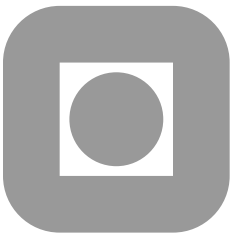
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

A finite element implementation of the level set equation

by
Samuel R. Ransau, Bjarte Hægland and Jens Holmen

PREPRINT
NUMERICS NO. 4/2004



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/numerics/2004/N4-2004.ps
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

A finite element implementation of the level set equation

Samuel R. Ransau[1], Bjarte Hægland[2] and Jens Holmen[3]

[1] Department of Marine Technology
Norwegian University of Science and Technology
N-7491 Trondheim, Norway

[2] Department of Mathematical Sciences
Norwegian University of Science and Technology
N-7491 Trondheim, Norway

[3] Sintef Applied Mathematics
N-7465 Trondheim, Norway

Abstract

A finite element implementation of the level set method is presented. The level set method is a capturing type of method, the interface is never found explicitly. The interface is embedded in the zero level set of a continuous function. A reinitialisation procedure is used to ensure that the level set field remains a distance field. Some results for gravity driven two-phase flows, where the density ratio is relatively small, are presented.

1 Introduction

Multiphase flows with moving interfaces are present in a large number of engineering and industrial problems, and represent a great challenge from Computational Fluid Dynamics (CFD) point of view. Development of efficient and robust numerical methods for representing accurately such phenomenas is still a major problem in CFD.

In the present report, we present an interface capturing method allowing to follow the evolution of complex moving interfaces : the level set technique. This approach was proved to be useful in a wide range of cases, and specially in extreme situations like breaking waves, that other techniques might not be able to handle. In the last years, scientists and engineers have shown an increasing interest for the level set technique, see for instance [4], [7], [10, 11]. Most of the published papers use the finite difference/volume method for spatial discretization. This might be explained by the fact that the equations present in the level set technique are purely hyperbolic and require robust and accurate upwinding strategies. Many of the existing upwinding methods have been devised and tailored for finite difference/volume type of schemes. In addition finite difference schemes are relatively computational efficient and easy to implement. However, complex geometries and mesh movement are quite complicated and cumbersome to deal with in the finite difference context. Here, we present a

finite element implementation of the level set method.
The present paper is organised as follows: first the mathematical model used is reviewed. Then, in section 3 the numerical formulation is presented. Section 4 outlines some implementation aspects. And finally, in 5 the algorithm is tested on different examples.

2 Mathematical formulation

In this section, the equations associated with the level set technique will be presented as well as the flow governing equations. These two sets of equations are solved in a uncoupled manner. The solution of the flow governing equations is first computed using an incompressible Navier-Stokes solver and then the velocity field obtained is used in order to solve the level set problem.

In the level set technique, the interface between two fluids is represented as the zero level set of a smooth function ϕ, which is advected by the fluid velocity. The level set technique [5], [8], [9] is usually used in an Eulerian reference frame, and the governing equations are solved in both fluids. Assuming that both fluids are governed by the incompressible Navier-Stokes equations, one has:

$$\nabla \cdot \mathbf{u}_1 = 0 \quad \text{and} \quad \nabla \cdot \mathbf{u}_2 = 0 \tag{1}$$

$$\rho_1 \left(\frac{\partial \mathbf{u}_1}{\partial t} + (\mathbf{u}_1 \cdot \nabla)\, \mathbf{u}_1 \right) = \nabla \cdot \boldsymbol{\sigma}_1 + \mathbf{f}_1 \quad \text{and} \quad \rho_2 \left(\frac{\partial \mathbf{u}_2}{\partial t} + (\mathbf{u}_2 \cdot \nabla)\, \mathbf{u}_2 \right) = \nabla \cdot \boldsymbol{\sigma}_2 + \mathbf{f}_2 \tag{2}$$

where the subscripts 1 and 2 refer to fluid 1 and fluid 2 respectively, $\boldsymbol{\sigma}$ is the stress tensor given by:

$$\sigma_{ij} = -p\delta_{ij} + 2\mu S_{ij} \tag{3}$$

p is the pressure, S the rate-of-strain tensor and μ the dynamic viscosity. Further, $\mathbf{u}$ is the fluid velocity, ρ the fluid density, ν the kinematic viscosity, and $\mathbf{f}$ an external force considered as known.
As mentioned above, the interface, that will be denoted Γ, is represented as the zero level set of the function ϕ: $\Gamma = \{\mathbf{x} | \phi(\mathbf{x}, t) = 0\}$.

In order to write the equation governing the evolution of ϕ, it can be assumed that, at time t, the interface is parametrised by $(x(s,t), y(s,t), z(s,t))$. The evolution of

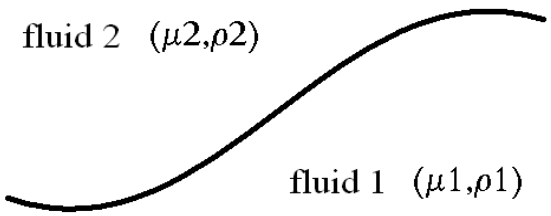


Figure 1: Sketch of the flow domain

(x, y, z) is then determined by:

$$\begin{aligned} \frac{dx(s,t)}{dt} &= u(x(s,t), y(s,t), z(s,t)) \\ \frac{dy(s,t)}{dt} &= v(x(s,t), y(s,t), z(s,t)) \\ \frac{dz(s,t)}{dt} &= w(x(s,t), y(s,t), z(s,t)) \end{aligned} \tag{4}$$

where u, v, w are the components of the velocity vector $\mathbf{u}$ defined as:

$$\mathbf{u} = \begin{cases} \mathbf{u_1} = (u_1, v_1, w_1)^T & \text{if } \phi \geq 0 \\ \mathbf{u_2} = (u_2, v_2, w_2)^T & \text{if } \phi < 0 \end{cases} \tag{5}$$

