INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **49**:1287–1304
Published online 23 August 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.963

Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions

A. H. Coppola-Owen[‡] and R. Codina[*,†]

Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain

SUMMARY

In this paper we present a problem we have encountered using a stabilized finite element method on fixed grids for flows with interfaces modelled with the level set approach. We propose a solution based on enriching the pressure shape functions on the elements cut by the interface. The enrichment is used to enable the pressure gradient to be discontinuous at the interface, thus improving the ability to simulate the behaviour of fluids with different density under a gravitational force. The additional shape function used is local to each element and the corresponding degree of freedom can therefore be condensed prior to assembly, making the implementation quite simple on any existing finite element code. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: incompressible two-phase flows; stabilized finite element methods; level set; discontinuous gradient shape functions

1. INTRODUCTION

Flows with moving interfaces (free surface and two-fluid interface problems) appear in numerous engineering applications. CFD approaches for such problems can be categorized into two main groups: fixed mesh or interface capturing techniques and moving mesh or interface tracking techniques.

In interface capturing techniques a fixed computational domain is used and an interface function is used to capture the position of the interface. The interface is captured within the resolution of the fixed mesh and the boundary conditions at the interface are somehow approximated.

*Correspondence to: R. Codina, Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.
†E-mail: ramon.codina@upc.edu, http://www.rmee.upc.es/homes/codina
‡E-mail: howen@cimne.upc.edu

Received 15 November 2004
Revised 10 February 2005
Accepted 11 February 2005

In interface tracking techniques the mesh is updated in order to track the interface. The simplest approach is to deform the mesh without changing its topology, but it is valid only for very simple flows. As the flow becomes more complex and unsteady remeshing and consequently the projection of the results from the old to the new mesh are needed. In 3D calculations, these operations can introduce costs that can render moving mesh techniques unfeasible.

For the same mesh size moving grid techniques lead to a more accurate representation of the interface at a higher computational cost. In this paper we will use a fixed grid method and introduce modifications to the basic formulation to enhance the representation of the interface.

The initial motivation for this work came from the impossibility to model a terribly simple flow; in fact, not even a flow, but two different density fluids at rest (the lighter one on top) inside a closed cavity. The hydrostatic pressure gradient is discontinuous at the interface. This cannot be correctly represented by the usual finite element shape functions when the front crosses an element. Since the velocity and pressure are coupled, the error in the representation of the pressure gives rise to spurious velocities that, depending of the properties used, can completely distort the interface that should otherwise remain horizontal.

The idea of enriching the representation of an unknown at a material discontinuity is not new and several approaches can be found in the literature [1, 2]. In Section 3 we will outline some of the differences between the previous methods and ours.

Fixed mesh methods generally share two basic steps. One where the motion in both phases is found as the solution of the Navier–Stokes equations for a one phase flow with variable properties. The other one, where an equation for an interface function that allows to determine the position of the interface, and thus the properties to be assigned in the previous step, is solved. The different methods differ mainly in the method used to determine the position of the interface but also differences can be found in the way to approximate the properties to be used close to the interface.

Referring to the evolution of the fluid interface, we update it using the so-called *level set* method (see References [3–5] for an overview), also called *pseudo-concentration technique* [6] and very similar to the volume of fluid (VOF) technique [7]. This formulation has been widely used to track-free surfaces in mould filling (see, for example, References [8, 9], among other references) and other metal forming processes.

The level set method leads to a transport partial differential equation the solution of which determines the position of the free surface as an isovalue of the unknown of this equation, which we will call ψ. This equation is hyperbolic and therefore it is also necessary to use a stabilized finite element method to solve it. The contribution we intent to introduce in this paper does not depend on the approach used to capture the interface and should also be valid using any of the other cited techniques.

The numerical formulation presented here to solve the incompressible Navier–Stokes equations uses a time discretization based on the standard trapezoidal rule and a stabilized finite element method referred to as algebraic sub-grid scales (ASGS) [10]. It is designed to allow both *equal velocity–pressure interpolations* (thus avoiding the need to satisfy the classical inf–sup condition) and to deal with *convection-dominated flows*, that is, situations in which the cell Reynolds number is greater than unity. $P1$–$P1$ elements will be used. In the elements cut by the interface the $P1$ pressure shape functions are supplemented with an additional shape function that is zero at all the element nodes, continuous within the element and has a constant gradient on each side of the interface. This shape function is local to each element

and the corresponding degree of freedom can therefore be condensed prior to assembly, making the implementation quite simple on any existing finite element code. The details will be discussed in Section 3.

The remainder of the paper is organized as follows. In Section 2 we describe the mathematical model used to solve Navier–Stokes equations when no enrichment functions are used and in Section 3 we briefly describe the Level Set Method used. In Section 4 we present the enrichment functions used and some implementation details. Finally, in Section 5 we present three simple numerical examples where the improvements obtained with the proposed formulation show up clearly.

2. FINITE ELEMENT APPROXIMATION OF THE TWO-FLUID NAVIER–STOKES EQUATIONS

The velocity and pressure fields of two incompressible fluids moving in the domain $\Omega = \Omega_1 \cup \Omega_2$ during the time interval (t_0, t_f) can be described by the incompressible two-fluid Navier–Stokes equations [3]:

$$\rho \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right] - \nabla \cdot [2\mu \boldsymbol{\varepsilon}(\mathbf{u})] + \nabla p = \mathbf{f} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

where ρ is the density, $\mathbf{u}$ the velocity field, μ the dynamic viscosity, p the pressure, $\boldsymbol{\varepsilon}(\cdot)$ the symmetric gradient operator and $\mathbf{f}$ the vector external body forces, which includes the gravity force $\rho\mathbf{g}$ and buoyancy forces, if required. The density, velocity, dynamic viscosity and pressure are defined as

$$\mathbf{u}, p, \rho, \mu = \begin{cases} \mathbf{u}_1 \;\; p_1, \;\; \rho_1, \;\; \mu_1, & \mathbf{x} \in \Omega_1 \\ \mathbf{u}_2 \;\; p_2, \;\; \rho_2, \;\; \mu_2, & \mathbf{x} \in \Omega_2 \end{cases}$$

where Ω_1 indicates the part of Ω occupied by fluid number 1 and Ω_2 indicates the part of Ω occupied by fluid number 2. The extent of Ω_1 and Ω_2 is given by the level set function ψ.

