HAL
open science

Simulation of vesicle using level set method solved by high order finite element

Vincent Doyeux, Vincent Chabannes, Christophe Prud'Homme, Mourad Ismail

To cite this version:

Vincent Doyeux, Vincent Chabannes, Christophe Prud'Homme, Mourad Ismail. Simulation of vesicle using level set method solved by high order finite element. ESAIM: Proceedings, 2012, 38, pp.335-347. 10.1051/proc/201238018 . hal-00665007

HAL Id: hal-00665007

https://hal.science/hal-00665007

Submitted on 1 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ESAIM: PROCEEDINGS, Vol. ?, 2012, 1-10
Editors: Will be set by the publisher

SIMULATION OF VESICLE USING LEVEL SET METHOD SOLVED BY HIGH ORDER FINITE ELEMENT

Vincent Doyeux[1], Vincent Chabannes[2], Christophe Prud'homme[2] and Mourad Ismail[1]

Abstract. We present a numerical method to simulate vesicles in fluid flows. This method consists of writing all the properties of the membrane as interfacial forces between two fluids. The main advantage of this approach is that the vesicle and the fluid models may be decoupled easily. A level set method has been implemented to track the interface. Finite element discretization has been used with arbitrarily high order polynomial approximation. Several polynomial orders have been tested in order to get a better accuracy. A validation on equilibrium shapes and "tank treading" motion of vesicle have been presented.

1. Introduction

Simulations of individual biological objects have been of major interest for several years now. Indeed, many biological fluids follow complex laws due to the behaviors of their individual components. The understanding of those microscopic behaviors can lead to a better knowledge of the macroscopic behavior of complex biological fluids. Blood is a good example of such a fluid, red blood cells (**RBC**) are biological objects with complex individual behaviors which lead to a complex macroscopic rheology of the blood.

Vesicles are two-layer phospholipidic membrane [1] having the property to mimic some mecanical behaviors of **RBC** under flow. They are often used as a simple model for **RBC** and thus, these objects are studies experimentally [2,3], theoretically [4] and numerically [5,6].

The numerical simulation of vesicles consists of a two-fluid flow simulation (for the inner and outer fluids) and a fluid-structure interaction (for the membrane-fluids interaction). For these reasons, several methods have been developped such as lattice Boltzmann methods [7], integral boundary method [8], phase field method [9], level set method [6,10,11] and a numerical method based on a contact algorithm to simulate RBC clusters in bifurcations [12]. These methods have their own advantages and drawbacks. Many numerical and mathematical investigations have to be done to explore the large field of vesicle simulation, improving the existing methods and finding new ones. The following paper takes place in this context. We propose to implement a method close to the one of [6] using finite element methods (**FEM**). The main advantage of coupling the level set method and the **FEM** is that it allows for more flexibility, simplicity and accuracy in the handling of the geometry of the computational domains. Even though it is common to use the lowest order continuous finite element approximation for this type of coupling, we have used various higher order approximations and found a good

[1] Université Grenoble 1 / CNRS, Laboratoire Interdisciplinaire de Physique / UMR 5588. Grenoble, F-38041, France e-mail: vincent.doyeux@ujf-grenoble.fr mourad.ismail@ujf-grenoble.fr

[2] Université Grenoble 1 / CNRS, Laboratoire Jean Kuntzman / UMR 5224 Grenoble, F-38041, France e-mail: vincent.chabannes@ujf-grenoble.fr christophe.prudhomme@ujf-grenoble.fr

© EDP Sciences, SMAI 2012

compromise between accuracy and computational cost. Although the **FEM** has many advantages, several issues remains to be solved, namely (i) high order derivative (up to fourth order) of the level set function, (ii) storing the stretching information of the interface. We propose solutions that indeed overcome those issues and validate our model on existing results and known behaviors of vesicles under flow.

2. Two-fluid flow

Simulation of vesicles always requires to be able to solve a two-fluid flow. Moreover, since the interface between the two fluids is a complex object (membrane), a special care must be taken to track it accurately. To reach this goal, level set methods have been studied for several years and it has been proved to give good results for tracking interfaces, especially in two-fluid or two-phase flows [13, 14]. In this section, we first introduce the necessary standard vocabulary regarding levelset methods and then we describe the fluid model which we used.

2.1. Level set generalities

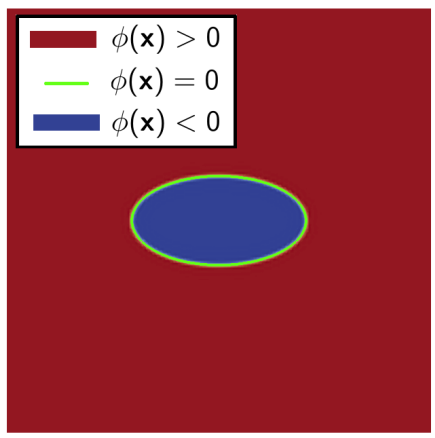


Figure 1. Sign of ϕ according to the position of the interface

Let's define a bounded domain $\Omega \subset \mathbb{R}^p$ $(p = 2, 3)$ decomposed into two sub-domains Ω_1 and Ω_2. We denote Γ the interface between the two partitions. The level set method, as described in [13–15], defines a continuous scalar function ϕ (the level set function) in Ω.

- The sub-domains Ω_1 and Ω_2 and the interface Γ are defined thanks to the level set function (see figure 1) :

$$\begin{aligned} \Omega_1 &= \{\boldsymbol{x} | \phi(\boldsymbol{x}) > 0\}, \\ \Omega_2 &= \{\boldsymbol{x} | \phi(\boldsymbol{x}) < 0\}, \\ \Gamma &= \{\boldsymbol{x} | \phi(\boldsymbol{x}) = 0\}. \end{aligned}$$

- The evolution of the level set is governed by the following advection equation :

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0, \tag{1}$$

  where $\boldsymbol{u}$ stands for a given divergence-free velocity field (the fluid velocity in our case).
- Usually, ϕ is taken as a signed distance function. So that $|\nabla \phi| = 1$.

