COMPLETE RELAXATION STRATEGIES APPLIED TO EMULSIONS FLOW SIMULATION USING THE BOUNDARY INTEGRAL METHOD

Ivan Rosa de Siqueira, ivan.rosa.siqueira@gmail.com
Taygoara Felamingo de Oliveira, taygoara@unb.br
UnB Gama College - University of Brasília

Francisco Ricardo da Cunha, frcunha@unb.br
Department of Mechanical Engineering - University of Brasília

Abstract. *In this work, we apply self adaptation methods to regularize the mesh over the droplet in boundary integral simulations of very disperse emulsions. Both disperse and continuous phases are Newtonian fluids, and the drops deform under the action of a simple shear flow. The velocity of the fluid particles right over the surface is represented by a boundary integral formulation, for drops of unitary viscosity ratios. A Lagrangian method is employed to evolve the position of the fluid particles in the interface, which are represented by the nodes of a triangular mesh over the drop surface. The elements of the mesh deforms due to the movement of its nodes and, after a short time interval, several numerical problems arises from the crescent distortion of the triangles. A strategy for a complete mesh relaxation is presented. This methodology allows the mesh to be independent of the flow history, even in a Lagrangian simulation.*

Keywords: *Emulsions, Boundary Integral Method, Relaxation Methods.*

1. INTRODUCTION

An emulsion is a biphasic mixture of two immiscible viscous liquids in which one is dispersed in the other in the form of drops. Emulsions are produced by a process of generation and breakup of drops in a couple of fluids which sets out two phases, one dispersed and other continuous. In most applications, the small size of the drop and its spacial distribution, allow the mixture to be treated as a homogeneous equivalent fluid whose mechanical medium behavior is intrinsically linked to the fluids dynamics in the scale of the droplets. The diameter of a droplet in an emulsion is typically between $1\mu m$ and $100\mu m$. Considering the thermophysical properties of fluids in typical emulsions and common speeds in flows of practical interest, it can be stated that, in the scale of the droplets, the flow is free from inertia effects, whether related to the particle (droplet) or related to the continuous phase. Whereas the drop is convected by the flow, we may assume that a characteristic scale of velocity experienced by the particle, U_a, is given by $U_a = \dot{\gamma}_c a$, where $\dot{\gamma}_c$ is is a typical shear rate of the flow and a is the droplet diameter before deformation. In those circumstances, Reynolds number is defined as $Re_a = a^2\dot{\gamma}_c/\nu$, where ν denotes the kinematic viscosity of the continuous phase. Whereas, for example, an oil droplet with diameter equal to $a = 10\mu m$, in water, and assuming that $\dot{\gamma}_c \approx 1s^{-1}$, then the Reynolds number of the external flow is $Re_a \approx 10^{-4}$, so $Re_a \ll 1$. Therefore, the Reynolds number of the internal flow is also very small and it's given by $Re_g = Re_a/\lambda \ll 1$, since $\lambda = \mathcal{O}(1)$, where λ denotes the ratio between the viscosities of the dispersed phase and continuous, respectively.

The study of mechanical behavior of emulsions begins with the analysis of diluted suspensions. Among the first works in this area is worth mentioning the calculation of the effective viscosity of a dilute suspension of rigid spherical particles, developed by Einstein (1906). Taylor (1932) extended Einstein's results for the case of dilute emulsions of spherical drops of high surface tension. Deformation of droplets in simple shear flows and extensional was also studied by Taylor (1934). In this paper, the author defined a scalar measure of deformation of drops with ellipsoidal shape. Numerical strategies for the study of emulsions are available in a great number of articles found in the area. Most of them use the boundary integral formulation for Stokes' flow, described by Ladyzheskaya (1969). The boundary integral formulation has the advantage of transforming the three-dimensional problem of flow around a droplet in a surface problem in two dimensions, reducing the computational effort. However, the Boundary Integral Method presents issues related to the singularity of the kernel of the boundary integral and to the distortion of the mesh, due to the Lagrangian method.

Within this context, this work aims to present a methodology for relaxation of the mesh that solves, in certain limits, the problem of distortion of the elements associated with the running time of simulation. This methodology is an extension of a method already known in literature (Rallison and Acrivos, 1978). The work is organized in an introduction, a mathematical description of the formulation used, including details of the subtraction of singularity, a section describing the main aspects of the numerical method, a section devoted to the mesh relaxation procedure, a brief conclusion and references.

