathbf{v}}{Dt}$ is the material derivative, \mathbf{g} the **gravity**, ρ the density, μ the dynamic viscosity and $\epsilon = \frac{\nabla\mathbf{v} + \nabla^T\mathbf{v}}{2}$ - the deviatoric strain rate.

Boundary conditions. At the mold walls Γ_d and other fixed domain parts (e.g. bottle neck during the gravity stretching), homogeneous Dirichlet boundary conditions are prescribed, i.e.

$$\mathbf{v} = 0 \quad \text{at } \Gamma_d \quad (4)$$

Two type of Neumann boundary conditions will be distinguished. The first one is the so-called “free-surface” condition that can be approximated for vanishing velocity gradients as: $p = 0$ at Γ_n . This condition will be prescribed at the evolving outer surface of the glass prior to the contact with the mold and at the inner surface prior to the application of the compressed air. A Neumann boundary condition will be used in order to account for the air pressure p_a at Γ_n :

$$\mathbf{t}_p = p_a \mathbf{n} \quad \text{at } \Gamma_n \quad (5)$$

This condition is prescribed at the inner surface of the glass.

3.1.1. Finite Element formulation

An equal order linear interpolations for the velocity and pressure variables over 4-noded tetrahedra (3D) is used here for the space discretization of the governing equations Eqs. (2), (3). We use Backward Euler time discretization scheme exclusively for the sake of simplicity. All the arguments presented in the paper are valid for any implicit time integration scheme. Being standard, the space and time discretization are not discussed here (see e.g. [10], [11]). A pressure stabilization term is added due to the use of the equal order velocity-pressure interpolation (see e.g. [12] or [13]).

Given $\bar{\mathbf{v}}_n$ and $\bar{\mathbf{p}}_n$ at t_n , the time discrete problem consists in finding $\bar{\mathbf{v}}_{n+1}$ and $\bar{\mathbf{p}}_{n+1}$ at t_{n+1} as the solution of

$$\mathbf{M} \frac{\bar{\mathbf{v}}_{n+1} - \bar{\mathbf{v}}_n}{\Delta t} + \mu \mathbf{L} \bar{\mathbf{v}}_{n+1} + \mathbf{G} \bar{\mathbf{p}}_{n+1} = \bar{\mathbf{F}} \quad (6)$$

$$\mathbf{D} \bar{\mathbf{v}}_{n+1} + \mathbf{S} \bar{\mathbf{p}}_{n+1} = 0 \quad (7)$$

where \mathbf{M} is the mass matrix, \mathbf{L} is the Laplacian matrix, \mathbf{G} is the gradient matrix, $\bar{\mathbf{v}}$ and $\bar{\mathbf{p}}$ are the velocity and pressure respectively and $\bar{\mathbf{F}}$ is the body force vector. Note the absence of the convective term due to Lagrangian kinematic framework.

The matrices and vectors are assembled from the element contributions defined as

$$\mathbf{M} = \rho \int_{\Omega_e} \mathbf{N} \mathbf{N}^T d\Omega_e \quad (8)$$

$$\mathbf{L} = \int_{\Omega_e} \nabla \mathbf{N} \nabla \mathbf{N}^T d\Omega_e \quad (9)$$

$$\mathbf{G} = - \int_{\Omega_e} \nabla \mathbf{N} \mathbf{N} d\Omega_e \quad (10)$$

$$\bar{\mathbf{F}} = \int_{\Omega_e} \mathbf{N} \rho \mathbf{g} d\Omega_e + \int_{\Gamma_n} p_a \mathbf{n} d\Gamma_n \quad (11)$$

$$\mathbf{D} = -\mathbf{G}^T \quad (12)$$

$$\mathbf{S} = \tau \mathbf{L} \quad (13)$$

\mathbf{N} stands for standard linear FE shape functions vector, Ω_e is the element integration domain, τ is an algorithmic stabilization coefficient defined

as $\tau = \left(\frac{2\|\tilde{\mathbf{v}}\|}{h} + \frac{4\nu}{h^2} \right)^{-1}$, where h is the element size [12]. Note also that the discrete operators given by Eqs. (8)-(13) correspond to the unknown current configuration \mathbf{X}_{n+1} according to the updated Lagrangian approach [1], [14]. Thus, the system is non-linear and must be solved in an iterative manner by updating the discrete operators at every non-linear iteration.

Modified fractional step technique for the efficient solution of governing equations. Solving the governing equation system monolithically (i.e. for velocities and pressure simultaneously) has a considerable computational cost [15]. Pressure segregation or fractional step methods are known for their high computational efficiency, however they often lead to mass conservation problems [16]. We propose to use the modified version of the fractional step approach presented in [2] and further developed in [3]. On the one hand, it inherits the high computational efficiency of the original fractional step procedure ([17], [18], [19]) due to the decoupling of the velocity and the pressure. On the other hand, it has much better mass conservation properties.

The fractional step split is applied here at purely algebraic level to the governing equations system defined by Eqs. (6)-(7). The momentum equation is split into two parts by introducing the intermediate velocity $\tilde{\mathbf{v}}$ and the original monolithic system is replaced by

$$\mathbf{M} \frac{\tilde{\mathbf{v}} - \bar{\mathbf{v}}_n}{\Delta t} + \mu \mathbf{L} \bar{\mathbf{v}}_{n+1} + \mathbf{G} \bar{\mathbf{p}}_{n+1}^g = \bar{\mathbf{F}} \quad (14)$$

$$\mathbf{M} \frac{\bar{\mathbf{v}}_{n+1} - \tilde{\mathbf{v}}}{\Delta t} + \mathbf{G} (\bar{\mathbf{p}}_{n+1} - \bar{\mathbf{p}}_{n+1}^g) = 0 \quad (15)$$

$$\mathbf{D} \bar{\mathbf{v}}_{n+1} + \mathbf{S} \bar{\mathbf{p}}_{n+1} = 0 \quad (16)$$

where $\tilde{\mathbf{v}}$ is an auxiliary vector, representing the intermediate or ‘‘fractional’’ nodal velocities and $\bar{\mathbf{p}}_{n+1}^g$ is the guess of the end-of-step nodal pressure vector. Eq. (14) is known as ‘‘fractional momentum’’ and Eq. (15) as ‘‘end-of-step momentum’’ equations. The novelty of [2] in contrast to the classical fractional step approach consists in using $\bar{\mathbf{p}}_{n+1}^g$ instead of $\bar{\mathbf{p}}_n$ in the fractional momentum equation.

The pressure Poisson’s equation is obtained by enforcing the incompressibility condition (Eq. (16)) with the end-of-step momentum equation (Eq. (15)), leading to

$$\mathbf{D} \tilde{\mathbf{v}} = \Delta t \mathbf{D} \mathbf{M}^{-1} \mathbf{G} (\bar{\mathbf{p}}_{n+1} - \bar{\mathbf{p}}_{n+1}^g) + \mathbf{S} \bar{\mathbf{p}}_{n+1} \quad (17)$$

Using the typical approximation $\mathbf{D} \mathbf{M}^{-1} \mathbf{G} \approx \mathbf{L}$, we arrive at the final system of discretized equations to be solved:

$$\left(\mathbf{M} \frac{\tilde{\mathbf{v}} - \bar{\mathbf{v}}_n}{\Delta t} + \mu \mathbf{L} \tilde{\mathbf{v}} + \mathbf{G} \bar{\mathbf{p}}_{n+1}^g \right) = \bar{\mathbf{F}} \quad (18)$$

$$\mathbf{D} \tilde{\mathbf{v}} = \Delta t \mathbf{L} (\bar{\mathbf{p}}_{n+1} - \bar{\mathbf{p}}_{n+1}^g) + \mathbf{S} \bar{\mathbf{p}}_{n+1} \quad (19)$$

$$\mathbf{M} \frac{\bar{\mathbf{v}}_{n+1} - \tilde{\mathbf{v}}}{\Delta t} + \mathbf{G} (\bar{\mathbf{p}}_{n+1} - \bar{\mathbf{p}}_{n+1}^g) = 0 \quad (20)$$

