INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:293–301 (DOI: 10.1002/fld.277)

A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique

R. Codina[1,*,†] and O. Soto[2,‡]

[1] *Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain*
[2] *SCS/Laboratory for Computational Fluid Dynamics, George Mason University, MS 4C7,*
4400 University Drive, Fairfax, VA 22030-4444, USA

SUMMARY

The objective of this work is to present a stabilized finite element formulation for transient incompressible flows and to apply it to the tracking of two-fluid interfaces. The stabilization technique employed allows us to use equal velocity–pressure interpolations and to deal with convection–dominated flows. The tracking of the fluid interface is based on the level set technique. A novel smoothing technique of this surface based on ideas of signal processing is presented. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: incompressible flows; stabilized finite element methods; level set; surface smoothing

1. INTRODUCTION

The purpose of this paper is two-fold. First, a stabilized finite element method to solve the transient incompressible Navier–Stokes equations is presented. Second, we describe a particular implementation of the so-called level set method to track interfaces between two fluids.

The numerical formulation presented here to solve the incompressible Navier–Stokes equations has three main features: the time discretization based on the standard trapezoidal rule, the stabilized finite element method and the iterative procedure, which in particular aims to uncouple the velocity–pressure calculation.

The important point of the numerical model is the stabilized finite element method we use. 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. This formulation, originally presented in References [1, 2], is briefly described in the following section. It consists of

[*] Correspondence to: R. Codina, Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain.
[†] E-mail: ramon.codina@upc.es, Web page: http://www.rmec.upc.es/homes/codina
[‡] E-mail: soto@scs.gmu.edu, Web page: http://www.scs.gmu.edu/˜soto

Received May 2001
Revised September 2001

adding to the Galerkin terms of the discrete variational equations a least-square form of the component of some terms *orthogonal* to the finite element space. These terms are those responsible for the instabilities of the Galerkin method, namely, the convective term and the pressure gradient.

Another important aspect of the numerical formulation is the iterative scheme. The iterations proposed here aim to deal with the non-linearity of the convective term, the projections involved in the stabilization terms, and also the uncoupling of the pressure from the momentum equation. This is particularly important from the computational standpoint, and needs to be treated carefully. Our scheme is motivated by the pressure segregation of some popular fractional step methods.

Referring to the evolution of the fluid interface, it is tracked 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 important point in this case is that, even if the initial condition ψ_0 is a smooth function, if ψ is maintained unmodified over several time steps it may begin to loose its smoothness and numerical problems may be encountered. Since the only important factor is the location of the critical contour that defines the front, it is possible to smooth ψ while maintaining the position of this critical contour. Several smoothing techniques can be found in the literature. In this work, we present a completely novel smoothing algorithm based on methods of signal processing. The idea is to compute first a 'noisy' distance function from each node to the fluid interface using a very fast algorithm. Once this is done, another algorithm is used to fair this function.

It is not the intention of this paper to assess the accuracy of the level set approach, for which numerous references exist (see References [5, 10–12], where also a finite element formulation is used). The purpose of this article is to show how to use it with our stabilized finite element method and to demonstrate the effectiveness of the smoothing technique proposed here. This will be done in the numerical simulation of an industrial casting process.

2. FINITE ELEMENT APPROXIMATION OF THE INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

The velocity and pressure fields of an incompressible fluid moving in the domain Ω during the time interval (t_0, t_f) are governed by the incompressible Navier–Stokes equations

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

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

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

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

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

for $t\in(t_0,t_f)$. Vectors $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ (for the three-dimensional case) span the space tangent to Γ_{mu}. Initial conditions have to be appended to the problem.

The condition on Γ_{mu} allows the fluid to slip on the boundary and is very often used for solid walls. Likewise, Γ_{nu} will be taken as the *outflow* boundary, where boundary conditions are in fact artificial, in the sense that the correct traction is unknown. On these boundaries we will also prescribe the pressure to zero, which together with $\boldsymbol{n}\cdot\boldsymbol{\sigma}=\boldsymbol{0}$ implies that we consider negligible the normal component of the velocity gradients.