Since Γ is defined by $\phi = 0$, one must have $\dfrac{d\phi}{dt} = 0$ on Γ:

$$\frac{d\phi\,(x(s,t), y(s,t), z(s,t), t)}{dt} = \frac{d\phi}{dx}\frac{dx}{dt} + \frac{d\phi}{dy}\frac{dy}{dt} + \frac{d\phi}{dz}\frac{dz}{dt} + \frac{d\phi}{dt} = 0 \tag{6}$$

ie:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \tag{7}$$

Osher and Sethian [5] showed that equation (7) moves the zero level set with the fluid velocity $\mathbf{u}$. A common choice is to take the function ϕ initially as a distance function such that (cf. figure 1):

$$\phi > 0 \rightarrow \text{fluid 1}$$
$$\phi = 0 \rightarrow \text{Surface}$$
$$\phi < 0 \rightarrow \text{fluid 2}$$

As a consequence, the flow properties (density and viscosity) are defined as functions of ϕ:

$$\rho(\phi) = \rho_2 + (\rho_1 - \rho_2)H(\phi) \tag{8}$$
$$\mu(\phi) = \mu_2 + (\mu_1 - \mu_2)H(\phi) \tag{9}$$

where H is the Heavyside function defined by:

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < 0 \\ 1 & \text{if } \phi > 0 \end{cases} \tag{10}$$

3 Numerical formulation

In this section, the strategies chosen to solve the problem numerically will be briefly presented.

3.1 Bulk flow solution

The projection method [1], [13] is used to solve equations (1) and (2) in the following manner:

$$\frac{\tilde{\mathbf{u}} - \mathbf{u}^n}{\delta t} + (\mathbf{u}^n \cdot \nabla)\,\tilde{\mathbf{u}} = \nu \Delta \tilde{\mathbf{u}} + \mathbf{f}^{n+1} \tag{11}$$
$$\frac{\mathbf{u}^{n+1} - \tilde{\mathbf{u}}}{\delta t} = -\nabla p^{n+1} \tag{12}$$
$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{13}$$

Equation (11) is solved and an intermediate velocity field $\tilde{\mathbf{u}}$ is obtained. Equations (12) and (13) are then combined in order to get the following Poisson equation for the pressure:

$$\nabla \cdot \tilde{\mathbf{u}} = \delta t \Delta p^{n+1} \tag{14}$$

Finally, the end-of-step velocity field is simply found simply:

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \delta t \nabla p^{n+1} \tag{15}$$

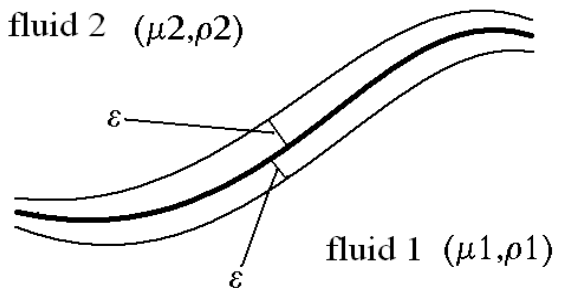


Figure 2: Smooth interface

3.2 Solution of the level set equation

Even though the level set function ϕ is continuous over the whole domain (and in particular across the interface), this is not the case for the quantities used by the fluid solver, ie. the density ρ and the viscosity ν. Those quantities are indeed discontinuous across the interface. When the ratio between the fluid properties is very large, severe numerical instabilities appear in the solution of the bulk flow equations due to the rapid change in fluid properties across the interface. Sussmann et al. [10, 11] solved this problem by giving the interface a finite thickness where the fluid properties change smoothly (cf. figure 2). This is realised by introducing a smooth Heavyside function in equations (8) and (9):

$$\rho(\phi) = \rho_2 + (\rho_1 - \rho_2)H_\epsilon(\phi) \tag{16}$$

$$\mu(\phi) = \mu_2 + (\mu_1 - \mu_2)H_\epsilon(\phi) \tag{17}$$

where

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2}[1 + \frac{\phi}{\epsilon} - \frac{1}{\pi}sin(\frac{\pi\phi}{\epsilon})] & \text{if } |\phi| < \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \tag{18}$$

Other ways of defining this smooth Heavyside function are also possible.
The thickness of the interface can be tuned through the parameter ϵ. It is common to choose ϵ proportional to the grid size. Typicallly, ϵ is taken to be 2-3 times larger than the grid size close to the interface [12].

For problems where the velocity vector is such that large gradients appear in the level

set field, solving only equation (7) is not sufficient to ensure accurate results. Indeed, ϕ will then not remain a distance function under its evolution by equation (7). This results in an interface with varying thickness, a feature which is undesirable. Therefore, a step where the function ϕ is reinitialised as a distance function is necessary. Sussman et al. [11] proposed a technique in order to reinitialise the function ϕ. They proposed to seek for the steady state solution of the following problem:

$$\frac{\partial d}{\partial \tau} + \text{sign}(\phi)\left(|\nabla d| - 1\right) = 0 \tag{19}$$

with initial conditions:

$$d(x, 0) = \phi(x) \tag{20}$$

where

$$\text{sign}(\phi) = \begin{cases} -1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi = 0 \\ 1 & \text{if } \phi > 0 \end{cases} \tag{21}$$

and τ is an artificial time. When the steady state solution is achieved, one will have $\frac{\partial d}{\partial \tau} = 0$ and consequently, $|\nabla \phi| = 1$ will hold.
A nice feature of equation (19) is that its solutions $d(x, \tau)$ have the same zero level set as $\phi(x)$ since $\text{sign}(0) = 0$. They also mention that it is not necessary to solve problem (19)-(20) in the whole domain since equation (19) is an hyperbolic equation with characteristic velocities pointing outward from the interface, and therefore only few reinitialisation steps are necessary.

3.3 Discretisation of the level set equation

3.3.1 Temporal discretisation

The time discretisation is performed using both the θ-method and the mid-point rule.
Applied to equation (7), the θ-method yields to:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \theta\left[-\mathbf{u}^{n+1} \cdot \nabla \phi^{n+1}\right] + (1-\theta)\left[-\mathbf{u}^{n+1} \cdot \nabla \phi^n\right], \qquad 0 \leq \theta \leq 1 \tag{22}$$

This scheme is generally first order accurate except for $\theta = \frac{1}{2}$, where it is second order accurate. Note that for $\theta = 0$, the forward Euler scheme is obtained, while for $\theta = 1$, one gets the backward Euler scheme.