It's known that the advection equation (1) does not conserve the property $|\nabla\phi| = 1$. To overcome this problem, we use a fast marching method (**FMM**) when $|\nabla\phi|$ goes far from 1. More precisely, **FMM** resets ϕ as a distance function without moving the interface (see [16] more details). We now introduce the definition of the smoothed delta and Heaviside functions that allow easy computations on the interface Γ without having to explicitely compute the interface

$$H_\varepsilon(\phi) = \begin{cases} 0, & \phi \leq -\varepsilon, \\ \frac{1}{2}\left(1 + \frac{\phi}{\varepsilon} + \frac{\sin(\frac{\pi\phi}{\varepsilon})}{\pi}\right), & -\varepsilon \leq \phi \leq \varepsilon, \\ 1, & \phi \geq \varepsilon, \end{cases}$$

$$\delta_\varepsilon(\phi) = \begin{cases} 0, & \phi \leq -\varepsilon, \\ \frac{1}{2\varepsilon}\left(1 + \cos(\frac{\pi\phi}{\varepsilon})\right), & -\varepsilon \leq \phi \leq \varepsilon, \\ 0, & \phi \geq \varepsilon, \end{cases}$$

where ε is a parameter defining a "numerical thickness" of the interface (a typical value of ε is 1.5 times the mesh size).

Using these two functions allows then to define (*e.g.* physical) properties having different values in Ω_1 and Ω_2. For example we define the viscosities and densities of the inner and outer fluid as follows:

$$\begin{aligned} \rho_\phi &= \rho_2 - (\rho_2 - \rho_1)H_\varepsilon \\ \mu_\phi &= \mu_2 - (\mu_2 - \mu_1)H_\varepsilon. \end{aligned}$$

As to the integrals over Γ, they are approximated by integrals over the whole domain Ω as follows:

$$\int_\Gamma 1 \simeq \int_\Omega \delta_\varepsilon$$

2.2. Navier Stokes equations

We consider that both the internal and the external fluids are governed by incompressible Navier Stokes equations :

$$\rho_\phi\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right) - \nabla\cdot(\mu_\phi D(\boldsymbol{u})) + \nabla p = \boldsymbol{F} \quad \text{in } \Omega, \tag{2}$$

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{3}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \partial\Omega \tag{4}$$

where $\boldsymbol{u}$ is the velocity of the fluid, p its pressure, $D(\boldsymbol{u}) = \nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T$, and $\boldsymbol{F}$ some external force and $\boldsymbol{g}$ a given function related to the boundary conditions.

A validation of our two-fluid model based on Navier-Stokes equations and the level set advection has been presented in [17] on a bubble simulation benchmark. In this work we also consider the Stokes equations. Indeed, many blood flow applications are in a very low Reynolds number regime. Thus, we use a stationary Stokes model for the fluids

$$-\nabla\cdot(\mu_\phi D(\boldsymbol{u})) + \nabla p = \boldsymbol{F} \quad \text{in } \Omega, \tag{5}$$

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{6}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \partial\Omega \tag{7}$$

3. Vesicles Model

A vesicle is a system of two fluids separated by a membrane. It has been studied by many previous numerical studies, see section 1. In this work, we consider a model of membrane with a zero thickness and has three main properties :

- **Constant Volume** : as both fluids (inside and outside the vesicle) are incompressibles and as the membrane is not porous, it results that the total volume (surface in 2D) of the vesicle is constant in time
- **Bending Energy** : it costs some energy to bend the membrane. This energy has to be taken into account in the model.
- **Local Inextensibility** : the membrane has to keep the same area (perimeter in 2D) during the simulation time.

We choose to use the model introduced in [6, 18] in which the bending energy and the local inextensibility are represented by interfacial forces: the membrane is represented by a force distribution along the interface between inner and outer fluids. In the following section, we briefly describe some details of this model.

3.1. Bending Force

The bending energy has been introduced by Helfrich in [19] as :

$$E_c = \int_\Gamma \frac{k_B}{2}\kappa^2, \tag{8}$$

where k_B is the bending modulus of the membrane and κ is the local curvature of the interface. By differentiating the expression (8), Maitre *et al* [6, 18] found the resulting local force. It reads

$$\boldsymbol{F_c} = \int_\Gamma k_B \nabla \cdot \left[\frac{-\kappa^2}{2}\boldsymbol{n} + \frac{1}{|\nabla\phi|}(\mathbb{I} - \boldsymbol{n}\otimes\boldsymbol{n})\nabla\{|\nabla\phi|\kappa\}\right], \tag{9}$$

where $\mathbb{I}$ stands for the identity tensor and $\boldsymbol{n}$ stands for the normal unit vector to the interface Γ, *i.e.* $\frac{\nabla\phi}{|\nabla\phi|}$. This expression can be then rewritten in the level set formulation by replacing the normal vector by its expression (as a function of ϕ) and the integral over Γ by an integral in Ω. Thus, we obtain :

$$\boldsymbol{F_c} = \int_\Omega k_B \nabla \cdot \left[\frac{-\kappa^2}{2}\frac{\nabla\phi}{|\nabla\phi|} + \frac{1}{|\nabla\phi|}\left(\mathbb{I} - \frac{\nabla\phi\otimes\nabla\phi}{|\nabla\phi|^2}\right)\nabla\{|\nabla\phi|\kappa\}\right]\delta_\epsilon(\phi). \tag{10}$$

3.2. Inextensibility Force

In many models, the local area of the vesicle is conserved by adding a Lagrange multiplier which ensures the membrane inextensibility as in [5, 17, 20, 21]. In [22, 23], the authors show the possibility to get the quasi inextensibility of the membrane by adding an elastic force using the derivative of an elastic energy. They show that the stretching of the interface is recorded in the gradient modulus of ϕ. Indeed, under the assumption that the flow is incompressible, one can write a constitutive law of the stretching related to $|\nabla\phi|$. The elastic energy reads :

$$E_{el} = \int_\Omega E(|\nabla\phi|)\delta_\epsilon,$$

with $E(|\nabla\phi|)$ a constitutive law of the membrane such that $E(1) = 0$ (no initial stretching). A simple definition of $E(|\nabla\phi|)$ is a quadratic law : $E(|\nabla\phi|) = \frac{\Lambda}{2}(|\nabla\phi| - 1)^2$ so that $E'(\nabla\phi)$ (the derivative of E) is a linear law : $E'(|\nabla\phi|) = \Lambda(|\nabla\phi| - 1)$ as for a spring. In our case, we take $E'(|\nabla\phi|) = \Lambda\max(|\nabla\phi| - 1, 0)$ which ensures that there is no force when the stretching is not positive (as for a tensile), Λ represents the *stiffness* of the membrane,

it has to be taken large enough to keep almost constant area. By differentiating the energy, it has been shown in [6, 22] that one can write the elastic force as :

$$\boldsymbol{F_{el}} = \int_\Omega \left\{ \nabla E'(|\nabla\phi|) - \nabla \cdot \left[E'(|\nabla\phi|)\frac{\nabla\phi}{|\nabla\phi|} \right] \frac{\nabla\phi}{|\nabla\phi|} \right\} \delta_\epsilon(\phi). \tag{11}$$