Let $\boldsymbol{\sigma}$ be the stress tensor and $\mathbf{n}$ the unit outward normal to the boundary $\partial\Omega$. Denoting by an over-bar prescribed values, the boundary conditions to be considered are

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_{\text{du}}, \quad \mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \text{on } \Gamma_{\text{nu}}, \quad \mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_1 = 0, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_2 = 0 \quad \text{on } \Gamma_{\text{mu}}$$

for $t \in (t_0, t_f)$. Vectors $\mathbf{g}_1$ and $\mathbf{g}_2$ (for the 3D case) span the space tangent to Γ_{mu}. Observe that Γ_{du} is the part of the boundary with Dirichlet velocity conditions, Γ_{nu} the part with Neumann conditions (prescribed stress) and Γ_{mu} the part with mixed conditions. These three parts do not intersect and are a partition of the whole boundary $\partial\Omega$. Initial conditions have to be appended to the problem.

ASGS [10] stabilization is used to deal with convection-dominated flows and to circumvent the well known div-stability restriction for the velocity and pressure finite element spaces [11], allowing in particular equal interpolation for both unknowns.

Let $\mathbf{V}_h^*$ and Q_h^* be the finite element spaces to interpolate vector and scalar functions, respectively, constructed in the usual manner and using the same interpolation from a finite

element partition $\Omega = \bigcup \Omega^e$, $e = 1, \ldots, n_{\text{el}}$, where n_{el} is the number of elements. From these spaces one can construct the subspaces $\mathbf{V}_{h,u}$ and Q_h for the velocity and the pressure, respectively. The former incorporates the Dirichlet conditions for the velocity components and the latter has one pressure fixed to zero if the normal component of the velocity is prescribed on the whole boundary. The space of velocity test functions, denoted by $\mathbf{V}_h$, is constructed as $\mathbf{V}_{h,u}$ but with functions vanishing on the Dirichlet boundary. Let also θ, with $0 < \theta \leqslant 1$, be the parameter of the trapezoidal rule for time discretization and δt the time step size, for simplicity constant. The algorithmic solution to the problem will be computed at $t^n = n\delta t$, $n = 1, 2, \ldots$. The ASGS monolithic discrete problem associated with the Navier–Stokes equations (1)–(2), discretizing in time using the generalized trapezoidal rule, and linearizing the convective term using a Picard scheme, can be written as follows: Given a velocity $\mathbf{u}_h^n$ at time t^n and a guess for the unknowns at an iteration $i-1$ at time t^{n+1}, find $\mathbf{u}_h^{n+\theta,i} \in \mathbf{V}_{h,u}$ and $p_h^{n+\theta,i} \in Q_h$, by solving the discrete variational problem:

$$
\begin{aligned}
&\int_\Omega \rho \frac{\mathbf{u}_h^{n+\theta,i} - \mathbf{u}_h^n}{\theta \delta t} \cdot \mathbf{v}_h \, \mathrm{d}\Omega + \int_\Omega \rho (\mathbf{u}_h^{n+\theta,i-1} \cdot \nabla) \mathbf{u}_h^{n+\theta,i} \cdot \mathbf{v}_h \, \mathrm{d}\Omega \\
&\quad + \int_\Omega \mu \boldsymbol{\varepsilon}(\mathbf{u}_h^{n+\theta,i}) : \boldsymbol{\varepsilon}(\mathbf{v}_h) \, \mathrm{d}\Omega - \int_\Omega \nabla \cdot \mathbf{v}_h \, p_h^{n+\theta,i} \, \mathrm{d}\Omega - \int_\Omega \mathbf{v}_h \cdot \mathbf{f} \, \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_1^{n+\theta,i-1} \left[\frac{\mu}{\rho} \Delta \mathbf{v}_h + (\mathbf{u}_h^{n+\theta,i-1} \cdot \nabla) \mathbf{v}_h \right] \cdot \left[\frac{\rho}{\theta \delta t} (\mathbf{u}_h^{n+\theta,i} - \mathbf{u}_h^n) \right. \\
&\quad \left. - \mu \Delta \mathbf{u}_h^{n+\theta,i} + \rho (\mathbf{u}_h^{n+\theta,i-1} \cdot \nabla) \mathbf{u}_h^{n+\theta,i} + \nabla p_h^{n+\theta,i} - \mathbf{f} \right] \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_2^{n+\theta,i-1} (\nabla \cdot \mathbf{v}_h)(\nabla \cdot \mathbf{u}_h^{n+\theta,i}) \, \mathrm{d}\Omega = 0 \quad \forall \ \mathbf{v}_h \in \mathbf{V}_h \\
&\int_\Omega \rho q_h \nabla \cdot \mathbf{u}_h^{n+\theta,i} + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \tau_1^{n+\theta,i} \nabla q_h \cdot \left[\frac{\rho}{\theta \delta t} (\mathbf{u}_h^{n+\theta,i} - \mathbf{u}_h^n) \right. \\
&\quad \left. - \mu \Delta \mathbf{u}_h^{n+\theta,i} + \rho (\mathbf{u}_h^{n+\theta,i-1} \cdot \nabla) \mathbf{u}_h^{n+\theta,i} + \nabla p_h^{n+\theta,i} - \mathbf{f} \right] \mathrm{d}\Omega = 0 \quad \forall \ q_h \in Q_h
\end{aligned}
$$

for $i = 1, 2, \ldots$ until convergence, that is to say, until $\mathbf{u}_h^{n+\theta,i-1} \approx \mathbf{u}_h^{n+\theta,i}$ and $p_h^{n+\theta,i} \approx p_h^{n+\theta,i-1}$ in the norm defined by the user.

The parameters τ_1 and τ_2 are chosen in order to obtain a stable numerical scheme with optimal convergence rates (see Reference [10] and references therein for details). They are computed within each element domain Ω^e. We take them as:

$$
\tau_1 = \frac{\rho (h^e)^2}{4\mu + 2\rho h^e |\mathbf{u}^e|} \quad \text{and} \quad \tau_2 = \mu + \tfrac{1}{2} \rho h^e |\mathbf{u}^e|
$$

where h^e and $|\mathbf{u}^e|$ are a typical length and a velocity norm of element e, respectively.

Once the algorithm has produced a converged solution, the velocity field at t^{n+1} can be updated from the velocity at $t^{n+\theta}$ by using the relation $\mathbf{u}^{n+1} = [\mathbf{u}^{n+\theta} - (1-\theta)\mathbf{u}^n]/\theta$.

Two options have been used to approximate the material properties in the elements cut by the interface. In the first one, which we will call *variable*, the material properties (μ, ρ) are approximated at each integration point (k) according to

$$\pi_k = \psi_k \pi_1 + (1 - \psi_k)\pi_2$$

where π_k stands for the value of μ or ρ to be used at an integration point k, ψ_k is the value of ψ at the integration point and π_1, π_2 are the material properties of fluids 1 and 2, respectively. In this equation, it is implied that ψ varies between 0 and 1. Its construction is described in the following section. In the second option, which we will call *jump*, π_k is taken as either π_1 or π_2 depending on the values of ψ_k.

The enrichment technique presented in Section 4 can be understood as a modification of the pressure space Q_h to $\hat{Q}_h$, with $Q_h \subset \hat{Q}_h$. Apart from this, the resulting formulation follows exactly the previous setting.