As for the mid-point rule, its application to equation (7) yields:

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = -\frac{1}{2}\left[\mathbf{u}^{n+\frac{1}{2}} \cdot \nabla \phi^{n+1}\right] + \frac{1}{2}\left[-\mathbf{u}^{n+\frac{1}{2}} \cdot \nabla \phi^n\right] \tag{23}$$

We take $\mathbf{u}^{n+\frac{1}{2}} = \dfrac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2}$.

3.3.2 Spatial discretisation

The space discretisation is performed by the standard Galerkin finite element method. The numerical solution of equation (7) is seeked for on the form:

$$\phi(\mathbf{x}, t) = \sum_j \phi_j N_j(\mathbf{x}, t) \tag{24}$$

where ϕ_j are the nodal values and $N_j(\mathbf{x}, t)$ are the shape functions.
The Galerkin formulation of equation (22) is:

$$\sum_j \int_\Omega \phi_j^{n+1} N_j N_i \, d\Omega - \int_\Omega \phi^n N_i \, d\Omega = -\Delta t \theta \sum_j \int_\Omega \left[\mathbf{u}^{n+1} \cdot \nabla N_j \; N_i\right] \phi_j^{n+1} \, d\Omega \\ -\Delta t(1-\theta) \int_\Omega \left[\mathbf{u}^{n+1} \cdot \nabla \phi^n \; N_i\right] \, d\Omega \tag{25}$$

Equation (25) can be reorganised as:

$$\sum_j \int_\Omega \left[N_j + \Delta t \theta \mathbf{u}^{n+1} \cdot \nabla N_j\right] N_i \phi_j^{n+1} \, d\Omega = \int_\Omega \left[\phi^n - \Delta t(1-\theta)\mathbf{u}^{n+1} \cdot \nabla \phi^n\right] N_i \, d\Omega \tag{26}$$

Equation (26) can be put in the following matrix form:

$$\sum_j \left(M_{ij} + C_{ij}\right) \phi_j^{n+1} = \mathbf{b}_i^n \tag{27}$$

where the matrices M, C and the vector $\mathbf{b}$ are defined as:

$$M_{ij} = \int_\Omega N_i N_j \, d\Omega \tag{28}$$

$$C_{ij} = \Delta t \; \theta \int_\Omega \mathbf{u}^{n+1} \cdot \nabla N_j N_i \, d\Omega \tag{29}$$

$$\mathbf{b}_i = \int_\Omega \left[\phi^n - \Delta t(1-\theta)\mathbf{u}^{n+1} \cdot \nabla \phi^n\right] N_i \, d\Omega \tag{30}$$

The spatial discretisation of equation (23) gives an equation very similar to equation (26) and is not written here.

3.4 Numerical solution of the reinitialisation equation

Equation (19) is a nonlinear Hamilton-Jacobi type of equation. Special solution techniques need to be used in order to solve equation (19) numerically. We use the well-known Newton-Raphson method, as well as the successive substitution method (also referred as Picard iteration), that we briefly recall in the present section.

3.4.1 Newton-Raphson method

Let us write equation (19) in the form $F(d) = 0$, with $F(d) := \dfrac{\partial d}{\partial \tau} + \text{sign}(\phi)\,(|\nabla d| - 1)$. Assuming that an approximation d^k to d is available, the objective is to approximate $F(d)$ in the vicinity of d^k by a function $M(d, d^k)$, such that the equation $M(d, d^k) = 0$ is easy to solve. The solution of the latter equation is taken as the approximation d^{k+1} to the root of $F(d) = 0$. An easy expression can be found for $M(d, d^k)$ by taking the first two terms of a Taylor-series approximation to F at point $d = d^k$:

$$M(d, d^k) = F(d^k) + \frac{dF(d^k)}{dd}(d - d^k) \tag{31}$$

d^{k+1} can be then computed by:

$$d^{k+1} = d^k - \frac{F(d^k)}{\frac{dF(d^k)}{dd}} \tag{32}$$

A inital value d^0 is required.

3.4.2 Successive substitution method

It is now assumed that equation (19) has already been discretised, and that we are left with the following algebraic system of equations to solve:

$$A(\mathbf{d})\mathbf{d} = \mathbf{b} \tag{33}$$

This can be solved with the following iteration technique:

$$A(\mathbf{d^k})\mathbf{d}^{k+1} = \mathbf{b}, \qquad k = 0, 1, 2, ... \tag{34}$$

until $||\mathbf{d}^{k+1} - \mathbf{d^k}||$ is sufficiently small. As in the Newton-Raphson case, an initial guess d^0 is needed.

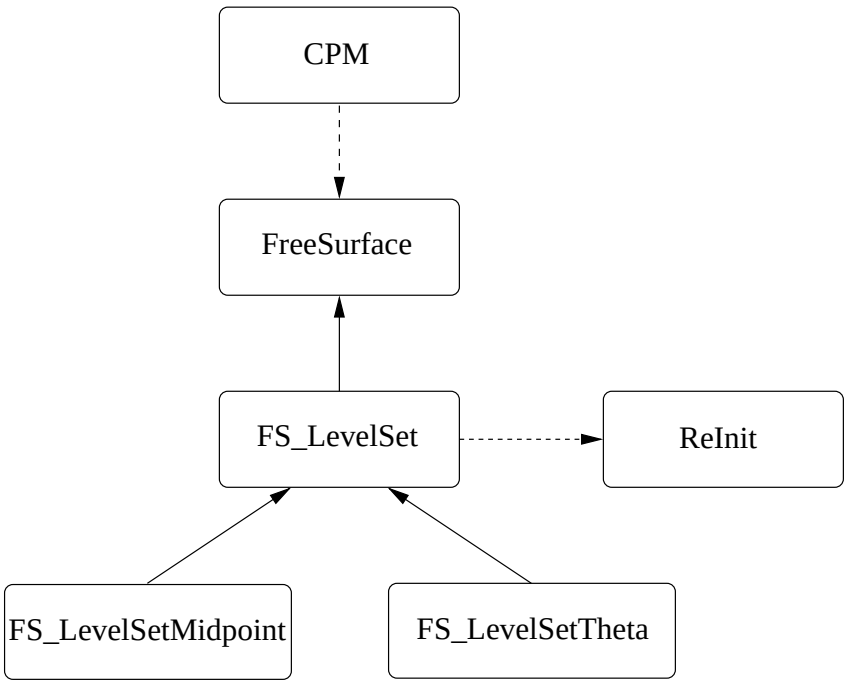


Figure 3: Flow chart for Level Set implementation

4 Implementation Issues

The code is implemented in the object-oriented programming language C++ with the numerical library Diffpack [3]. The overall flow chart of the implementation is shown i Figure 3.

