ECI 8th International Conference on Boiling and Condensation Heat Transfer
Ecole Polytechnique Fédérale de Lausanne, 3-7 June 2012
Lausanne, Switzerland

3D ALE FINITE ELEMENT METHOD FOR TWO-PHASE FLOWS WITH PHASE CHANGE

Gustavo ANJOS*, Norberto MANGIAVACCHI, Navid BORHANI and John R. THOME
*Author for correspondence
Heat and Mass Transfer Laboratory (EPFL - STI - IGM – LTCM, ME Station 9)
Swiss Federal Institute of Technology - EPFL
Lausanne, 1015 - Switzerland
gustavo.rabello@epfl.ch, norberto.mangiavacchi@uerj.br, navid.borhani@epfl.ch, john.thome@epfl.ch

ABSTRACT

We seek to study numerically two-phase flow phenomena with phase change through the Finite Element Method (FEM) and the Arbitrary Lagrangian-Eulerian (ALE) framework. This method is based on the so-called 'one-fluid' formulation, thus only one set of equations is used to describe the flow field at the vapour and liquid phases. The equations are discretized on an unstructured tetrahedron mesh and the interface between the phases is defined by a triangular surface, which is a sub-set of the 3-dimensional mesh. The Navier-Stokes equation is used to model the fluid flow and an additional energy equation is used to model the heat transport between the phases. Such a methodology can be employed to study accurately many problems such as oil extraction and refinement in the petroleum area, design of refrigeration systems, modelling of biological systems and efficient cooling of electronics for computational purposes, which is the aim of this research. A comparison of the obtained numerical results to the analytical solutions is performed, thus proving the capability of the proposed new methodology.

INTRODUCTION

Today, most of the cooling devices found in personal computers and datacenters use either single-phase air or water-cooling systems. With microprocessor performance increasing exponentially, an efficient and better way to cool and decrease the computer chip temperature is of utmost importance. Furthermore, a substantial increase of the number of chips per motherboard plans to go to multi-layer stacks of chips with internal cooling channels, since higher computational resources are continuously required. It is known that the heat exchange of two-phase flow systems are much higher than those using single-phase flow, mainly due to the nature of the thermal behavior of each phase in presence of an interface layer separating both fluids. Therefore, a new cooling technique is proposed to maintain simultaneously the temperature of two or more stacked microprocessors, within an optimal working range, by flowing and evaporating two-phase environmentally friendly refrigerants in-between. These operating fluids are responsible for removing the excessive heat produced by the processors, however the cooling channels are limited to the order of 100 microns size.

Despite the available cutting-edge experimental techniques, a deeper insight into the microscale flow field is necessary. However, to access such a small length scale accurately, different techniques are required. In this context, numerical analysis has become an useful tool to simulate the mechanisms of two-phase flows, due to the fast growth of computer resources and the reduction of cost compared to those of experimental facilities. In fact, the modeling of such conditions is not an easy task due to the complexity of the non-linear set of equations that govern the flow field. Moreover, the characterization of surface tension forces and the interfacial deformation between the vapor and liquid phases adds another level of complexity, all of which require significant efforts to resolve in two-phase flow simulations.

The attempt to solve numerically two-phase flows splits the research in two main categories, namely 'one-fluid' and 'two-fluids' formulations. The former uses one set of equations to describe all the phases and it is assisted by a color function, which defines their regions in the domain. The latter describes each phase as a separate fluid with its own set of governing equations, thus requiring an additional mathematical formulation for their. Within the 'one-fluid' formulation, the interface description may be divided in two sub-areas, namely *Eulerian* and *Lagrangian* descriptions. The basic difference between both methodologies is reflected on the modeling of the interface between the phases. In the *Eulerian* description, the computational mesh is fixed on the space and an additional hyperbolic equation is required to describe the motion of the interface. Despite its relative ease of handling strong interface distortions, the discretization of such an equation may introduce artificial diffusion, thus leading to leak of accuracy, for instance the Volume of Fluid (Hirt and Nichols, 1981) and the Level-Set (Sussman et al. (1994) methods. On the other hand, the *Lagrangian* formulation describes the interface between fluids explicitly by computational elements. Such a description allows a sharp representation of the front, but its drawback is the proper treatment of topological changes in the interface, since coalescence and break-ups are not inherent to its methodology. The Volume-tracking Harlow and Welch, 1965 and the Front-tracking (Glimm et al. 1988) methods are far the most widely used. Due to the shortcomings of purely *Eulerian* and purely *Lagrangian* formulations, the Arbitrary Lagrangian-Eulerian description allows these two frameworks to be combined in one single formulation so that the best aspect of each separately approach can be used in conjunction, that is, the computational mesh nodes may move with the continuum in normal Lagragian fashion or to be held fixed in Eulerian manner. The ALE description has shown to be suitable to describe fluid flow problems (see, for instance Anjos et al. 2012) and this work extends its capability to two-phase flows with phase change.

Two-phase flow problems become even more interesting if phase change occurs, where the mass transfer from one phase to another adds significantly complexity to the dynamics of bubbles and droplets. However, the modeling of such phenomena is not an easy task and should be treated with extreme care. Unfortunately, the related literature, in the numerical domain, is not so widely developed for mass transfer in single and two-phase flows, compared to that for no phase change taking place. Despite the problems related on the modeling of phase change, efforts have been invested to develop tools capable to predict, to a certain extent, boiling and condensation processes in two-phase flows. The pool boiling process was extensively reviewed by Dhir, 1991, in which 4 basic mechanisms were identified that contribute to the total heat flux, namely evaporation at the liquid interface, enhanced natural convection, natural convection and transient conduction at a nucleation site. It was also found that these mechanisms are strongly linked to the temperature of the superheated wall. In Juric and Tryggvason, 1998, they presented a new model to simulate two-phase flows with phase change in two- dimensional domains. The new formulation was included in the previous front-tracking adiabatic code developed by Unverdi and Tryggvason, 1992, to extend its computation to boiling flows. Since the interface is represented by geometrical objects, special treatment of phase change was considered. They also studied the influence of several parameters in the interface temperature. To validate their model, they compared it to the exact solution of a 1-dimensional test case, followed by the simulation of film boiling with different fluid properties.