Fractional momentum equation solution and pressure prediction. The momentum equation (Eq. (18)) is non-linear due to the dependence of the discrete operators on the unknown current configuration \mathbf{X}_{n+1} . Thus, it must be solved iteratively. For this reason, let us define the residual of the fractional momentum equation as

$$\tilde{\mathbf{r}}_m = \bar{\mathbf{F}} - \left(\mathbf{M} \frac{\tilde{\mathbf{v}} - \bar{\mathbf{v}}_n}{\Delta t} + \mu \mathbf{L} \bar{\mathbf{v}}_{n+1} + \mathbf{G} \bar{\mathbf{p}}_{n+1}^g \right) \quad (21)$$

The modified fractional step method proposed in [2] consists in considering the pressure variation in the fractional momentum equation. Thus, $\bar{\mathbf{p}}_{n+1}^g$ cannot be considered constant in the residual expression. In order to obtain the fractional momentum equation dependent on the velocity exclusively (and thus maintain the decoupling of the velocities and the pressure), a prediction for the pressure p_{n+1}^g can be computed using an assumption of slight compressibility. Following this assumption the current-step pressure is obtained by adding the term proportional to the divergence of velocity to the pressure of the previous step (see [14], [2] for details):

$$p_{n+1}^g = p_n + \kappa \int_{t_n}^{t_{n+1}} \nabla \cdot \mathbf{v} dt \quad (22)$$

where κ is the compressibility parameter of the fluid. The discrete form of Eq. (22) using linear velocity-pressure finite elements reads

$$\mathbf{M}_p \bar{\mathbf{p}}_{n+1}^g = \mathbf{M}_p \bar{\mathbf{p}}_n + \kappa \int_{t_n}^{t_{n+1}} \mathbf{D} \bar{\mathbf{v}} dt \quad (23)$$

where \mathbf{M}_p is the pressure mass matrix.

In order to avoid matrix inversion for obtaining the current step pressure, the pressure mass matrix \mathbf{M}_p will be taken in the lumped form. Performing the integration, Eq. (23) can be re-written as

$$\bar{\mathbf{p}}_{n+1}^g = \bar{\mathbf{p}}_n + \kappa \Delta t \mathbf{M}_p^{-1} \mathbf{D} \bar{\mathbf{v}}_{n+1} \quad (24)$$

Expressing the pressure in terms of velocity according to Eq. (24) allows to define the iterative solution of the non-linear equation $\tilde{\mathbf{r}}_m = 0$ (with $\tilde{\mathbf{r}}_m$ defined by Eq. (21)) exclusively in terms of the nodal velocities:

$$\mathbf{H} \delta \bar{\mathbf{v}} = \tilde{\mathbf{r}}_m(\bar{\mathbf{v}}^i, \bar{\mathbf{p}}^i) \quad (25)$$

$$\bar{\mathbf{v}}^{i+1} = \bar{\mathbf{v}}^i + \delta \bar{\mathbf{v}} \quad (26)$$

$$\bar{\mathbf{p}}_{n+1}^g = \bar{\mathbf{p}}_n + \mathbf{M}_p^{-1} \kappa \Delta t \mathbf{D} \bar{\mathbf{v}}^{i+1} \quad \mathbf{X}^{i+1} = \mathbf{X}^i + \Delta t \bar{\mathbf{v}}^{i+1} \quad (27)$$

where “i” stands for the non-linear iteration index at time t_{n+1} and \mathbf{H} is the tangent matrix defined as

$$\mathbf{H} = -\frac{\partial \tilde{\mathbf{r}}_m}{\partial \tilde{\mathbf{v}}} \quad (28)$$

As the non-constant pressure term is now included in the residual, it must be accounted for in the linearization.

According to [3] the dynamic tangent matrix can be approximated as:

$$\mathbf{H} = \frac{\mathbf{M}}{\Delta t} + \mu \mathbf{L} + \Delta t \int_{\Omega_e} \mathbf{B}^T \mathbf{C}_\kappa \mathbf{B} d\Omega_e \quad (29)$$

where the operator \mathbf{B} and the volumetric constitutive matrix \mathbf{C}_κ are defined (in 2D) as

$$\mathbf{B} = \begin{pmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} \end{pmatrix} \quad (30)$$

$$\mathbf{C}_\kappa = \frac{1}{2} \begin{pmatrix} \kappa & \kappa & 0 \\ \kappa & \kappa & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (31)$$

Elimination of the unknown pressure variables from the left-hand-side of the modified fractional momentum equation enables us to solve it for the velocities only (as it is done in the standard fractional step), thereby preserving the decoupling of the nodal velocities and pressures. However at every non-linear iteration the nodal pressure needs to be updated using Eq. (24).

Pressure Poisson' equation and the correction step. The next step to be carried out is the correction of the pressure, i.e. obtaining the end-of-step incompressible pressure field using Eq. (19). Solution of Eq. (19) requires to impose the pressure boundary conditions due to the presence of the Laplacian \mathbf{L} . According to the methodology presented in [2], $\bar{\mathbf{p}}_{n+1} = \bar{\mathbf{p}}_{n+1}^g$ can be used as an essential boundary condition for the pressure necessary for solving the Poisson's equation for the pressure. The quality of this approximation is related exclusively to the value of the compressibility constant κ used in the prediction step. Having the pressure fixed to the predicted value $\bar{\mathbf{p}}_{n+1}^g$ at the free surface, the pressure Poisson's equation is solved elsewhere in the domain to yield the end-of-step pressure vector $\bar{\mathbf{p}}_{n+1}$.

This step can be thus viewed as a correction of the predicted pressure $\bar{\mathbf{p}}_{n+1}^g$ to the correct end-of-step one everywhere except for the free surface, where the "slightly compressible" pressure is maintained. Consequently, the projection step is carried out according to Eq. (20) and returns the end-of-step divergence-free velocity everywhere in the domain except for the pressure boundary, where the divergence-free velocity is approximated.

The implementation of the modified fractional step scheme is very similar to that of the classical fractional step method. It is summarized in Table 1.

- | |
|--|
| <ol style="list-style-type: none"> 1. Solve the fractional momentum, Eq. (18) <ul style="list-style-type: none"> • Update the nodal pressure vector according to Eq. (24) and add it to the fractional momentum residual • Update the discrete operators according to the new nodal positions \mathbf{X}_{n+1} • Repeat two previous steps until convergence for the velocity field is achieved 2. Solve the pressure Poisson's Eq. (19), using the predicted pressure as the boundary condition 3. Solve the end-of-step momentum equation (Eq. (20)) |
|--|

Table 1: Implementation procedure for the modified fractional step scheme

3.2. Contact modeling and boundary mesh

Contact modeling. Accurate modeling of contact between the glass and the wall of the mold is essential for predicting the wall thickness of the glass object correctly. Literature exhibits a large number of numerical methods. In Eulerian non-matching grid formulations a large variety of methods has been proposed ranging from penalty approaches ([20], [21]) to augmented Lagrange multipliers [22]. For the matching grid methods the contact modeling is usually done using simple geometric considerations [4]. In this study we assume that the mold is a rigid body and the glass sticks to the mold. This is a reasonable and commonly accepted approximation (see e.g. [6], [4]).

We propose here a simple algorithm that allows preserving the overall strategy of the PFEM, but at the same time does not lead to the incorrect prediction of the wall thickness. For the sake of minimizing the computational cost associated with re-meshing, PFEM utilizes unconstrained Delaunay triangulation/tetrahedrization ([23]) equipped with the alpha-shape technique for detecting boundaries.

Modified PFEM contact algorithm. Considering that the nodes follow a variable distribution $h(x)$, which is the minimum distance between two nodes, the *alpha-shape* technique (see [24], [1]) applied to the unconstrained Delaunay mesh allows to distinguish whether an element defined by the four nodes must be created or not. The radius r of a sphere defined by the examined nodes is compared to the corresponding size h (average size of the inter-nodal distances). If the ratio $\frac{r}{h} > \alpha$, where α is the alpha-shape parameter typically taken as 1.5 in 3D, the element is not created and the nodes are marked as belonging to the boundary (for details on the alpha-shape techniques one can see [1]). For the correct contact representation we propose to combine the alpha-shape technique with the boundary markers.