3. IMPLEMENTATION OF THE LEVEL SET METHOD

The basic idea of the level set method is to define a scalar function, say $\psi(\mathbf{x}, t)$, over the computational domain Ω that determines the extent of subdomains Ω_1 and Ω_2. For instance, and as it has been mentioned above, we may assign the value 1 to the points belonging to Ω_1 and the value of 0 to the points belonging to Ω_2 . The position of the fluid front will be defined by the isovalue contour $\psi(\mathbf{x}, t) = \psi_c$, where $\psi_c \in [0, 1]$ is a critical value defined *a priori*. The value $\psi_c = 0.5$ is usually taken. This value is immaterial if ψ is a true step function, but is needed in the finite element discretization and for the smoothing to be described later. The conservation of ψ (assumed to be sufficiently smooth) in any control volume $V_t \subset \Omega$ which is moving with the divergence-free velocity field $\mathbf{u}$ leads to

$$\frac{\partial \psi}{\partial t} + (\mathbf{u} \cdot \nabla)\psi = 0 \quad \text{in } \Omega \times (t_0, t_f) \tag{3}$$

This equation is hyperbolic and therefore boundary conditions for ψ have to be specified at the inflow boundary, defined as:

$$\Gamma_{\text{inf}} := \{\mathbf{x} \in \partial\Omega \mid \mathbf{u} \cdot \mathbf{n} < 0\}$$

Function ψ is the solution of the hyperbolic equation (3) with the boundary conditions:

$$\psi = \overline{\psi} \qquad \text{on } \Gamma_{\text{inf}} \times (t_0, t_f)$$

$$\psi(\mathbf{x}, 0) = \psi_0(\mathbf{x}) \quad \text{in } \Omega$$

The initial condition ψ_0 is chosen in order to define the initial position of the fluid front to be analysed. The boundary condition $\overline{\psi}$ determines whether fluid enters or not through a certain point of the inflow boundary.

Due to the pure convective type of the equation for ψ, we use the SUPG technique for the spatial discretization. Again, the temporal evolution is treated via the standard trapezoidal rule.

If ψ is taken as a step function, as indicated before, numerical problems may be encountered when it is transported. It is known that small oscillations in the vicinity of sharp gradients still remain using the SUPG formulation. These oscillations may propagate and yield to distorted front shapes, specially near corners. Since the basic idea of the method does not depend on the choice of the function ψ, it is preferable to use a smooth function instead of one with abrupt changes [8]. This can be achieved by redefining ψ for each node of the finite element mesh according to the following expression:

$$\psi = \psi_c + \mathrm{sgn}(\psi^0 - \psi_c) f(d) \tag{4}$$

where ψ^0 stands for the calculated value of ψ, $f(d)$ is a function proportional to the distance (d) from the node under consideration to the front, and $\mathrm{sgn}(\cdot)$ is the signum of the value enclosed in the parenthesis.

Since the objective of this paper is to analyse the improvements that can be obtained in the solution of the two-fluid Navier–Stokes equations and not to optimize the solution of the Level Set equations, a very simple algorithm has been used to calculate the distance d. Using linear elements, the free surface is approximated by triangular planes p (lines in 2D). Then the perpendicular distance $d_i p$ of each grid point i to each plane p can be computed. The minimum distance from each nodal point to the planes is the required distance between the point and the front ($d_i = \min_p\{d_i p\}$).

4. DISCONTINUOUS GRADIENT PRESSURE SHAPE FUNCTIONS

In fixed grid finite element methods the whole domain Ω is subdivided into elements Ω^e. Within each element the unknowns are interpolated as

$$\phi_h|_{\Omega^e} = \sum_{I=1}^{\mathrm{NNODE}} N_e^I \Phi_e^I$$

where NNODE is the number of element nodes.

In typical finite element methods, ∇N_e^I are continuous within each element and therefore $\nabla \phi_h|_{\Omega^e}$ is continuous. When the interface crosses an element the discontinuity in the material properties leads to discontinuities in the gradients of the unknowns that the interpolation used cannot capture. For example, as mentioned previously, for two different density fluids at rest the interpolation errors in the pressure give rise to spurious velocities that can render the solution meaningless. Also, viscosity discontinuities can lead to discontinuous velocity gradients.

Enrichment methods add degrees of freedom at elements cut by the interface in order to reduce interpolation errors. In our particular case we add only one pressure degree of freedom per cut element. Therefore the pressure in elements cut by the interface is interpolated as

$$p_h|_{\Omega^e} = \sum_{I=1}^{\mathrm{NNODE}} N_e^I P_e^I + N_e^{\mathrm{ENR}} P_e^{\mathrm{ENR}} \tag{5}$$

The shape function N_e^{ENR} we introduce has a constant gradient on each side of the interface, its value is zero at the element nodes and is C^0 continuous in Ω^e. The added degree of freedom is local to the element and can therefore be condensed after the element matrix has

been computed and before assembly. The resulting pressure finite element space is made of functions discontinuous across interelement boundaries, and thus it is a subspace of $L^2(\Omega)$, but not of $H^1(\Omega)$, as would be the case using $P1$–$P1$ elements. However, our method is still conforming. If we had tried to use the previous enrichment functions for the velocity we would have obtained a non-conforming method.

Minev and co workers [1] also use enrichment functions for two-fluid flows. They use discontinuous gradient velocity shape functions and discontinuous pressure shape functions because they include surface tension. No discontinuous gradient pressure shape functions are included. The velocity enrichment functions are not only local to each element, as in our case, but also continuous across elements. This is possible because they are the product of a bubble function times some other function. Tempted by the fact of being able to condense the enriched degree of freedom while using H^1 functions, we initially tried to use the enrichment they use for the velocity for the pressure, but obtained very unsatisfactory results in the hydrostatic two-fluid case. This led us to look for enrichment functions that could model constant gradients on each side of the interface.

Also for two-fluid problems, Chessa and Belytschko [2] use an enrichment method called XFEM, initially developed by the second author for modelling cracks. As in the case of Minev *et al.*, they use discontinuous gradient velocity shape functions and not discontinuous gradient pressure shape functions. The enrichment they use is continuous and cannot be condensed prior to matrix assembly. It is computationally much more expensive because the mesh graph needs to be updated as the interface moves from one element to another.