Using forces to maintain incompressibility has some clear advantages: (i) we can add complexity to the model such as changing the constitutive law of E' (*e.g.* a more rigid part which could take into account the cytoskeleton of a red blood cell) or add a new force like a surface tension force, (ii) and the fact that the fluid solver is used as a "black box": the vesicle model is seen only through the external forcing terms, the density and viscosity which allows for easily changing the fluid solver — *e.g.* use a non-newtonian model. — However the main drawback is that this technique leads to a larger variation of the surface area than the ones using Lagrange multipliers. Note also that in this model, we need to keep the information of $\nabla\phi$. Reinitialization is then difficult since it resets $|\nabla\phi| = 1$ and *forgets* the stretching information. We explain how to get over this issue in section 4.3. The conservation of the total surface of the vesicle is better than the conservation of the local perimeter since it is enforced by the incompressibility of the fluid. Finally, we just have to add forcing terms $\boldsymbol{F_b}$ and $\boldsymbol{F_{el}}$ to equation (2) to simulate the presence of the membrane.

4. Implementation

4.1. Discretization

4.1.1. *Time Discretization*

For stability and accuracy reasons, we choose to use the Crank-Nicolson scheme (**CN**) to discretize equation (1). It reads

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + \frac{1}{2}(\boldsymbol{u} \cdot \nabla\phi^{n+1} + \boldsymbol{u} \cdot \nabla\phi^n) = 0. \tag{12}$$

We could use another scheme like backward differencing scheme of order 2, **BDF2**, but this method needs ϕ^{n-2} to compute ϕ^n and in this case, at reinitialization step, ϕ changes roughly and breaks the second order of **BDF2** scheme.

4.1.2. *Space Discretization*

Both the advection equation (12) and the Stokes equations (5)- (6) are descretized using the finite element method. To do that, we use the `C++` Finite Element Library `Feel++` [24–26] which allows us to use high order polynomial approximation and work indifferently in 1, 2 or 3 dimensions without changing the code except for the configuration of the problem.

The advection problem is known to be unstable when solved with finite element method without stabilization. Thus, we implemented several known methods as StreamLine Upwind Diffusion (SUPG), Galerkin Least Square (GLS), Subgrid Scale (SGS) [27–29] and Continuous Interior Penalty method (CIP) [30].

4.1.3. *Weak formulation*

The problem (1)-(2)-(3) is equivalent to find $(\boldsymbol{u}, p, \phi) \in V \times L^2_0(\Omega) \times H^1(\Omega)$ which verifies $\forall(\boldsymbol{v}, q, \psi) \in H^1_0(\Omega)^2 \times L^2_0(\Omega) \times H^1(\Omega)$:

$$\rho_\phi \int_\Omega \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right)\cdot \boldsymbol{v} + \mu_\phi \int_\Omega D(\boldsymbol{u}) : D(\boldsymbol{v}) - \int_\Omega p\nabla\cdot\boldsymbol{v} = \int_\Omega (\boldsymbol{F_c}(\phi) + \boldsymbol{F_{el}}(\phi))\cdot\boldsymbol{v}, \tag{13}$$

$$\int_\Omega q\nabla\cdot\boldsymbol{u} = 0, \tag{14}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \partial\Omega, \tag{15}$$

$$\int_\Omega \partial_t\phi\psi + \int_\Omega (\boldsymbol{u}\cdot\nabla\phi)\psi + \int_\Omega S(\phi,\psi) = 0, \tag{16}$$

where $S(\phi,\psi)$ stands for a stabilization term and $V = \{\boldsymbol{v} \in [H^1(\Omega)]^2 | \boldsymbol{v}|_{\partial\Omega} = \boldsymbol{g}\}$.

As to the discretization, we introduce $\mathbb{U}_h^{n+1}$, $\mathbb{P}_h^n$ and $\mathbb{Q}_h^m$ the corresponding discrete finite elements spaces depending on mesh size h and respectively based on Lagrange polynomials of degree $n+1$, n and m. To handle the fact that the pressure is known up to a constant, we introduce a pseudo compressibility term, ϵp with $\epsilon \approx 1e-6$, to the Stokes equations.

Let $(\boldsymbol{u}_h, p_h, \phi_h) \in \mathbb{U}_h^{n+1} \times \mathbb{P}_h^n \times \mathbb{Q}_h^m$ be the discretization of $(\boldsymbol{u}, p, \phi)$. The discrete version of problem (13)-(14)-(16) becomes : find $(\boldsymbol{u}_h, p_h, \phi_h) \in \mathbb{U}_h^{n+1} \times \mathbb{P}_h^n \times \mathbb{Q}_h^m$ which verifies $\forall(\boldsymbol{v}_h, q_h, \psi_h) \in \mathbb{U}_h^{n+1} \times \mathbb{P}_h^n \times \mathbb{Q}_h^m$:

$$\rho_\phi \int_\Omega \left(\frac{\partial \boldsymbol{u}_h}{\partial t} + (\boldsymbol{u}_h\cdot\nabla)\boldsymbol{u}_h\right)\cdot \boldsymbol{v}_h + \mu_\phi \int_\Omega D(\boldsymbol{u}_h) : D(\boldsymbol{v}_h) - \int_\Omega p_h\nabla\cdot\boldsymbol{v}_h = \int_\Omega (\boldsymbol{F_c}(\phi_h) + \boldsymbol{F_{el}}(\phi_h))\cdot\boldsymbol{v}_h, \tag{17}$$

$$\int_\Omega q_h\nabla\cdot\boldsymbol{u}_h + \epsilon p_h q_h = 0, \tag{18}$$

$$\boldsymbol{u}_h = \boldsymbol{g} \quad \text{on } \partial\Omega, \tag{19}$$

$$\int_\Omega \partial_t\phi_h\psi_h + \int_\Omega (\boldsymbol{u}_h\cdot\nabla\phi_h)\psi_h + \int_\Omega S(\phi_h,\psi_h) = 0. \tag{20}$$

