

Adaptive BEM Simulation for the Formation of Parisons

K. Wang, R.M.M. Mattheij

*Department of Mathematics and Computing Science,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
EMail: wang@win.tue.nl*

Abstract

In manufacturing bottles or jars, the motion of the glass between the *mould* and the *plunger* can be modelled as an axi-symmetric, time-dependent Stokes flow with free and “moving” boundaries. The Stokes flow can be reduced to a 1- D boundary integral equation (BIE) due to its axi-symmetry. In this paper, we present a h -adaptive boundary element method (BEM) to solve this BIE. The time integration is carried out by a sort of predictor-corrector scheme. Numerical Results illustrate the efficiency and accuracy of the method.

1 Mathematical Model

In manufacturing bottles or jars, several processes are known, one of them being *pressing/blowing*, the pressing phase is schematically shown in Figure 1. A certain amount of glass falls into a configuration consisting of a so-called *mould* and *plunger*. As soon as the entire glass drop falls into the mould, the plunger starts moving. The phase in which the plunger is moving, is referred to as the pressing phase. The intermediate product after the pressing phase is called *parison*.

The glass can be considered an incompressible Newtonian fluid. The flow of the glass in the pressing phase is now governed by the

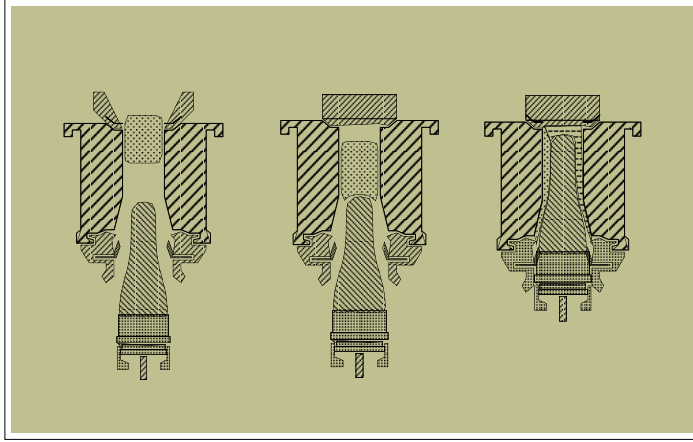


Figure 1: Pressing phase

following Navier–Stokes equations which describe the *conservation of mass* and *conservation of momentum*

$$\begin{cases} \nabla \cdot \mathbf{v} = 0, \\ \rho(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v}) = \eta \Delta \mathbf{v} - \nabla p + \rho \mathbf{g}. \end{cases} \quad (1)$$

where ρ is the density of the fluid, $\mathbf{g} = (g_i)$ is the body force (here only gravitational force is considered), $\mathbf{v} = (v_i)$ is the velocity vector and p is the pressure.

After dimension analysis (Laevsky at el.[5]), it turns out that the *Reynolds number* and the quotient of the Reynolds number and the *Froude number* are both so small that we effectively end up with the *Stokes equations*

$$\begin{cases} \nabla \cdot \mathbf{v} = 0, \\ \Delta \mathbf{v} - \nabla p = 0. \end{cases} \quad (2)$$

In order to find a unique velocity \mathbf{v} and a pressure p , a set of boundary conditions have to be imposed (see Figure 2, note that we have turned over the geometry by 180° for convenience). As in Laevsky at el.[5], the boundary condition of the free boundary is given by

$$\sigma \mathbf{n} = -p_0 \mathbf{n}. \quad (3)$$

where $\mathbf{n} = (n_i)$ is the outward unit normal, $\sigma = -p\mathbf{I} + (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)$ is the stress tensor. Boundary conditions on the mould and the plunger

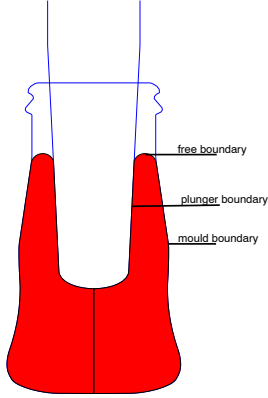


Figure 2: The Boundaries

depend on the roughness of the mould and the plunger. In general we might impose the following boundary conditions

$$\begin{cases} \mathbf{v} \cdot \mathbf{n} = \mathbf{v}_b \cdot \mathbf{n}, \\ \sigma \mathbf{n} \cdot \mathbf{t} = -\beta_b (\mathbf{v} - \mathbf{v}_b) \cdot \mathbf{t}. \end{cases}$$

where \mathbf{t} is the unit tangential direction, \mathbf{v}_b is the velocity of the boundary and β_b is the *slip parameter* indicating the amount of friction. The boundary conditions mean that the normal component of the velocity of the flow equals the normal component of the velocity of the boundary wall and the shear stress is proportional to the tangential velocity difference.

