

# Development of a computer simulation model for blowing glass containers

**Citation for published version (APA):**

Giannopapa, C. G. (2006). *Development of a computer simulation model for blowing glass containers*. (CASA-report; Vol. 0607). Technische Universiteit Eindhoven.

**Document status and date:**

Published: 01/01/2006

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# DEVELOPMENT OF A COMPUTER SIMULATION MODEL FOR BLOWING GLASS CONTAINERS

**C. G. Giannopapa**

Dept. of Mathematics and Computer Science  
Technische Universiteit Eindhoven  
PO Box 513, 5600 MB Eindhoven  
The Netherlands  
Email: c.g.giannopapa@tue.nl

## ABSTRACT

In glass container manufacturing (e.g. production of glass bottles and jars) an important process step is the blowing of the final product. This process is fast and is characterized by large deformations and the interaction of a hot glass fluid that gets into contact with a colder metal, the mould. The objective of this paper is to create a robust finite element model to be used for industrial purposes that accurately captures the blowing step of glass containers. The model should be able to correctly represent the flow of glass and the energy exchange during the process. For tracking the geometry of the deforming inner and outer interface of glass, level set technique is applied on structured and unstructured fixed mesh. At each time step the coupled problem of flow and energy exchange is solved by the model. Here the flow problem is only solved for the domain enclosed by the mould, whereas in the energy calculations, the mould domain is also taken into account in the computations. For all the calculations the material parameters (like viscosity) are based on the glass position, i.e. the position of the level sets. The velocity distribution, as found from this solution procedure, is then used to update the two level sets by means of solving a convection equation. A re-initialization algorithm is applied after each time step in order to let the level sets re-attain the property of being a signed distance function. The model is validated by several examples focusing on both the overall behavior (such as conservation of mass and energy) and the local behavior of the flow (such as glass-mould contact) and temperature distributions for different mesh size, time step, level set settings and material parameters.

## NOMENCLATURE

$A$	-	Lakatos coefficients
$B$	$K$	Lakatos coefficients
$c_p$	$J/kgK$	specific heat of glass

$f$	$m/s^2$	body force (gravitation)
$k_c$	$W/mK$	conductivity of glass
$p$	$Pa$	pressure
$q$	$W/m^2$	heat flux
$t$	$sec$	time
$T$	$K$	glass temperature
$T_L$	$K$	Lakatos coefficients
$U$	$m/sec$	velocity
$\eta$	$m^2/s$	dynamic viscosity
$\rho$	$kg/m^3$	density
$\sigma$	$Pa$	Cauchy stress tensor

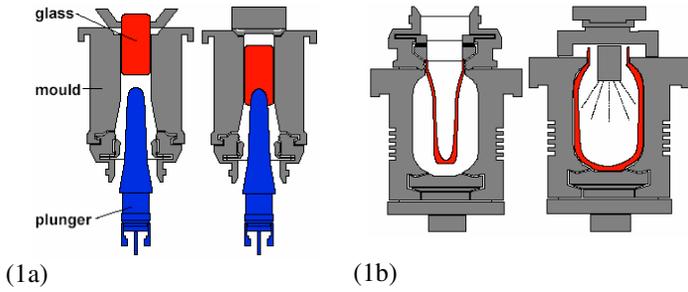
## INTRODUCTION

Glass is a material used in many products such as windows, bottles, art articles, etc, which all have their own unique manufacturing process. In the past, glass technology has been a craft mainly based on empirical knowledge and hands on experience. Over the last twenty years there has been significant progress in understanding and modeling the glass manufacturing process [1,2].

This paper concentrates on modeling the blowing stage of the forming process of glass containers. In particular it is interested in the accurate representation of the glass-air interface during blowing [3].

For glass containers manufacturing, the forming process is typically comprised of two stages: a pressing stage (Figure 1a) and a blowing stage (Figure 1b). As the glass flows down entering the forming machinery it is cut into gobbs. At the start of the process the pressing mould is open from above and a gob of molten glass falls from the gobbing device into the mould then the plunger is pushed up by a piston (Figure 1a). This mould usually produces an intermediate form, called the preform or parison. This preform is then transferred into a

second mould (Figure 1b), the blowing mould. There the preform sags due to gravity sufficiently far without touching the mould. Finally pressurized air is used to inflate the preform to the final container shape.



**Figure 1: Press-blow process**

In spite of the fact that the glass manufacturing process has been used for many years, glass manufacturers still experience difficulties in optimizing and controlling this process. Most glass forming is carried out in closed constructions involving high temperatures. It is difficult to measure and control in those ranges. Computer simulations would be very helpful to gain a better insight and assist in designing optimal bottle shapes. Simulation could offer a good alternative to time consuming and expensive trial and error procedures common to factories. Representative numerical simulations could help minimize unwanted variations in wall thickness of containers and reduce their weight while maintaining the strength. It can also help optimize cooling conditions and increase the production speed. All this could result in decreasing manufacturing cost.