2. MATHEMATICAL FORMULATION

2.1 Integral representation of the velocity over a drop surface

Our model consider a deformable drop of a newtonian fluid of viscosity μ in an other immiscible newtonian fluid with the same viscosity. Both fluid are considered incompressible and inertial forces are negligible, and the ambient fluid is considered unbounded. In this case, the problem is governed by the Stokes equations. Under these conditions it's possible to find a dimensionless boundary integral representation for the velocity vector $\boldsymbol{u}(\boldsymbol{x}_o)$, just at the drop surface in the form (Rallison and Acrivos (1978) & Oliveira (2007))

$$\boldsymbol{u}(\boldsymbol{x}_o) \quad = \quad Ca\,\boldsymbol{u}^{\infty}(\boldsymbol{x}_o) - \frac{1}{4\pi}\int_S \kappa(\boldsymbol{x})\boldsymbol{\mathcal{G}}(\boldsymbol{x}-\boldsymbol{x}_o)\cdot\boldsymbol{n}(\boldsymbol{x})\,dS(\boldsymbol{x}). \tag{1}$$

In Equation(1), the integral is taken over the drop surface S; $\boldsymbol{x}_o$ is a point over S; Ca represent the capillary number, defined by $Ca = \dot{\gamma}_o a\mu/\sigma$, where $\dot{\gamma}_o$ is the characteristic shear rate, a is the drop ratio and σ the interfacial tension coefficient; $\boldsymbol{u}^{\infty}(\boldsymbol{x}_o)$ refers to the non-disturbed flow at the position $\boldsymbol{x}_o$; $\kappa(\boldsymbol{x})$ and $\boldsymbol{n}(\boldsymbol{x})$ are the local curvature and normal vector of the the drop surface, respectively. The tensorial function $\boldsymbol{\mathcal{G}}(\boldsymbol{x}-\boldsymbol{x}_o)$ is the Stokeslet, defined by

$$\boldsymbol{\mathcal{G}}(\boldsymbol{x}-\boldsymbol{x}_o) = \frac{1}{r}\boldsymbol{I} + \frac{(\boldsymbol{x}-\boldsymbol{x}_o)(\boldsymbol{x}-\boldsymbol{x}_o)}{r^3}, \tag{2}$$

where, $\boldsymbol{I}$ is the identity tensor and $r = \sqrt{(\boldsymbol{x}-\boldsymbol{x}_o)\cdot(\boldsymbol{x}-\boldsymbol{x}_o)}$. The integral representation of the velocity field given by Eq.(1) and Eq.(2) are singular for $\boldsymbol{x} = \boldsymbol{x}_o$, as can bee seen in the definition of the Stokeslet.

2.2 Singularity subtraction

The singularity subtraction of the kernel of the integral in Eq.(1) is an algebraic procedure which allows the surface integrals, present in the integral representation of a Stokes flow, to be numerically evaluated with precision (Zinchenko *et al.*, 1997). The continuity equation for the drop surface in the boundary integral form is give by (Rallison and Acrivos (1978) & Oliveira (2007))

$$\int_S \boldsymbol{\mathcal{G}}(\boldsymbol{x}-\boldsymbol{x}_o)\cdot\boldsymbol{n}(\boldsymbol{x})\,dS(\boldsymbol{x}) = \boldsymbol{0}. \tag{3}$$

Considering that $\kappa(\boldsymbol{x}_o)$ is a constant in relation to the integration with respect to the variable $\boldsymbol{x}$, it is possible to write that

$$\frac{1}{4\pi}\int_S \kappa(\boldsymbol{x}_o)\boldsymbol{\mathcal{G}}(\boldsymbol{x}-\boldsymbol{x}_o)\cdot\boldsymbol{n}(\boldsymbol{x})\,dS(\boldsymbol{x}) = \boldsymbol{0}. \tag{4}$$

Subtracting (4) from (1) one obtains

$$\boldsymbol{u}(\boldsymbol{x}_o) \quad = \quad Ca\,\boldsymbol{u}^{\infty}(\boldsymbol{x}_o) - \frac{1}{4\pi}\int_S [\kappa(\boldsymbol{x})-\kappa(\boldsymbol{x}_o)]\boldsymbol{\mathcal{G}}(\boldsymbol{x}-\boldsymbol{x}_o)\cdot\boldsymbol{n}(\boldsymbol{x})\,dS(\boldsymbol{x}). \tag{5}$$

The Equation (5) provides a boundary integral representation for the velocity vector over the drop surface with an eased singularity in the point $\boldsymbol{x}_o$. Strictly, the integral in Eq.(5) is still singular for $\boldsymbol{x} = \boldsymbol{x}_o$. However, the singularity subtraction procedure assures that numerator and denominator of the fraction resultant of the product $[\kappa(\boldsymbol{x})-\kappa(\boldsymbol{x}_o)]\boldsymbol{\mathcal{G}}(\boldsymbol{x}-\boldsymbol{x}_o)$ have the same order of magnitude, promoting more accurate numerical calculations.

3. NUMERICAL METHODOLOGY

3.1 Spatial Discretization

The spatial discretization of the drop interface is carried out by positioning control points over the surface S. These points are connected forming triangular elements, in a typical finite element mesh, which is initially spherical. The method used to generate the initial mesh starts from a regular icosahedron. Each face of the initial solid is subdivided in equilateral triangles, and each new node is projected over the sphere which circumscribes the icosahedron. The mesh refinement is controlled by a parameter f, which represents the number of subdivisions of the edge of the icosahedron, as suggest Fig.(1).