In this work, we present an extension of the previous developed 3-dimensional code Anjos, 2012, to simulate two-phase flows with phase change that allows the simulation of complex problems involving liquid-vapor interface. The equations are written in the Arbitrary Lagrangian-Eulerian description and discretized by the Finite Element method. An additional equation is required to model the temperature field and, due to the mass transfer occurring in the interface, the mass conservation equation is slightly modified to take into account boiling and/or condensation. In the following sections of this paper, the mathematical formulation used to model two-phase flows with phase change will be detailed. Moreover, the interface representation and the employed re-meshing technique will be shortly described. Latter, validations for the surface tension term and simulations of the implemented phase change algorithm will be presented. Finally, conclusions from this study will be discussed.

MATHEMATICAL FORMULATION

Let us consider vapor and liquid occupying a domain where phase change occurs. In the 'one-fluid' approach, one set of governing equations, namely momentum, conservation of mass and energy, are written for both phases with the assistance of a step function, which takes into account the jump of properties in the interface separating the phases. Assuming that the fluids in each phase are incompressible and the change of volume, led by the phase change, occurs only at the interface, the momentum equation is written in the Arbitrary Lagrangian-Eulerian non-dimensional form as:

$$\rho\left[\frac{\partial \mathrm{u}}{\partial \mathrm{t}}+(\mathrm{u}-\hat{\mathrm{u}})\cdot\nabla\mathrm{u}\right]=-\nabla\mathrm{p}+\frac{1}{\mathrm{Re}}\nabla\cdot\left[\mu\left(\nabla\mathrm{u}+\nabla\mathrm{u}^{\mathrm{T}}\right)\right]+\frac{1}{\mathrm{Fr}^2}\rho\mathrm{g}+\frac{1}{\mathrm{We}}\kappa\nabla\mathrm{H}$$

On the left hand side of Eq. 1, the term $\mathrm{u}-\hat{\mathrm{u}}$ represents the relative velocity between the flow field u and the mesh $\hat{\mathrm{u}}$ at one time step. The gravity term is represented by **g**, while the system pressure by **p** and time by **t**. **Re**, **We** and **Fr** are the Reynolds, Weber and Froude numbers respectively. The term $\kappa\nabla\mathrm{H}$ represents the surface tension force, which is computed based on a new 3-dimensional extension of the Frenet's formula. Here, κ is the surface's curvature and **H** is the Heaviside function, which defines the region of each fluid in the domain, so that the inner fluid is 1, the outer fluid is 0 and the surface is 0.5. The fluid properties density ρ and viscosity μ are kept constant at each phase, but not necessarily assume equal values. Thus, the Heaviside function is used to defined a generic property φ distribution along the domain as follow:

$$\varphi=\varphi_1\mathrm{H}+\varphi_2(1-\mathrm{H})$$

Within the ALE context, the energy transport equation should take into account the relative velocity $\mathrm{u}-\hat{\mathrm{u}}$, thus representing the heat convection. The energy equation is written in the Arbitrary Lagrangian-Eulerian non-dimensional form as:

$$\rho c_p\left[\frac{\partial T}{\partial t}+(u-\hat{u})\cdot\nabla T\right]=\frac{1}{\mathrm{Re}\,\mathrm{Pr}}\left[\nabla\cdot k\nabla T+\dot{q}\left|\nabla H\right|\right]$$

In the above equation, **T** is the temperature distribution, $\mathbf{c_p}$ is the heat capacity and **k** is the heat conductivity, both distributed on the domain in the same manner as the density and viscosity. **Pr** is the Prandtl number and $\dot{q}$ is the mass flux defined as:

$$\dot{q}=k\nabla T$$

In mass transfer problems, the interface should be moved according to evaporation or condensation effects, thus the velocity field is no longer divergence free in the vicinity of the interface. Therefore, the continuity equation is slightly modified to take into account such a phenomena, so that:

$$\nabla\cdot u=\left(1-\frac{\rho_l}{\rho_v}\right)\dot{q}\left|\nabla H\right|$$

Since the gradient of the Heaviside function is only different of zero near the interface, the fluid equation becomes incompressible far from it. The normal velocity of the interface $\mathbf{u_I}$ is found to be:

$$u_I=\frac{1}{2}\left(u_l+u_v\right)+\frac{\dot{q}}{2}\left(1+\frac{\rho_l}{\rho_v}\right)$$

The new position of the interface nodes $\mathbf{x_I}$ may be found by integrating:

$$\frac{dx_I}{dt}=u_I n$$

where **n** is the outward normal vector**.**

INTERFACE DESCRIPTION AND REMESHING

Unlike the descriptions found in most front-tracking codes, the interface mesh implemented in this work is a sub-set of the tetrahedron domain mesh, i.e. each triangle shares a face of two adjacent tetrahedral elements, thus the interface is sharply defined. Moreover, due to the finite element formulation, the fluid properties are kept constant inside the mesh element and no smoothing functions is required to treat the jump conditions at the interface. As can be seen in Fig. (1a), the interface between the phases is not part of the computational mesh and it should be defined by a function, which is advected using an additional equation. Moreover, the transition of properties (Φ_1, Φ_2) should be calculated using a smooth function to avoid numerical instabilities. On the other hand, in the Lagrangian approach (Fig. (1b)), the properties Φ_1 and Φ_2 are sharply defined in each region, thus not requiring any additional function.

In two-phase flows within the moving mesh context, a re-meshing process is extremely necessary since the flow field, either due to the motion of a single bubble or due to the imposed velocity condition, tends to move the mesh nodes from one region to another, damaging the uniformity of node distributions. In this work, the interface between the phases is part of the computational mesh and a surface re-meshing is also required to keep the element connectivity consistently while the simulation is running. Thus, two sets of data are stored during the simulations, namely volumetric nodes and surface mesh. The former consists in the mesh nodes distributed on the 3-dimensional space and the later stands for the interface and boundary triangular meshes. Each set of data is treated separately and exported to an external mesh generator library (Si, 2008), which is responsible to generate the tetrahedron connectivity array at each time step. The volumetric node distribution is done by solving Helmholtz equation, in which the boundary conditions consist in the characteristic edge lengths of each surface mesh, thus a smooth transition between the interface and the boundary meshes is successfully achieved. Note that the treatment of the volumetric nodes does not require the maintenance of the tetrahedron's connectivity. On the other hand, the surface meshes should be modified, at each time step, to keep the triangle's shape bounded to good aspect ratios. This is achieved by the insertion/deletion of points, edge flipping and edge contractions.