In correctly modeling the blow stage of the forming process one has to accurately capture the glass-air interface. There are a number of numerical techniques that advocate for solving moving interface problems. They can be classified into two main categories: *interface-tracking techniques (ITT)* and *interface-capturing techniques (ICT)*.

### Interface-tracking techniques

The interface tracking techniques (ITT) are based on the Deforming-Spatial-Domain/Stabilized Space-Time formulation, where the mesh is updated as the flow evolves in order to track the interface. The major challenge of this method is the mesh update. They focus on how the mesh should be updated to accommodate for the changes in the spatial domain depends on several factors such as the complexity of the interface and the overall geometry, how unsteady the interface is and how the initial mesh was generated. Mesh updating can become a very computationally expensive matter especially when the number of elements becomes very large or when the frequency of re-meshing has to be high. In the mesh moving strategies, the only rule in the mesh motion one needs to follow is that at the interface the normal velocity of the mesh has to match the normal velocity of the fluid. The strongest motivation in mesh updating techniques is to minimize the frequency of re-meshing.

### Interface-capturing techniques

The interface-capturing techniques (ICT) function on fixed meshes. These methods are based on formulations, of both the flow equations and the advection equation governing the time-

evolution of an interface function. The interface function is the one that marks the location of the interface. In these methods the major concern is the accurate representation of the interface. This involves the accurate representation of the interface geometry as well as the representation of the flow near the interface. The interface is captured within the resolution of the mesh used. The interface capturing techniques use a massless particle or an indicative function to mark the continuum on either side of the interface. It is based on an Eulerian approach where the mesh once generated is maintained and gives the flexibility to apply it in complicated topologies; thus, it is very attractive for this project. The main interface capturing techniques are the Volume-of-Fluid method [3] and Level Set Method [4,5].

As this project is driven by industry, computational time is very essential. The interface tracking techniques due to the computational overhead related with the re-meshing are not attractive as for this application as the interface capturing techniques. The Level Set Method has been chosen for the interface tracking in this project and a detailed description of the method will be presented in the relevant sections.

The objective of this paper is to create a robust computational model to be used for industrial purposes which can accurately captures the blowing step of glass containers. The finite element model should be able to correctly represent the free surface flow of glass with high temperature dependent viscosity and the energy exchange during the process. The models should be able to take as an input information a glass perform shape, a temperature distribution for the perform and the mould and a prescribed inlet air pressure. It should output the containers final shape, the products final wall thickness as well as the stress and thermal deformations the pre-form and the mold undergo during the process.

## **MATHEMATICAL FORMULATION**

This section is concerned with presenting the mathematical model used to describe the blowing stage of the production process. This involves solving the motion of the glass, the heat exchange in the glass and the position of the glass-air interface.

### Flow problem

The motion of the hot glass can be described by the Navier-Stokes equations for fluids.

*Momentum equation:*

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f} \quad (1)$$

*Continuity equation or mass conservation:*

$$\nabla \cdot \mathbf{U} = 0. \quad (2)$$

For Newtonian fluids the constitutive equation for the stress tensor,  $\boldsymbol{\sigma}$  is given by:

$$\boldsymbol{\sigma} = 2\eta \text{dev}(\dot{\boldsymbol{\varepsilon}}) - p \mathbf{I}, \quad (3)$$

where  $\boldsymbol{\varepsilon}$  is the strain tensor,  $\dot{\boldsymbol{\varepsilon}}$  is the strain rate tensor and  $\text{dev}(\cdot)$  is the deviatoric part of the a tensor. The dynamic viscosity depends on the position

$$\eta = \eta(x). \quad (4)$$

Substituting (10) into (1) the momentum equation can be rewritten as:

$$\rho \frac{\partial \mathbf{U}}{\partial t} = \nabla \cdot (\eta \nabla \mathbf{U}) - \rho \nabla \cdot (\mathbf{U}\mathbf{U}) - \nabla p + \rho \mathbf{f} \quad (5)$$

In order to further analyze the problem quantitatively equation (5) is made dimensionless. Let us consider a typical: velocity  $\mathbf{U}_0$ , length scale  $L_0$ , viscosity  $\eta_0$ , density  $\rho_0$ , and pressure  $p_0$ . The characteristic pressure for the glass domain can be derived from:

$$p_0 = \eta_0 \mathbf{U}_0 / L_0, \quad (6)$$

and for the air domain by:

$$p_0 = \rho_0 \mathbf{U}_0^2. \quad (7)$$

Defining dimensionless time, length, velocity and pressure:

$$t^* = \frac{\mathbf{U}_0 t}{L_0}, \quad x^* = \frac{x}{L_0}, \quad \mathbf{U}^* = \frac{\mathbf{U}}{L}, \quad p^* = \frac{p}{p_0}. \quad (8)$$