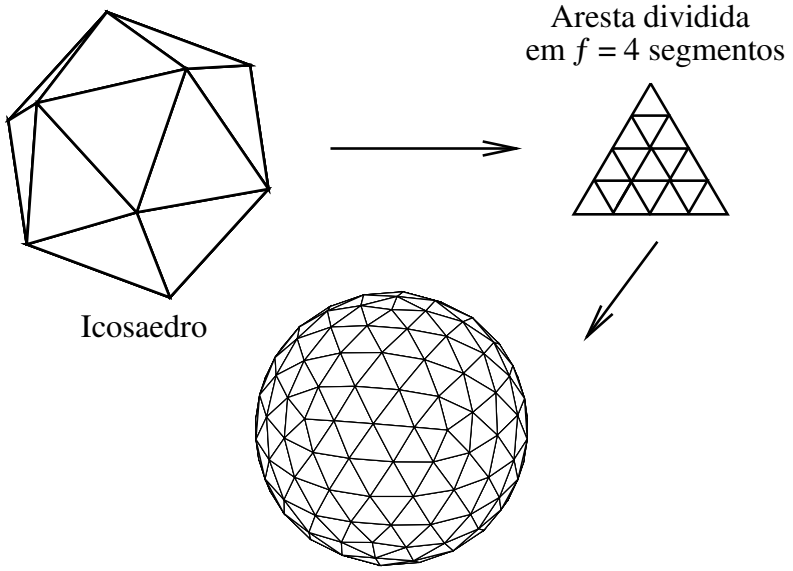

Figure 1. Process of mesh generation on a spherical surface from a regular icosahedron.

The quantities of the mesh are related to f, so that $N_\Delta = 20f^2$, $N_\bullet = 10f^2 + 2$ and $N_\ell = 30f^2$, where N_Δ, $N_\bullet$ and N_ℓ are the number of triangular elements, nodes and edges, respectively. Subdividing the surface into triangular elements is convenient to calculate the integrals associated to the Boundary Integral Method. Thus, the control points are connected using a data structure that consists of a table of coordinates of the nodes and a table of connection of the elements, where the nodes that make up each triangle on the surface are listed in counterclockwise direction. Structures to describe the neighboring elements and the neighboring nodes of each node on the surface are also required. These tables were also generated in the counterclockwise direction. It is important to note that the number of elements and neighboring nodes of each control point can vary between 5 and 6, due to the method of mesh generation.

This discretization process is widespread in the literature ((Loewenberg and Hinch, 1996), (Zinchenko *et al.*, 1997), (Cunha and Loewenberg, 2003), (Cunha *et al.*, 2003) & (Bazhlekov *et al.*, 2004)) for its simplicity and the good quality of mesh generated, by having almost equilateral triangles and little variation in the number of neighboring elements.

To calculate the normal vector and mean curvature at each local node in the mesh is possible to use a variation of Stokes' Theorem such that

$$\bar{\kappa}(\boldsymbol{x}_c)\boldsymbol{n}(\boldsymbol{x}_c) = -\frac{1}{S_x}\int_C \boldsymbol{t}\, d\ell, \tag{6}$$

where S_x is a surface that contains the control point $\boldsymbol{x}_c$, C is the contour of this surface and $\boldsymbol{t}$ is a unit vector tangent to the triangular face and orthogonal to C path, as illustrated in Fig.2. This method will be referred as *Line-Integral Method* in the course of this text and was developed by Loewenberg and Hinch (1996).

The definition of the trajectory C is very relevant to the accuracy achieved in calculating the curvature by the Line-Integral Method. Loewenberg and Hinch (1996) have defined C as the trajectory passing through the bisectors lines [1] of each neighbor triangle to the node $\boldsymbol{x}_c$. However, this procedure becomes inaccurate when the triangle is obtuse, because, in this case, the circumcenter of the triangle is outside its perimeter. Therefore, this definition is not complex enough to be used in a code to simulate arbitrarily deforming drops. Alternatively, the contour of S_x can be defined by the path consisting of segments joining the midpoints of the edges of the triangle to its centroid $\boldsymbol{x}_o$ (Bazhlekov *et al.*, 2004). Thus, and using the nomenclature used in Fig.3, the point $\boldsymbol{x}_o$ is always internal to the triangle.