Let us distinguish the outer boundary of the glass domain and the mold wall with an *interface* and *wall* flags, respectively. As the *interface* nodes approach

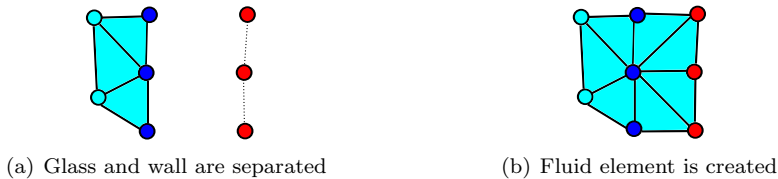


Figure 3: Schematic representation of the standard PFEM contact algorithm

the *wall* nodes, the distance between them diminishes and an element is created. According to the standard PFEM algorithm such an element immediately obtains the properties of the fluid, thus bringing the fluid and the solid into contact (see Fig. 3).

However, this newly created element (see Fig. 3(b)) defines a “premature” contact (both mechanical and thermal) between the glass and the wall, as the distance between the original fluid boundary nodes and the wall are still of order h . This implies introducing an error of order h in the determination of the thickness of the fluid layer (glass wall in our case). In this work we propose a simple change in the contact algorithm that leads to a considerable improvement in the accuracy of thickness determination.

We propose to consider such the newly created contact element a purely geometrical entity, that does not contribute to the governing equation system and thus does not result in any resistance to the fluid motion. This can be formally described as: if the mesher creates an element that contains both *interface* and *wall* nodes, mark it as a *fictitious element* and multiply its contribution (residual and tangent matrix) by zero during the finite element assembly process. Nevertheless, fictitious elements can be used for tracking the distance between the *interface* nodes and the wall. Moreover, creating fictitious elements as geometric entities enables maintaining the original PFEM unconstrained meshing strategy.

The distance d between the *interface* node and the wall is computed within each fictitious element. As long as d is larger than a given tolerance ϵ , the fictitious element is maintained and no actual contact between the fluid and the solid is established. As soon as d reached a value lower than a given tolerance ϵ , the *interface* node is erased, thus the glass domain gets connected to the wall via a real element in a fully sticking contact (see Fig. 4 c) and d)). This way, the error in the determination of the thickness of the bottle becomes controlled by the user-defined tolerance ϵ . In the simulations used in the present work, ϵ was taken as 10 % of the element size.

Preserving the boundary mesh quality. The domain deformation occurring in a glass forming process can be viewed as a stretch in different directions. This leads to the danger of decreasing the mesh resolution and possibility of creating non-physical “holes” at the glass domain boundary due to alpha-shape (see stretching of a domain fragment fixed at the top on Fig. 5(a)). In order to preserve the mesh quality and avoid the creation of holes, an adaptive refinement

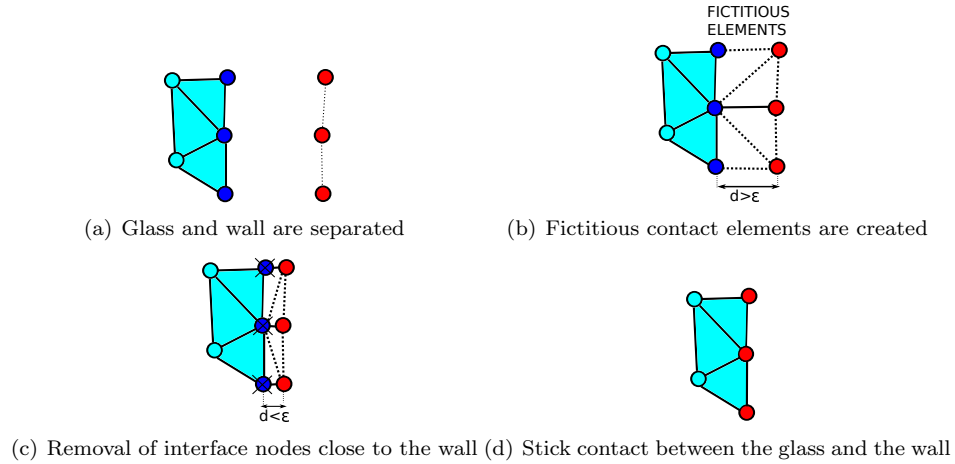


Figure 4: Schematic representation of the modified PFEM contact algorithm

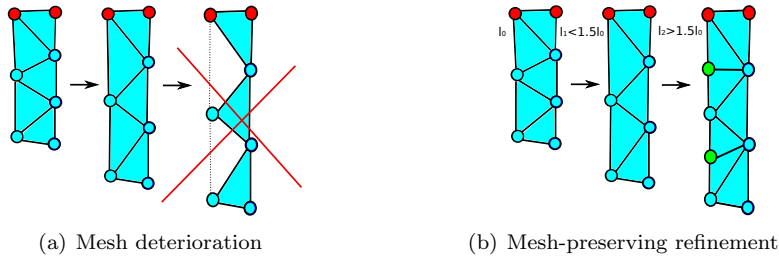


Figure 5: Mesh refinement at the boundary

of glass domain boundary is used. A new node is introduced at the center of the boundary elements edge as the corresponding length reaches of 1.5 times the original value (see Fig. 5(b)).

3.3. Thermal model

During glass forming process the temperature distribution within the glass is highly non-uniform. First of all, the temperature varies in the thickness direction of the glass domain due to cooling of the free surfaces. Second, when the glass reaches the mold wall a local rapid cooling occurs, resulting in a strong increment of viscosity, thus altering the shape evolution: zones with the higher temperature will undergo stronger stretching and thinning [25]. Thus, inclusion of the heat equation is mandatory and purely mechanical models ignoring the heat transfer must be discarded when modeling the problem of interest.

Assuming that the conduction obeys the Fourier law which relates the heat flux with the temperature and the thermal conductivity, the heat transfer equation can be written in the Lagrangian reference frame as:

$$\rho c \frac{DT}{Dt} = k \nabla^2 T + q \quad (32)$$

where c is the specific heat, T is the temperature, k is the thermal conductivity and q is the internally generated heat flux. Using linear temperature finite elements and the Backward Euler scheme for space and time discretization, respectively, leads to the following equation:

$$\rho c \frac{\bar{\mathbf{T}}_{n+1} - \bar{\mathbf{T}}_n}{\Delta t} \mathbf{M} = k \mathbf{L} \bar{\mathbf{T}}_{n+1} + \mathbf{Q} \quad (33)$$

where \mathbf{M} and \mathbf{L} are the mass and Laplacian matrices and $\mathbf{Q} = \int_{\Omega_e} \mathbf{N} q d\Omega_e$. Note the absence of the convective term in the Lagrangian framework. Heat convection is automatic in the Lagrangian model, as the heat is convected by the moving Lagrangian grid “attached to the material”.

The problem must be equipped with the Dirichlet boundary conditions (fixed temperature $T = T_d$ at mold in the locations to be specified) and Neumann boundary condition (adiabatic condition will be considered in this work. Heat conduction prior to the contact with the mold (occurring due to convection of the air in contact with the glass) will not be considered here. Once the contact with the mold has taken place, the direct heat conduction between the glass and the mold takes place. Temperature field T_{n+1} obtained as a solution of Eq. (33) is used for computing the viscosity distribution (Eq. (1)) for the mechanical model. More details on the thermo-mechanical coupling can be found in [26], [27].

3.4. Overall solution algorithm

To this end all the “ingredients” of the thermo-mechanical model are defined. Table 2 presents the overall solution algorithm for the bottle forming process.

4. Examples

In this section two numerical examples are solved. We mentioned previously that there exist practically no established benchmark for the glass forming models. In the majority of literature on the numerical modeling of the glass manufacturing process the data provided is insufficient for reproducing the tests. Thus, in this section we first validate the proposed formulation by solving one of the very few examples suitable for the validation of the glass forming simulation model on a simple rectangular geometry. Next we propose a numerical test dealing with the bottle-forming that could serve as a reference in the future. The main aim is to accurately define all the data necessary for reproducibility of this example.