The stabilized formulation that will be used is a monolithic version of the orthogonal subscale stabilization method, hereafter referred to as OSS [1, 2]. The original motivation of this method is that it allows to circumvent the well-known div-stability restriction for the velocity and pressure finite element spaces [13], allowing in particular equal interpolation for both unknowns. Likewise, it allows to deal with convection-dominated flows, eliminating the instability problems encountered when viscous (or diffusive) terms are small.

Let $\boldsymbol{V}_h^*$ 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 these spaces one can construct the subspaces $\boldsymbol{V}_{h,u}$ 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 $\boldsymbol{V}_h$, is constructed as $\boldsymbol{V}_{h,u}$ 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 discretisation 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 OSS monolithic discrete problem associated with the Navier–Stokes equations (1) and (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 $\boldsymbol{u}_h^n$ at time t^n and a guess for the unknowns at an iteration $i-1$ at time t^{n+1}, find $\boldsymbol{u}_h^{n+\theta,i}\in\boldsymbol{V}_{h,u}$, $p_h^{n+1,i}\in Q_h$, $\boldsymbol{\xi}_u^i\in\boldsymbol{V}_h^*$, $\boldsymbol{\xi}_p^i\in\boldsymbol{V}_h^*$ and $\xi_d^i\in Q_h^*$, by solving the discrete variational problem:

$$\begin{aligned}
&\int_\Omega \rho\,\frac{\boldsymbol{u}_h^{n+\theta,i}-\boldsymbol{u}_h^n}{\theta\delta t}\cdot\boldsymbol{v}_h\,\mathrm{d}\Omega+\int_\Omega \rho(\boldsymbol{u}_h^{n+\theta,i-1}\cdot\nabla)\boldsymbol{u}_h^{n+\theta,i}\cdot\boldsymbol{v}_h\,\mathrm{d}\Omega\\
&\quad+\int_\Omega \mu\boldsymbol{\varepsilon}(\boldsymbol{u}_h^{n+\theta,i}):\boldsymbol{\varepsilon}(\boldsymbol{v}_h)\,\mathrm{d}\Omega-\int_\Omega \nabla\cdot\boldsymbol{v}_h\,p_h^{n+1,i-1}\,\mathrm{d}\Omega-\int_\Omega \boldsymbol{v}_h\cdot\boldsymbol{f}\,\mathrm{d}\Omega\\
&\quad+\int_\Omega \tau_1\rho(\boldsymbol{u}_h^{n+\theta,i-1}\cdot\nabla)\boldsymbol{v}_h\cdot[(\boldsymbol{u}_h^{n+\theta,i-1}\cdot\nabla)\boldsymbol{u}_h^{n+\theta,i}-\boldsymbol{\xi}_u^{i-1}]\,\mathrm{d}\Omega\\
&\quad+\int_\Omega \tau_2(\nabla\cdot\boldsymbol{v}_h)(\nabla\cdot\boldsymbol{u}_h^{n+\theta,i}-\xi_d^{i-1})\,\mathrm{d}\Omega=\boldsymbol{0},\quad \forall\boldsymbol{v}_h\in\boldsymbol{V}_h
\end{aligned}\tag{3}$$

$$\int_\Omega \rho q_h \nabla \cdot \boldsymbol{u}_h^{n+1,i} \,\mathrm{d}\Omega + \int_\Omega \delta t \nabla q_h \cdot (\nabla p^{n+1,i} - \nabla p^{n+1,i-1})\,\mathrm{d}\Omega$$

$$+ \int_\Omega \tau_1 \nabla q_h \cdot (\nabla p^{n+1,i} - \boldsymbol{\xi}_p^{i-1})\,\mathrm{d}\Omega = 0, \quad \forall q_h \in Q_h \tag{4}$$

$$\int_\Omega \boldsymbol{v}_h^* \cdot \boldsymbol{\xi}_u^i \,\mathrm{d}\Omega = \int_\Omega \boldsymbol{v}_h^* \cdot (\boldsymbol{u}^{n+\theta,i} \cdot \nabla)\boldsymbol{u}_h^{n+\theta,i}\,\mathrm{d}\Omega, \quad \forall \boldsymbol{v}_h^* \in \boldsymbol{V}_h^* \tag{5}$$