Applying the above boundary conditions to the mould and the plunger yields

$$\begin{cases} \mathbf{v} \cdot \mathbf{n} = 0 \\ \sigma \mathbf{n} \cdot \mathbf{t} = -\beta_m \mathbf{v} \cdot \mathbf{t} \end{cases} \text{ on the mould,} \quad (4)$$

and

$$\begin{cases} \mathbf{v} \cdot \mathbf{n} = \mathbf{v}_p \cdot \mathbf{n} \\ \sigma \mathbf{n} \cdot \mathbf{t} = -\beta_p (\mathbf{v} - \mathbf{v}_p) \cdot \mathbf{t} \end{cases} \text{ on the plunger.} \quad (5)$$

where \mathbf{v}_p is the velocity of the plunger.

Since at least some part of the boundary is moving, we use a quasi-static approach to describe the movement of a material fluid particle, i.e.

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}(t)). \quad (6)$$

Now our objective is to solve the problem (2) – (6). Although this type of problem can also be solved by FEM techniques, the particular interest for the displacement of the (free) boundary makes a BEM approach very attractive. In section 2 we develop an h -adaptive BEM scheme to solve the Stokes equation in an axi-symmetric domain based on residual computation. In section 3 we briefly discuss the time integration, finally some examples are given in section 4, to show the efficiency and accuracy of this method.

2 Adaptive BEM Strategy

As is well known one can define a boundary integral equation for an axi-symmetric Stokes flow (Brebbia et al.[1])

$$\mathbf{c}(\mathbf{x})\mathbf{v}(\mathbf{x}) + \int_{\Gamma} \mathbf{q}(\mathbf{x}, \mathbf{y})\mathbf{v}(\mathbf{y})d\Gamma_{\mathbf{y}} = \int_{\Gamma} \mathbf{u}(\mathbf{x}, \mathbf{y})\mathbf{b}(\mathbf{y})d\Gamma_{\mathbf{y}}. \quad (7)$$

where $\mathbf{v}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ are velocity and stress vector respectively.

This is solved by a BEM. Obviously the velocity field has very large gradients near the plunger if the parameter β_p is large; the stress vector changes rapidly around the corners (this has been verified by FEM results). This means we need a fine mesh on some part of the boundary, but on other parts of the boundary where the geometry and the variables are smooth enough, the discretization error is relatively small, so a fine mesh is not necessary on these parts of the boundary. We now like to introduce a self-adaptive strategy.

There exists ample literature. Kita et al.[4] gives a good review on this subject; Charafi et al.[2] proposes a new way based on the local reanalysis (LR); Paulina et al.[7] suggests that nodal sensitivities can be used as estimator or/and indicator; while Kamiya et al.[3] use the so-called *extended error* as the error indicator. Here we employ the error indicator of residual type (cf. Kita et al.[4]). For simplicity, linear elements are adopted in the remaining parts of this work.

Let $\hat{\mathbf{v}}$ and $\hat{\mathbf{b}}$ be the approximate solution obtained for the given boundary discretization. We substitute these into Eqn. (7) and define the residual $\mathbf{r}(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{b}})$ at the point \mathbf{x} as

$$\mathbf{r}(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{b}}) = \mathbf{c}(\mathbf{x})\hat{\mathbf{v}}(\mathbf{x}) + \int_{\Gamma} \mathbf{q}(\mathbf{x}, \mathbf{y})\hat{\mathbf{v}}(\mathbf{y})d\Gamma_{\mathbf{y}} - \int_{\Gamma} \mathbf{u}(\mathbf{x}, \mathbf{y})\hat{\mathbf{b}}(\mathbf{y})d\Gamma_{\mathbf{y}}. \quad (8)$$

Once $\hat{\mathbf{v}}$ and $\hat{\mathbf{b}}$ are obtained by the initial BEM simulation, the residual $\mathbf{r}(\mathbf{x}, \hat{\mathbf{v}}, \hat{\mathbf{b}})$ is computable at any point \mathbf{x} . Of course, the residual

vanishes at the initial collocation points, but does not vanish at the other points. The residuals at middle points of the elements are most likely give a good idea of the actual (local) error.