Define dimensionless gravitational body forces.

$$\mathbf{f}^* = \frac{\mathbf{f}}{\|\mathbf{f}\|}. \quad (9)$$

The dimensionless form of the momentum (Equation 5) and the continuity equation (Equation 2) are:

$$\rho^* \frac{\partial \mathbf{U}^*}{\partial t^*} = \frac{1}{\text{Re}} \nabla \cdot (\eta^* \nabla \mathbf{U}^*) - \rho^* \nabla \cdot (\mathbf{U}^* \mathbf{U}^*) - \frac{p_0}{\rho_0 \mathbf{U}_0} \nabla p^* + \frac{1}{\text{Fr}_2} \rho^* \mathbf{f}^*, \quad (10)$$

$$\nabla \cdot \mathbf{U}^* = 0. \quad (11)$$

Here the relation between the internal and viscous forces is described by the Reynolds number  $\text{Re} = \rho_0 \mathbf{U}_0 / \eta_0$  and the ratio of the internal force to the gravitational force is give by the Froude number  $\text{Fr}_2 = \mathbf{U}_0^2 / \mathbf{f} L_0$ .

### Heat exchange

The heat flow can be described by the energy equation and for an incompressible fluid in an Eulerian description it reads:

$$\rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{U} \cdot \nabla T \right) = -\nabla \cdot \mathbf{q}. \quad (12)$$

Here  $c_p$  is the specific heat of the glass,  $\mathbf{q}$  is the heat flux and  $\phi$  is the source term from the internal heat generation by viscous and volume forces. Neglecting the heat transfer due to radiation and assuming that the conduction obeys the Fourier law which relates the heat flux vector to the temperature  $T$  by thermal conductivity  $k$  as

$$\mathbf{q} = -k \nabla T. \quad (13)$$

Thus, the energy equation can be written as:

$$\rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{U} \cdot \nabla T \right) = k \nabla^2 T. \quad (14)$$

Define dimensionless viscosity and density:

$$\eta^* = \frac{\eta}{\eta_0}, \quad \rho^* = \frac{\rho}{\rho_0}. \quad (15)$$

Define dimensionless temperature:

$$T^* = \frac{T - T_0}{\Delta T}, \quad (16)$$

where  $T_0$  is the temperature at the mould and  $\Delta T = T_g - T_0$  is the temperature jump between the glass and the mould.

The dimensionless form of the energy equation (14) reads:

$$\text{Pe} \rho^* c_p^* \left( \frac{\partial T^*}{\partial t^*} + \mathbf{U}^* \cdot \nabla \mathbf{U}^* \right) = k^* \nabla^2 T^*, \quad (17)$$

where the forced convection of a system is related to its heat conduction by the Péclet number  $\text{Pe} = \rho_0 c_p L_0 \mathbf{U}_0 / k_0$ .

It should be mentioned that the viscosity  $\eta$  which typically represents the fluids' resistance to the flow. In the glass the viscosity strongly depends on the temperature. As the temperature rises the viscosity decreases. Its values can be obtained by the so-called *Vogel-Fulcher-Tammann (VFT) relation* [1]:

$$\eta(T) = 10^{-A+B/(T-T_L)} \quad (18)$$

where  $A, B, T_L$  are the Lakatos coefficients and are empirically determined depending on the type of the glass.

### Glass-air interface position – Level set method-

The problems this paper intends to model involves moving interfaces. The *level set method* has been chosen to capture the moving interface between the two continua, glass and air [4,5]. This method allows to model flows with moving interfaces using a fixed spatial domain. Two continua A and B with different densities and viscosities are considered.

The basic idea is to embed a moving interface as the zero level set of the interface function  $\phi$ , which serves as a marker identifying the interface between the two continua. The evolution of the zero level set, thus the motion of the interface, is governed by the convention equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{U} \cdot \nabla \phi = 0. \quad (19)$$

So, at any time  $t$  the corresponding interface  $\Gamma(t)$  is given implicitly by  $\phi(x,t)=0$ . In this project, for the position of the glass-air interface two levels sets are used one describing the inner interface of the perform and the other the outer one.

### DISCRETISATION METHOD

The finite element (FE) method has been used for the discretization of the Stokes problem, the level set functions and the energy equation. The finite-element formulation has been implemented in SEPRAN finite element package. The mini element is used for the computations [8].

The resulting equations for the Stokes problem are solved using the BI-CGSTAB method [14]. This iteration method terminates after a finite number of iteration steps. As a stepping criterion the absolute value of the residual is less than  $10^{-8}$  is uses. As an initial guess for the iteration process the solution obtained in the previous time step is used.