- | |
|--|
| <ol style="list-style-type: none"> 1. Discretize the glass domain with a finite element mesh. 2. Identify the external boundaries for the fluid and glass-mold wall contact elements (using the alpha-shape technique + boundary markers) 3. Solve the Lagrangian equations for the glass. Obtain nodal velocities, pressure and displacements. 4. Move the mesh nodes to a new position (according to the computed displacements). 5. Solve the heat transfer equation. Obtain the temperature distribution of the glass. 6. Update viscosity according to the obtained temperature distribution using Eq. 1 7. Re-generate the mesh for the glass domain (nodes are generally maintained) 8. Go back to the next time step. Start the solution from Step 2 |
|--|

Table 2: Solution strategy for the bottle forming simulation

4.1. TV panel pressing

This example reported by Berndhauser in [28] models the manufacturing of a TV glass panel. Even though it does not deal explicitly with the bottle manufacturing process, the characteristics of the example are very similar (pressing the glass gob into a mold, accounting for thermo-mechanical nature of the process). Moreover, it provides simple nearly rectangular geometry which facilitates the validation. A portion of molten glass in the shape of cylindrical gob, is pressed into the mold by the plunger (see Fig 6). The plunger is moving downwards with a prescribed time-dependent velocity. During pressing, the glass is flowing while it cools down because of the contact with the colder mold and plunger. The problem settings and material properties presented below are taken from [28].

The problem to be simulated accounts for the thermo-mechanical process involving the plunger, the glass and the mold. The mold and the plunger are considered to be rigid. The overall time of 3 seconds is simulated. The plunger descends with the vertical velocity $v_z = (-0.0842e^{-1.535t} + 0.00842)$ m/s for $0.0s < t < 1.5$ s. For $1.5s < t < 3.0s$ it remains still. The no-slip condition at the glass-wall interface is considered.

The thermal conditions are as follows: the initial glass temperature $T_g = 1000$ °C. The initial temperature of the plunger and the mold is $T_p = T_m = 500$ °C. Dirichlet boundary condition for the temperature is prescribed at the outer surface of the mold and the outer surface of the plunger. The rest of the mold and the plunger surfaces are considered to be adiabatic (heat flux $Q = 0$).

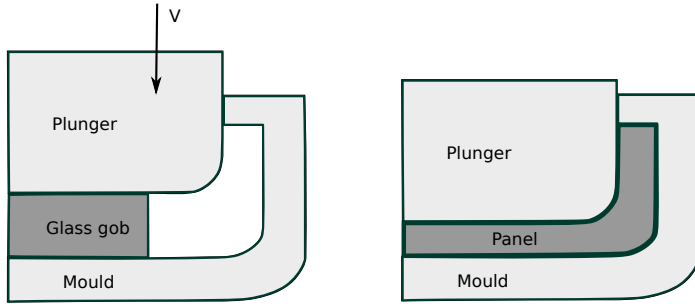


Figure 6: Cross section of the glass pressing process: initial and final states

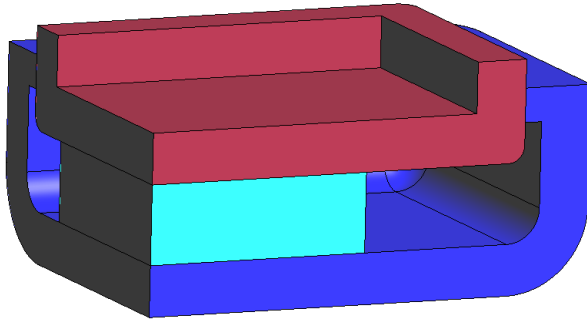


Figure 7: TV screen forming set-up (one quarter): mold, gob and plunger

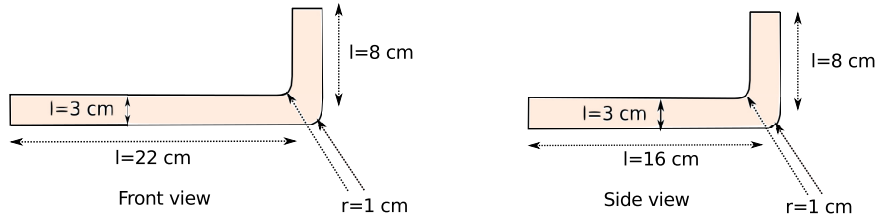
The geometry of the model is shown in Fig. 7. Geometrical details can be found in Fig. 8. The initial distance between the mold and the plunger $d_{pl}^0 = 4.67376$ cm is equal to the original height of the glass gob. The gob has a cylindrical shape with a radius of $r = 13.243$ cm.

Material properties of the plunger, the mold and the glass are summarized in Table 3. Gravity of 9.8 m/s^2 is considered.

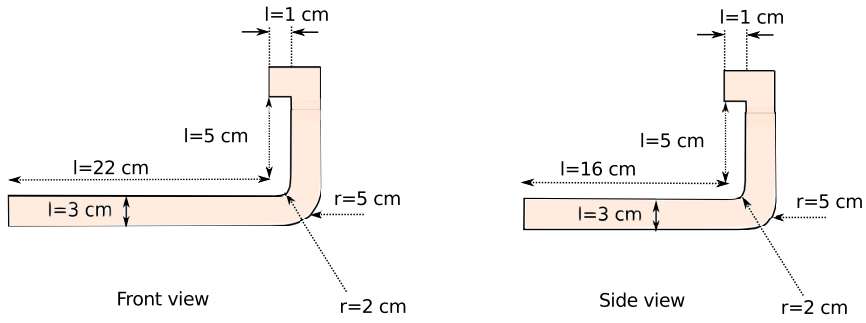
Results. According to the benchmark proposal, the temperature, pressure and the glass front evolution are analyzed. The values of the variables of interest are inspected at several locations indicated in Fig. 9. “Sensor” lines are located at four corners of the plain part of the mold bottom as L1: $x=0, y=0$; L2: $x=0.22 \text{ m}, y=0$; L3: $x=0, y=0.16 \text{ m}$; L4: $x=0.22, y=0.16 \text{ m}$.

Fig. 10 displays the evolution of the temperature at L1 reported in [28]. One can see the compressing gob due to the plunger descend. In the beginning the temperature field exhibits a sharp discontinuity at the glass/mold and the glass/plunger interface. As time evolves, the transition region appears. At $t=1.5$ s the plunger stops and the process is governed by diffusion only.

A comparison of the results obtained in the present work with the reference data is displayed in Fig. 11. One can see a good agreement between the com-



(a) Cross section of the plunger; front and side view



(b) Cross section of the mold; front and side view

Figure 8: Dimensions of the mould and the plunger.

| Glass | |
|--------------------------|---|
| Viscosity μ | $10^{-2.8 + \frac{4700}{T - 220}}$ Pa s |
| Conductivity λ_g | 5 W/m K |
| Specific heat c_g | 1400 J/kg K |
| Density ρ_g | 2500 kg/m ³ |
| Plunger and mold | |
| Conductivity λ_m | 20 W/m K |
| Specific heat c_m | 500 J/kg K |
| Density ρ_m | 8000 kg/m ³ |

Table 3: Material properties of the model components

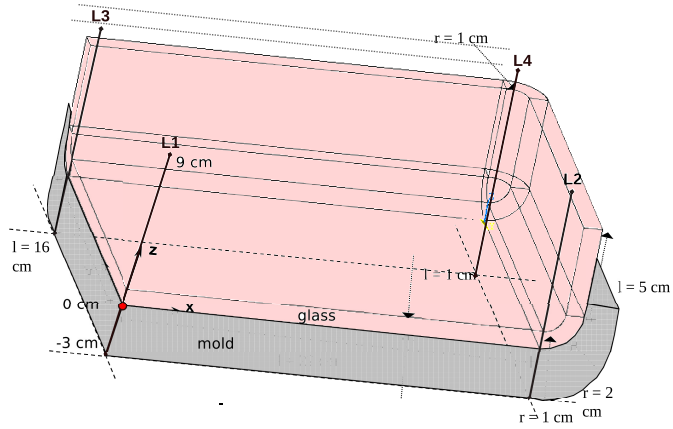


Figure 9: Location of the inspection lines (“sensors”) [28]

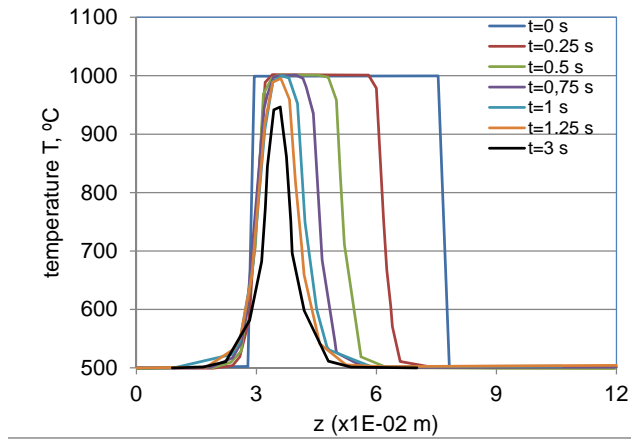


Figure 10: Temperature evolution at L1 according to [28]

pared values. In the beginning the temperature is distributed almost uniformly across the glass thickness (z -direction). Around 0.75 s the maximum temperature concentrates in the middle plain of the gob. At the end of the simulation the maximum temperature in the middle of the glass thickness reduces from 1000 °C to some 950 °C.