Thus, the Residual can be used as an error indicator. Indeed, from the St. Venant principle in elasticity (a Stokes problem can be considered as a particular elasticity problem) it is known that a local residual force generates only local influence on the displacement (cf. Rank[8]). Secondly, For some integral operators, a *pseudo-local property* can be defined which states that the influence of a local disturbance decays sufficiently fast, which reflects the decay property of Green’s function. Furthermore, in collocation BEM, the global error can be controlled by the residual, which is confirmed by numerical examples although it is difficult to mathematically prove in some cases (cf. Kita[4]).

We can use a very simple strategy: if the residual $\|\mathbf{r}^i\|$ is larger than a given reference tolerance r_{ref} , then the i -th element should be refined.

To effectively refine, we first have to decide how many points should be inserted into the element which needs to be refined. This is done using the convergence order of the residual. Specifically, in the case of linear elements, the convergence order is 2, we reasonably subdivide the element into $\sqrt{\|\mathbf{r}^i\|/r_{ref}}$ elements (approximately). In the meantime, if two adjacent elements both have residuals less than $r_{ref}/4$, then these two elements merge into one element.

Once we decide to insert k points into the i -th element the next problem is to locate the k points, the simplest way is to distribute them uniformly. But in some region where the solution changes rapidly, this method is not efficient enough. We should take into account the residual’s variation on the boundary so such that the grid distance reflects the residual’s variation. We will see that this tactics is much more efficient than the uniform distribution.

3 Time Integration

In this section, we discuss how the geometry of the glass is updated. At time t_0 , we can determine the velocity field \mathbf{v}^0 , which is used to determine the geometry of the glass in cylindrical coordinates at time $t_1 := t_0 + \Delta t$ from discretising the ordinary differential equation (6).

We rewrite this initial value problem as follows

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}(t)), \\ \mathbf{x}(t_0) = \mathbf{x}^0. \end{cases} \quad (9)$$

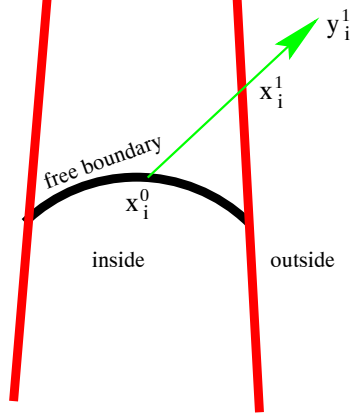


Figure 3: The clipping algorithm

We denote the area between the mould and the plunger at time t by \mathcal{A}_t , the boundary of the glass by \mathcal{B}_t and the domain of the glass by \mathcal{G}_t . Of course the glass body at time t should be in \mathcal{A}_t for all t . If the initial geometry of the glass, represented by its boundary points \mathbf{x}^0 , satisfies

$$\mathbf{x}_i^0 \in \mathcal{A}_{t_0}, \quad \text{for each } i.$$

the new geometry of the glass \mathbf{x}^1 at time $t_1 := t_0 + \Delta t$ has to satisfy the constraint

$$\mathbf{x}_i^1 \in \mathcal{A}_{t_1}, \quad \text{for each } i. \quad (10)$$

Suppose we employ, say, the Euler forward scheme to discretize the problem (9)

$$\mathbf{y}^1 = \mathbf{x}_0 + \Delta t \mathbf{v}_0.$$

In general, constraint (10) will not be satisfied. That means we need a strategy to reposition those points which are not in \mathcal{A}_{t_1} : if $\mathbf{y}_i^1 \in \mathcal{A}_{t_1}$, then no further action is needed; otherwise they are redefined to be at the intersection of the boundary of \mathcal{A}_{t_1} and the line which runs from \mathbf{x}_i^0 to \mathbf{y}_i^1 (see Figure 3). We refer to this repositioning step as the *clipping algorithm*. Applying the clipping algorithm to \mathbf{y}^1 we obtain the new geometry of the glass at time t_1 .