The stabilized formulation of the of the energy equation is obtained using the streamline-upwind Petrov-Galerkin (SUPG) method [9,15]. The essence in this method is that next to the standard Galerkin equation an extra term of the following type is added

$$\int (Dc - f) p d\Omega. \quad (20)$$

Here  $e$  is the element,  $Dc$  represent the differential equation applied to  $c$  and  $f$  is the right hand side and  $\Omega$  is the computational domain. The upwind parameter  $p$  is defined by

$$p = \frac{h\xi}{2} \frac{\mathbf{U} \cdot \nabla \phi_i}{\|\mathbf{U}\|}. \quad (21)$$

Here  $h$  is the width of the element in the direction of the flow,  $\phi_i$  is the  $i^{\text{th}}$  basis function,  $\xi$  is the choice of parameter defining the unwinding. In the classical case  $\xi=1$  and in the modified case we have [16]:

$$p = \left( \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2\|\mathbf{U}\|}{h} \right)^2 + \left( \frac{4\mathbf{U}^T \alpha \mathbf{U}}{h} \right)^2 \right)^{-1/2} \mathbf{U} \cdot \nabla \phi, \quad (22)$$

where  $\alpha$  is a matrix. The BI-CGSTAB method [14] is used to solve of the equations up to an accuracy of  $10^{-6}$ . As before, the values obtained from the previous time step are used as an initial guess. For the level set solution the accuracy used is  $10^{-8}$ .

## RE-INITIALISATION OF THE LEVEL SET FUNCTION

The Level Set Method is one of the methods used for capturing the two-continua flow interface. It is adequate to compute geometric properties of highly complicated boundaries without necessarily explicitly tracking the interface. So the moving interface can undergo topological changes i.e. develop corners quite naturally.

One of the difficulties encountered in this method is to set the level set function keeping the desired shape. When the velocity in the domain is not constant the function itself can deform such that it introduces additional numerical problems. There are two solutions to this problem: (a) To adapt the velocity field, this procedure of extension velocities, is mainly used for problems where the velocity is only know at the interface. (b) To re-initialize the level set function; in this case the velocity field is unchanged.

One of the drawbacks of most existing re-initialization techniques is that they are based on a structured grid. The one that was developed for this project and is presented in this paper is based on an unstructured grid.

The level set function  $\phi$  is initially a Euclidean distance function to the corresponding interface  $\Gamma(t)$  where

$$\|\nabla \phi(\mathbf{x}, 0)\| = 1. \quad (23)$$

While the interface is convected by the interface velocity according to Equation (19), the gradient of the function  $\phi$  is not equal to 1 and  $\phi$  may become irregular after some time. The initially well shaped level set function can develop steep gradients on one side and can become almost constant on the other side of the domain. Thus, it is desirable to compute a function  $d(\mathbf{x})$ , that has the following property while the zero level of the distance function coincides with the interface

$$\|\nabla d(\mathbf{x})\|_2 \leq 1. \quad (24)$$

A numerical re-initialization procedure is developed and implemented to correct the level set function. The re-initialization in this paper is based on fast marching methods and is adopted for computations in unstructured meshes i.e. triangles and tetrahedrons.

## Description of the fast marching method

The fast marching methods [11,12] solve the following boundary value equations in the computational domain:

$$\|\nabla d(\mathbf{x})\|_2 = 1, \quad (25)$$

$$d(x) = 0 \quad \text{on } \Gamma. \quad (26)$$

The fast marching level set method is based on the idea that the surface front is propagating outward with a velocity in an upwinding fashion. The interface is carried along the computational domain by considering a narrow band around it and marching along the grid points. During this propagation some of the values of the grid points that start becoming further away from the narrow band around the interface become irrelevant and are ‘frozen’ and new ones that are coming closer are included.

Figure 2 represents the algorithmic representation of the method where a two dimensional grid is considered with the dashed line representing the location of the interface (Figure 2a). The distance between two grid points is  $\Delta x$ . The algorithm is initialized [13] by computing the  $d$  at grid points close to the interface  $\Gamma$ . These immediate points around the interface are tagged as ‘proximate’ and in Figure 2b are color marked as grey. All neighbor points to the ‘proximate’ points have know values and are tagged as ‘known’ and color marked in Figure 2c as white. Points located further than the ‘known’ points marked as ‘far’ and are not of interest. The ones that are in the upfront of the interface will become of interest as the interface propagates. The points that are candidates to be added to the set of ‘known’ points or to the set of ‘far’ points are the ‘trial’ points. These points have a trial value  $d$  assigned to them that might not yet be the correct one.