The slightly smaller temperature observed at $t=0.01$ s in the lowest edge of the moving plunger ($z=0.075$) is a spurious overshoot, likely to occur due to the steep temperature gradient (this phenomenon is addressed, for example, [29]). Steep temperature gradient is present at both contact interfaces, namely at $z=0.03$ and $z=0.075$. However, since mold and the adjacent glass at the bottom ($z=0.03$) are not moving, no convection occurs there and thus the overshoot is observed only at the contact above ($z=0.075$), where glass is in contact with the moving plunger. We note that the overshoot is only of some 3 C (which is less than 1 % of the temperature difference between the mold and the glass).

Evolution of glass front in the vertical direction of the xz -cross section (long side, $x=0.23$ m, $y=0$) is shown in Fig. 4.1. In the reference [28] the location of the glass front is reported only for $t>1$ s. We included the results spanning over the total simulation time. Since the plunger reaches zero velocity at $t=1.5$ s, the glass front position remains constant afterwards. One can see good agreement between the results of the present work and the ones obtained in [28]. Slightly slower increment in height may be attributed to either different mesh resolutions or presence of the gravity in the present work (gravity was neglected in [28]). One other factor that may influence this result is the volume conservation of the method. Spurious volume gain would lead to the faster domain evolution, while volume losses would lead to a slower growth. In order to exclude this source of inaccuracy of our method, the glass volume evolution had been recorded throughout the simulation (see Fig. 4.1). One can see excellent volume conservation features of the present method. Maximum volume variation observed was below 4 %. Volume conservation features of the method are further analyzed in the next example.

The pressure evolution at four inspection lines is shown in Fig. 4.1. It is important to note that pressure varies insignificantly in the thickness direction. Thus a single value is representative. Originally, glass does not reach the position of sensors L2, L3 and L4 (thus, zero values are observed). First, the glass front reaches L2, then L3 and, finally, the further most corner L4. The maximum pressure of appx. 1.1 MPa is reached at L1 ($t=1.25$ s). An example of the pressure distribution at $t=0.50$ s is shown in Fig. 4.1.

4.2. Bottle manufacturing: final blow process

In this example we concentrate our attention on the second stage of the bottle manufacturing process, namely the gravity stretching and the final blow. The objective of this example is to test the capability of the presented numerical approach to solving this challenging problem. Moreover, we strive to establish an example that may be used as a first approximation to a benchmark for the thermo-mechanical simulations of the bottle forming process. In order to facilitate the modeling and reduce computational time, the mold is modeled

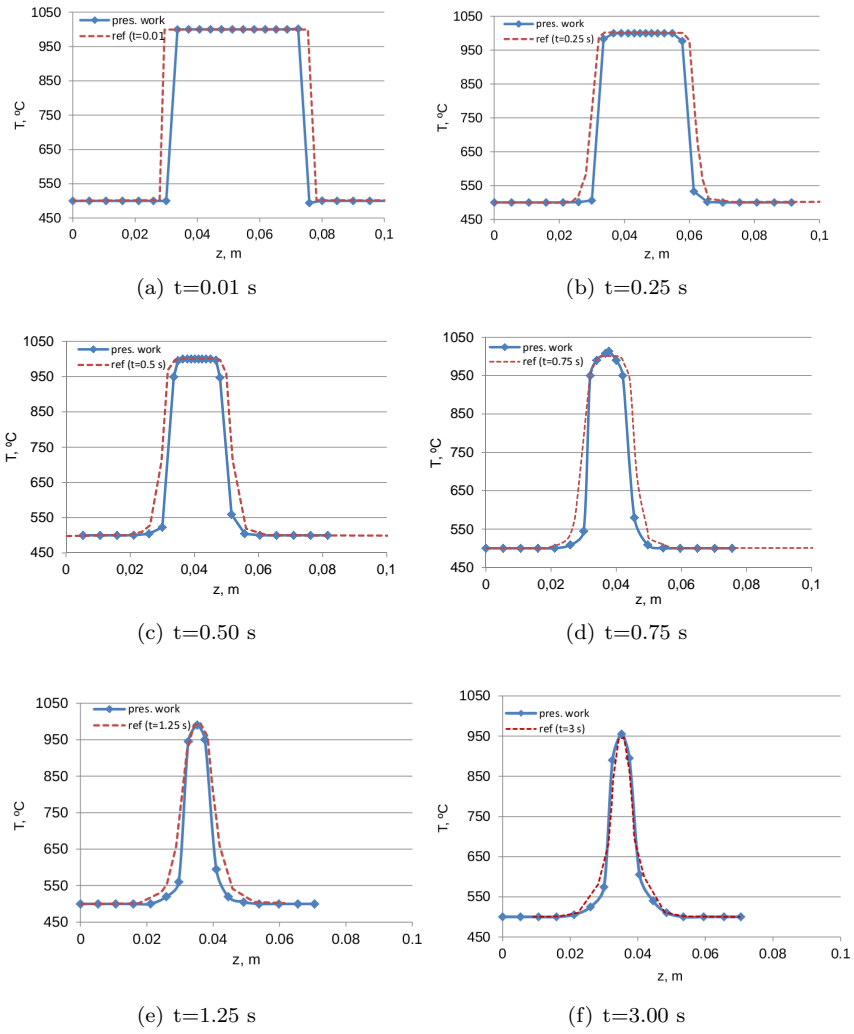


Figure 11: Temperature along L1 at different time steps. Comparison with [28]

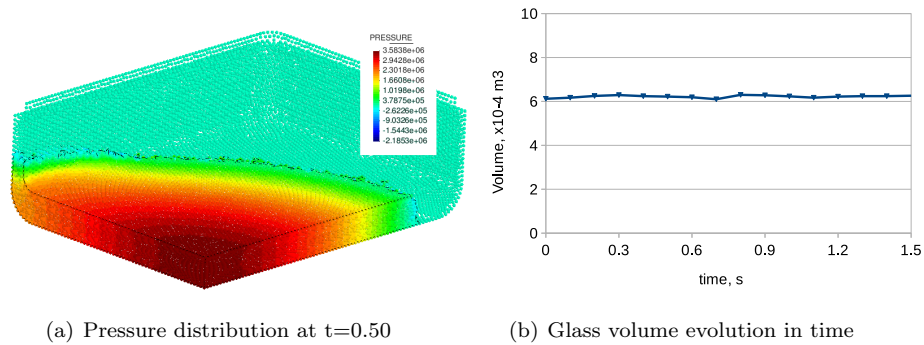


Figure 12: TV screen pressing: pressure and volume data.