CPM is a class for computing viscous incompressible fluid flow by using finite elements and the continuous projection method [2]. `FreeSurface` is a general top-class for implementing tracking or capturing of interfaces between two fluids. The level-set method is just one of many possible realisation, implemented as a sub-class of `FreeSurface`. `FS_LevelSetMidpoint` and `FS_LevelSetTheta` are sub-classes of `FS_LevelSet` implementing the midpoint-method and the θ-method , respectively, as temporal discretisations of the level-set equation.

The solution algorithm we use can be summarised as:

1. Solve for $\mathbf{u}^{n+1}$ and p^{n+1} in the Navier-Stokes equations.

2. Solve for ϕ^{n+1} in the level-set equation, taking $\mathbf{u} = \mathbf{u}^{n+1}$. Update density ρ^{n+1} and viscosity μ^{n+1}.

4.1 Reinitialisation

The reinitialisation of the scalar field ϕ in the level-set problem (7) is implemented in the separate class `ReInit`. The user may choose to use reinitialisation or not. If one chooses to include reinitialisation this problem is solved directly after the level-set equation is solved for each time-step.

The temporal discretisation chosen for equation (19) is the θ-method. As equation (19) is non-linear, we have implemented both a Picard-iteration technique and the a Newton-Raphson method for this problem.

5 Numerical examples

In this section, some numerical examples are presented. The first three examples are based on analytical velocity fields. They are realised in order to insure that the transport equation (7) is properly solved, and are chosen such that the evaluation of the solution of this pure advection equation is possible without the need of reinitialisation. In those three examples, the densities and viscosities of the two fluids are set equal. Even though, there is no difference in the fluid properties, a level set field is initialised and the level set equation is solved. The level set field is then used as a tracer for following the deformation of the fluid.
Then two-phase flow examples with physical velocity fields and differences in the fluid properties will be presented.

5.1 Translation of a circle

The level set method is used to follow a circle which is advected by a uniform velocity. This velocity field is given by:

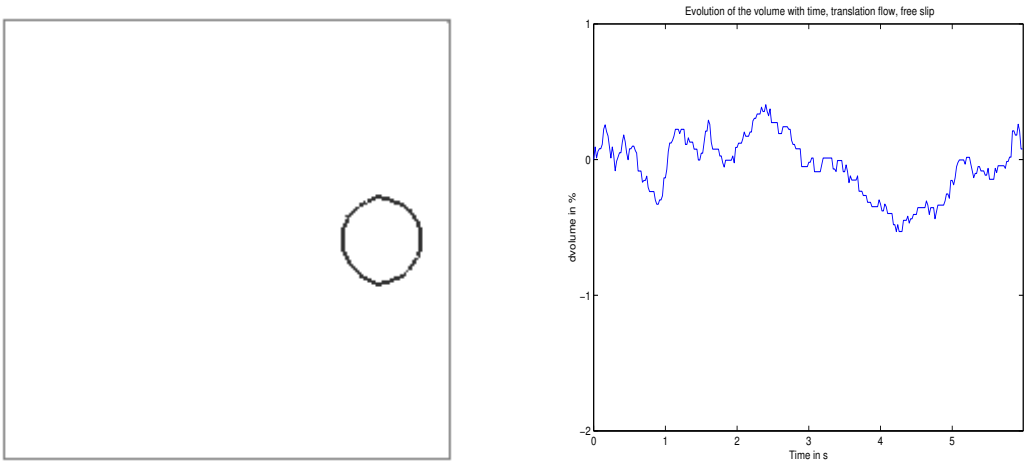
$$\mathbf{u} = \mathbf{i} \tag{35}$$

The flow domain is assumed to be infinite and uniform velocity is also prescribed on the top and bottom boundaries. The mesh is regular with element length equal to 0.016 in both x- and y-directions. The time step is set to 0.005, and $\theta = \frac{1}{2}$ is used in equation (22). In this example, the reinitialisation is turned off.
We look at the conservation of the circle. The volume evolution is defined as follows:

$$\Delta V = \frac{V(t) - V_0}{V_0} \tag{36}$$

where the volume at time t is defined by:

$$V(t) = \int_\Omega H(\phi)\, d\Omega \ , \tag{37}$$



(a) Solution at time t=1.995s

(b) Evolution of the volume with time

Figure 4: $\mathbf{u} = \mathbf{i}$. Computed solution at time t=1.995s and evolution of the volume of the circle for $\Delta x = \Delta y = 0.016$ and $\Delta t = 0.005$.

V_0 is the initial volume, Ω the total fluid domain and H is the Heavyside function:

$$H(\phi) = \begin{cases} 1 & \text{if } \phi < 0 \\ 0 & \text{if } \phi > 0 \end{cases} \tag{38}$$