$$\int_\Omega \boldsymbol{v}_h^* \cdot \boldsymbol{\xi}_p^i \,\mathrm{d}\Omega = \int_\Omega \boldsymbol{v}_h^* \cdot \nabla p^{n+1,i}\,\mathrm{d}\Omega, \quad \forall \boldsymbol{v}_h^* \in \boldsymbol{V}_h^* \tag{6}$$

$$\int_\Omega q_h^* \xi_d^i \,\mathrm{d}\Omega = \int_\Omega q_h^* \nabla \cdot \boldsymbol{u}_h^{n+\theta,i}\,\mathrm{d}\Omega, \quad \forall q_h^* \in Q_h^* \tag{7}$$

for $i = 1, 2, \ldots$ until convergence, that is to say, until $\boldsymbol{u}_h^{n+\theta,i-1} \approx \boldsymbol{u}_h^{n+\theta,i}$ and $p_h^{n+1,i} \approx p_h^{n+1,i-1}$ in the norm defined by the user. Subscript h has been introduced to refer to the discrete finite element problem, while superscript n refers to time t^n, and superscript i to the iteration within the specific time step. In these equations, $\boldsymbol{u}_h$ and p_h are approximations to the velocity and pressure fields, $\boldsymbol{\xi}_u$, $\boldsymbol{\xi}_\mathrm{p}$ and ξ_d the L^2 projections of the convective, pressure and divergence terms onto the finite element subspace, and $\boldsymbol{v}_h$, q_h, $\boldsymbol{v}_h^*$ and q_h^* the test functions.

The first five terms in (3), and the first term in (4) are the standard Galerkin terms. The rest of terms are added to obtain a stable and consistent scheme [1, 2]. Basically, the stabilization of the convection term is provided by the sixth term of (3), which introduces streamline diffusion in a consistent manner, while the stabilizing effect on the velocity–pressure interpolation comes from the third term of (4).

A remark is needed for the fourth term in (3) and the second term in (4). If in both cases the pressure $p_h^{n+1,i-1}$ is replaced by p_h^n, we would have a second-order fractional step scheme, leaving the pressure gradient at a given time level in the first step and computing its increment in the second one, with the discrete divergence of the pressure gradient approximated by the standard Laplacian (see References [14–16] for different ideas related to second-order fractional step schemes for the Navier–Stokes equations and Reference [17] for the stability analysis of this pressure treatment). Since we have used $p_h^{n+1,i-1}$ instead of p_h^n, our scheme can be viewed as a *predictor–corrector* method rather than a pressure splitting algorithm.

The parameters τ_1 and τ_2 are chosen in order to obtain a stable numerical scheme with optimal convergence rates (see References [1, 2, 18, 19] for details). They are computed within each element domain Ω^e, with $e = 1, \ldots, n_\mathrm{el}$, where n_el is the number of elements of the finite element partition. We take them as [19]

$$\tau_1 = \frac{\rho (h^e)^2}{4\mu + 2\rho h^e |\boldsymbol{u}^e|} \quad \text{and} \quad \tau_2 = \mu + \frac{1}{2}\rho h^e |\boldsymbol{u}^e| \tag{8}$$

where h^e and $|\boldsymbol{u}^e|$ 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 $\boldsymbol{u}^{n+1} = [\boldsymbol{u}^{n+\theta} - (1-\theta)\boldsymbol{u}^n]/\theta$.

The accuracy of this formulation in the treatment of convection and pressure stabilization is discussed in Reference [1]. Likewise, the implementation and accuracy of the pressure stabilization is thoroughly discussed in Reference [20].