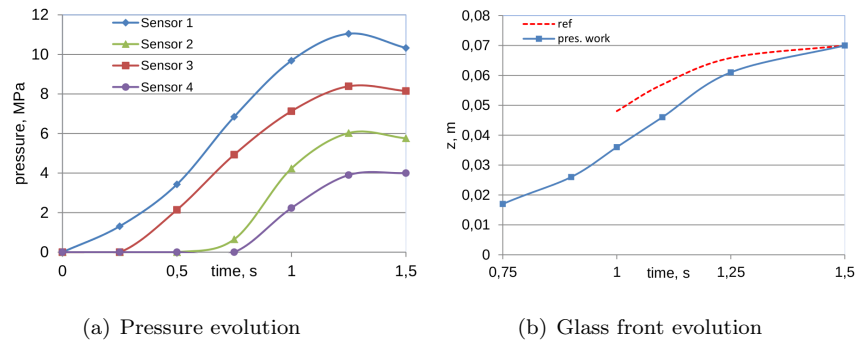


Figure 13: Pressure and glass front evolution in the gob

| Glass | |
|--------------------------|---|
| Viscosity | $\mu = 265677693762693e^{-0.0233569026*T}$ Pa s |
| Conductivity λ_g | 1.5 W/m K |
| Specific heat c_g | 1409 J/kg K |
| Density ρ_g | 2400 kg/m ³ |

Table 4: Material properties of glass in the final blow simulation

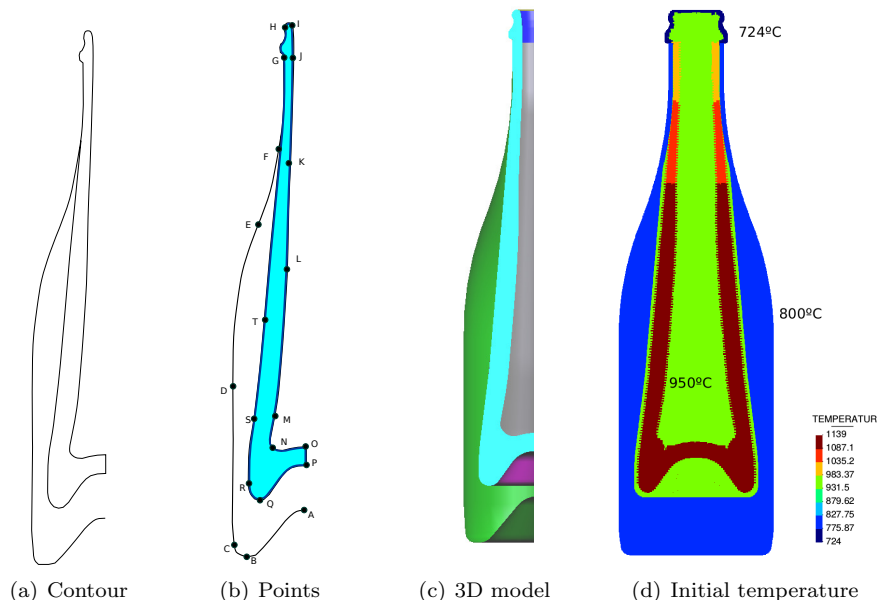


Figure 14: Model for the final blow simulation

merely as a set of nodes discretizing its inner wall providing Dirichlet boundary for both the mechanical and the thermal problem. The thermal problem is modeled considering convective and diffusive heat transfer modes (radiation is neglected). We note again that adopting the Lagrangian framework, convection in the glass phase is automatically solved due to the mesh motion. No convection condition at the glass domain boundary is prescribed.

The initial geometry consists of the mold wall and the parison. The geometry data is shown in Fig. 14(c).

The 3D model can be obtained by applying rotation to the surface (representing the parison) and the curve (mold wall) around the vertical axis (see Fig. 14(a)). The location of the points necessary for reproducing the geometry is shown in Fig. 14(b) and the corresponding coordinates are summarized in Table 5. The mold and the glass domain share the point G . The bottle neck and the mold are fixed. The domain boundaries are defined as follows: Dirichlet

| Points | x | y | z |
|--------|----------|-----|----------|
| A | 0 | 0 | -0.2815 |
| B | -0.0346 | 0.0 | -0.30848 |
| C | -0.042 | 0 | -0.3020 |
| D | -0.0429 | 0 | -0.20632 |
| E | -0.0275 | 0 | -0.1127 |
| F | -0.0163 | 0 | -0.0662 |
| G | -0.0144 | 0 | -0.0185 |
| H | -0.0145 | 0 | -0.0014 |
| I | -0.0100 | 0 | 0 |
| J | -0.0086 | 0 | -0.0185 |
| K | -0.01004 | 0 | -0.0803 |
| L | -0.01139 | 0 | -0.1420 |
| M | -0.01874 | 0 | -0.2241 |
| N | -0.01963 | 0 | -0.2440 |
| O | 0 | 0 | -0.2447 |
| P | 0 | 0 | -0.2815 |
| Q | -0.028 | 0 | -0.2276 |
| R | -0.034 | 0 | -0.27 |
| S | -0.029 | 0 | -0.22 |
| T | -0.0246 | 0 | -0.169 |

Table 5: Model for the final blow simulation

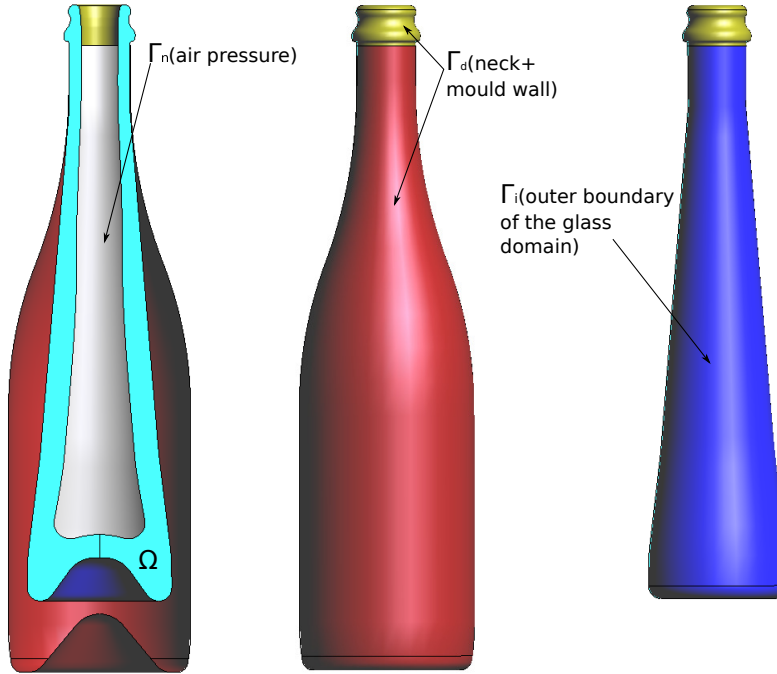


Figure 15: The initial domain and the location of Dirichlet and Neumann boundaries in the *final blow mold*.

boundary Γ_d comprises of the bottle neck and the mold wall. The mold wall becomes part of the computationally domain Ω only once the contact with the glass is established (see Section 3.2 for details). The mold wall, the glass domain as well as Dirichlet and Neumann boundaries are shown in Fig. 15.

We shall also distinguish the free outer boundary of the glass domain Γ_i . Note, that as soon as the glass node belonging to Γ_i touches the wall, it becomes a part of Γ_d .

The temperature at mold walls is fixed to 800 °C. At the bottle neck the temperature is fixed to 724 °C. At the inner and outer surfaces of the glass the initial temperature is set to 950 °C. Elsewhere in the glass domain, the initial temperature distribution is prescribed according to $T_{init} = 950 + (1140 - 950)(1 - \frac{z - z_{bot}}{z_{top} - z_{bot}})^3$, where z is the vertical coordinate of the given node, z_{bot} and z_{top} are the minimum and maximum vertical coordinates of the parison. The initial temperature distribution is shown in Fig. 14(d). During first two seconds the parison is exposed to gravity exclusively. At $t=2$ s air pressure of 0.14 MPa is applied at the inner cavity. At $t=2.6$ the process is completed. The glass density is $\rho = 2400 \text{ kg/m}^3$. The viscosity is computed from the temperature field according to: $\mu = 265677693762693e^{-0.0233569026 \cdot T}$. The industrial partner (see Acknowledgements) has provided the authors directly with the viscosity values

for a number of temperatures, rather than the values of Fulcher constants. Thus, simple exponential fitting to find a function approximating this data was used.

Results. Fig. 16 shows the evolution of the glass object. One can see that around 0.3 seconds glass reaches the mold and accommodates there until the air pressure is applied. It takes less than 0.5 seconds to press the glass into the mold.

Fig. 17 shows the temperature distribution along the vertical coordinate at the inner bottle surface. In the beginning of the simulation temperature of 950 °C is set at inner (and outer) surfaces (green line). The initial temperature distribution in the middle of the wall (thickness) is shown in blue. The temperature (and, consequently, viscosity) at the surface of the glass evolves due to convection and diffusion leading to the final temperature distribution at the inner surface (shown in red).