4.2. **High order derivative**

The formulations of bending and elastic forces given by equations (10) and (11) could be problematic when using finite element approximation. Indeed, F_{el} needs a second order derivative of the level set. Moreover, recalling that κ is of second order derivative, F_c needs some 4th order derivative of ϕ. Differentiating several times ϕ leads to oscillations, especially when low order polynomials are used to approximate it.

In [11] the authors introduced two mixed variables ($\tilde{\psi} = \kappa|\nabla\phi|$ and $\tilde{\gamma} = \nabla|\nabla\phi|$) and two new equations to lower the degree of the derivatives. In [10], the authors used an augmented level set (as described in [31]) in which they solve by finite difference schemes an advection equation for ϕ and an other one for $\nabla\phi$. Moreover, three equations are solved at reinitialization step to take into account the higher derivatives of ϕ.
We choose to solve an additional advection equation to calculate the gradient modulus of ϕ. Let's introduce a scalar function $e = |\nabla\phi|$ which verifies

$$\partial_t e + \boldsymbol{u}\cdot\nabla e = -e\frac{\nabla\phi\otimes\nabla\phi}{|\nabla\phi|^2} : D(\boldsymbol{u}). \tag{21}$$

Note that using this equation, we lower the degree of derivative of one.

Another motivation to introduce the variable e as an independent parameter was that, as one can see from the expression (11), we need to keep the gradient modulus information during all the simulation time to compute properly the elastic force. Consequently, re-initializing the level set to a distance function would reset $|\nabla\phi|$ to 1 and thus artificially vanishing the elastic force. Using e as an independently advected variable allows us to

reinitialize the level set without introducing errors in the calculation of $\boldsymbol{F_{el}}$. However, there is still a problem to achieve 4th order derivative smoothly.

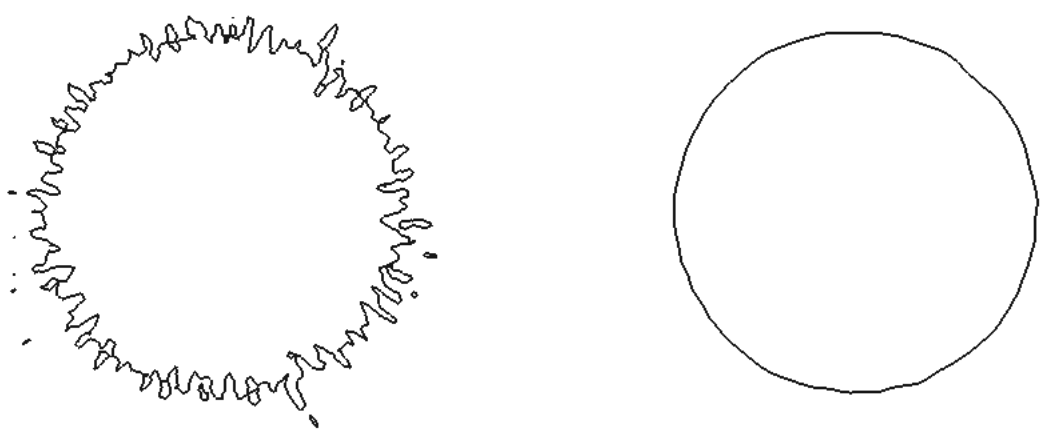
In order to solve this issue, one can do a L^2 projection of the gradient of any function to derive, which mean, having a gradient $\nabla f \in [L^2(\Omega)]^2$, find $\boldsymbol{f'} \in [L^2(\Omega)]^2$ such that $\forall \boldsymbol{g} \in [L^2(\Omega)]^2$

$$\int_\Omega \boldsymbol{f'} \cdot \boldsymbol{g} = \int_\Omega \nabla f \cdot \boldsymbol{g}. \tag{22}$$

This L^2 projection can lead to oscillations, but we can add a smoothing term proportional to a parameter ε_s which has to be small enough so that it does not perturb the accuracy of the projection. This projection leads to the following formulation : find $\boldsymbol{f'} \in [H^1(\Omega)]^2$, such that $\forall \boldsymbol{g} \in [H^1(\Omega)]^2$,

$$\int_\Omega \boldsymbol{f'} \cdot \boldsymbol{g} + \varepsilon_s \int_\Omega \nabla \boldsymbol{f'} : \nabla \boldsymbol{g} - \varepsilon_s \int_{\partial\Omega} (\nabla \boldsymbol{f'} \cdot \boldsymbol{n}) \cdot \boldsymbol{g} = \int_\Omega \nabla f \cdot \boldsymbol{g}. \tag{23}$$

Figure 2 shows the iso line $\kappa = 1$ when $\phi = 0$ is a circle of radius 1. One can see that the projected curvature without smoothing (figure 2(a)) leads to oscillations, whereas the smoothed curvature (figure 2(b)) leads to the expected result.



(a) projected curvature without smoothing

(b) projected curvature with smoothing

FIGURE 2. Iso line 1 of curvature of the level set when the iso line $\phi = 0$ is a circle of radius 1

4.3. **Reinitialization**

When $|\nabla\phi|$ is too different from 1, the support of the Heaviside and Dirac functions (which are defined according to ϕ) are not symmetric with respect to the interface. Indeed, they are narrowed on the side where $|\nabla\phi| > 1$ and enlarged on the side where $|\nabla\phi| < 1$. It has been shown in [22] that $\tilde{\phi} = \dfrac{\phi}{|\nabla\phi|}$ is a first order approximation of a distance function close to the interface. Thus by replacing ϕ by $\tilde{\phi}$ in the definition of H_ε

and δ_ε, we keep the constant size of the interface and avoid the need of very frequent reinitialization. This projection is called Interface Local Projection (**ILP**).