3. IMPLEMENTATION OF THE LEVEL SET METHOD

The basic idea of the level set method is to define a scalar function, say $\psi(\boldsymbol{x},t)$, over the computational domain Ω in such a manner that its value at a certain point $\boldsymbol{x}\in\Omega$ indicates the presence or absence of fluid. This function may be considered as a fictitious fluid property which we will call pseudo-concentration. For instance, we may assign the value 1 to regions where fluid has already entered and the value of 0 to air-filled regions. The position of the fluid front will be defined by the iso-value contour $\psi(\boldsymbol{x},t)=\psi_{\mathrm{c}}$, where $\psi_{\mathrm{c}}\in[0,1]$ is a critical value defined *a priori*. The value $\psi_{\mathrm{c}}=0.5$ is usually taken. This value is immaterial if ψ is a true step function, but is needed in the finite element discretisation and for the smoothing to be described later.

The conservation of the pseudo-concentration (assumed to be sufficiently smooth) in any control volume $V_t\subset\Omega$ which is moving with the divergence free velocity field $\boldsymbol{u}$ leads to

$$\frac{\partial\psi}{\partial t}+(\boldsymbol{u}\cdot\nabla)\psi=0 \quad \text{in } \Omega\times(t_0,t_f) \tag{9}$$

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

$$\Gamma_{\mathrm{inf}}:=\{\boldsymbol{x}\in\partial\Omega \mid \boldsymbol{u}\cdot\boldsymbol{n}<0\} \tag{10}$$

The definition of the position of the fluid front will be given by the physical properties. Let π be any of these, i.e., density (ρ), or viscosity (μ). We will have that

$$\pi=\begin{cases}\pi_{\mathrm{fluid}}(\boldsymbol{x},t) & \text{if } \boldsymbol{x}\in\Omega_t\\ \pi_{\mathrm{air}} & \text{if } \boldsymbol{x}\in\Omega\backslash\Omega_t\end{cases} \tag{11}$$

where $\Omega_t:=\{\boldsymbol{x}\in\Omega \mid \psi(\boldsymbol{x},t)\geqslant\psi_{\mathrm{c}}\}$, and the pseudo-concentration function ψ is the solution of the following problem:

$$\begin{aligned}&\frac{\partial\psi}{\partial t}+(\boldsymbol{u}\cdot\nabla)\psi=0 && \text{in } \Omega\times(t_0,t_f)\\ &\psi=\bar{\psi} && \text{on } \Gamma_{\mathrm{inf}}\times(t_0,t_f)\\ &\psi(\boldsymbol{x},0)=\psi_0(\boldsymbol{x}) && \text{in } \Omega\end{aligned}$$

The initial condition ψ_0 is chosen in order to define the initial position of the fluid front to be analyzed. The boundary condition $\bar{\psi}$ 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 discretisation. 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 the pseudo-concentration for each node of the finite element mesh according to the following expression:

$$\psi = \psi_c + \mathrm{sgn}(\psi_0 - \psi_c) f(d) \tag{12}$$

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.

The crucial point is how to calculate the distance d from a point under consideration to the front. A totally new smoothing algorithm has been developed to compute $f(d)$ in (12). It is only based on the mesh topology, and on an approach to fair surfaces given in Reference [21]. To explain this technique, let us introduce first the concept of nodal levels.

Given a pseudo-concentration distribution, and therefore a front position, at each grid point i of the mesh one may assign a level l_i based on the following rules:

(i) A nodal point i belongs to the level $l_i = 0$, if it is filled with real fluid $(\psi_i > \psi_c)$, but it is connected to at least one node j filled with air $(\psi_j < \psi_c)$.
(ii) A nodal point i belongs to a level $l_i > 0$, if it is filled with air $(\psi_i < \psi_c)$, and it is connected to at least one node of level $l_i - 1$.
(iii) A nodal point i belongs to a level $l_i < 0$, if it is filled with real fluid $(\psi_i > \psi_c)$, and it is connected to at least one node of level $l_i + 1$.

The algorithm to assign the levels is very fast, and it only needs the pseudo-concentration field and the arrays defining the mesh graph (the list of points surrounding each point). Observe that the levels defined through this algorithm are integers.