We'd like to use the midpoint rule due to its *symplecticness* (cf. Mattheij et al. [6]). For the initial value problem (9) the midpoint rule reads

$$\mathbf{x}^1 = \mathbf{x}^0 + \Delta t \mathbf{v}(\mathbf{x}(t_{\frac{1}{2}})). \quad (11)$$

where $t_{\frac{1}{2}} := t_0 + \frac{\Delta t}{2}$. But the information $\mathbf{v}(\mathbf{x}(t_{\frac{1}{2}}))$ is unknown. To approximate $\mathbf{v}(\mathbf{x}(t_{\frac{1}{2}}))$, we first use the Euler forward scheme with time step $\frac{\Delta t}{2}$ to compute $\mathbf{x}(t_{\frac{1}{2}})$, i.e.

$$\mathbf{x}(t_{\frac{1}{2}}) = \mathbf{x}^0 + \frac{\Delta t}{2} \mathbf{v}^0. \quad (12)$$

then $\mathbf{v}(\mathbf{x}(t_{\frac{1}{2}}))$ is achieved by solving a Stokes flow with a boundary represented by $\mathbf{x}(t_{\frac{1}{2}})$. In summary we have the following stages at each time step

1. Solve a Stokes problem on the domain \mathcal{G}_{t_0} to get velocity \mathbf{v}^0 at time t_0 .
2. Use the Euler forward scheme to predict the geometry of $\mathcal{B}_{t_{\frac{1}{2}}}$ at time $t_{\frac{1}{2}}$ numerically.
3. Solve a Stokes problem on the domain $\mathcal{G}_{t_{\frac{1}{2}}}$ to predict the velocity $\mathbf{v}^{\frac{1}{2}}$ at time $t_{\frac{1}{2}}$.
4. Use the midpoint rule to determine the new geometry of \mathcal{B}_{t_1} at time t_1 numerically.

Note that, when necessary the clipping algorithm should be used to get the boundary \mathcal{B}_t . Stages 3–4 may be used in an iterative procedure if desired.

4 Numerical Results

We investigate some cases to illustrate that the method really works. The solution to this problem is strongly related to the roughness of the plunger and the mould, i.e. the slip parameters β_p and β_m . If the slip parameters are very large then the boundary conditions can be considered as no-slip boundary conditions. In this case, we expect the velocity to have large gradients on the interfaces between the glass and the mould or the plunger, and thus the region

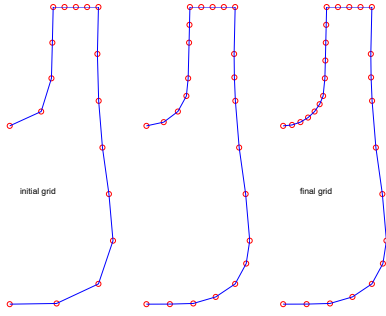


Figure 4: $\beta_m = \beta_p = 0.1, r_{ref} = 0.01$

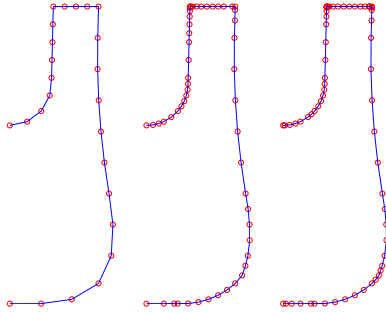


Figure 5: $\beta_m = 1000, \beta_p = 100, r_{ref} = 0.01$

including the interfaces should be refined. On the other hand, if the slip parameters are relatively small then the boundary conditions can be approximately considered as slip boundary conditions. This situation is more tractable since the velocity gradient is relatively small at the free boundaries.

case 1: *Small slip parameters* $\beta_m = \beta_p = 0.1$: The refinement procedure is displayed in Figure 4. In this case, the refinement just reflects the curvature of the geometry, i.e. the refinement procedure is trying to approximate the boundary more accurately.

case 2: *Large slip parameters* $\beta_m = 1000, \beta_p = 100$: The refinement procedure is displayed in Figure 5. In this case, large velocity gradients are expected in some regions, the refinement procedure tells us where these regions are. If we uniformly distribute the inserted points then we need more than 6 times refinement procedures, but now by using a grid which is determined by the residual's variation, we only need to refine 2 times. This is much more efficient.