Nevertheless, for long time simulation, the gradient of ϕ can become really high so that, at some point, $\tilde{\phi}$ vanishes and creates new artificial interfaces. To overcome this issue, some reinitializations are needed at a low frequency. The method used to reinitialize ϕ is a Fast Marching Method described in [16].

5. Validation

5.1. Equilibrium Shape

The first system on which we validate our model is the equilibrium shape of a vesicle in a fluid at rest. Indeed, if we consider a vesicle in a shape which is not the one minimizing its energy, the curvature force makes the membrane move as the fluid around it. This dynamic stops when the vesicle reaches its equilibrium state. In our simulations, we initialize the vesicles as ellipses of perimeter p (which does not correspond to the equilibrium shape).

We solve the Navier-Stokes equations in a rectangular domain containing the vesicle with homogeneous Dirichlet boundary conditions and the external forces modeling the vesicle behavior discussed in the previous section.

To analyze the results of the simulations, we need to define a dimensionless number representing the relative strength of the involved forces : $R_f = \dfrac{R_0^2\Lambda}{k_B}$, with $R_0 = \dfrac{p}{2\pi}$ the radius of a circle having the same area of the vesicle. Note that R_f has to be large enough so that the incompressibility force keeps the perimeter of the vesicle almost constant. In practice we choose $R_f = 5 \times 10^4$ which gives a variation of the total perimeter of the order of 5% and we use $(\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^1)$ as Finite Element Space approximation.

The parameter which controls the final shape that takes a vesicle is the reduced area α. It is defined as the ratio between the area of the vesicle (A) and the area of a disk having the same perimeter, *i.e.* $\alpha = \dfrac{4\pi A}{P^2}$. It can be shown that for a reduced area of 1, the shape minimizing the Helfrich energy is a circle. Then decreasing α leads to attain a biconcave shape which is also a characteristic of red blood cells (see figure 3(d)). Figure 3 shows our results (blue dots) plotted vs the one obtained in [7] (red line) with a Lattice Boltzmann method. We see a good agreement between the two models. We reported on the graphs the final values of the reduced volume, which has moved from the initial state since we do not conserve exactly the perimeter during all the simulation. This difference in the reduced area can explain the small differences which can be seen between our shapes and the one from [7].

5.2. "Tank Treading"

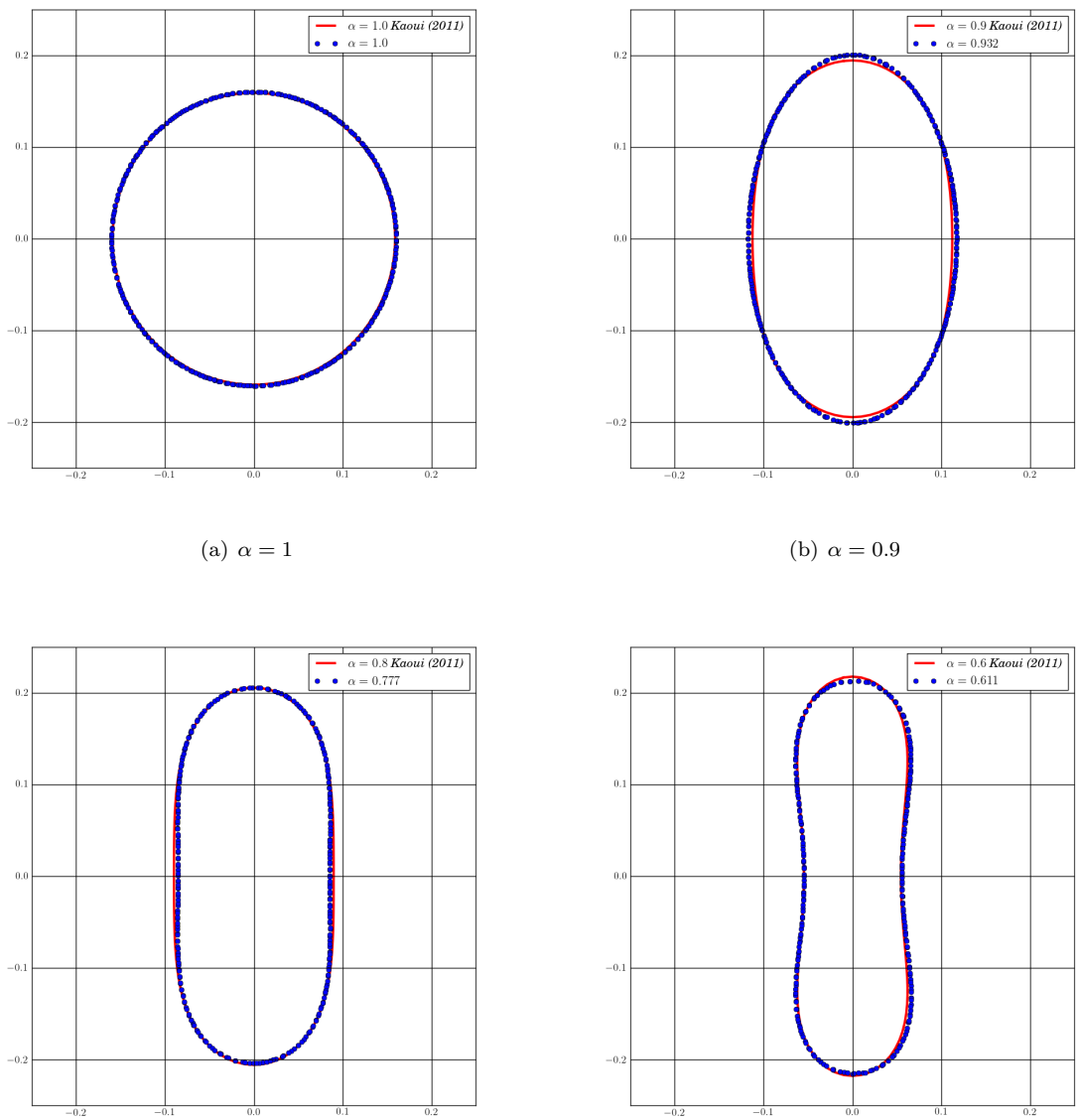
5.2.1. *Description*

The second benchmark that we use to validate our model is the so called tank treading regime. In a shear flow, if the ratio $\frac{\mu_2}{\mu_1}$ between the inner and the outer fluid viscosities is smaller than a certain critical value (around 5 [32]), the vesicle takes a steady angle with respect to the horizontal axis and the fluid inside the vesicle starts to rotate like a chain of a tank.