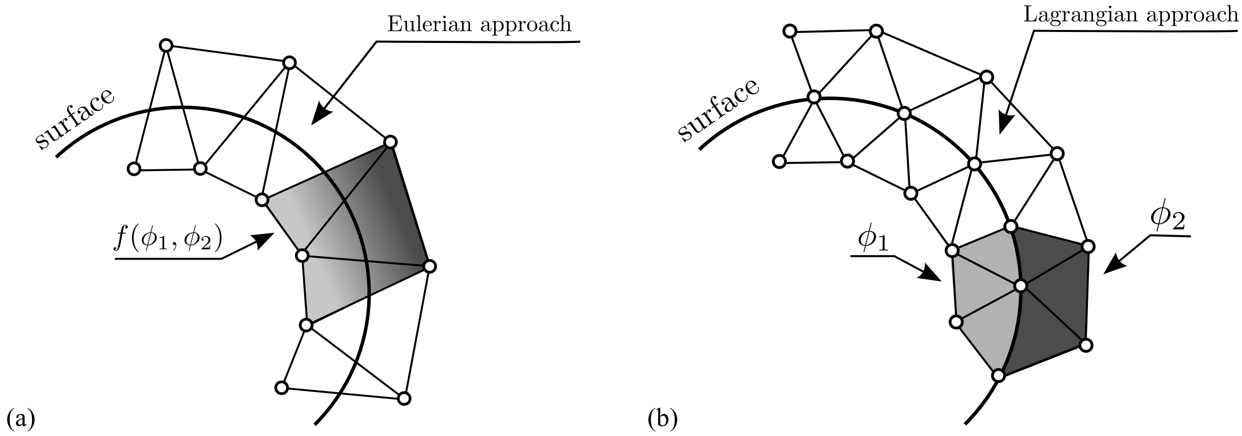

Figure 1 2-dimensional representation of the interface between the phases. (a) In the Eulerian approach, the interface is implicitly defined as a scalar function, which is advected by an additional equation. (b) A set of geometrical elements is used to represent the interface in the Lagrangian approach, thus no additional equation is required.

Figure (2a) shows the insertion procedure, which aims to occupy ”barren” areas and to increase the accuracy in certain regions where a higher precision is required. If an edge e > e_{max}, a new point v is inserted by considering the curvature of the adjacent nodes. Figure (2b) shows the deletion of a surface node, which is required when the moving mesh causes the clustering of points in the surface mesh. The strategy attempts to find an edge e that is smaller than a predefined length e_{min}, which is defined according to the simulation requirements. Once such an edge is detected, the sum of the edge length of the 1-ring neighbors is performed for both extremity vertices. The one that has the lowest value is then eliminated from the surface mesh. Once the vertex deletion is performed, its neighbors form a polyhedron, which must be subdivided to recover the triangular structure. Such a task is done based on the work of Devillers, 2002 and Xu et al. 1998 and extended here to triangular surface meshes.

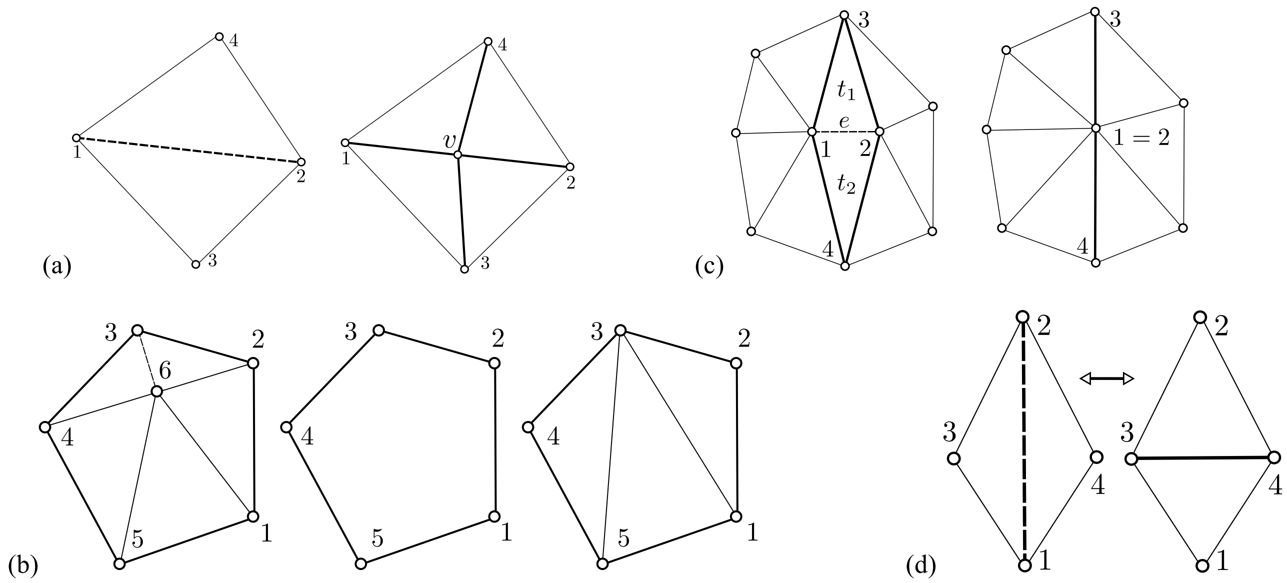

Figure 2: Basic surface mesh operations (a) point insertion, (b) edge contraction, (c) point deletion and (d) edge flipping.

Figure (2c) shows the strategy of edge contraction adopted, which is based on the work of Negami, 1994. Once an edge e < e_{min} is detected, such a scheme aims to collapse the two extremity vertices into the midpoint of the edge e, thus the two adjacent triangles are removed from the surface mesh. After been contracted, the edge e is no longer part of the surface mesh, therewith points 1 and 2 occupies the same position while points 3 and 4 remains in the same location. The benefit of such an approach compared to the previous point deletion strategy is its geometrical simplicity since the surrounding connectivity of the mesh is not affected. Note that the same consideration of the curvature of two adjacent points fount in the insertion strategy should be taken into account when collapsing two vertices, thus avoiding displacement errors and non-desired losses of mass. The last strategy, but not less important, is shown in Fig (2d). The edge flipping strategy attempts to restore the mesh quality changing the connectivity of two consecutive elements by comparing the quality of the pair of triangles (1-2-3 and 1-4-2) to the new pair (1-4-3 and 2-3-4). This quality is based on the sum of triangle aspect ratio, the curvature of neighbors points, the sum of area and the circumcenter of each triangle and performed for the entire surface mesh. All these operations are made systematically, thus the mesh is kept regularized and the simulation can reach the final solution.