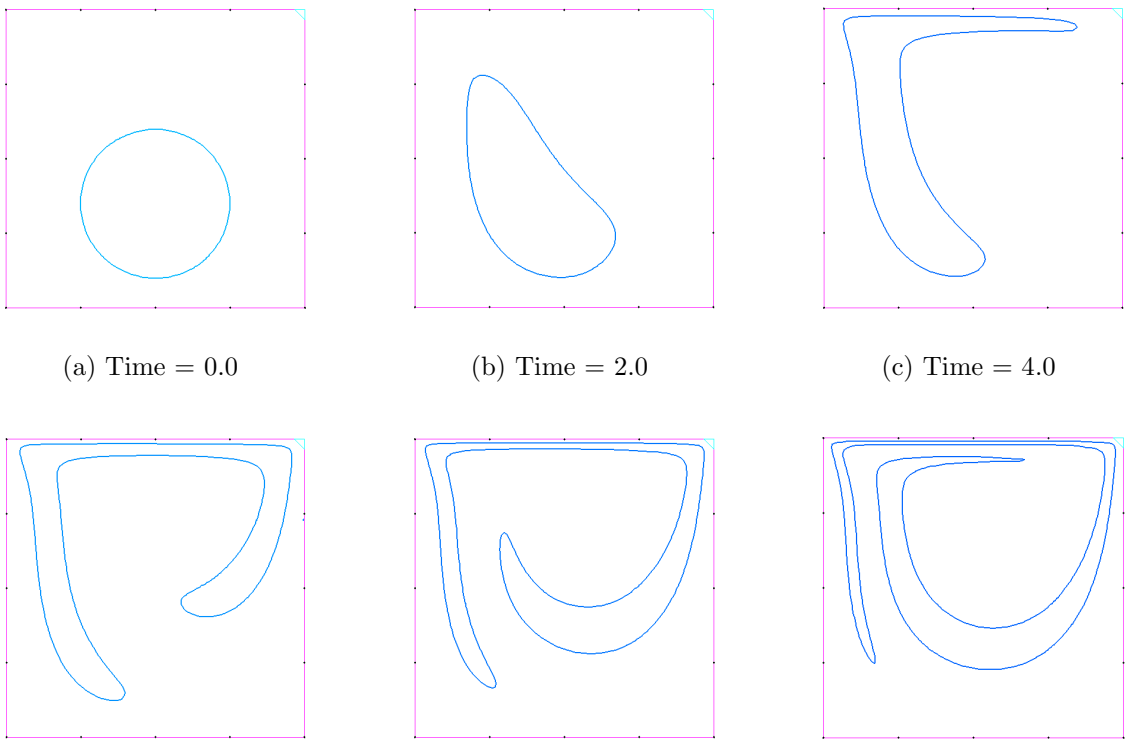
Figures 4(a) and 4(b) show the circle after 1.995 seconds (ie 400 time steps) and the evolution of the volume. The accuracy is satisfactory, but at the cost of a relatively fine grid and small time step. In addition the velocity field used in this example is the simplest one possible.

5.2 Cavity flow

The next example is the classical cavity flow in a square domain. No-slip condition is prescribed at all boundaries. The left, right and bottom boundaries are fixed, while the top one is moving with a velocity given by:

$$\mathbf{u} = 2\mathbf{i} \tag{39}$$

The Reynolds number based on this velocity, the length of one side of the square and the kinematic viscosity is equal to 100.
The deformation of the circular region (figure 5) follows the expected pattern. One problem seems however present: the mass conservation principle is not respected (see



(a) Time = 0.0 (b) Time = 2.0 (c) Time = 4.0

(d) Time = 6.0 (e) Time = 8.0 (f) Time = 10.0

Figure 5: Time series of driven cavity test problem. Both fluids have same density and viscosity.

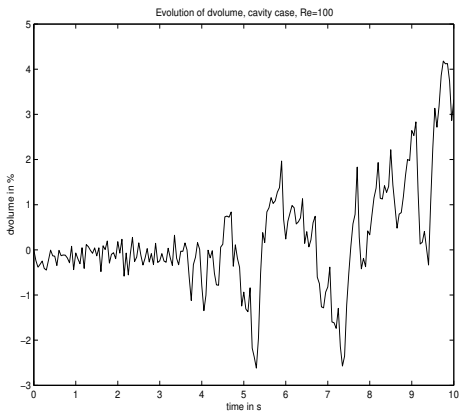
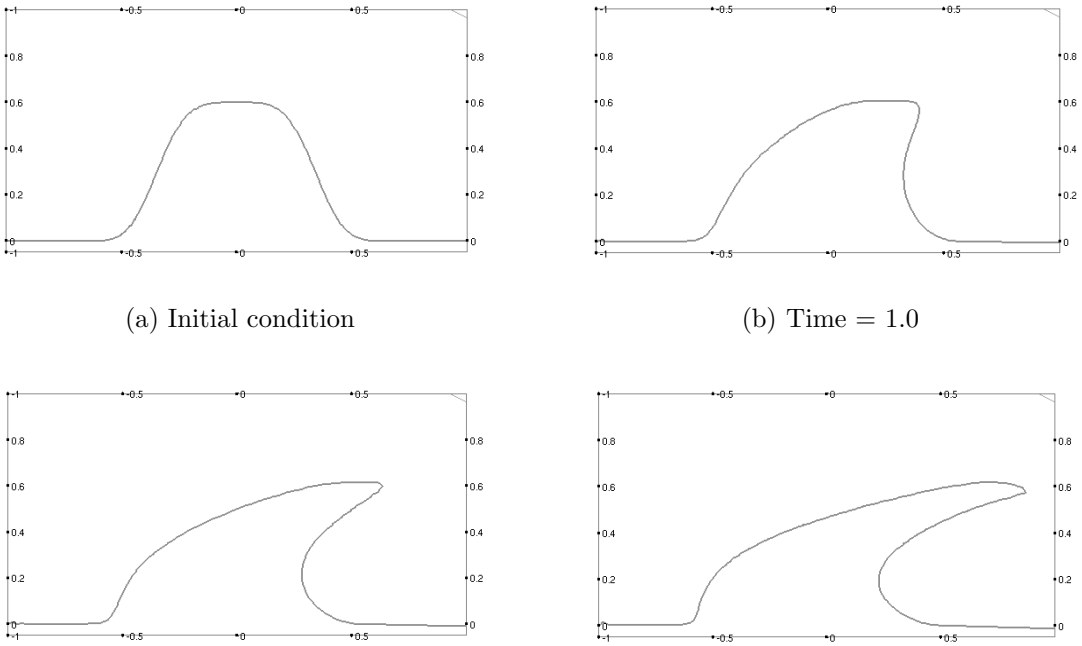


Figure 6: Evolution of ΔV for the cavity problem



(a) Initial condition

(b) Time = 1.0

(c) Time = 2.0

(d) Time = 3.0

Figure 7: Time series of the cusping test problem. Both fluids have same density and viscosity.

figure 6). (The change in volume is defined as in the previous example, see equation (36)). This confirms the well-known weakness of the level set method: poor mass conservation properties. In cases where the two fluids have different properties, the mass loss will be increased.