In Figure 1 we show a sketch of the enrichment function we use for an element cut by the interface in the 2D case. The element has nodes named 1, 2 and 3 and the interface cuts the element edges at points A and B. A way to build such function is as follows. Suppose that node 1 belongs to Ω_1 and nodes 2 and 3 belong to Ω_2. Let $\Omega_1^e = \Omega_1 \cap \Omega^e$ and $\Omega_2^e = \Omega_2 \cap \Omega^e$. In Ω_2^e we want N^{ENR} to have constant gradient and to have a zero value at $\mathbf{x}_2$ and $\mathbf{x}_3$. We can therefore define

$$N^{\mathrm{ENR}}|_{\Omega_2^e} = k^1 N^1|_{\Omega_2^e}$$

where k^1 is a constant to be defined. By definition we want $N^{\mathrm{ENR}}(\mathbf{x}_A) = 1$. As we are using linear elements to interpolate the level set function we have that

$$N^1(\mathbf{x}_A) = \frac{\Psi^2 - \psi_{\mathrm{c}}}{\Psi^2 - \Psi^1}$$

where Ψ^i is the value of ψ at node i, and therefore

$$k^1 = \frac{\Psi^2 - \Psi^1}{\Psi^2 - \psi_{\mathrm{c}}}$$

Now we have k^1 we can find

$$N^{\mathrm{ENR}}(\mathbf{x}_B)|_{\Omega_2^e} = k^1 N^1(\mathbf{x}_B)|_{\Omega_2^e} = k^1 \frac{\Psi^3 - \psi_{\mathrm{c}}}{\Psi^3 - \Psi^1}$$

We can proceed to find $N^{\mathrm{ENR}}|_{\Omega_1^e}$. We want it to have a constant gradient in Ω_1^e and to be zero at $\mathbf{x}_1$. Then

$$N^{\mathrm{ENR}}|_{\Omega_1^e} = k^2 N^2|_{\Omega_1^e} + k^3 N^3|_{\Omega_1^e}$$

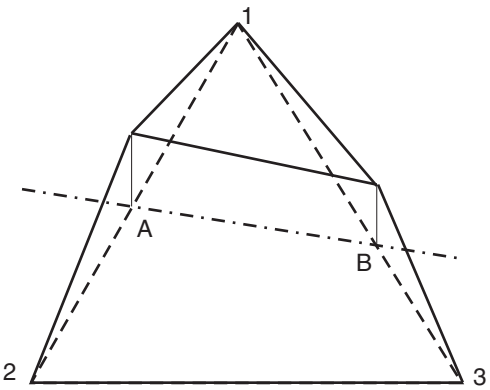

Figure 1. 2D Enrichment function for a cut element.

Using once more that $N^{\mathrm{ENR}}(\mathbf{x}_A)=1$ and the fact that $N^3(\mathbf{x}_A)=0$ we get

$$k^2=\frac{1}{N^2(\mathbf{x}_A)}=\frac{\Psi^1-\Psi^2}{\Psi^1-\psi_{\mathrm{c}}}$$

Since we want the enrichment function to be continuous in Ω^e we need

$$N^{\mathrm{ENR}}(\mathbf{x}_B)|_{\Omega_2^e}=N^{\mathrm{ENR}}(\mathbf{x}_B)|_{\Omega_1^e}$$

then, as $N^2(\mathbf{x}_B)=0$,

$$k^3=N^{\mathrm{ENR}}(\mathbf{x}_B)|_{\Omega_2^e}\frac{1}{N^3(\mathbf{x}_B)}=k^1\frac{\Psi^3-\psi_{\mathrm{c}}}{\Psi^3-\Psi^1}\frac{\Psi^1-\psi_{\mathrm{c}}}{\Psi^1-\Psi^3}$$

$$k^3=-k^1\frac{(\Psi^3-\psi_{\mathrm{c}})(\Psi^1-\psi_{\mathrm{c}})}{(\Psi^3-\Psi^1)^2}$$

We have obtained an enrichment function that is proportional to N^1 on Ω_2^e and a linear combination of N^2 and N^3 on Ω_1^e, where the values of k^1,k^2,k^3 only depend on the values of the level set function at the element nodes. It is very easy to obtain the enrichment function an its Cartesian derivates from the usual shape function. It seems worthwhile to remark that $N^{\mathrm{ENR}}|_{\Omega^e}$ does not belong to the space formed by $N^1|_{\Omega^e}$, $N^2|_{\Omega^e}$, $N^3|_{\Omega^e}$. The same ideas have been used to obtain N^{ENR} for 3D elements.

In order to capture the discontinuities and take advantage of the enrichment functions used, the integration rules need to be modified in elements cut by the front. The method we use is to divide each tetrahedral (triangular in 2D) element into up to six tetrahedral (three triangular in 2D) sub elements. For each sub element the same integration rule as for the non-cut elements is used.

When using enrichment functions for the pressure, the material properties μ,ρ are taken as μ_1,ρ_1 or μ_2,ρ_2 depending on which part of the domain (Ω_1 or Ω_2) the integration point is found.

Since the pressure space is enriched, a remark is needed concerning pressure stability. If we had used a velocity–pressure interpolation satisfying the inf–sup condition, the enrichment of the pressure could have led to an unstable velocity–pressure pair. However, we are using a stabilized finite element formulation. Even though we have no stability analysis for the enriched pressure space, we have not encountered any type of stability misbehaviour.

A final remark is required concerning the extension of the proposed enrichment to higher order elements. Since the intention is to add a pressure field able to deal with discontinuous pressure gradients, but constant in each fluid phase, exactly the same methodology as described for P_1 elements can be applied to higher order elements. The construction of the enriched pressures can be based only in the linear part of the interpolation basis functions of these higher order elements. This is particularly simple when they are implemented using a hierarchical basis. The case of quadrilateral elements (or hexahedra in 3D) can be dealt with by splitting the quadrilateral into triangles (or hexahedra).

5. NUMERICAL EXAMPLES

In this section we present three numerical examples where the improvements obtained with the proposed formulation show up clearly. The first two examples are related to the original two-fluid hydrostatic problem but modified so that they have non-zero velocities. The results obtained with the enriched formulation are compared with those obtained with a typical finite element formulation with no enrichment nor improved integration. Since we are trying to prove that the benefits come from the pressure enrichment and not from the improved integration, an intermediate case where no enrichment is used and we only modify the integration is also presented. Regarding the approximation of the material properties in elements cut by the interface, we have mentioned two possibilities in Section 2, *variable* and *jump* properties. When we use improved integration, with or without enriched pressures, the natural choice is to use *jump* properties. When no improved integration nor pressure enrichment is used *jump* or *variable* properties can be used.

The first example, called the two-fluid cavity, is a square domain filled with equal amounts of two different density fluids and a fixed horizontal velocity (0.1 m/s) on the bottom wall. The walls are supposed frictionless. Jump properties are used to approximate the material properties in elements cut by the interface. It is a very simple example but it can be representative of the numerical problems that can appear in much more complex problems such as the two-phase flow in a stirred reactor.