Due to the separation of the domain and the surface mesh in the above procedure, the mesh distribution treatments may be combined into a scheme and adjusted by parameters varying from 0 to 1, which is a 3-dimensional generalization of the approach presented by Souza and Mangiavacchi 2005 for 2-dimensional simulations. The domain and surface velocities are therefore treated as follows:

$$\hat{\mathbf{u}}(\mathbf{x}) = \begin{cases} \mathbf{u} - \gamma_1(\mathbf{u}\cdot t)t + \gamma_2(\mathbf{u}_e\cdot t)t & \text{if } \mathbf{x} \text{ belongs to the interface} \\ \beta_1\mathbf{u} + \beta_2\mathbf{u}_v + \beta_3\mathbf{u}_e & \text{if } \mathbf{x} \text{ does not belong to the interface} \end{cases}$$

In such a method, due to the description of the interface mesh by computational elements, the surface should move according to the fluid motion. In the above equation, if **x** belongs to the interface, we can define its velocity as $\mathbf{u_I}$. Thus, it is convenient to decompose it into two orthogonal components: $\mathbf{u_{In}}$ and $\mathbf{u_{It}}$ which represent the normal and tangential velocities, respectively. To decrease the displacement of nodes in the tangential direction, one may remove partially, or even totally, its velocity from the total interface's velocity. This can be achieved by either projecting the interface's velocity $\mathbf{u}_{\mathrm{In}}$ to the normal vector associated to the node or, in a simpler manner, by removing the tangent component from the total surface mesh velocity $\mathbf{u}_{\mathrm{It}} = \mathbf{u}-(\mathbf{u}\cdot t)t$. Therewith, the parameter γ_1 controls the magnitude of the tangent velocity in the total interface's velocity. Letting $\gamma_1=1$, only the normal interface's velocity is taken into account in the surface mesh motion, and therefore the surface nodes are not allowed to move in the tangent direction. Additionally, the parameter γ_2 includes the smoothing scheme on the surface mesh nodes, thus keeping them all bounded within good aspect ratio. More details about the smoothing technique used in this work, see Anjos, 2012). The parameter β_1 controls the Lagrangian motion of the inner and outer volumetric mesh velocity. By setting $\beta_1=1$, the flow velocity **u** is fully included in the moving mesh velocity $\hat{\mathbf{u}}$ and, consequently, the volumetric nodes move according to the flow field. Otherwise, letting $\beta_1=0$, the flow velocity **u** is not taken into account on the moving mesh velocity. The parameters β_2 and β_3 control the intensity of the velocity smoothing scheme $\mathbf{u_v}$ and the Laplacian smooth scheme $\mathbf{u_e}$ into the moving mesh velocity. Thus, setting both parameters to null, the volumetric mesh smoothing is not performed. Note that the parameters γ and β may vary from 0 to 1 to achieve a desirable node distribution according to the simulation requirements.

NUMERICAL PROCEDURE

The Navier-Stokes and energy equations are discretized through the Finite Element method. A summary of the underlying principles is given below and further details can by found at Anjos, 2012.

The Galerkin method is applied to the variational form of the governing equations to discretize all terms except the non-linear convective term. To overcome the obstacle of modelling and implementing a numerical scheme for this term, the semi-Lagrangian technique is employed (for details, see Robert, 1981 and Pironneau, 1982) based on the idea of representing the acceleration field by a Lagrangian point of view instead of the well-known Eulerian derivative. For each time step the points are moved towards the flow and, once the task is accomplished the coordinate system is reinitialized and the original mesh is recovered. The substantial derivative is evaluated in the strong form along the characteristic trajectory, by estimating the position of a point and solving the equation Dx/Dt backwards in time $t^{n+1} \geq t \geq t^n$ with the initial condition $x(t^{n+1})=x_i$ than an integration method is used to evaluate the previous point position in the grid. A first order discretization scheme is adopted assuming the trajectory as a straight line. The choice of element types for coupled PDEs problems must take into account the *Babuska-Brezzi* condition in order to preserve the stability properties intrinsic to the discretization scheme (Cuvelier, 1986 and Zienkiewicz, 2000). Such a condition is not mandatory, but must be fulfilled if used in conjunction with the Galerkin method. For these reasons and considering the excellent mass conservation property, the Taylor-Hood $P1^+ - P1$ element (*Mini*-element) was used to represent pressure and temperature, both calculated at the tetrahedron vertices, and the velocities, by evaluating them at the tetrahedron vertices and its centroid. Thus avoiding critical numerical instabilities and pressure oscillations. Once the discretization of the domain is accomplished, the system matrices are assembled and the solution of the time dependent 3-dimensional equations is then found by successively solving the linear system in each time step for pressure, velocity and temperature. Due to the strong coupling between pressure and velocity, the numerical procedure implemented to solve the mentioned linear system uses the Projection method based on the LU decomposition, which was first introduced by Chorin, 1968. The aim of this method is to uncouple pressure and velocity and solve each quantity separately, thus reducing the large linear system size into smaller ones. Additionally, the temperature equation is solved separately and it does not require the same methodology. The solution of the linear system for pressure and velocity is described below followed by the solution of the temperature field.

Let us define the matrix **B** and the right hand side vector $\mathbf{r^n}$ according to the Finite Element method as:

$$B = \frac{M_\rho}{\Delta t} + \frac{K}{Re} \qquad r^n = \frac{M_\rho}{\Delta t} u_d^n$$

where M_ρ and **K** are the mass and stiff matrices respectively. The time step is represented by Δt, **Re** is the Reynolds number and u_d^n is the velocity calculated in the previous time step at the departure points from the semi-Lagrangian method. The system is uncoupled and solved in the following way by computing the trial velocity $\tilde{u}$ and solving the linear system with proper velocity boundary conditions a $\mathbf{bc_1}$ as:

$$B\tilde{u} = r^n + bc_1$$

An update of $\tilde{u}$ is performed while considering the gravity **g** and the surface tension force **f**:

$$\tilde{u}_{corr} = \tilde{u} + \Delta t M_{\rho L}^{-1}\left(M_\rho g + \frac{1}{We} Mf\right)$$