Figure 4 shows a vesicle in a tank treading regime at different times. In this simulation, we set $R_f = 2 \times 10^3$. The simulation was run on 10^4 elements with a time step $\Delta t = 3 \times 10^{-3}$. On figures 4(a), 4(b), membrane forces are not compensating yet the hydrodynamical forces, it is a transient state. From figure 4(d) to the end ($t = 4.5$), the vesicle is at steady state.

5.2.2. *Order of approximation*

As expected, increasing the order of approximation of fluid solver does not lead to gain much accuracy on the vesicle behavior. This is due to the fact that the pressure is not smooth enough to use higher order polynomials.



(a) $\alpha = 1$

(b) $\alpha = 0.9$

(c) $\alpha = 0.8$

(d) $\alpha = 0.6$

FIGURE 3. Equilibrium shape of vesicles in a fluid at rest for different reduced areas

Figure 5(a) shows the relative variation of perimeter p of the vesicle $\Delta p = 100 \times \dfrac{p - p_0}{p_0}$ as a function of time for two different polynomial approximations : $\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^1$ and $\mathbb{U}_h^4 \times \mathbb{P}_h^3 \times \mathbb{Q}_h^1$. The small *jumps* in the perimeter are due to the reinitializations of level set function.

Three regions can be observed, firstly, the perimeter decreases quickly due to the fact that the vesicle is initialized like an ellipse and has to relax to its equilibrium shape thanks to the curvature force. Secondly,

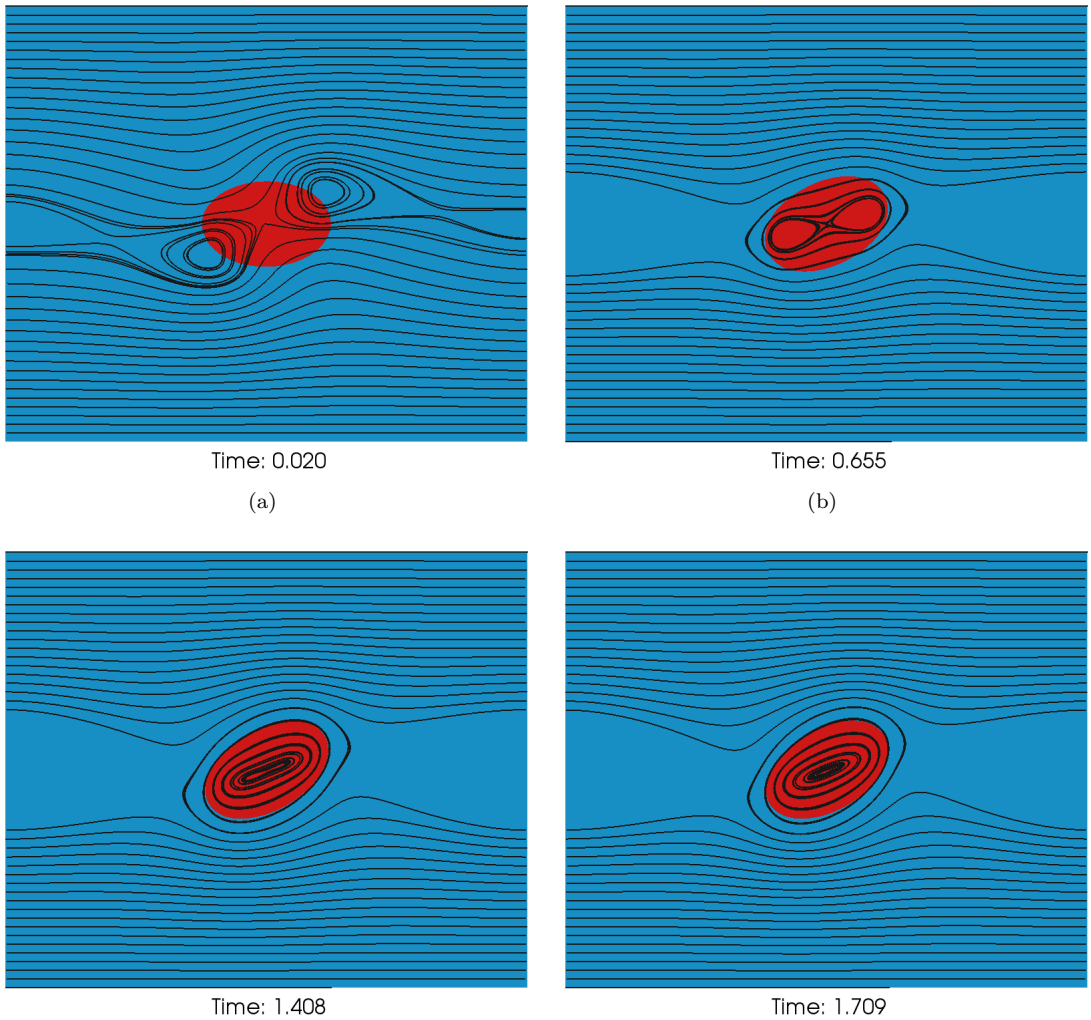
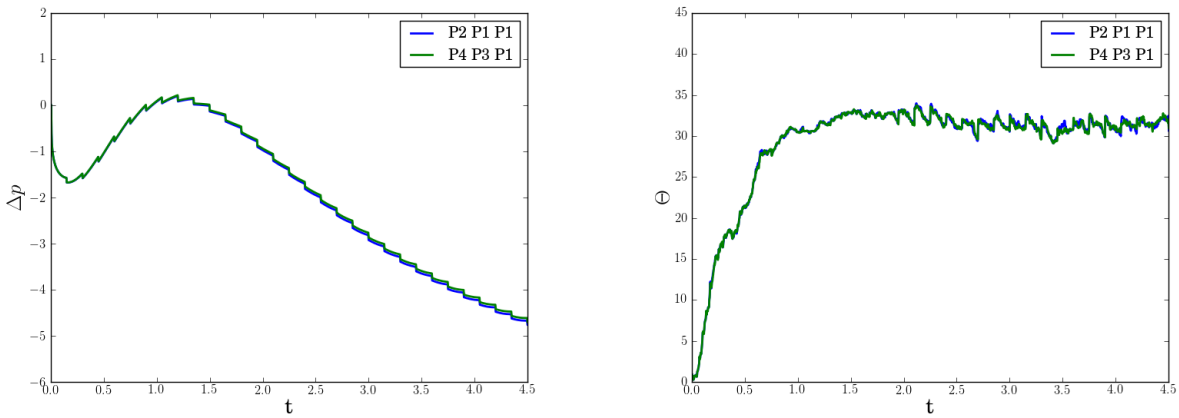