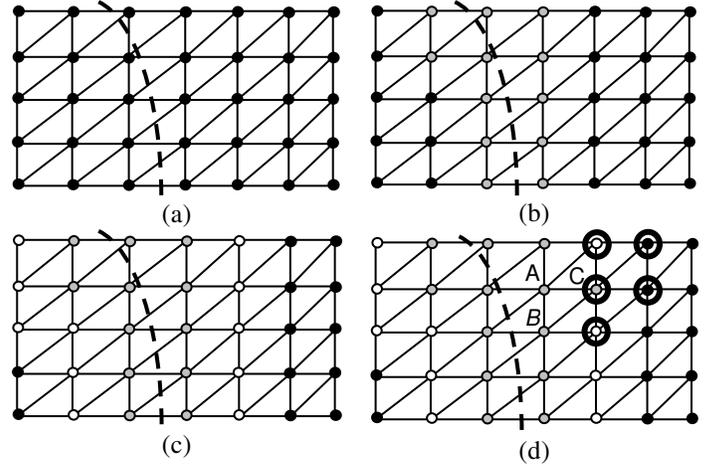


Figure 2: Fast marching methods.

Point C in Figure 2d with two known neighbors A and B is considered. The trial value at this point considering that  $d_C \geq d_A$ ,  $d_B$  is the solution of the quadratic equation:

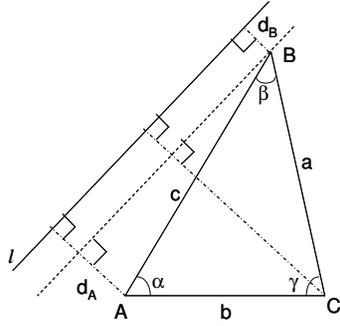
$$(d_C - d_A)^2 + (d_C - d_B)^2 = \Delta x^2. \quad (27)$$

After having assigned values  $d$  to all the ‘trial’ points, the points with the smallest value are the ones with the correct value. If this is the case for the point C it can be removed from the ‘trial’ set of points and be added to the ‘known’ one.

Thus, the algorithm can further proceed with the neighbors of point C denoted in Figure 2d as the big circles. In this case the ‘far’ points are transferred to the ‘trial’ point set and trial values of  $d$  are computed.

### Fast marching method on unstructured grid

For the description of the re-initialization method developed and used in the project described Figure 3 is considered.



**Figure 3: Fast marching methods re-initialization.**

A triangle ABC with angles  $\alpha, \beta, \gamma$  is considered and a line  $l$  is drawn with distance  $d_A$  from point A and distance  $d_B$  from point B, assuming that  $d_B \leq d_A$ . Angle  $\delta$  is can be computed by  $\sin(\delta) = (d_A - d_B)/c$  and  $\sin(\delta + \beta) = (d_C - d_B)/a$ . Thus, the distance  $d_C$  of point C from line  $l$  is give by:

$$d_C = \alpha \sin(\delta + \beta) + d_B. \quad (28)$$

In order to make sure that the shortest distance to the line  $l$  from C passes through the middle of the triangle the following requirement has to be fulfilled:  $0 \leq \alpha \cos(\delta + \beta) / \cos(\delta) \leq c$ . If it is not satisfied then the update is performed by taking  $d_C = \min\{d_A + b, d_B + a\}$ . It is also possible to consider triangles with only one value known. Let us assume that only the value at A is known, then the values at B and C can be computed from the assigned distance values using

$$d_B = d_A + c, \quad (29)$$

$$d_C = d_A + b. \quad (30)$$

It should be mentioned that from Equation (28) the exact distances to line  $l$  from point C can be calculated, whereas in practice the line  $l$  is just a segment of finite length and  $d_A$  and  $d_B$  are distance approximations. Thus, distance  $d_C$  is also a distance approximation.

### COMPUTER SIMULATION MODEL

The models concerned for the simulations are a quarter of a bottle in a horizontal direction and half a bottle in the length direction of real dimensions of a beer bottle. The quarter circle was chosen as the prime test case because it has known analytical solution, and thus can be used for the computational model developed.

The typical values for glass and air used for the simulations are:  $L_0 = 10^{-2} \text{m}$ ,  $U_0 = 10^{-2} \text{m/sec}$ ,  $\eta_{0(\text{glass})} = 10^3 \text{Pa s}$ ,  $\eta_{0(\text{air})} = 10^{-5} \text{Pa s}$ ,  $\rho_{0(\text{glass})} = 2.5 \cdot 10^{-3} \text{kg/m}^3$ ,  $\rho_{0(\text{air})} = 2.5 \cdot 10^{-3} \text{kg/m}^3$ ,  $p_0 = 10^3 \text{Pa}$ ,  $\Delta T = 350^\circ \text{C}$ ,  $c_{p(\text{glass})} = 1.2 \cdot 10^3 \text{ J/(kg K)}$ ,  $c_{p(\text{air})} = 10^3 \text{ J/(kg K)}$ ,  $k_{\text{glass}} = 2.75 \text{ W/(m K)}$ ,  $k_{\text{air}} = 10^{-2} \text{ W/(mK)}$ ,  $Re_{\text{glass}} = 2.5 \cdot 10^{-4}$ ,  $Re_{\text{air}} = 10$ ,  $Fr = 10^{-3}$ ,  $Pe_{\text{glass}} = 1.1 \cdot 10^2$ ,  $Pe_{\text{air}} = 10$ .