5.3 Cusping interface

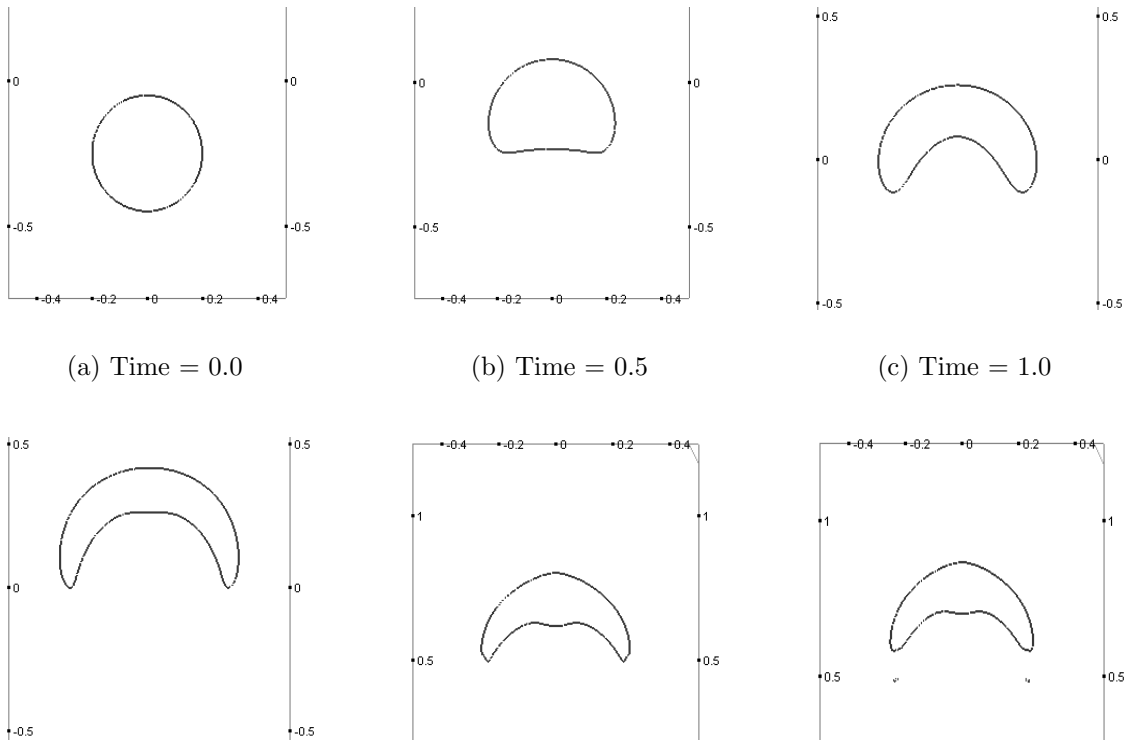
Our next example simulates the development of a sharp cusp in the interface. The initial condition is given by the following expression:

$$\phi(x, y, 0) = 0.6 \exp(-50x^4) \tag{40}$$

The velocity is chosen as:

$$\mathbf{u} = y^3\mathbf{i} \tag{41}$$

This example is relevant in, for instance, run-up problems on beaches, where an incoming wave will cusp, steepen and eventually break. Here also, the results (figure 7) follow the expected pattern (see for example [4]).



(a) Time = 0.0 (b) Time = 0.5 (c) Time = 1.0

(d) Time = 1.5 (e) Time = 2.7 (f) Time = 2.9

Figure 8: Oil bubble rising in water with reinitialisation.

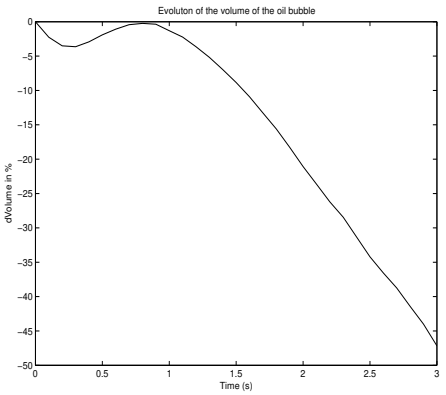
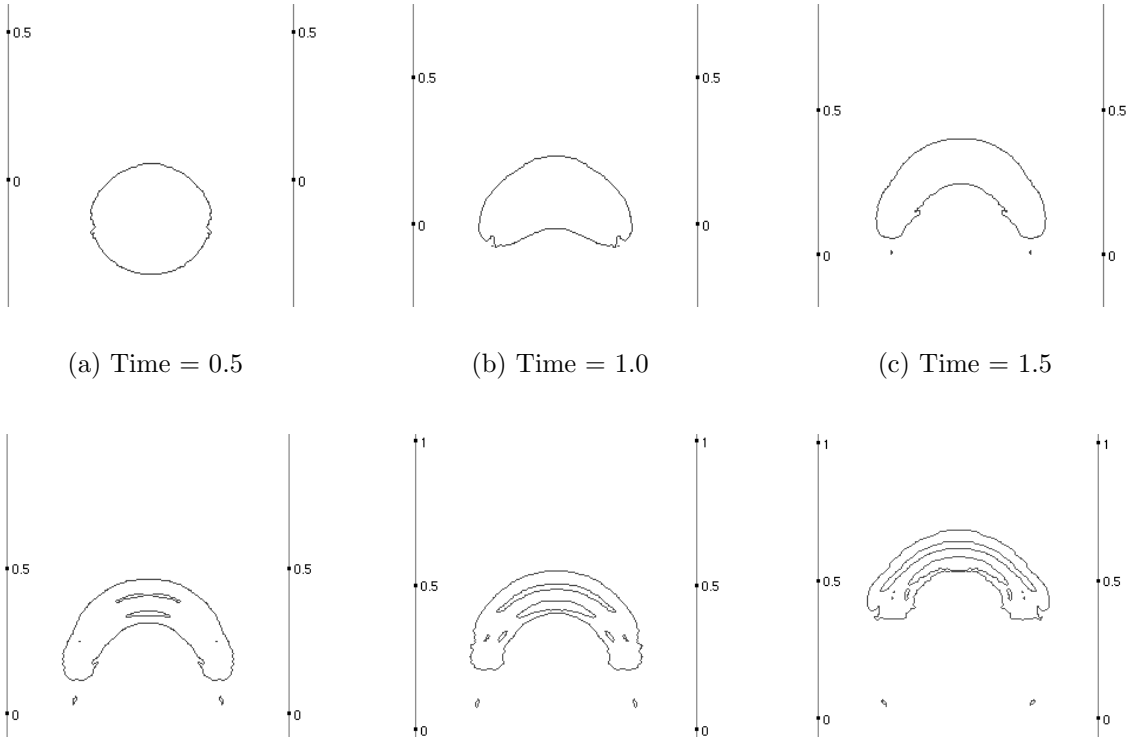


Figure 9: Evolution of the volume of the oil bubble with reinitialisation.