Three 2D unstructured triangular meshes were used (see Table I). The square domain has a side length $L = 10\,\text{m}$. The material properties used (SI units) are $\rho_1 = 1000$, $\mu_1 = 10$ for the fluid on the bottom, and $\rho_2 = 900$, $\mu_2 = 9$ for the one on top. The viscosity of the bottom fluid is 1000 times the viscosity of water so as to obtain a relatively low Reynolds number

Table I. Meshes used for the two fluid cavity flow.

Mesh	N_{poin}	N_{elem}	Elem length
Coarse	128	214	1.0
Medium	472	862	0.5
Very fine	11615	22828	0.1

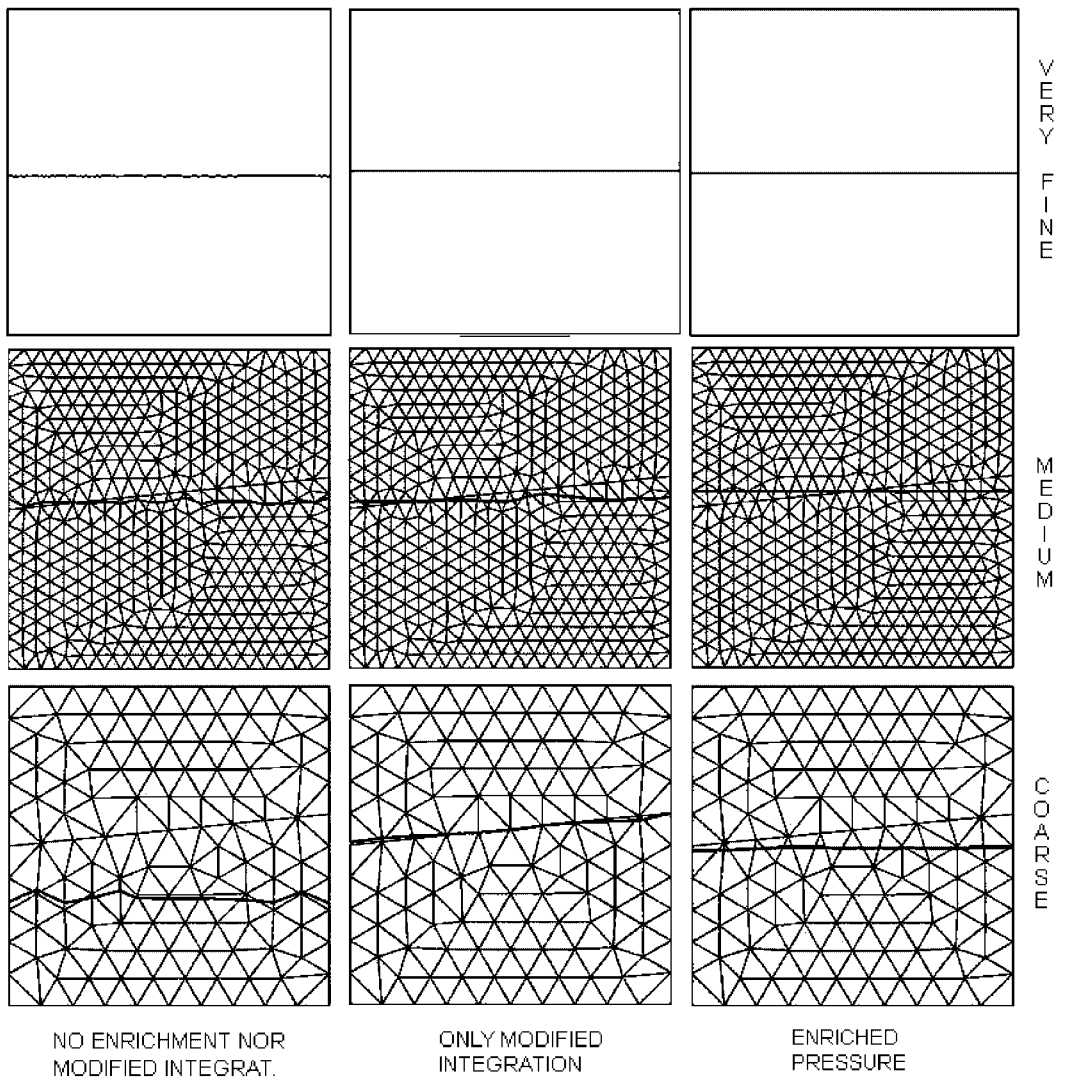

Figure 2. Shape of the interface for the different meshes and numerical conditions at $t = 100$ s.

($Re = 100$) in order to avoid unnecessary complications. The simulations where run for 100 s with a 0.5 s time step size. In all the examples presented in this paper the acceleration of gravity is $g = 10$.

In Figure 2 we show the shape of the interface for the three meshes in the three different conditions mentioned previously: with no enrichment nor improved integration, using only modified integration but no pressure enrichment and finally with both pressure enrichment and improved integration. Using the finest mesh the three methods give nearly the same result. Only in the case with no enrichment nor improved integration, slight oscillations can be observed. Despite we have not got physical measurements, the solutions with this mesh can