The momentum equation (Equation 10) can be simplified based on the following arguments. The Reynolds number in the glass domain is low and  $Re < Fr \ll 1$ ; thus the problem is dominated by viscous and gravitational forces. For the

simulations the air is replaced by a fictitious fluid with viscosity 4 Pa and the same density as the air. So, the viscosity of the fictitious fluid remains a lot smaller than that of the glass. From the dimensionless numbers one can conclude that the inertia terms in the fictitious fluid domain can be neglected, while the pressure drop is still negligible compared to the pressure drop in the glass domain. Thus, the system of equations (momentum, continuity, heat) used for the computations in the while domain are (\* is omitted):

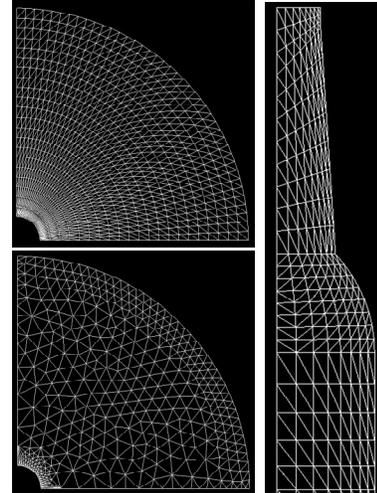
$$\nabla \cdot (\eta \nabla \mathbf{U}) + \rho \frac{Re_{\text{glass}}}{Fr} \mathbf{f} - \nabla p = 0, \quad (31)$$

$$\nabla \cdot \mathbf{U} = 0, \quad (32)$$

$$Pe \rho c_p \left( \frac{\partial T}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} \right) = k \nabla^2 T. \quad (33)$$

Two level set functions were used to capture the moving interface; one for the outer interface and one for the inner interface. Simulations were performed using both structured and unstructured meshes. Graphs of the finite element mesh can be found in Figure 4.

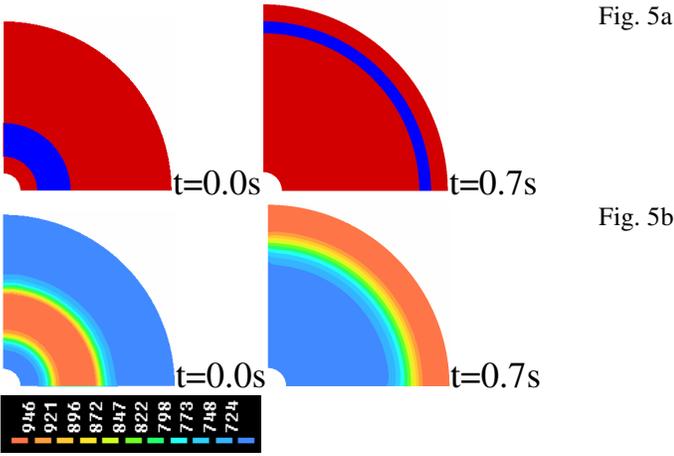
An inflow pressure is prescribed at the boundary and a free-stress condition applies to the air at the walls of the mould. In the quarter circle the glass can freely move outside the computational domain whereas in the case of the half bottle no-slip condition applies at the mould walls.



**Figure 4: Finite element mesh.**

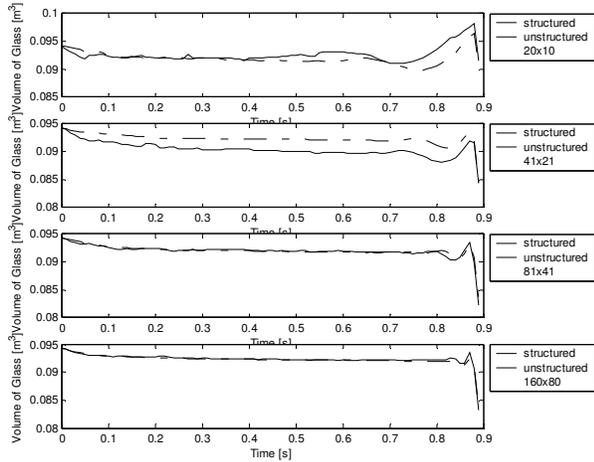
### RESULTS

The simulations considered are: a quarter bottle in horizontal direction and half a bottle in the length direction. The quarter of a bottle is used for the investigation of the accuracy and robustness of the method. In the quarter circles and initial velocity field is prescribed at the inflow and a stripe of glass is propagating outwards. In this simulation when the glass reaches the computational domain boundaries it can freely move outside the domain. A temperature profile was also calculated. Figure 5a, 5b the glass air interface propagation and the temperature profiles can be seen.