FIGURE 4. Vesicle in shear flow under tank treading regime.

the vesicle is sheared by the flow and moves to its equilibrium angle. During this time the perimeter increases because the vesicle is stretched. The elastic force increases to minimize the stretching. Finally, the vesicle attains its equilibrium position and one can see that the perimeter decreases because of the error accumulation at steady state. This last effect can be controlled by decreasing the mesh size and/or increasing the polynomial order of the level set approximation.

Moreover, figure 5(b) shows the variation of the angle as a function of time. As we can see on figure 5, increasing the polynomial order of fluid variables approximation doesn't lead to better results.

At the opposite, figure 6 shows the same tank treading simulation for three sets of approximation spaces $(\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^1)$, $(\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^2)$ and $(\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^3)$ in which the perimeter conservation is improved by increasing the polynomial order of the level set approximation. It can be understood by the fact that the forces depend strongly on the level set. Thus, increasing the degrees of freedom of the level set approximation leads to a better approximation of the forces. Moreover, since the forces requires several derivative of ϕ (or e), increasing

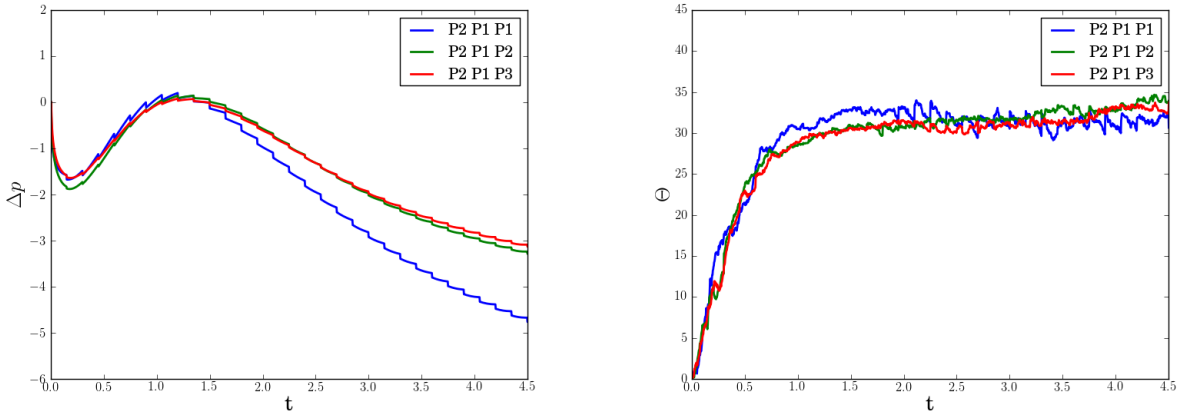


(a) Relative perimeter of the vesicle

(b) Angle from horizontal

FIGURE 5. Vesicle in tank treading regime for two polynomial approximations : $\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^1$ and $\mathbb{U}_h^4 \times \mathbb{P}_h^3 \times \mathbb{Q}_h^1$

the polynomial order of ϕ's approximation gives a better accuracy in its derivative (up to 4th order) and thus in the forcing terms. The interesting point to retain from fig 6(a) is that increasing the order of level set function from 1 to 2 leads to gain a good accuracy (1.5% of the total loss of perimeter in this particular case) and is not too costly in term of computational time (4 times longer than in order 1). At the opposite, increasing the order polynomial of ϕ from 2 to 3 leads to a small gain in accuracy but the computational time starts to be prohibitive (12 times longer than in order 1). For now we found that $\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^2$ is a good compromise between computational time and accuracy.



(a) Relative perimeter of the vesicle

(b) Angle from horizontal

FIGURE 6. Vesicle in tank treading regime for two polynomial approximations : $\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^1$, $\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^2$ and $\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^3$

Acknowledgements

Mourad Ismail and Vincent Doyeux acknowledge the financial support from ANR MOSICOB. Vincent Chabannes and Christophe Prud'homme thank the support of the Région Rhône-Alpes through the project ISLE/CHPID.

Conclusion

We have presented a new finite element implementation of a vesicle model. This model has the advantage to be independant of the fluid solver used since it is expressed only by external forces applied on the interface between inner and outer fluids. We tested this model with a Stokes and a Navier-Stokes fluid. It is possible to add more complexity and use it on a high Reynolds fluid flow or a non Newtonian one. However, this method needs to have a two-fluid interface localized with a good accuracy. To this aim, we developped a level set method in which we do not have to reinitialize too often thanks to the interface local projection. Moreover, when reinitialization is needed we use **ILP** to initialize a Fast Marching Method. Since the **ILP** doesn't move the position of the interface, we can reinitialize without loss of mass which is usually the main drawback of level set methods. We also described how to handle high order derivative of the level set function without necessarily add as much equations and intermediate variables as the highest derivative degree.