Note that the subscript **L** refers to the Lumped matrix technique, in which a diagonalization is performed to the consistent matrix M_ρ to reduce numerical costs in its inversion. The mass transfer contribution across the interface **Z** is represented by the expression below:

$$Z = \left(1 - \frac{\rho_l}{\rho_v}\right)\dot{q}\left|G_T H\right|$$

where ρ_l and ρ_v are the densities for the liquid and vapour phases respectively. $\dot{q}$ is the mass flux and $G_T H$ is the gradient of the Heaviside function defined by the temperature mesh nodes. The pressure p^{n+1} is calculated by solving the linear system with pressure boundary conditions bc_2:

$$DB_L^{-1}Gp^{n+1} = -D\tilde{u}_{corr} + Z + bc_2$$

where G is the gradient matrix, D is the divergence matrix and B_L^{-1} is the inverted lumped matrix B. The velocity solution u^{n+1} is found from:

$$u^{n+1} = \tilde{u}_{corr} + B_L^{-1}Gp^{n+1}$$

On the other hand, the solution of the linear system of the temperature equation is straightforward. Let define the matrix B_T and r_T^n as:

$$B_T = \frac{M_T}{\Delta t} + \frac{K_T}{PrRe} \qquad r_T^n = \frac{M_T}{\Delta t} T_d^n + \dot{q}\left|G_T H\right|$$

where $\mathbf{M_T}$ and $\mathbf{K_T}$ are the mass and stiff matrices of the temperature equation respectively. **Pr** is the Prandtl number and **Re** is the Reynolds number. The temperature calculated in the previous time step at the departure points is represented by u_d^n. Thus, the solution of the temperature is found by solving the following linear system for **T** with temperature boundary conditions $\mathbf{bc_T}$ as:

$$B_T T = r_T^n + bc_T$$

The solutions of the velocity and temperature linear systems are obtained iteratively at each time step by the conjugate gradient method and preconditioned by the incomplete Cholesky factorization. The pressure is solved by the generalized minimum residual method and preconditioned by an incomplete LU factorization. Such a methodology has shown to be suitable to solve the linear systems of two-phase flow problems with phase change.

RESULTS

In this section, a numerical test has been carried out to validate the code against the terminal velocity and liquid film thickness of elongated bubbles in circular channels. The wall effects are included in the bubble dynamics and the results are compared to the well-known flow pattern map of White and Beardmore, 1962 and the Brown's theoretical solution for the film thickness Brown, 1965. The modeling of such flows increases significantly the obstacles of the remeshing process, since the formation of the thin liquid film between the bubble and wall requires a fine mesh to capture the flow mechanisms. The flexibility of the Finite Element method is explored in the development of the computational meshes, in which different mesh element sizes are used in the 3-dimensional domain.

- ### Isothermal Rising of Taylor Bubbles

Figure (3) shows the time progression of a Taylor air bubble immersed in a sucrose solution (see Table 1). The same bubble shape and film thickness, as the previous cases, is used as an initial shape. In the transient evolution, the bubble's velocity reached its maximum velocity at time $t \approx 1$, and its terminal velocity at time $t \approx 3.7$ Also, it was shown that the bottom part of the bubble was pulled in and oscillated until convergence at $t \approx 7.4$ The mesh parameters used in this simulation were $\beta_1 = 0.0$, $\beta_2 = 1.0$, $\gamma_1 = 1.0$ and $\gamma_2 = 0.1$ and the dimensionless numbers were set to $M=10^{-7}$, Fr=1 and We=40. Figure (4) presents the transient solution of the bubble's center of mass velocity. It was observed, an overshooting of the ascension velocity from time $t=0$ to $t=1.1$, due to the initial deformation of the bottom part of the bubble, and consequently acceleration of the center of mass. The result agreed to the prediction of the flow pattern map, obtaining an error of 1.1%.

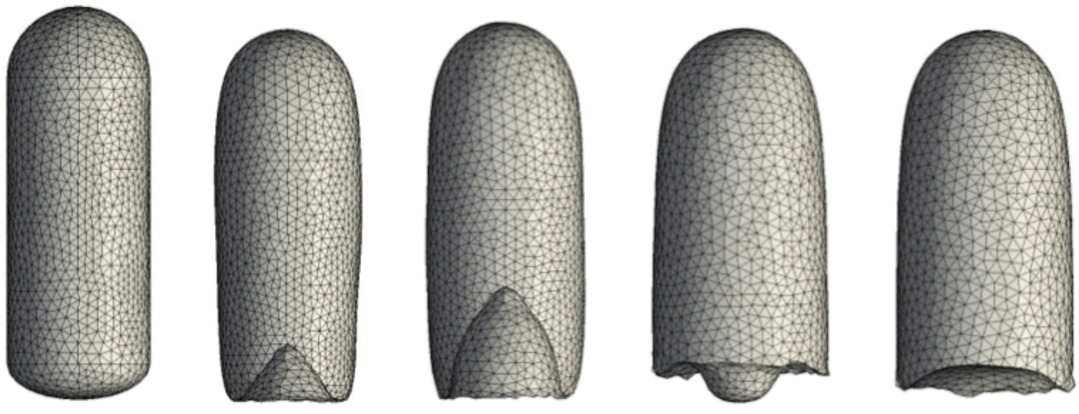

Figure 3 Bubble shape evolution with time for an air bubble in a sucrose solution with dimensionless numbers $M=10^{-7}$, Fr=1 and We=40. The adaptive mesh refinement proposed in this work captures accurately the strong shape distortion produced by the high ascension velocity. (a) Initial bubble shape with t=0. (b-d) Bubble shape change during transient solution. (e) Terminal bubble shape with t=7.41.

Table 1 Fluid properties

Fluid	ρ [kg/m^3]	μ [μPa.s]	σ [N/m]	k [W/mK]	c_p [J/kgK]	δ [-]
R134a - vapor	37.54	12.04	0.0074	0.0173	1065	-
R134a - liquid	1187	185.4	0.0074	0.079	1446	-
sucrose	1.172	5.650	77.7	-	-	0.0617
glycerol	1.260	712.0	63.1	-	-	0.1483
air	1.789	1.225	-	-	-	-