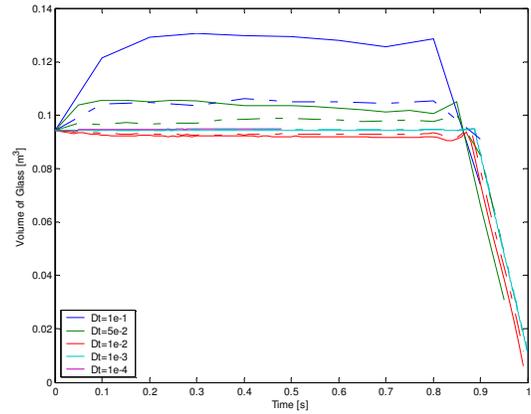
**Figure 5: Quarter circles glass blowing with even temperature distribution at inflow. (a) Glass area propagation: glass area propagation is denoted with blue and air domain with red. (b) Temperature profiles.**

In order to be able to properly assess the accuracy of the simulation model presented in this paper, the glass volume conservation is used. The quarter bottle is used and computations are performed with structured and unstructured mesh. Computations were performed using different mesh sizes 20x10, 41x21, 51x41 and 160x80. Figure 6 shows that the that for resolutions higher than 51x41 the results of the structure and unstructured mesh have about the same accuracy. In this computations the Euler Implicit scheme was used and the time step was 1e-2.



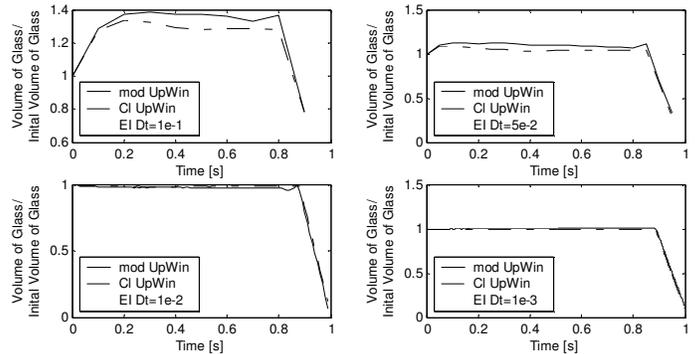
**Figure 6: Comparison of volume conservation for structured and unstructured grids with mesh sizes 20x10, 41x21, 81x41, 160x80 in the circular example.**

Euler implicit and Crank Nicholson computations were compared for various time steps: 1e-1, 5e-2, 1e-2, 1e-3, 1e-4. The mesh used was 81x41. The results from the computations in Figure 7 show that for larger time steps the Crank Nicholson method is more accurate than the Euler Implicit. That is expected as the Crank Nicholson method is second order accurate whereas the Euler Implicit is first order. For small time steps the results are very close to each other.



**Figure 7: Comparison of volume conservation between Euler implicit and Crank Nicholson for different time steps: 1e-1, 5e-2, 1e-2, 1e-3, 1e-4.**

Two different upwind schemes were compared: the classical upwind (Equation 21) and the modified upwind (Equation 22). The time step used was 1e-2 and the mesh grid 81x41. The results are presented in Figure 8. The classical upwind scheme gives more accurate results.



**Figure 8: Comparison of volume conservation between classical upwind scheme and modified upwind scheme in the circular example.**

It is common practice in glass manufacturing that there is a temperature variation in the glass perform, which results in wall thickness variations. For this reason a temperature gradient was implemented at the inflow. In Figure 9 the results are presented. It can be seen that at the side where a higher temperature was imposed, this side became thinner during the evolution.

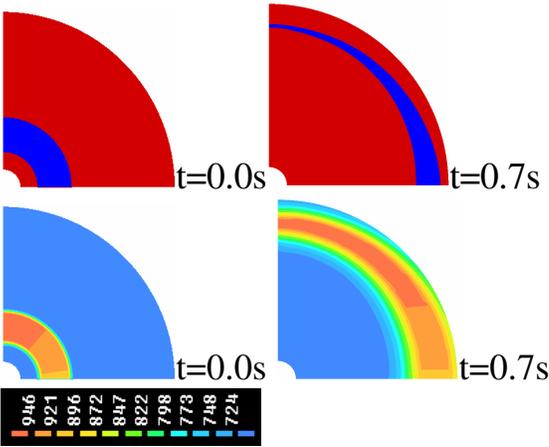


Fig. 9a

Fig. 9b

**Figure 9: Quarter circles glass blowing with temperature gradient at inflow. (a) Glass area propagation: glass area propagation is denoted with blue and air domain with red. (b) Temperature profiles.**

Finally, the two dimensional model for an unstructured grid was implemented in realistic beer bottle simulations. A pressure profile is imposed at the top inlet of the mold. The results are presented in Figure 9.