We then validated our model on a first benchmark, the equilibrium shape of a vesicle in a fluid at rest. We compared our results with existing results in litterature and found a good agreement. Finally, we reproduced an other behavior of vesicle, the tank treading regime, in which the vesicle takes a steady angle and the membrane rotates around it. We tried different polynomial order approximations to find a good compromise. At the moment increasing only the polynomial approximation of the fluid variables do not lead to significantly better results and we found that $\mathbb{U}_h^2 \times \mathbb{P}_h^1 \times \mathbb{Q}_h^2$, for the various confiugrations we tested, gives the best ratio between accuracy and computational time. These results have to be confirmed in a near future by rigorous mathematical analysis and new numerical experimentations. Otherwise we need to explore other directions such as perform 3D simulations, handle complex geometries or use more complex fluid models. `FEEL++` already allowed us to perform 3D simulations without re-coding the 2D programs.

References

[1] Udo Seifert. Configurations of fluid membranes and vesicles. *Advances in Physics*, 46(1):13–137, 1997.

[2] M. Mader, V. Vitkova, M. Abkarian, A. Viallat, and T. Podgorski. Dynamics of viscous vesicles in shear flow. *The European Physical Journal E: Soft Matter and Biological Physics*, 19(4):389–397, 2006-04-01.

[3] G. Coupier, B. Kaoui, T. Podgorski, and C. Misbah. Noninertial lateral migration of vesicles in bounded Poiseuille flow. *Physics of Fluids*, 20(11):111702–+, November 2008.

[4] Chaouqi Misbah. Vacillating breathing and tumbling of vesicles under shear flow. *Phys. Rev. Lett.*, 96(2):028104, Jan 2006.

[5] Badr Kaoui, Alexander Farutin, and Chaouqi Misbah. Vesicles under simple shear flow: Elucidating the role of relevant control parameters. *Phys. Rev. E*, 80(6):061905, Dec 2009.

[6] E. Maitre, C. Misbah, P. Peyla, and A. Raoult. Comparison between advected-field and level-set methods in the study of vesicle dynamics. *ArXiv e-prints*, May 2010.

[7] B. Kaoui, J. Harting, and C. Misbah. Two-dimensional vesicle dynamics under shear flow: Effect of confinement. *pre*, 83(6):066319–+, June 2011.

[8] Ghigliotti, Giovanni, Selmi, Hassib, Kaoui, Badr, Biros, George, and Misbah, Chaouqi. Dynamics and rheology of highly deflated vesicles. *ESAIM: Proc.*, 28:211–226, 2009.

[9] D. Jamet and C. Misbah. Towards a thermodynamically consistent picture of the phase-field model of vesicles: Local membrane incompressibility. *Phys. Rev. E*, 76(5):051907, Nov 2007.

[10] D. Salac and M. Miksis. A level set projection model of lipid vesicles in general flows. *Journal of Computational Physics*, 230(22):8192–8215, September 2011.

[11] Aymen Laadhari. *Modelisation numerique de la dynamique des globules rouges par la methode des fonctions de niveau.* PhD thesis, Universite de Grenoble, 2011.

[12] Bui, C, Lleras, V., and Pantz, O. Dynamics of red blood cells in 2d. *ESAIM: Proc.*, 28:182–194, 2009.

[13] J.A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 1996.

[14] Ronald Fedkiw Stanley Osher. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, S.S. Antman, J.E. Marsden, L. Sirovich.

[15] Stanley Osher and James A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[16] Christophe Winkelmann. *Interior penalty finite element approximation of Navier-Stokes equations and application to free surface flows*. PhD thesis, 2007.

[17] V. Doyeux, V. Chabannes, C. Prud'homme, and M. Ismail. Simulation of two fluid flow using a level set method application to bubbles and vesicle dynamics. *Journal of Computational and Applied Mathematics (Submitted)*, 2012.

[18] Emmanuel Maitre, Thomas Milcent, Georges-Henri Cottet, Annie Raoult, and Yves Usson. Applications of level set methods in computational biophysics. *Mathematical and Computer Modelling*, 49(11-12):2161–2169, June 2009.

[19] Ou-Yang Zhong-can and Wolfgang Helfrich. Bending energy of vesicle membranes: General expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders. *Phys. Rev. A*, 39(10):5280–5288, May 1989.

[20] T. Biben and C. Misbah. Tumbling of vesicles under shear flow within an advected-field approach. *Phys Rev E*, 67(3):031908–+, March 2003.

[21] A. Laadhari, C. Misbah, and P. Saramito. On the equilibrium equation for a generalized biological membrane energy by using a shape optimization approach. *Physica D: Nonlinear Phenomena*, 239(16):1567 – 1572, 2010.

[22] Emmanuel Maitre Georges-Henri Cottet. A level set method for fluid-structure interactions with immersed surfaces. *Mathematical Models and Methods in Applied Sciences*.

[23] Thomas Milcent. *Une approche eulerienne du couplage fluide-structure, analyse mathematique et applications en biomcanique.* PhD thesis, Universite de Grenoble, 2009.

[24] C. Prud'homme, V. Chabannes, and G. Pena. Feel++: Finite Element Embedded Language in C++. Free Software available at http://www.feelpp.org. Contributions from A. Samake, V. Doyeux, M. Ismail and S. Veys.

[25] Christophe Prud'homme. A domain specific embedded language in C++ for automatic differentiation, projection, integration and variational formulations. *Scientific Programming*, 14, 2006.

[26] Christophe Prud'homme. Life: Overview of a unified C++ implementation of the finite and spectral element methods in 1d, 2d and 3d. 2007.

[27] Codina Ramon. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 156(1-4):185–210, April 1998.

[28] G. Hauke. A simple subgrid scale stabilized method for the advection-diffusion-reaction equation. *Computer Methods in Applied Mechanics and Engineering*, 191(27-28):2925–2947, April 2002.

[29] R.P. Bonet Chaple. Numerical stabilization of vonvection-diffusion-reaction problems. Technical report, 2006.

[30] Erik Burman and Peter Hansbo. Edge stabilization for the generalized stokes problem: A continuous interior penalty method. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22):2393–2410, April 2006.

[31] Jean-Christophe Nave, Rodolfo Ruben Rosales, and Benjamin Seibold. A gradient-augmented level set method with an optimally local, coherent advection scheme. *Journal of Computational Physics*, 229(10):3802 – 3827, 2010.

[32] J. Beaucourt, F. Rioual, T. Seacuteon, T. Biben, and C. Misbah. Steady to unsteady dynamics of a vesicle in a flow. *Phys. Rev. E*, 69(1):011906–, January 2004.