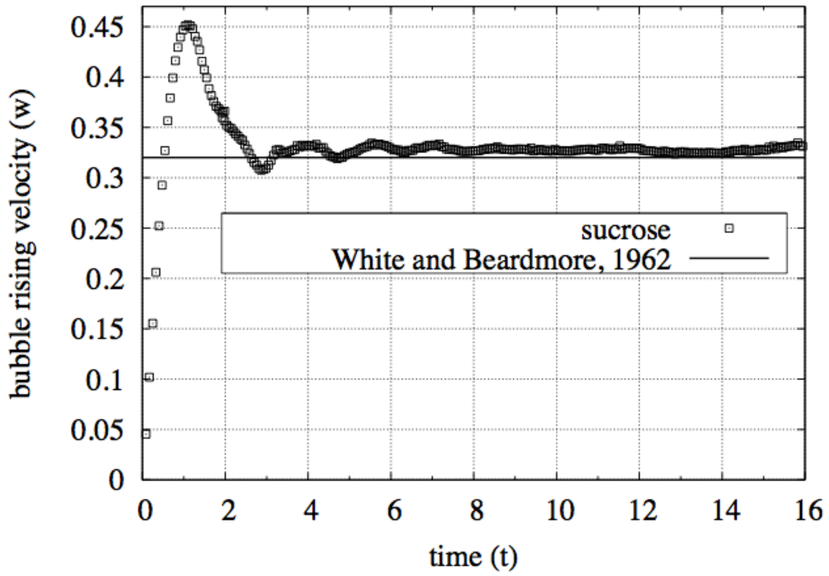

Figure 4 Rising of an air Taylor bubble immersed in a sucrose solution with dimensionless numbers set to M=10^{-7}, Fr=1 and W*e*=40. The time evolution of the Bubble's center of mass velocity is compared to the terminal bubble's velocity found in White and Beardmore, 1962. Velocity and time are non-dimensional.

Figure (5) shows the time progression of a Taylor air bubble immersed in a glycerol solution (see Table 1). The mesh parameters used in this simulation were β_1=0.0, β_2=0.5, γ_1=0.5 and γ_2=0.1 and the dimensionless numbers was set to M=10^{-2}, Fr=1 and We=100. In this case, the mesh parameters were slightly modified to test different mesh conditions in the simulation. The obtained result shows that the bubble shape and terminal velocity agrees well to experimental data. Figure (6) shows the evolution in time of the bubble's center of mass velocity, in which the computed terminal velocity approaches the value found in the flow pattern map. The error was found to be 6.5%.

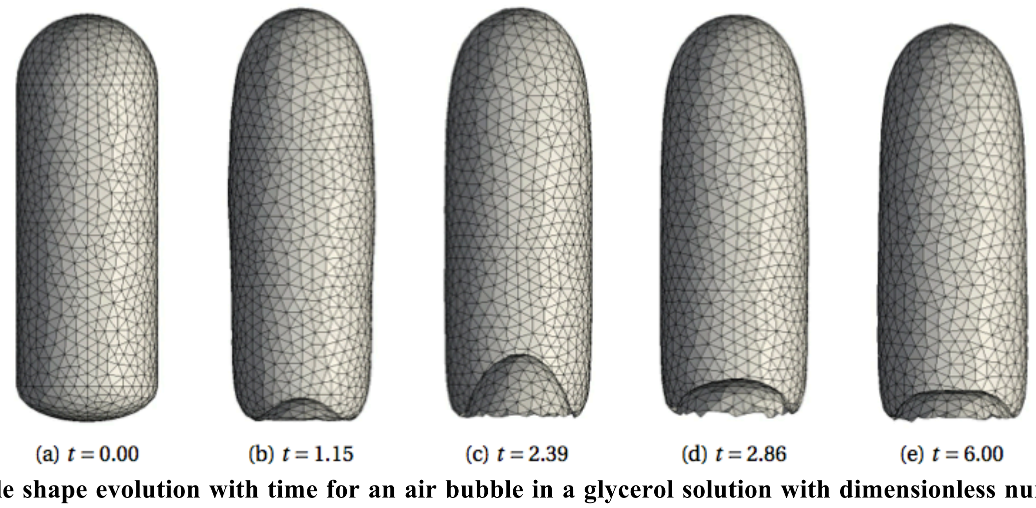

Figure 5 Bubble shape evolution with time for an air bubble in a glycerol solution with dimensionless numbers M=10^{-2}, Fr=1 and We=100. (a) Initial bubble shape with *t*=0. (b-d) Bubble shape transient solution. (e) Terminal bubble shape with *t* = 6.00.

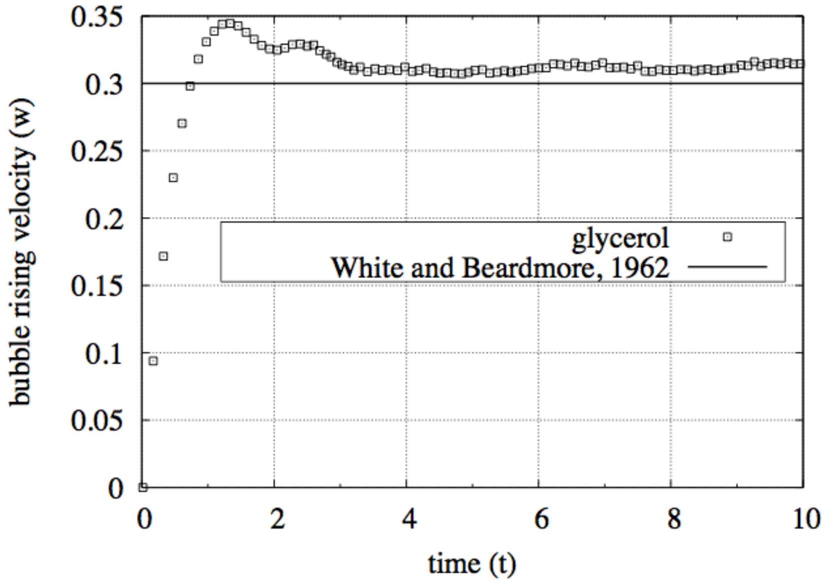

Figure 6 Rising of an air Taylor bubble immersed in a glycerol solution with dimensionless numbers set to M=10^{-2}, Fr=1 and We=100. The time evolution of the Bubble's center of mass velocity is compared to the terminal bubble's velocity found in White and Beardmore, 1962. Velocity and time are non-dimensional.

- ### Spherically growing bubble in a superheated liquid

To validate the proposed mass transfer model, a spherical vapor bubble is simulated in a superheated liquid with the fluid properties shown in Table (1). The numerical domain was set as a cubic shape and an outflow condition was imposed at all the walls, thus allowing the bubble to expand equally to all directions. Initially, a vapor bubble with its temperature equal to the saturation temperature *Tsat* is placed on the middle of the domain. The surface temperature is set to the saturation temperature, and thus the only difference in temperature is due to the superheated liquid surrounding the vapor bubble. Considering that the temperature is non-dimensionalized such that T^*=(T−*Ts*)/(Tw−*Ts*), the saturation temperature is set to *Ts*=0 and the temperature of the superheated liquid is set to *T*=0+. The superheated liquid at the given saturation pressure starts to evaporate, thus the total liquid volume decreases and the vapor volume increases. At this point, an analytical solution can be used to compare the change of vapor volume with time. Such an expression is given by:

$$R(t) = 2\beta\sqrt{kt}$$

where *R* is the bubble's radius, *k* is the thermal conductivity and β is a constant calculated in Scriven, 1959. Note that the viscous and surface tension effects are neglected during the simulation, thus the above is valid for the growth of the vapor bubble.