The velocity fields on the refined grids are shown in Figure 6. The left one is the slip boundary condition, while the right one corresponding to the no-slip boundary condition.

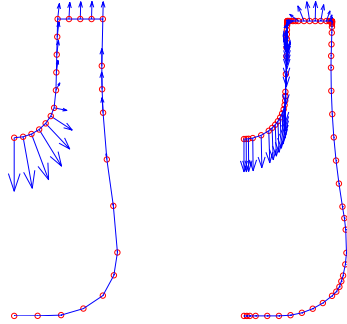


Figure 6: Velocity fields in two cases

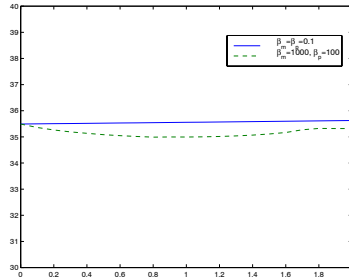


Figure 7: Mass as a function of time t

The refinement procedures coincide exactly with our expectation. To verify the time integration scheme, we compute the volume of the glass as a function of time t and display the volume in Figure 7. The volume should be conservative because of incompressibility. In case **1**, the mass increases by 0.38%; while in case **2**, the mass decreases by 0.49% in the whole pressing phase due to the clipping.

We'd like to compare the complexity of BEM with FEM. We may use the number of matrix elements (to be calculated) as a criterion. In case **2**, to get the same accuracy, a FEM (Laevsky at el.[5]) needs about 1500 triangular elements, while our BEM only has about 60 elements. This means that, to build the matrices the BEM should compute $60 \times 60 \times 16 = 57600$ *single* integrals while the FEM should compute at least $1500 \times (21 + 36) = 85500$ *double* integrals. We

find the BEM is cheaper. Furthermore, in the case of slip boundary condition, the BEM is much cheaper.

We conclude from the numerical results that the refinement tactics which takes the residual's variation into account is much more efficient; the BEM is cheaper than the FEM for this problem; the time marching scheme developed above works well.

References

- [1] Brebbia, C.A., Telles J.C.F. & Wrobel L.C., *Boundary Element Techniques*, Computational Mechanics Publications, Southampton, 1984.
- [2] Charafi, A. & Wrobel, L.C., *A New h-Adaptive Refinement Scheme for the Boundary Element Method Using Local Reanalysis*, App. Math. Comp., **82**, pp.239–271, 1997.
- [3] Kamiya, N. & kawaguchi, K., *Error analysis and adaptive refinement of boundary elements in elastic problem*, Adv. Eng. softw. **15**, pp. 223–230, 1992.
- [4] Kita, E. & Kamiya, N., *Recent studies on adaptive boundary element methods*, Adv. Eng. Softw., **19**, pp.21–32, 1994.
- [5] Laevsky, K. & R.M.M. Mattheij, *Development of a Simulation Model for Glass Glow in Bottle and Jar Manufacturing*, RANA, Eindhoven University of Technology, 1998.
- [6] Mattheij, R.M.M. & Molenaar, J., *Ordinary Differential Equations in Theory and Practice*, Wiley, 1996.
- [7] Paulino, G.H., Shi, F., Mukherjee, S. & Ramesh, P., *Nodal sensitivities as error estimates in computational mechanics*, Acta Mech., **121**, pp.191–213, 1997.
- [8] Rank, E., *Adaptive h-, p- and hp-Versions for Boundary Integral Element Methods*, Int. J. Numer. Methods Eng., **28**, pp. 1335–1349, 1989.
- [9] Vorst, G.A.L. van de, & Mattheij, R.M.M. *A BEM-BDF scheme for Curvature Driven Moving Stokes Flows*, J. Comput. Phys. **120**, pp.1–14, 1995.