The line integral in Eq.(6) is calculated by summing the contributions of each neighbor triangle to the node. Using the nomenclature described in Fig.3 and knowing the spatial coordinates of control points $\boldsymbol{x}_c$, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ it follows that $\boldsymbol{x}_o = (\boldsymbol{x}_c + \boldsymbol{x}_1 + \boldsymbol{x}_2)/3$, $\boldsymbol{a}_1 = (\boldsymbol{x}_1 - \boldsymbol{x}_c)/2$, $\boldsymbol{a}_2 = (\boldsymbol{x}_2 - \boldsymbol{x}_c)/2$, $\boldsymbol{b}_1 = \boldsymbol{x}_c - \boldsymbol{x}_o - \boldsymbol{a}_1$ and $\boldsymbol{b}_2 = \boldsymbol{x}_o - \boldsymbol{x}_c - \boldsymbol{a}_2$. The normal vector to the triangle is $\boldsymbol{n} = (\boldsymbol{a}_1 \times \boldsymbol{a}_2)/\|\boldsymbol{a}_1 \times \boldsymbol{a}_2\|$. The tangent vectors are calculated as $\boldsymbol{t}_1 = \boldsymbol{b}_1 \times \boldsymbol{n}$ and $\boldsymbol{t}_2 = \boldsymbol{b}_2 \times \boldsymbol{n}$. Thus, they have the desired orientation and norm equal to the length of the segment joining $\boldsymbol{a}_1$ to $\boldsymbol{x}_o$, in the case of $\boldsymbol{t}_1$, and $\boldsymbol{a}_2$ to $\boldsymbol{x}_o$, in the case of $\boldsymbol{t}_2$. Therefore, it's possible to write

$$\int_C \boldsymbol{t}\, d\ell = \sum \boldsymbol{t}_i, \tag{7}$$

[1] Bisector line: Segment joining the midpoint of an edge of the triangle to its circumcenter.

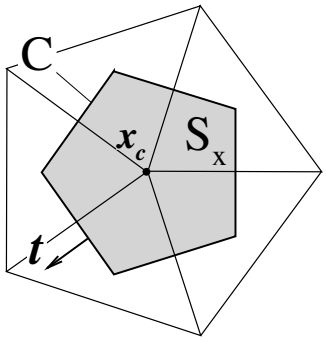

Figure 2. Geometric definitions for calculating the local mean curvature and normal vector by the Line-Integral Method.

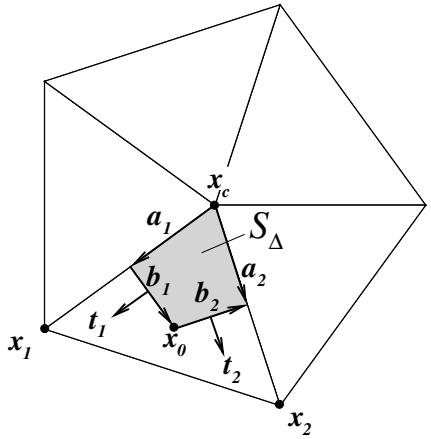

Figure 3. Details of the auxiliary vectors to calculate the local mean curvature by the Line-Integral Method.

where the sum is performed over all tangent vectors to all elements of the neighboring node.

3.2 Velocity Calculation

The velocity over the droplet surface is determined by solving the Eq.(5). It is necessary to find numerical solutions for surface integrals of vector equations of the form $\boldsymbol{F}(\boldsymbol{x}_o) = \int_S \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{x}_o)\, dS(\boldsymbol{x})$, where $\boldsymbol{f}(\boldsymbol{x})$ is a vectorial function of $\boldsymbol{x}$ with $\boldsymbol{x}_o$ as parameter. After discretization of the droplet surface into triangles, the solution of the integral over S can be obtained by summing the integrals over the elements defined by the surface $S^e,\ e = 1 \ldots N_\Delta$, so that

$$\int_S \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{x}_o)\, dS(\boldsymbol{x}) = \sum_{e=1}^{N_\Delta} \int_{S^e} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{x}_o)\, dS(\boldsymbol{x}). \tag{8}$$

For the integration over the surfaces S^e we used the Trapezoidal Rule to calculate integrals over a triangle of vertices $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ e $\boldsymbol{x}_3$ in space, so that

$$\int_S \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{x}_o)\, dS(\boldsymbol{x}) \approx \sum_{e=1}^{N_\Delta} \frac{S^e}{3} \sum_{i=1}^{3} \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{x}_o). \tag{9}$$

The Equation(9) introduces an error $\mathcal{O}(N_\Delta^{-1})$. This is the same numerical error in calculating the curvature by the Line-Integral Method with triangular elements.

The natural way to calculate the Eq.(9) is visit each element and add their contributions. However, considering that in most cases each node belongs to six triangles, the value of the integrand at each node is calculated repeatedly six times. One way to solve this problem is to perform a scan by nodes, such that

$$\int_S \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{x}_o)\, dS(\boldsymbol{x}) \approx \frac{1}{3} \sum_{i=1}^{N_\bullet} S_i^{viz} \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{x}_o), \tag{10}$$

where S_i^{viz} is the sum of the edges of neighboring elements of the node $\boldsymbol{x}_i$. Using this method to calculate the surface integrals, the time per iteration drops to about 17% of the time it would take if the integration was performed according to Eq.(9).