Figure (7) shows the growth of the vapor bubble with time due to evaporation process. From t≈0.02 to t≈0.12, the vapor bubble expands linearly due to heat conduction. From t≈0.13 to the end of the simulation, the bubble asymptotically reduces its expansion and begins to deviate from the analytical solution, due to different assumptions used to derive the above equation with respect to those in the numerical simulation. Figure (8) depicts the three velocity components showing the vapor bubble expansion. As can be seen, the velocity increases with time as well as the bubble radius.

CONCLUSIONS

This article presents a new methodology for simulating incompressible two-phase flows within the Finite Element Method context in which the mesh moves in an Arbitrary Lagrangian-Eulerian fashion. The coupling ALE-FEM methodology provides a sharp representation of the interface between the phases, not only for the geometrical representation itself but also for the definition of the phase properties, thus resulting in a model, which accurately describes the actual physical conditions. The proposed treatment of the computational mesh, splitting the surface meshes and the volumetric points, has shown to be an excellent choice, thus avoiding the obstacles of handling the remeshing process over the tetrahedron mesh, allowing the utilization of a standard Delaunay tetrahedralization library. Moreover, the new adaptive meshing strategy achieves good control of the mesh quality during the simulations by keeping the volumetric and surface elements bounded to a satisfactory shape, thus preserving the accuracy of the calculation. However, excessive linear interpolations on the volumetric and surface mesh may lead to a lack of accuracy in certain domain areas and this should be used with care. The heat and mass transfer was implemented into the code using the same strategy of the fluid flow solver. However, it requires an extensive reformulation of the isothermal method presented here. The thin thermal boundary layer requires an additional number of nodes and elements to be physically resolved, thus increasing significantly the processing time. The preliminary tests presented for evaporation in square microchannels qualitatively agreed to those observed experimentally. However, a deeper analysis is still required to compare the actual state of the phase change model with different benchmarks available in the literature.

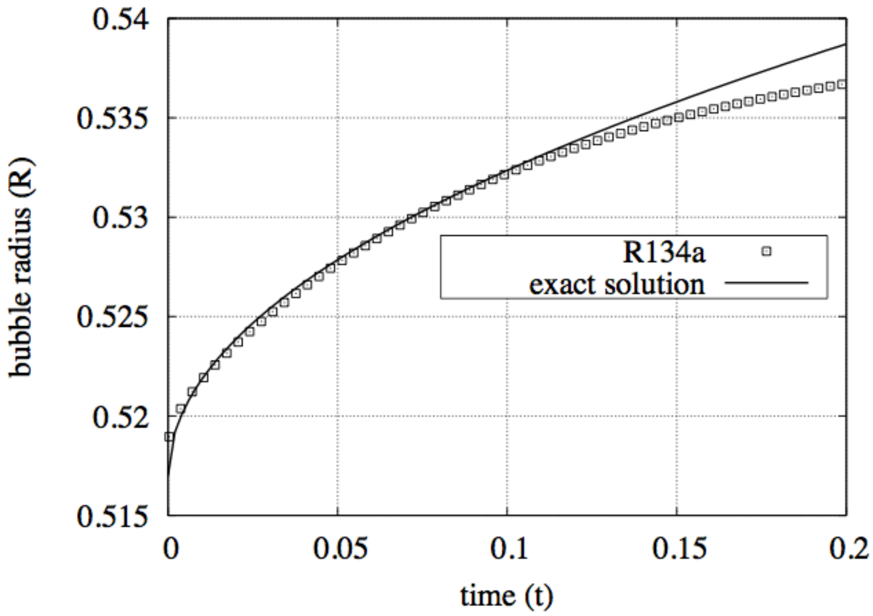

Figure 7 Growth of a vapor bubble due to evaporation.

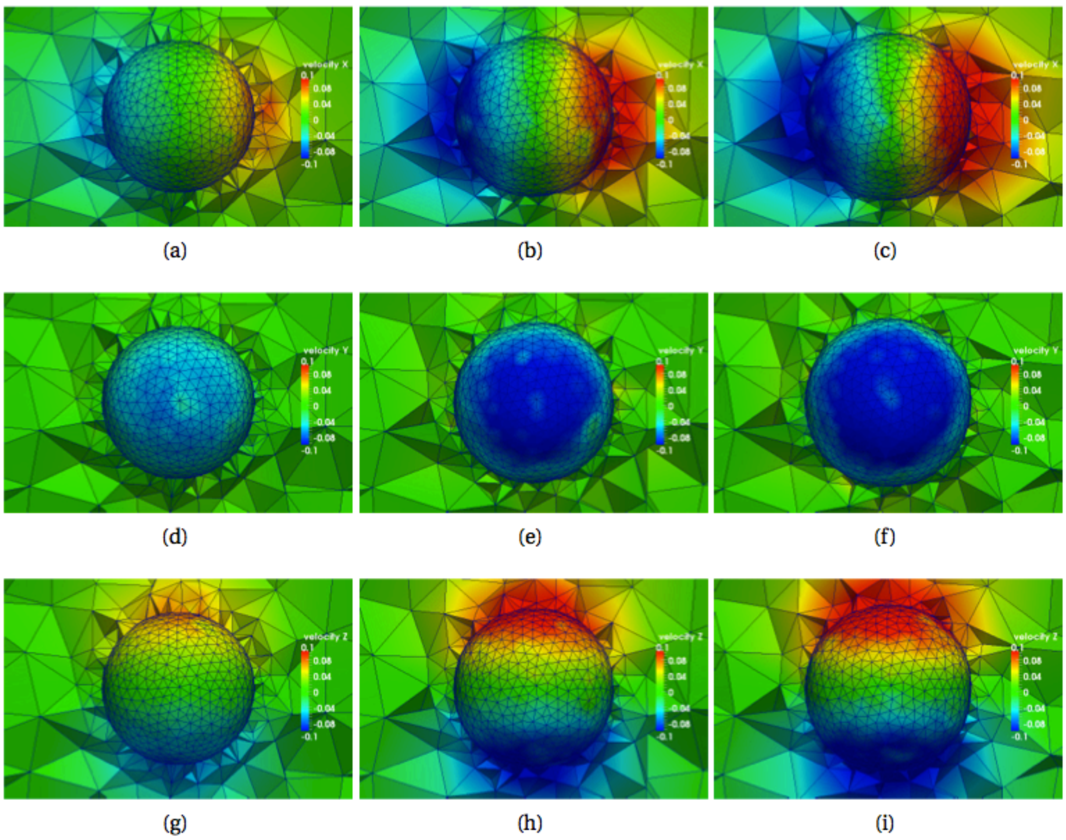

Figure 8 Velocity components of a vapor bubble growing due to evaporation of the superheated liquid. (a)-(c) x-component, (d)-(f) y-component and (g)-(i) z-component.