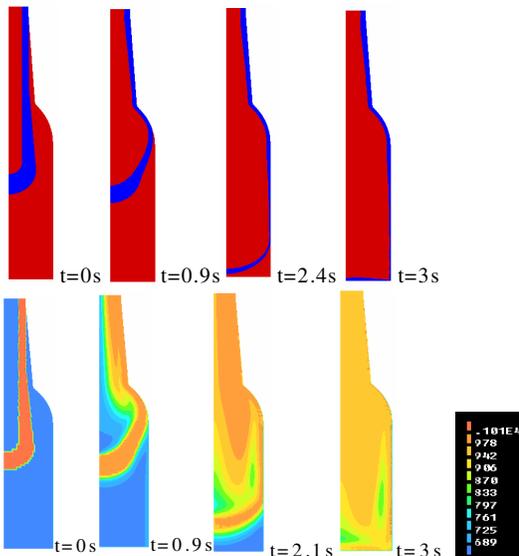


Fig. 9a

Fig. 9b

**Figure 9: Two dimensional glass blowing (a) Glass area propagation: glass area propagation is denoted with blue and air domain with red. (b) Temperature profiles.**

## CONCLUSIONS

In this paper a model was derived and implemented for the blowing stage of glass containers. The fluid flow is described by the Stokes equations and the temperature by the energy equations. The glass position is modeled using two level set function representing the inner and outer glass-air interface.

The method was implemented for structured and unstructured grids giving the flexibility to manufacturers to use it for any form and shape the end product might be. For

accurate computations a re-initialization technique for unstructured grids was developed and implemented.

A parametric analysis was performed using different grid resolution, time steps and different discretisation methods. The results showed that the method was robust and adequately accurate. The mass loss of glass during the simulations was less than 1%, which give confidence that the two dimensional model can be used for industrial blowing simulations. The next step of development is to implement the model in 3D and extend the re-initialization technique to 3D. Comparison between the model and experiment would be a future consideration.

## ACKNOWLEDGMENTS

The author would like to thank Marie Currie Research Training Networks Fellowships, European Commission for funding this project.

## REFERENCES

- [1] Tooley, F. V., 1984, "The handbook of glass manufacture", Vol II, Aslee Publishing Co.. New York.
- [2] Loch, H. , Krause, D. , 2002, "Mathematical simulation in glass technology", Springer-Verlag Berlin and Heidelberg GmbH & Co. K.
- [3] Hirth, C.W., Nichols, B.D., 1981, "Volume of fluid (VOF) method for the dynamics of free boundaries", *J. Comp. Phys.*, **39**, 201-225.
- [4] Sethian, J.A, 1999, "Level Set Methods and Fast Marching Methods", Cambridge university press, USA.
- [5] Sussman, M., Smereka, P., Osher, S.J., 1994, "A level set method for computing solutions to incompressible two-phase flow", *J. Comp. Phys.*, **114**, 146-159.
- [6] Cuvelier, C., Segal, A., van Steenhoven, A., 1986, "Finite Element Methods and Navier Stokes Equations, Mathematics and its applications", D. Reidel Publishing Co., The Netherlands.
- [7] Gunzberger, M., 1989, "Finite element methods for viscous incompressible flows", *Computer Science and Scientific Computing*, Academic Press.
- [8] Bathe, K. J., 1997, "Finite element procedures", Prentice Hall, Englewood Cliffs, New Jersey.
- [9] Johnson, C., Nävert, U., Pitkäranta, J., 1984, "Finite element methods for linear hyperbolic problems", *Computer methods in applied mechanics and engineering*, **45**, 285-312.
- [10] L. A. Hageman and D. Young, 1981, "Applied Iterative Methods", 1st ed. New York: Academic.
- [11] Sethian, J.A., 1996, "A marching level set method for monotonically advancing fronts", *Proc. Nat. Acad. Sci.*, **93**, 1591-1595.

- [12] Sethian, J.A., 1999, "Fast marching methods", *SIAM review*, **41(2)**, 199-235.
- [13] Chopp, D.L., 2001, "Some improvements of the fast marching method", *SIAM Journal of Scientific Computing*, **23(1)**, 230-244.
- [14] van der Vorst., 1992, "BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems", *SIAM J. Sci. Stat. Comput*, **13(2)**, 631-644.
- [15] Brooks, A.N., Hughes, T.J.R., 1982, "Stream-line upwind/Petrov-Galerking formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", *Comput. Methods Appl. Mech. Eng.*, **32**, 199-259.
- [16] Shakib, F., 1989, "Finite element analysis of the compressible Euler and Navier-Stokes equation", PhD thesis, Stanford University.