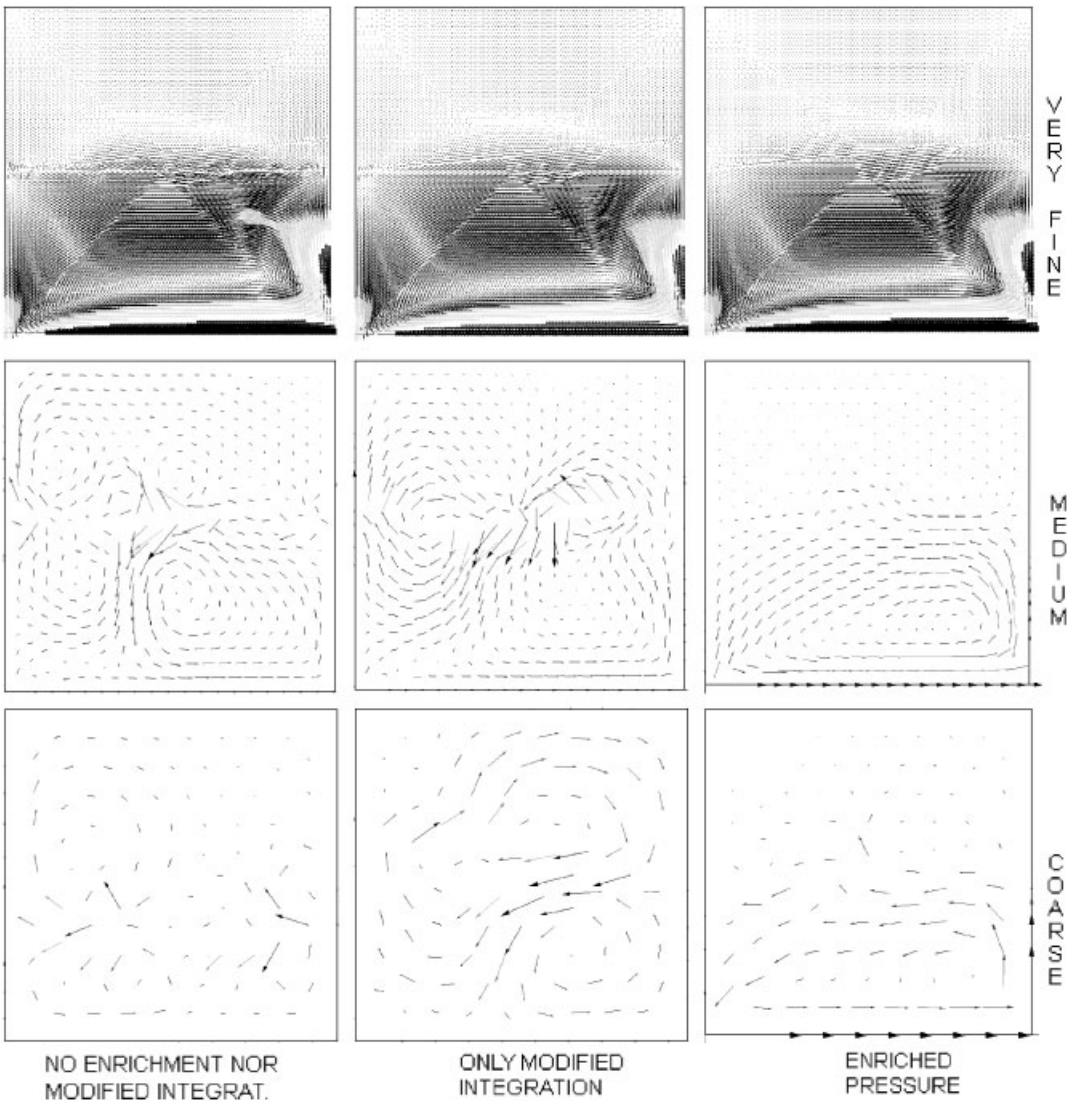

Figure 3. Flow pattern for the different meshes and numerical conditions at $t = 100$ s.

be taken as a reference against which we can compare the results obtained with the other two meshes. Using the medium mesh, only the simulation with both pressure enrichment and improved integration attains results nearly as good as those obtained with the very fine mesh. The other two cases show a distorted interface shape. Finally, using the coarse mesh, the shape of the interface using pressure enrichment and improved integration shows slight errors but is much better than the other two cases, where it can be clearly observed that mass is not conserved.

The flow pattern is compared in Figure 3. Using the finest mesh there is not much difference between the three methods. Two recirculations can be found, one in the bottom fluid and one

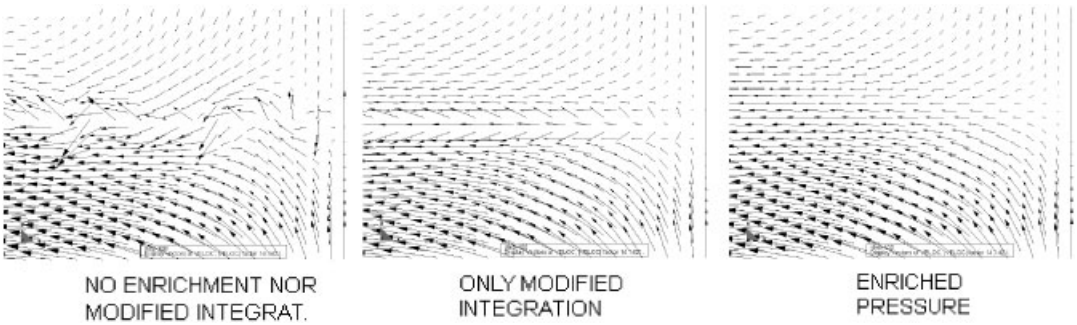

Figure 4. Detailed flow pattern for the finest meshes and different numerical conditions close to the interface.

in the top one. With the medium mesh, the results obtained with pressure enrichment remain very similar to the ones obtained with the previous mesh. In the other two cases the flow pattern is strongly modified by the errors that originate close to the interface. With the coarse mesh the results deteriorate in all three cases as expected, but it can be observed that in the case with pressure enrichment the solution is still better than that obtained with a medium mesh in the other two cases. Even though the global flow pattern obtained with the very fine mesh is nearly the same in all three cases, a zoom at the velocity vectors close to the interface (see Figure 4) reveals that even in this case, the pressure enrichment produces a better solution. The spurious oscillatory behaviour obtained close to the interface in the cases without pressure enrichment has been observed not only in space but also in time. The effect of reducing the time step size has been analysed, but no significant improvements have been obtained compared to those resulting from the pressure enrichment.

Finally, it has been observed that the convergence in the L^2 velocity norm within each time step is much better using the enriched formulation than without it, for all three meshes. Using a 0.0001 relative convergence tolerance for the velocity, the two formulations without enrichment converge in twice or more iterations than the enriched one. The enriched case takes 2 iterations to converge with the fine mesh, 3 with the medium mesh, and 4 with the coarse one.

The second example is a 20 m high vertical channel with a square cross section (side length $L = 5$ m). The channel is fed from the bottom with a heavier fluid at a constant (both in space and time) 1 m/s velocity and the upper face is left free so that the lighter fluid can escape. No friction is assumed on the walls. The initial interface is flat and at 2.5 m from the entrance. The solution for this problem is very simple. The velocity in the whole domain, included the interface, should be equal to the inlet velocity and the interface should remain flat.

A 3D unstructured tetrahedral mesh with 1106 nodes and 4921 elements is used. The material properties used (SI units) are $\rho_1 = 1000$, $\mu_1 = 100$ for the fluid on the bottom, and $\rho_2 = 10$, $\mu_2 = 1$ for the one on top. The time step size is 0.1 s. The Reynolds number based on the length of the square section is $Re = 50$. In the case with no improved integration nor pressure enrichment both jump or variable properties have been tested.

In Figures 5–8 we show the shape of the interface and the velocity field for four different cases. In Figure 5 the results with pressure enrichment are shown. The interface remains flat as expected and its displacement corresponds to the amount of injected fluid. When a modified

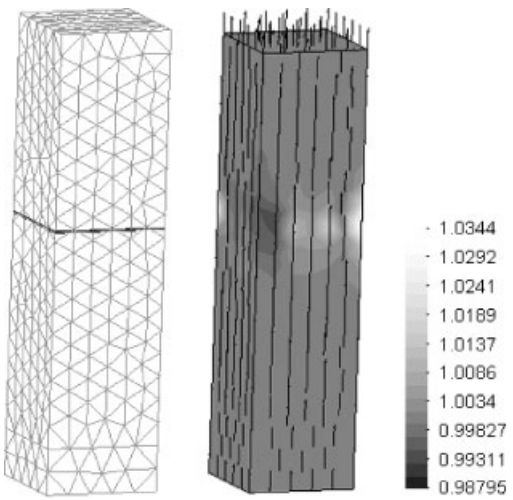

Figure 5. Interface shape and vertical velocity band plot together with velocity vectors using pressure enrichment.