4. EVOLUTION OF DROP SHAPE - RELAXATION METHOD

The temporal evolution of the position of material points on the droplet surface is made by a second order Euler algorithm applied to the solution of equation $d\boldsymbol{x}/dt = \boldsymbol{u}$, where the velocity vector $\boldsymbol{u}$ is given by Eq.(5). However, the Lagragian character of the spatial discretization, such that each point over the mesh is a material particle, gives rise to a problem of instability of the mesh. Even if the shape of the drop stops to vary, the flow on its surface still exists (as well as inside and outside the drop). Thus, the control points are always in motion, distorting the elements until the mesh collapse. A possible solution to this problem is applying a mesh relaxation method. Such methods are procedures used to remove the time dependence of the position of the nodes over the drop surface, such that the mesh does not depend of the flow history. Tanzosh *et al.* (1992) report this problem and suggest that the evolution of the mesh points must be made using only the component of the velocity vector orthogonal to the surface. In fact, this procedure reduces significantly the distortion of the mesh, but doesn't solve the problem, especially in long time simulations. Loewenberg and Hinch (1996) studied the control points by changing the evolutionary equation for $d\boldsymbol{x}/dt = \boldsymbol{u} + \boldsymbol{\Pi}$, where $\boldsymbol{\Pi}$ is a tangential velocity arbitrarily defined in order to contain the mesh distortion. The tangential velocity $\boldsymbol{\Pi}$ is calculated locally, so that the processing time needed for the relaxation procedure is $\mathcal{O}(N_\Delta)$. This method avoids the mesh distortion during long time simulations. Moreover, the definition of $\boldsymbol{\Pi}$ can take into account factors such as local mean curvature, variations in the area of elements, distance between two or more drops, velocity on the surface, concentration of surfactants and others, thus demonstrating an active approach to relaxation. Therefore, the reorganization of the control points over the mesh can consider regions of high curvature or high gradients of concentration of surfactants, for example. Zinchenko *et al.* (1997) used an algorithm based on minimizing an energy defined according to the length of the edges of the mesh. Their method is independent of any physical parameter of the problem (such as local curvature and concentration of surfactants, for example) and is therefore classified as a passive method. To the extent possible, it tends to hold the mesh elements with the same size. According to the authors, under optimized conditions, the procedure is $\mathcal{O}(N_\Delta)$, although it is a global method.

In this work we employed a method similar to that of Loewenberg and Hinch (1996). Thus, instead of solving the simple evolution equation, we used

$$\frac{d\boldsymbol{x}}{dt} = (\boldsymbol{u} \cdot \boldsymbol{n})\boldsymbol{n} + \boldsymbol{\Pi}, \tag{11}$$

where the value of $\boldsymbol{\Pi}$, for each node $\boldsymbol{x}_i$ of the mesh, is given by (Loewenberg and Hinch, 1996)

$$\boldsymbol{\Pi}(\boldsymbol{x}_i) = \psi^{-1} \frac{N_\Delta^{3/2}}{1+\lambda} [\boldsymbol{I} - \boldsymbol{n}(\boldsymbol{x}_i)\boldsymbol{n}(\boldsymbol{x}_i)] \sum_j \left(C_{r1} + C_{r2} |\bar{\kappa}_j|^{3/2} \right) S_j^e (\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{12}$$

where ψ, C_{r1} and C_{r2} are positives constants, j is an index that counts the neighboring nodes to the control point $\boldsymbol{x}_i$ and S_j^e is the area of the element on the left of the edge formed by $\boldsymbol{x}_i - \boldsymbol{x}_j$. According to the authors, Eq.(12) is heuristic and careful optimizations are unnecessary. The expression $\boldsymbol{I} - \boldsymbol{n}(\boldsymbol{x}_i)\boldsymbol{n}(\boldsymbol{x}_i)$ ensures that $\boldsymbol{\Pi}$ is a tangent velocity to the surface. The constant C_{r1} controls the tendency of the mesh in keeping the elements with the same area and C_{r2} controls the trend of the relaxation to concentrate the points in regions of greatest curvature. For the speed of relaxation be sensitive to other parameters is necessary to add some similar terms. For example, if there is a need to concentrate points in nearby regions of other drops, the term $C_{r3}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|/h(\boldsymbol{x}_j)$, where $h(\boldsymbol{x}_j)$ is the distance between $\boldsymbol{x}_i$ and the nearest drop, should be included. The constant is related to the magnitude of $\boldsymbol{\Pi}$. In this work we used $\psi = 300$, $C_{r1} = 10$ e $C_{r2} = 5$.