ACKNOLEDGEMENTS

This research is founded by the Nano-Tera RTD project CMOSAIC (ref. 123618) financed by the Swiss Confederation and scientifically evaluated by SNSF.

NOMENCLATURE

Re	Reynolds number
Fr	Froude number
We	Weber number
Pr	Prandtl number
M	Morton number
B	matrix
M_ρ	mass matrix
K	stiff matrix
Δt	time step, non-dimensional
r^n	residual vector from current time step
u_d^n	velocity vector calculated from the departure points in the previous time step
$\tilde{u}$	trial velocity
$\mathbf{f}$	surface tension force
$\mathbf{g}$	gravity
c_p	specific heat
k	thermal diffusity
$\mathbf{n}$	outward normal vector
$\mathbf{u_I}$	interface's velocity
$\boldsymbol{\beta}$	volumetric parameter
Υ	surface parameter
t	time
$\hat{u}$	mesh velocity

REFERENCES

G.R. Anjos, N. Barhani, N. Mangiavacchi, J.R. Thome (2012), A 3D ALE Finite Element Method for Two-Phase Flows, Journal of Computational Physics, submitted.

G.R. Anjos (2012), A 3D ALE Finite Element Method for Two-Phase Flows with Phase Change, Ph.D. Thesis, École Politechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2012.

R.A.S Brown, 1965. The mechanism of large bubbles intubes. i. bubbles velocities in stagnant liquids. *Canadian Journal of Chemical Engineering, Vol.*43, pp.217-223.

A. J. Chorin, 1968. Numerical solution of the Navier-Stokes equations. *Mathematics of Com- putation, Vol.*22, pp.745-762.

C. Cuvelier, A. Segal and A.A. van Steenhoven, 1986. Finite Element Method and Navier-Stokes *Equations.* Dordrecht, Holland.

O. Devillers, 2002. On deletion in Delaunay triangulations. *International Journal of Computa- tional Geometry & Applications, Vol.*12, pp.285.

V.K. Dhir, 1991. Nucleate and transition boiling heat transfer under pool and external flow conditions. *International Journal of Heat and Fluid Flow, Vol.* 21, pp.290-314.

J. Glimm, J.Grove, W.B.Lindquist, O.McBryan, and G.Tryggvadson (1988). The bifurcation of tracked scalar waves. *SIAM Journal of Computations, Vol.* 9(1), pp.61-79

FH.Harlow and J.E.Welch (1965), Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids, Vol.* 8, pp. 2182-2189.

C. Hirt, B. Nichols (1981), Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics, Vol. 39, pp. 201–225.

D. Juric and G.Tryggvason, 1998. Computations of boiling flows. Journal of Computational *Physics, Vol.*24(3), pp.387-410.

S. Negami, 1994. Diagonal flips in triangulation of surfaces. *Discrete Mathematics, Vol.*135, pp.225-232.

O.Pironneau, 1982. On the transport-diffusion algorithm and its applications to the Navier-Stokes equation. *Numerische Mathematik, Vol.*38, pp.309-332.

A. Robert, 1981. A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere Oceans, Vol.*19, pp.35-46.

L.E. Scriven, 1959. On the dynamics of phase growth. *Chemical Engineering Science, Vol.*10, pp.1-13.

H. Si, 2008. Adaptive tetrahedral mesh generation by constrained Delaunay refinement. *International Journal for Numerical Methods in Engineering, Vol.*46(7), pp.856-880.

M. Sussman, P. Smereka, S. Osher (1994), A Level-Set Approach for Computing Solutions to Incompressible Two-Phase Flow, Journal of Computational Physics, Vol. 114, pp. 146–159.

S.O.Unverdi and G.Tryggvason, 1992. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics, Vol.*100, pp.25–37.

X. Xu, C.C. Pain, A.J.H Goddard, and C.R.E. Oliveira, 1998. An automatic adaptive meshing technique for delaunay triangulations. *Computer Methods in Applied Mechanics and Engineering, Vol.*161, pp.297-303.

E.T. White and R.H. Beardmore, 1962. The velocity of rise of single cylindrical air bubbles through liquids contained in vertical tubes. *Chemical Engineering Science, Vol.*17, pp. 351-361.

O.C. Zienkiewicz and R.L.Taylor, 2000. The Finite Element Method Volume 1: The Basis. Wiley John and Sons, 5th edition.

APPENDIX

The non-dimensional numbers **Re**, **Fr**, **We**, **M** and **Pr** are the Reynolds, Froude, Weber, Morton and Prandtl numbers respectively which are important to characterize the fluid flow problem. These numbers are defined based on a characteristic length L and velocity **U** as follow:

$$\mathrm{Re} = \frac{\rho_0 \mathrm{UL}}{\mu_0} \qquad \mathrm{Fr} = \frac{\mathrm{U}}{\sqrt{\mathrm{g}_0 \mathrm{L}}} \qquad \mathrm{We} = \frac{\rho_0 \mathrm{U}^2 \mathrm{L}}{\sigma_0} \qquad \mathrm{Pr} = \frac{\mathrm{c_p}\mu_0}{\kappa_0} \qquad \mathrm{M} = \frac{\mu_0^4 \mathrm{g}}{\rho_0 \sigma_0^3} = \frac{\mathrm{We}^3}{\mathrm{Fr}\,\mathrm{Re}^4}$$

Here, g is the gravity, c_p is the fluid specific heat k is the thermal conductivity, σ is the surface tension coefficient and the subscript 0 denotes that the properties are taken from the referential fluid.