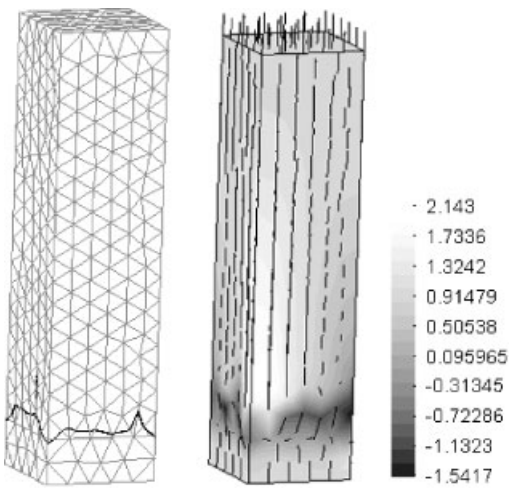

Figure 6. Interface shape and vertical velocity band plot together with velocity vectors using only improved integration.

integration but no pressure enrichment is used (see Figure 6) the results are very poor. The velocity shows important oscillations close to the interface and there is an important mass loss. The mass loss is so important that the free surface remains nearly at its initial height. Without using pressure enrichment nor modified integration and approximating the material properties with the option described as *jump* properties, the results (shown in Figure 7) are as bad the those described for the previous case. Finally, when variable properties are used

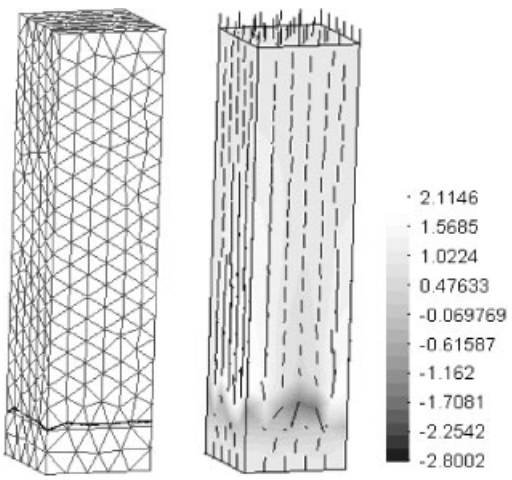

Figure 7. Interface shape and vertical velocity band plot together with velocity vectors using jump properties and no enrichment nor modified integration.

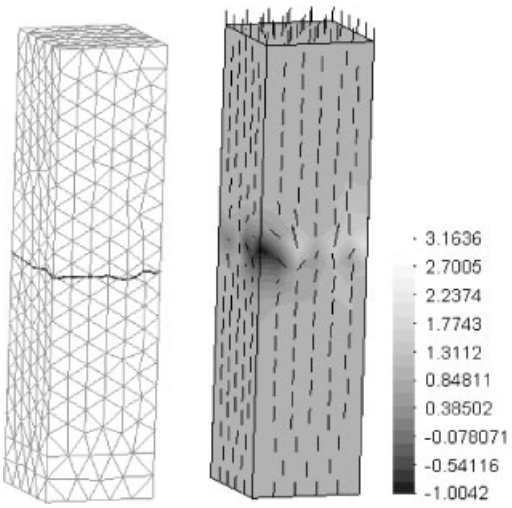

Figure 8. Interface shape and vertical velocity band plot together with velocity vectors using variable properties and no enrichment nor modified integration.

(see Figure 8), without pressure enrichment nor modified integration, the mass conservation improves with respect to the previous two cases, but is worse than that obtained with the formulation proposed in this paper. As in the other two cases without pressure enrichment, the errors in the prediction of the vertical velocity close to the interface can be twice the inlet velocity. The source of these errors is the impossibility of the shape functions to capture the discontinuous pressure gradient that exists at the interface. When the pressure is enriched the

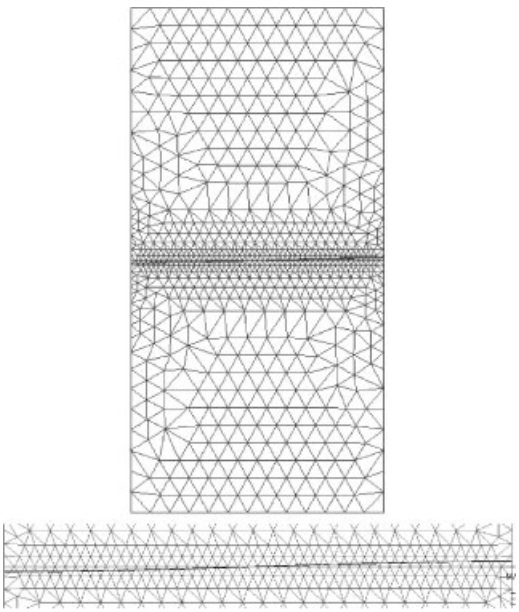

Figure 9. Mesh and initial interface position for the sloshing problem. Top: general view. Bottom: detail.

velocity errors nearly disappear. The slight errors that remain can be attributed to the fact that the numerical resolution of the level set is not exact and therefore the deviations in the shape of the interface give rise to small variations in the velocity.

As a final example, we consider a sloshing problem. It is a simple problem of free oscillation of an incompressible liquid in a container. Following References [12–14] we consider a liquid column of width b with an initial surface profile corresponding to the first antisymmetric mode of vibration. The height of the free interface is

$$\eta(x,0) = 1.0 + a \sin \frac{\pi}{b} x$$

where a is the amplitude of oscillation. Thus, the initial condition for the level set function is

$$\psi(x,y,0) = \eta(x,0) - y + \psi_c$$

where $x = 0$ at the middle of the container and $y = 0$ at the bottom. Using the same parameters as in the cited papers, the amplitude is taken as $a = 0.01$ and the cinematic viscosity as $v = 0.01$. The previous papers use a Lagrangian formulation and therefore only one fluid is solved and the computational domain is allowed to deform. Since we use an Eulerian formulation the computational domain does not deform but a second fluid is also simulated. The density and dynamic viscosity of the second fluid are 100 times smaller than those of the bottom one so that it does not affect significantly the flow of the heavier fluid. The container walls are assumed to be impermeable and allow for free slip. The domain we use has a width $b = 1$ and a height $h = 2$ and is filled with equal amounts of each fluid (see Figure 9). Jump properties are used to approximate the material properties in elements cut by the interface. The mesh has 1394 triangular elements and 747 nodes and is refined close to the interface.