Fig. 18 displays the evolution of the glass volume. In case of using the standard PFEM algorithm (not using the fictitious contact elements approach) the total volume increases by as much as 35 %. One can distinguish two stages: the one corresponding to the gravity stretch (the contact between the parison on the mold becomes established at around 0.3 seconds) and the one of the air pressure application. Since the contact area during the gravity stretch is mainly restricted to the bottom of the bottle, the overall volume growth due to contact is minor. On the other hand, air presses the entire outer surface of the bottle towards the mold wall. This process starts at $t=2$ seconds and at around $t=2.3$ seconds the entire outer surface of the glass is in contact with the wall. A major spurious volume increment is observed at this stage. From this point onwards no considerable volume change takes place.

On the other hand, if the fictitious contact elements algorithm is used, the volume conservation considerably improves. One can observe the line corresponding to the apparent volume (actual volume of the glass + volume of the fictitious contact elements): the apparent volume increases when the contact elements are created, but since these elements provide no resistance it reduces to the actual one when these contract and finally vanish. Thus the apparent and the actual glass volumes coincide at the end of the simulations (and, practically coincide at the end of each sub-stage: the gravity stretch and the air blow). In case of using the proposed contact algorithm, volume gain reduces to 5 %.

Fig. 19 shows the wall thickness distribution at $t=2.6$ s (vertical axis being the distance from the top measured along the wall coordinate L). This distribution can be used as the main reference data for the future comparisons.

Summary and conclusions. In this paper we have presented a 3D Lagrangian thermo-mechanical model for glass forming simulation. The model follows the PFEM philosophy combining the features of a classical FEM and a Lagrangian mesh-free method. A modified fractional step methodology has been introduced for obtaining a computationally efficient solution without compromising on the mass conservation.

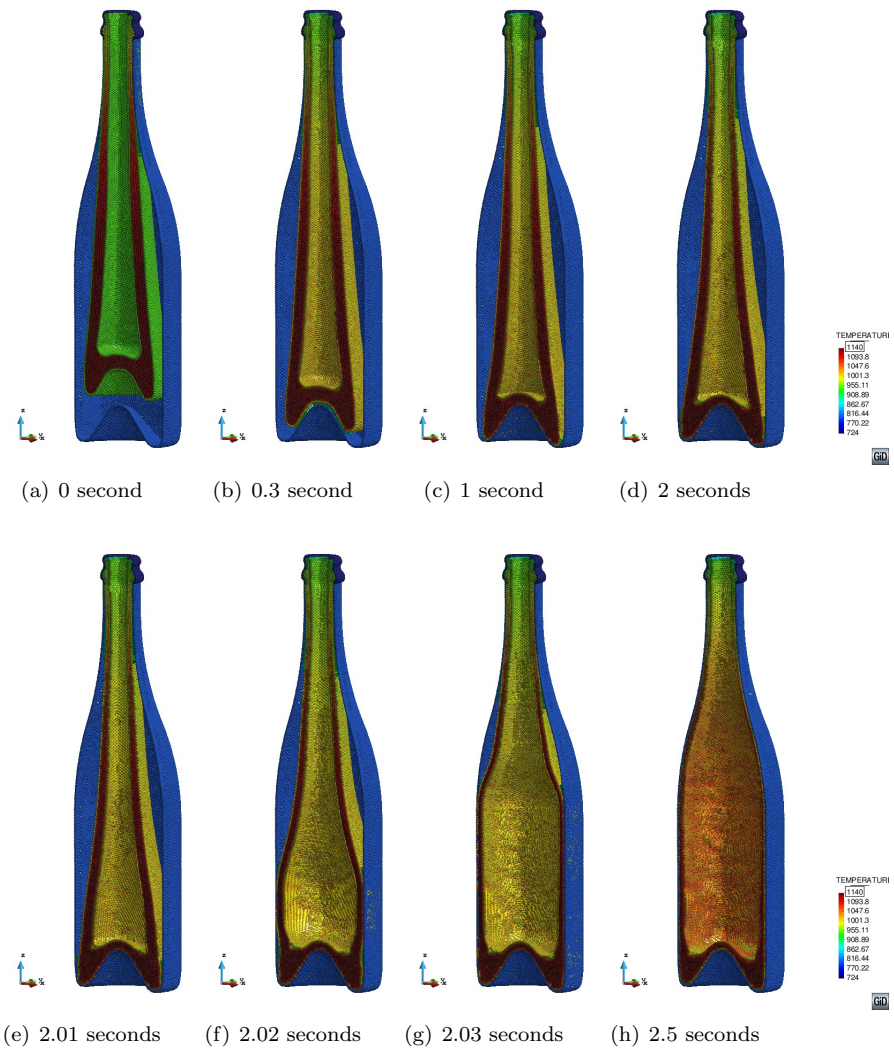


Figure 16: Gravity stretching and final blow

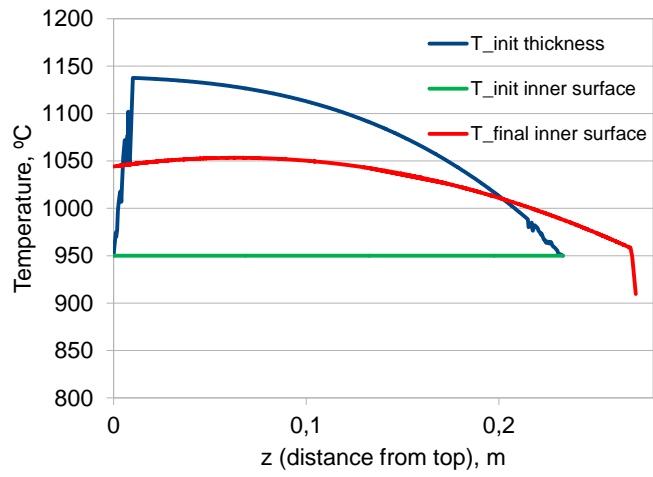


Figure 17: Temperature distribution along the bottle height

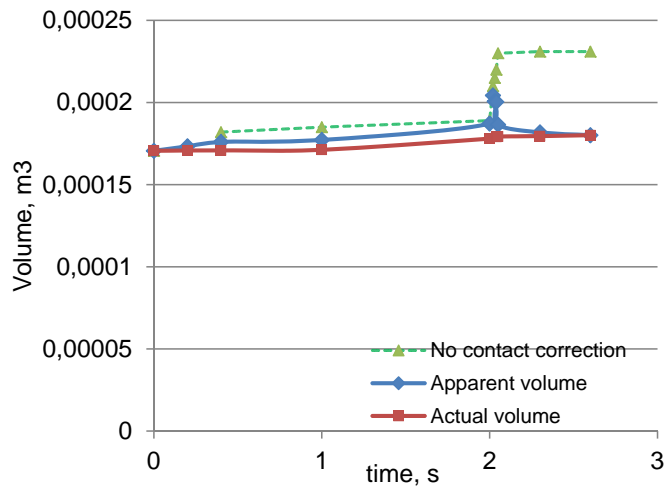


Figure 18: Temporal evolution of glass volume

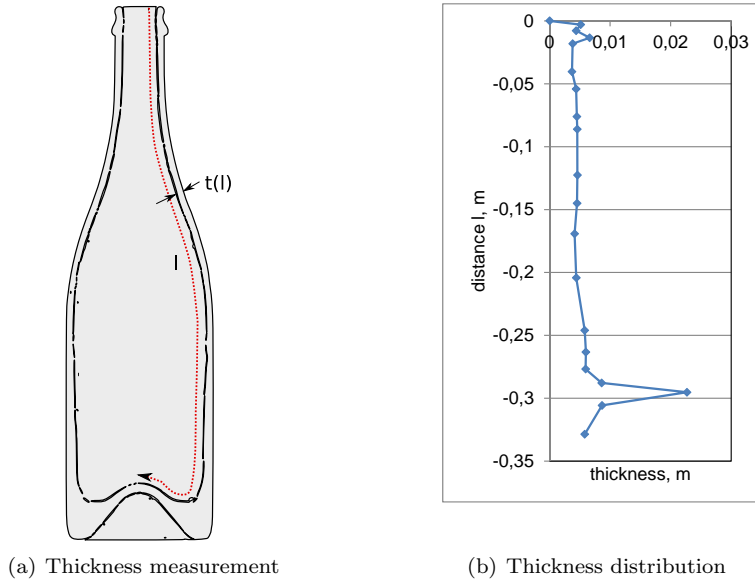


Figure 19: Wall thickness distribution along the bottle height

The model was validated using a TV-panel pressing example, a test characterized by features similar to bottle forming but with a simpler geometry. A numerical example dealing with the *gravity stretch* and *final blow* stage of the bottle manufacturing process has been proposed. We intended to create an example that would fulfill the need of a standardized reference lacking up-to-date in the field of bottle manufacturing simulation. All the data necessary for reproducing this test (geometry, material's properties, boundary conditions) has been provided. Even though several strong approximations are introduced in this example (such as simplified thermal contact, initial temperature distribution), the example defines a first approximation to a benchmark in the field.