Assume that $\boldsymbol{x}^p$ denotes the points over the surface of the drop at the instant p, N_t the number of time steps and $\boldsymbol{m}(\boldsymbol{x}^p) = (\boldsymbol{u} \cdot \boldsymbol{n})\boldsymbol{n} + \boldsymbol{\Pi}$. The use of a second order Euler method for the evolution of the equation Eq.(11), where an initial condition $\boldsymbol{x}^0$ is known, is based on the algorithm that follows

`Since` $p = 0$ `until` $p = N_t$ `do:`

$$\left[\begin{array}{ll} 1. & \boldsymbol{k}_1 \leftarrow \boldsymbol{m}(\boldsymbol{x}^p) \\ 2. & \boldsymbol{x}' \leftarrow \boldsymbol{x}^p + \Delta t \boldsymbol{k}_1 \\ 3. & \boldsymbol{k}_2 \leftarrow \boldsymbol{m}(\boldsymbol{x}') \\ 4. & \boldsymbol{x}^{p+1} \leftarrow \boldsymbol{x}^p + \frac{1}{2}(\boldsymbol{k}_1 + \boldsymbol{k}_2). \end{array}\right. \tag{13}$$

It is worth mentioning that in the first and third steps of the previous algorithm is necessary to calculate the normal vector, the curvature, the velocity vector $\boldsymbol{u}$ and the relaxation speed $\boldsymbol{\Pi}$. The dimensionless time step is adjusted considering the physical parameters of the problem, such that $\Delta t = min\left\{\frac{\Delta x}{(\lambda+1)Ca}, \omega\Delta x, 10^{-2}\right\}$ where $\Delta x = \sqrt{4\pi/N_\Delta}$, Ca denotes the capillary number and ω is the excitation frequency of the flow, if applicable. If the time evolution is accomplished simply by using the algorithm in Eq.(13) there isn't guarantee that the relaxation process was completely done in making the mesh independent of the flow history. For this, the points on the droplet surface are subjected to a new process of time evolution in which only the speed of relaxation is used. Then, are defined $\boldsymbol{m}^*(\boldsymbol{x}) = \boldsymbol{\Pi}$ and the parameter of the relaxation evolution t^*. This process is conducted in a virtual context, and its increments Δt^* are fictitious and do not contribute to advancing real time during simulation. To signal the end of the relaxation of the mesh we used the derivative with respect to t^* of the standard deviation of the areas of mesh elements. The relaxation algorithm can be described as follows

$$\left[\begin{array}{ll} 1. & \boldsymbol{k}_1 \leftarrow \boldsymbol{m}^*(\boldsymbol{x}, t) \\ 2. & \boldsymbol{x}' \leftarrow \boldsymbol{x} + \Delta t^* \boldsymbol{k}_1 \\ 3. & \boldsymbol{k}_2 \leftarrow \boldsymbol{m}^*(\boldsymbol{x}') \\ 4. & \boldsymbol{x} \leftarrow \boldsymbol{x} + \frac{1}{2}(\boldsymbol{k}_1 + \boldsymbol{k}_2) \\ 5. & \texttt{If } |std'(S^e)| < tol_r \texttt{ then} \rightarrow \textbf{END} \\ 6. & \texttt{Return to step 1} \end{array}\right. \tag{14}$$

where $std'(S^e)$ represents the derivative of the standard deviation with respect to t^*. We used $\Delta t^* = \Delta t$ though, apparently, these parameters are independent. The tolerance used was $tol_r = 10^{-3}$. Qualitatively, the result of the relaxation process can be seen in Fig.(4). In Figures (4a) and (4b) are shown two views of a mesh over a drop that deformed during a period of time without relaxation (using only the normal component of the velocity vector). In Figures (4c) and (4d) are shown the same views of the previous mesh after the relaxation process described in Eq.(14). We can observe that the relaxed mesh has a more uniform distribution of points on the surface. To obtain this result were required 250 iterations.

In this case, the relaxation acted on a mesh that originally evolved without any relaxation. However, in the simulations of this work, the relaxation process is applied to the mesh at each time step. To show the effect of this procedure, three meshes were generated under the same conditions of the mesh of Fig.(4). The first mesh relaxed only by adding $\boldsymbol{\Pi}$ to $(\boldsymbol{u}\cdot\boldsymbol{n})\boldsymbol{n}$, just like Eq.(13). The other meshes, moreover, undergo the process illustrated in Eq.(14) with different numbers of iterations (for that, the fifth step should be deleted and the number of iterations must be controlled), at each time step. Finally, the resulting meshes were completely relaxed and the standard deviation of the area of elements of the mesh, $sdt(S^e)$, was studied as a function of the number of iterations, n_{relax}, of this final relaxation process. The result of this comparison is shown in Fig.(5).

It is observed in Fig.(5) that using 10 iterations of relaxation per time step the distribution of points on the mesh, practically, is not altered by the extra relaxation. This suggests that the Lagragian characteristic of the mesh is canceled by the relaxation process and that the position of the nodes on the surface is independent of the flow history. In practice, when the stop criterion of the fifth step of the algorithm (14) is used, the number of relaxation iterations rarely exceeds 4.