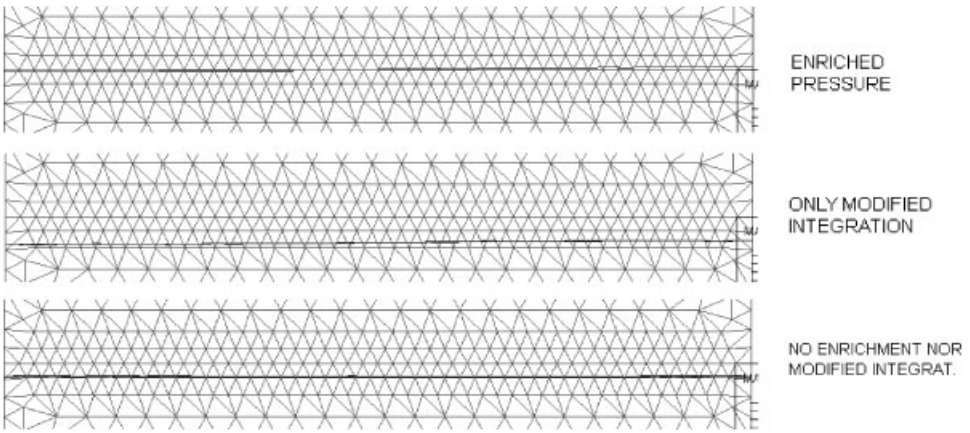

Figure 10. Interface position at $t = 11$ s for the sloshing problem.

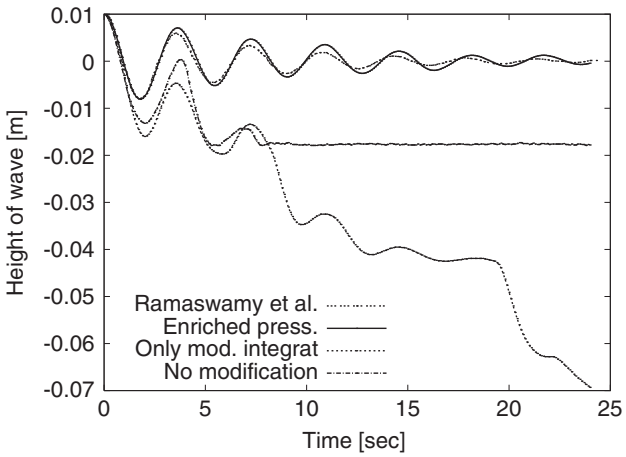

Figure 11. Time histories of surface elevation amplitude for the sloshing problem.

In Figure 10 we show the position of the interface after 11 s, using: (1) enriched pressures, (2) only improved integration and (3) no modification. Figure 11 shows the computed time history of $\eta(b/2, t)$ for the three cases together with the results presented by Ramaswamy *et al.* [13]. Both figures show that in the cases without pressure enrichment there is a significant mass loss. The enriched simulation agrees closely with the results reported by Ramaswamy *et al.* [13].

6. CONCLUSIONS

In this paper we have presented an enrichment for the pressure finite element shape functions that allows to improve the solution of two phase flows. The benefits introduced by the method

show up clearly as the gravitational forces increase. The Froude number

$$Fr = \frac{U^2}{gL}$$

characterizes the ratio of the inertial to the gravitational forces in free-surface flows, where U is a characteristic velocity, g is the value of the gravitational acceleration and L is a characteristic length. A modified Froude number can be used for two fluid flows

$$Fr_m = \frac{U^2}{gL}\frac{\rho_1}{(\rho_1 - \rho_2)}$$

where ρ_1 is the density of the heavier fluid and ρ_2 is the density of the lighter one. The errors without using the enrichment increase as the Froude number tends to zero. As the gravitational forces tend to dominate the flow, small errors in the pressure can give rise to big errors in the velocities. Therefore the type of enrichment presented in this paper is specially useful for low modified Froude number flows. The enrichment used is local to each element cut by the interface and can therefore be condensed prior to assembly, making the implementation quite simple on any finite element code.

It is interesting to note that, at least for the problems presented in this paper, the proposed solution can reduce the errors by one or two orders of magnitude (see vertical tube results) and thus make problems solvable with quite coarse meshes.

Despite we have not presented numerical results for the increase in CPU time for obtaining the system to be solved per iteration introduced by the enrichment and improved integration; we have observed that it is generally less or much less than 10%. Taking into account that the method allows to reduce the number of iterations per time step and also to use coarser meshes the mentioned CPU time increase is by far counter balanced.

We have not presented results for the pressure because it is difficult to do so with a typical post-processing program in the enriched elements, but it is evident from the improvements shown for the velocities and shape of the interface that the pressure, since it is the only unknown we have modified, must have also improved.

The examples shown in this paper demonstrate that the proposed enrichment can introduce significant improvements. It allows to avoid spurious velocities and enhances mass conservation.

REFERENCES

1. Minev PD, Chen T, Nandakumar K. A finite element technique for multifluid incompressible flow using Eulerian grids. *Journal of Computational Physics* 2003; **187**:255–273.
2. Chessa J, Belytschko T. A extended finite element method for two-phase fluids. *Journal of Applied Mechanics* 2003; **70**:10–17.
3. Chang YC, Hou TY, Merriman B, Osher S. A level set formulation of Eulerian interface capturing methods. *Journal of Computational Physics* 1996; **124**:449–464.
4. Sussman M, Almgren AS, Colella JB, Howell LH, Welcome ML. An adaptive level set approach for incompressible two phase flows. *Journal of Computational Physics* 1999; **148**:81–124.
5. Osher S, Fedkiw RP. Level set methods: and overview and some recent results. *Journal of Computational Physics* 2001; **169**:463–502.
6. Thompson E. Use of the pseudo-concentration to follow creeping viscous during transient analysis. *International Journal for Numerical Methods in Engineering* 1986; **6**:749–761.
7. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.

8. Codina R, Schäfer U, Oñate E. Mould filling simulation using finite elements. *International Journal of Numerical Methods for Heat and Fluid Flow* 1994; **4**:291–310.
9. Lewis RW, Usmani AS, Cross JT. Efficient mould filling simulation in metal castings by an explicit finite element method. *International Journal for Numerical Methods in Engineering* 1995; **20**:493–506.
10. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2681–2706.
11. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: Berlin, 1991.
12. Ramaswamy B. Numerical simulation of unsteady viscous free surface flow. *Journal of Computational Physics* 1990; **90**:396–430.
13. Ramaswamy B, Kawahara M, Nakayama T. Lagrangian finite element method for the analysis of two-dimensional sloshing problems. *International Journal for Numerical Methods in Fluids* 1985; **6**:659–670.
14. Radovitzky R, Ortiz M. Lagrangian finite element analysis of Newtonian fluid flows. *Journal of Computational Physics* 1998; **43**:607–619.