5.4 Oil bubble rising in water

The previous three examples were used to insure that the level set equation was properly solved. The next example is a physical flow: we let a circular bubble of



(a) Time = 0.5 (b) Time = 1.0 (c) Time = 1.5

(d) Time = 1.7 (e) Time = 2.0 (f) Time = 2.5

Figure 10: Oil bubble rising in water without reinitialisation.

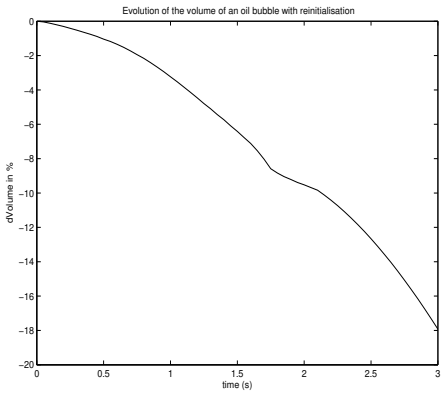
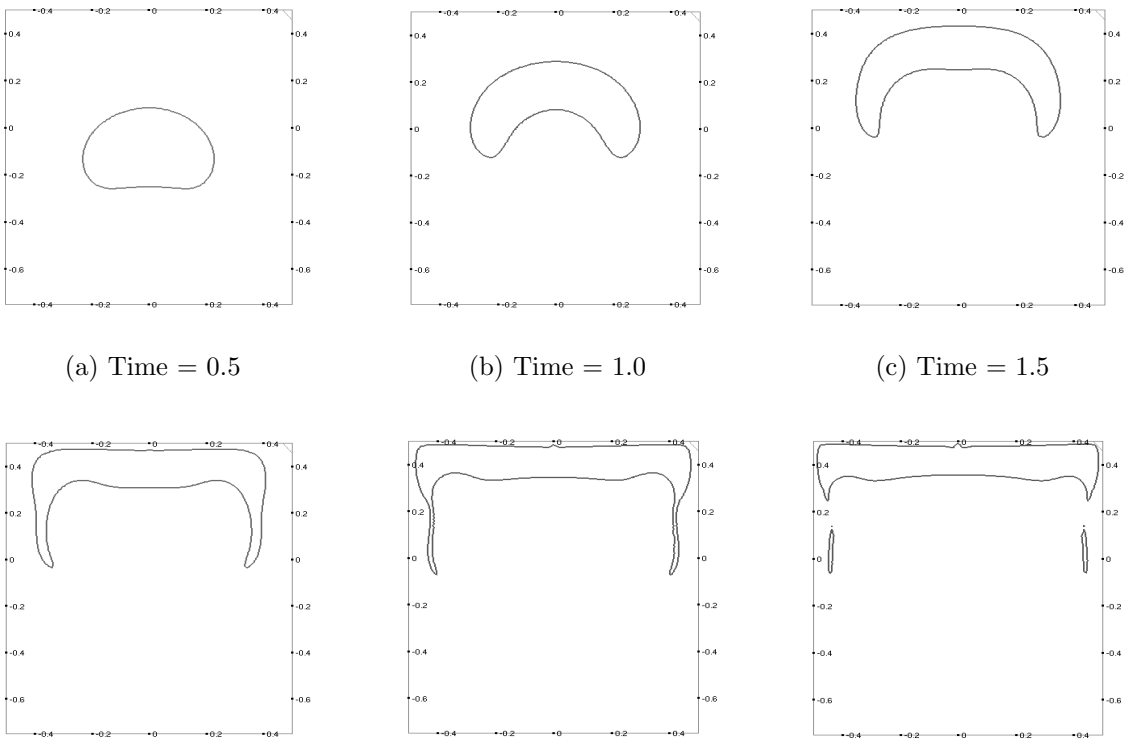


Figure 11: Evolution of the volume of the oil bubble without reinitialisation.

gasoline be in water under the influence of gravity. All quantities (ie water and oil properties and gravity) have been given physical correct values. The gasoline density



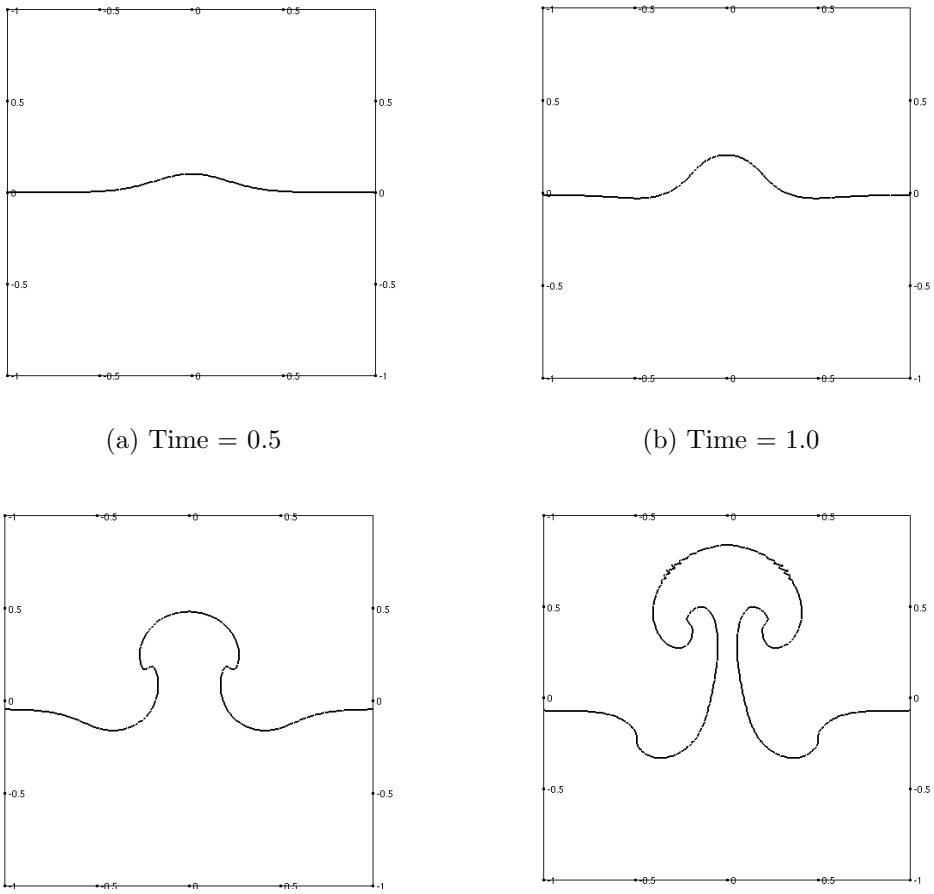
(a) Time = 0.5 (b) Time = 1.0 (c) Time = 1.5