It can be seen in Fig.(5) that $std(S^e)$ stabilizes at a value above its minimum value ($std(S^e)_{min} = 1,3 \times 10^{-3}$). This indicates that the relaxation method is modifying the mesh so that the size distribution of the elements is not the most homogeneous possible. This fact is associated to the tendency of the process of concentrating points in regions of greater curvature. To demonstrated this, the mesh in Fig.(4a) was relaxed with $C_{r2} = 0$. Thus, the only effect of relaxation is the homogenization of the elements size. In Figure (6) are compared the evolution of the standard deviation of S^e as function of n_{relax} for $C_{r2} = 0$ and $C_{r2} = 5$.

We can note that in the case of $C_{r2} = 0$ the stabilization of the mesh happens at the minimum value of $std(S^e)$, indicating that in this way, the procedure distributes the points in order to make the mesh more homogeneous as possible regarding the size of the elements. It was also noted that this stabilization needs more iterations to be obtained. In Figure (6) we can observe that the difference between the final mesh obtained is discreet, but noticeable, with greater concentration of points in the region of greatest curvature in the second case, where $C_{r2} = 5$.

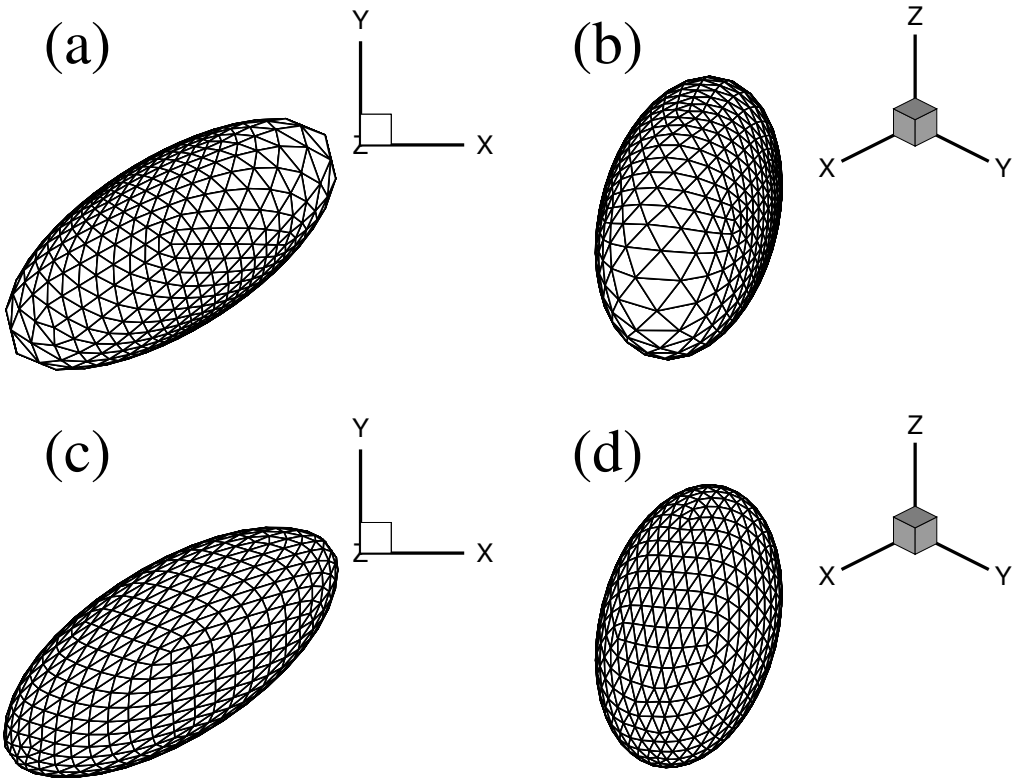

Figure 4. Mesh over a drop: $f = 8$, $N_\Delta = 1280$. (a) and (b) Views of the resulting mesh after a brief time without relaxation ($Ca = 1$, $\lambda = 1$, after 100 iterations). (c) and (d) resulting mesh after relaxation.

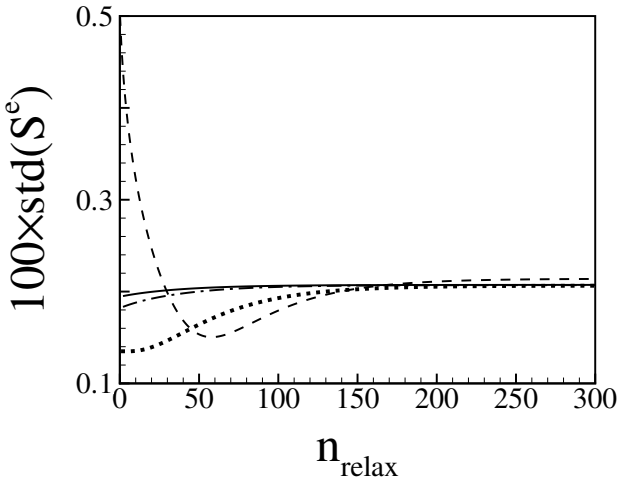

Figure 5. Standard deviation of the areas of mesh elements as function of the number of iterations of the relaxation process. Dashed line: initial mesh generated without relaxation; dotted line: initial mesh generated using only Eq.(13); dashes and dots line: initial mesh generated after 5 iterations of extra relaxation; full line: initial mesh generated after 10 iterations of extra relaxation;