The final blow process was simulated in a reasonable computational time (around 1 hour using a quad-core I7 PC) on a computational mesh containing around 300 000 four-noded tetrahedra elements. The facility of the free-surface evolution tracking together with the automatic transfer of the non-homogeneous and temperature-dependent material properties proves that Lagrangian approaches are more advantageous for the problem at hand than Eulerian ones. Thermal contact did not require any special technique, being an intrinsic feature of a matching-mesh method. The proposed mechanical contact methodology is promising, allowing the user to control the error in the thickness direction. In the present work the geometrical contact tolerance (i.e. the critical distance leading to erasal of the interface node of a fictitious element) was

set to 10 % of the element size. The simulations carried out have proven that the method possesses excellent mass conservation features (≈ 5 % of volume variation for the given contact tolerance).

Keeping in mind all the advantages of the formulation, it is important to note that it also has some limitations. For optimal functionality of the method the mesh size distribution must be as uniform as possible. Thus only the mesh quality can be easily controlled. Moreover, the time step size was restricted due to the danger of element inversion. Using a novel Jacobian-independent Lagrangian explicit stream-line temporal integration [30] is a promising option that must be investigated further. While keeping the overall architecture of the approach proposed here, this alternative Lagrangian formulation may lead to considerable advantages in computational efficiency removing the time step restrictions faced by the present method. It is important to note that bottle forming simulations deal with a problem involving very large changes in viscosity due to temperature variation. Thus, purely mechanical simulations are meaningless for the practical purposes. Even though the heat equation was included in the present model and viscosity was computed according to the changing temperature field, the present model did not account for radiative heat transfer. It is essential to include this feature in the next development step of this numerical method.

Acknowledgements

This work was supported under the auspices of the FPDI-2013-18471 grant of the Spanish *Ministerio de Economía y Competitividad*. It was also partially funded by the NUMEXAS project of the European Commission (FP7) and the COMETAD project of the National RTD Plan (ref. MAT2014-60435-C2-1-R) of the Spanish *Ministerio de Economía y Competitividad*. The authors of this work express their gratitude to Dr. J. Jimenez from Vidrio&Engineering COMM. V. for the consultations regarding the bottle forming technology and, in particular, the data provided for Example 2.

- [1] E. Oñate, S. Idelsohn, F. Del Pin, and R. Aubry. The Particle Finite Element Method: an overview. International Journal of Computational Methods, 1:267–307, 2004.
- [2] P. Ryzhakov, E. Oñate, R. Rossi, and S. Idelsohn. Improving mass conservation in simulation of incompressible flows. Int. Jour. for Num.Methods. in Eng., 90/12:1435–1451, 2012.
- [3] P. Ryzhakov. A modified fractional step method for fluid-structure interaction problems. Revista Intern. Met. Num. Ing. (RIMNI), 2016. DOI 10.106/j.rimni.2015.09.002.
- [4] J.M.A. Cesar de Sa. Numerical modelling of glass forming processes. Eng. Comput., 3:266–275, 1986.

- [5] H. Matthew. Numerical simulation of glass forming and conditioning. Journal of the American Ceramic Society, 85(5):1047–1056, 2002.
- [6] E. Feulvarch, N. Moulin, P. Saillard, T. Lornage, and J-M Bergheau. 3d simulation of glass forming process. Journal of materials processing technology, 164:1197–1203, 2005.
- [7] C.G. Giannopapa and J. Groot. Modeling the blow-blow forming process in glass container manufacturing: a comparison between computations and experiments. Journal of Fluids Engineering, 133(2):021103, 2011.
- [8] G.S. Fulcher. Analysis of recent measurements of the viscosity of glasses. Journal of the American Ceramic Society, 8(6):339–355, 1925.
- [9] B. Delaunay. Sur la sphere vide. Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793–800, 1934.
- [10] J. Donea and A. Huerta. Finite element method for flow problems. J. Wiley edition, 2003.
- [11] R. Löhner. Applied CFD Techniques, 2nd edition. J. Wiley and Sons edition, 2008.
- [12] R. Codina. A stabilized finite element method for generalized stationary incompressible flows. Computer Methods in Applied Mechanics and Engineering, 190(20-21):2681 – 2706, 2001.
- [13] Oñate E. A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. Computer Methods in Applied Mechanics and Engineering, 182:355–370, 2000.
- [14] P. Ryzhakov, R. Rossi, S. Idelsohn, and E. Oñate. A monolithic Lagrangian approach for fluid-structure interaction problems. Journal of Computational Mechanics, 46/6:883–399, 2010.
- [15] P. Ryzhakov, J. Cotela, R. Rossi, and E. Oñate. A two-step monolithic method for the efficient simulation of incompressible flows. International Journal for Numerical Methods in Fluids, 74(12):919–934, 2014.
- [16] S. Idelsohn and E. Oñate. The challenge of mass conservation in the solution of free-surface flows with the fractional-step method: problems and solutions. International Journal for Numerical Methods in Biomedical Engineering, 26:1313–1330, 2010.
- [17] A.J. Chorin. A numerical method for solving incompressible viscous problems. Journal of Computational Physics, 2:12–26, 1967.
- [18] R. Temam. Sur l'approximation de la solution des equations de Navier-Stokes par la methode des pase fractionaires. Archives for Rational Mechanics and Analysis, 32:135–153, 1969.

- [19] R. Codina. Pressure stability in fractional step finite element method for incompressible flows. Journal of Computational Physics, 170:112–140, 2001.
- [20] D. Perić and D. Owen. Computational model for 3-d contact problems with friction based on the penalty method. International journal for numerical methods in engineering, 35(6):1289–1309, 1992.
- [21] P. Papadopoulos and R. Taylor. A simple algorithm for three-dimensional finite element analysis of contact problems. Computers & Structures, 46(6):1107–1118, 1993.
- [22] J. Simo and T. Laursen. An augmented lagrangian treatment of contact problems involving friction. Computers & Structures, 42(1):97–116, 1992.
- [23] H. Si. Adaptive tetrahedral mesh generation by constrained Delaunay refinement. Internat. J. Numer. Methods Engrg., 75:856–880, 2008.
- [24] N. Akkiraju, H. Edelsbrunner, M. Facello, Fu P., E. P. Mucke, and Varela C. Alpha shapes: definition and software. Proceedings of International Computational Geometry Software Workshop, 1995.
- [25] S. Grégoire, J. Cesar de Sá, P. Moreau, and D. Locheignies. Modelling of heat transfer at glass/mould interface in press and blow forming processes. Computers & structures, 85(15):1194–1205, 2007.
- [26] Codina R., Vazquez M., and Zienkiewicz O.C. A general algorithm for compressible and incompressible flows. International Journal for Numerical Methods in Fluids, 27:13–32, 1998.
- [27] Pavel Ryzhakov, Riccardo Rossi, and Eugenio Oñate. An algorithm for the simulation of thermally coupled low speed flow problems. International Journal for Numerical Methods in Fluids, 70(1):1–19, 2012.
- [28] C. Berndhauser. Tc25 - modelling of glass forming processes short review of activities 2000 - 2009. In GlassTrend - ICG seminar and workshop INNOVATION IN GLASS PRODUCTION. International Commission on Glass, 2013.
- [29] Augustin M., Caiazzo A., A. Fiebach, J. Fuhrmann, Volker J., Linke A., and Umla R. An assessment of discretizations for convection-dominated convection–diffusion equations. Computer Methods in Applied Mechanics and Engineering, 200(47):3395–3409, 2011.
- [30] S.R. Idelsohn, N. Nigro, J. Gimenez, R. Rossi, and J Marti. A fast and accurate method to solve the incompressible Navier-Stokes equations. Engineering Computations, 30(2):197–222, 2013.