(d) Time = 2.0 (e) Time = 2.5 (f) Time = 2.8

Figure 12: Oil bubble rising in water in a closed tank (with reinitialisation).

and viscosity are set to 680kg/m^3 and $2.9 \cdot 10^{-4}$kg/(m·s) respectively. The water density and viscosity are 998kg/m^3 and $1 \cdot 10^{-3}$ kg/(m·s) respectively. A free-slip boundary condition is applied on all the boundaries.
Figure 8 shows how the bubble shape and position evolve. The bubble shape and position compare relatively well with results presented in [6]. However, figure 9 underlines the important drawback (already observed and mentioned in section 5.2 for the cavity problem) of the level set method: the poor mass conservation properties (and in particular when the reinitialisation is used, see figure 11). Figure 10 shows that without reinitialisation, the algorithm breaks down relatively fast.

Figure 12 shows a rising bubble of gasoline in water in a closed tank. Here, the no-slip condition is prescribed on the top boundary. We can observe the correct behaviour of the bubble when it gravitates against the roof of the tank and forms a horizontal layer on top of the water. We can also observe the pinch-off of oil. The pinch-off time which is approximatively 2.6 seconds is lower than the real time (see [10]). This is explained by the omission of surface tension in our simulations. Clearly,



(a) Time = 0.5

(b) Time = 1.0

(c) Time = 1.5

(d) Time = 2.0

Figure 13: Rayleigh instability.

the pinch-off would have been delayed if surface tension had been incorporated.

5.5 Rayleigh instability

The next example is the so-called Rayleigh instability where a heavier fluid rests over a lighter one. Any disturbance in the horizontal configuration of the interface will initiate a movement of the fluid where the lighter fluid will go up while the heavier one will go down, to achieve a equilibrium where the lighter fluid is above. Figure 13 shows a time series of the simulation. The expected mushroom shape, formed by the lighter fluid as it goes up and penetrate the heavier one, can be seen. For this simulation, water and the SAE 30 oil were used. The density and viscosity of the oil are set to 891kg/m^3 and 0.29kg/(m·s), respectively, while the properties

of the water are the same as in section 5.4. Free-slip condition were prescribed on all walls.

6 Concluding remarks

In summary, we presented our work on developing a numerical tool for simulation of two-phase flows with sharp interfaces based on the finite element method combined with the level set technique. As the latter is an Eulerian method, no re-meshing is necessary. Very promising results were obtained for gravity driven oil-water flows where the density ratio is of order of unity. The reinitialisation procedure proved to be efficient, and useful in order to obtain more accurate and correct results, even if it has a negative effect on the mass conservation properties.

Remarks on further work :

- A volume conservation procedure needs to be introduced in the algorithm. This is mostly needed to control the mass loss happening during the reinitialisation step.
- The solution of the Navier-Stokes equations needs to be stabilise in order to be able to compute flows where the density ratio between the two fluids is very large (1000:1 for example in the case of water/air). Indeed, when there is a density jump across the interface, a slip line is generated at the interface. This means that there is contact discontinuity where the tangential velocity jumps over a thickness which corresponds to the thickness of the interface. The slip line will cause instabilities (the larger will be the density ratio, the stronger will be the instabilities) unless some kind of "robust" upwind/stabilisation procedure for the discretization of the nonlinear terms in the Navier-Stokes equations.
- Surface tension needs to be incorporated.
- Generalisation of the code such that its application to multiphase flows (where the number of different fluids is > 2) is possible.

References

[1] A. J. Chorin. *Numerical Solution of the Navier-Stokes Equations.* Math. Comput., vol. 22, pp. 742-762, 1968

[2] Jens Holmen. *Some Numerical Methods for the Simulation of Laminar and Turbulent Incompressible Flows.* Ph.D. thesis no. 2002:6, Norwegian University of Science and Technology, Trondheim, Norway, 2002

[3] Hans Petter Langtangen. *Computational Partial Differential Equations Numerical Methods and Diffpack Programming.* Lecture Notes in Computational Science and Engineering, Springer, 1991

[4] Otto Munthe. *A Critical Evaluation of some Finite Element Level Set Schemes for Two-Phase Fluid Flow.* In Finite element algorithms and object-oriented simulator design in viscous fluid dynamics, Dr. Scient. thesis no. 22, University of Oslo, Department of Mathematics, Mechanics Division, 1999

[5] S. Osher and J.A. Sethian. *Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations.* Journal of Computational Physics, vol. 79, pp. 12-49, 1988

[6] Peter L. O'Sullivan. *Numerical simulation of two dimensional two-phase flows with sharp interfaces.* Research Report SINTEF no. STF42 A02016

[7] Il-Ryong Park and Ho-Hwan Chun. *Analysis of Viscous Free Surface Flow around a Ship by a Level-Set Method.* Journal of Ship & Ocean Technology, vol. 6, no. 2, pp.37-50, 2002

[8] Samuel R. Ransau. *Solution Methods for Incompressible Viscous Free Surface Flows: A Litreature Review.* Preprint Numerics no. 3/2002, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway, 2002

[9] J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 1999

[10] M. Sussman, P. Smereka and S. Osher. *A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow.* Journal of Computational Physics, vol. 114, pp. 146-159, 1994

[11] M. Sussman, E. Fatemi, P. Smereka and S. Osher. *An Improved Level Set Method for Incompressible Two-Phase Flows.* Computers and Fluid, vol. 27, no. 5, pp. 663-680, 1998

[12] M. Sussman *Private communications.* 2002, 2003

[13] R. Temam. *Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I).* Arch. Ration. Mech. Anal., vol. 32, pp. 135-153, 1969