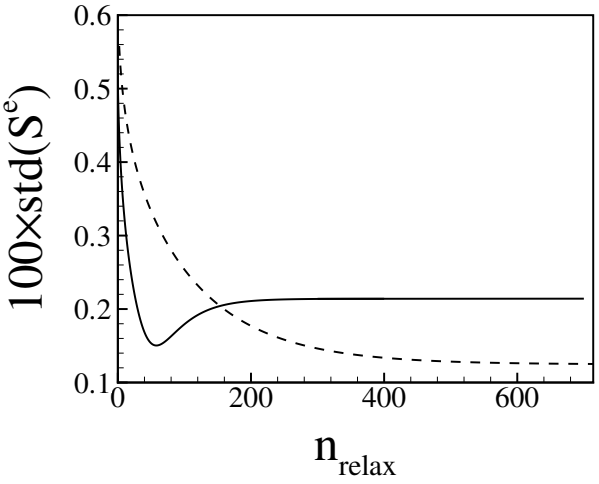

Figure 6. Evolution of $std(S^e)$ during the relaxation of the mesh illustrated in Fig.(4). Full line: $C_{r2} = 5$; Dashed line: $C_{r2} = 0$.

5. CONCLUSION

In the present work we have used a boundary integral formulation for the velocity vector over the surface of a drop, to simulate the evolution of the drop shape when it is subjected to a simple shear flow. A procedure to partially remove of the integral kernel singularity is described. Some details of the numerical procedures, including the spatial discretization, the methodology of calculation of the local curvature and normal vector, and the velocity calculation method are presented.

The temporal evolution of the drop shape is done by a second order Euler method. This procedure leads to numerical instabilities related to the progressive distortion of the superficial elements of the spatial grid. This phenomena is closely related to the fact that the positions of the grid nodes over drop surface are dependent of the history of the flow. Relaxation methods designed to remove this dependence are available in the literature. In this work, we show that a classical relaxation method, largely applied in the literature, actually doesn't completely remove the dependence of the grid in relation of the history of the flow. We propose an extension of that method which is capable of completely remove this dependence, in such a way that the nodes distribution over the drop surface becomes only a function of local geometric quantities, like the curvature.

6. ACKNOWLEDGEMENTS

The authors are thankful to Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq for financial support this work.

7. REFERENCES

Bazhlekov, I.B., Anderson, P.D. and Meijer, H.E.H., 2004. "Nonsingular boundary integral method for deformable drops in viscous flows". *Physics of Fluids*, Vol. 16, No. 4, pp. 1064–1081.

Cunha, F.R. and Loewenberg, M., 2003. "A study of emulsion expansion by a boundary integral method". *Mechanics Research Communications*, Vol. 30, No. 639.

Cunha, F.R., Sousa, A.J. and Loewenberg, M., 2003. "A mathematical formulation of the boundary integral equations for a compressible stokes flow". *Computational and Applied Mathematics*, Vol. 22, No. 53.

Einstein, A., 1906. "Eine neue bestimmung der moleküledimensionen". *Ann. Physik*, Vol. 19, No. 289.

Ladyzheskaya, O.A., 1969. *The mathematical theory of viscous inclompressible flow*. New York.

Loewenberg, M. and Hinch, E.J., 1996. "Numerical simulations of a concentrated emulsion in shear flow". *Journal Fluid Mechanics*, Vol. 321, No. 395.

Oliveira, T.F., 2007. *Microhidrodinâmica e reologia de emulsões*. Ph.D. thesis, Pontifícia Universidade Católica do Rio de Janeiro.

Rallison and Acrivos, 1978. "A numerical study of the deformation and burst of a viscous drop in an extensional flow". *Journal Fluid Mechanics*, Vol. 89, No. 191.

Tanzosh, J., Manga, M. and Stone, H.A., 1992. "Boundary integral methods for viscous free-boundary problems: Deformation of sinlgle and multiple fluid-fluid interfaces". In *BETECH 92 conference*. Albuquerque, New Mexico.

Taylor, G.I., 1932. "The viscosity of a fluid containing small drops of another fluid". *Proc. R. Soc. London Ser. A*, Vol. 138, No. 41.

Taylor, G.I., 1934. "The formation of emulsions in definable fields of flow". *Proc. R. Soc. London Ser. A*, Vol. 146, p. 501.

Zinchenko, A.Z., Rother, M.A. and Davis, R.H., 1997. "A novel boundary-integral algorithm for viscous interaction of deformable drops". *Physics of Fluids*, Vol. 9, No. 6, pp. 1493–1511.

8. RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.