Once the levels have been assigned to the mesh points, the distances d_i from the points i such that $0 \leqslant l_i \leqslant 1$ to the front are computed using the first scheme described above. This computation is very fast if for each plane p only the points surrounding the element in which p is contained are checked. It is clear that the points belonging to levels 0 and 1 are the points that belong to the elements cut by the front. Then, $f(d)$ for a given node i is defined by the following expression:

$$f_i = \begin{cases} d_i & \text{if } 0 \leqslant l_i \leqslant 1 \\ d_m + (l_i - 1)\bar{d} & \text{if } l_i > 1 \\ d_m - l_i \bar{d} & \text{if } l_i < 0 \end{cases} \tag{13}$$

where

$$\bar{d} = \frac{1}{N} \sum_i d_i \quad \text{and} \quad d_m = \max_i \{d_i\} \quad \forall i \text{ such that } 0 \leqslant l_i \leqslant 1 \tag{14}$$

and N is the number of nodes belonging to levels 0 and 1. Basically, the above equations define a scalar field that is proportional to l_i, and, therefore, that follows the mesh topology. However, such field has to be smoothed in some way to maintain its contour positions. Hence, an algorithm presented in Reference [21], and used to smooth surfaces without modifying the volume they enclose, was implemented. The method consists in redefining the smoothed

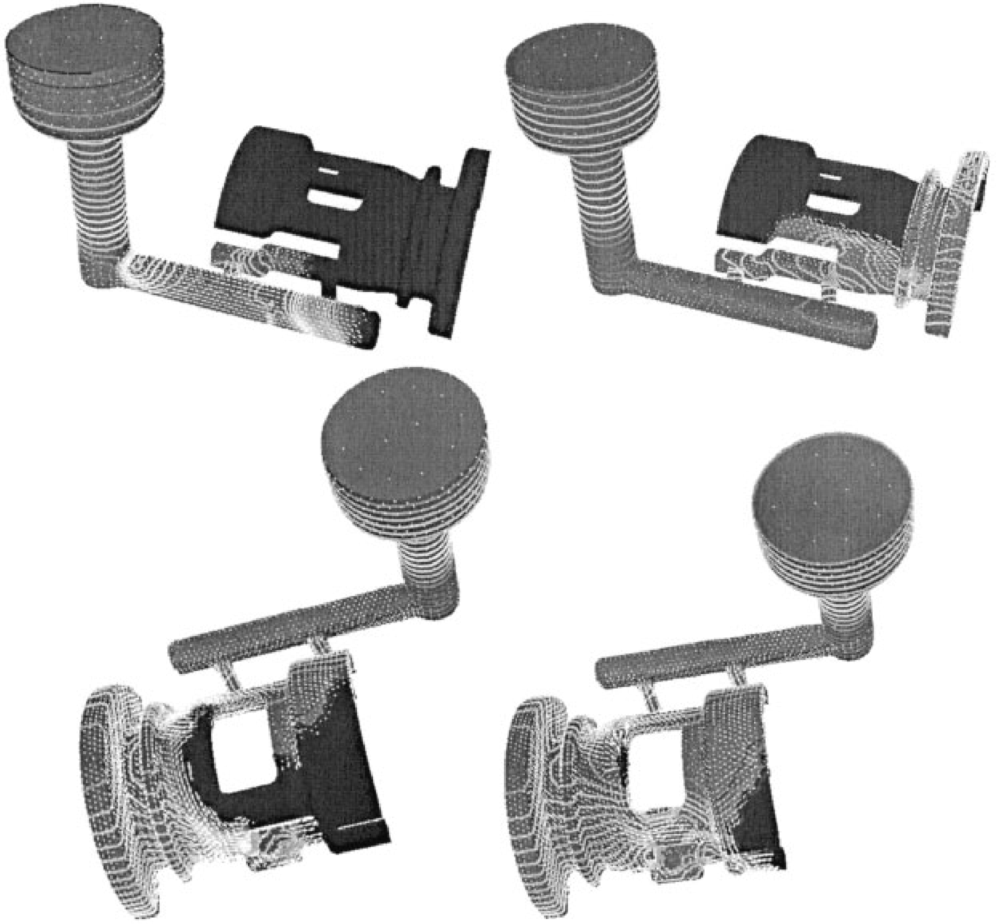

Figure 1. Position of the front of molten metal and pressure contours at $t = 18$ (top left), $t = 26$ (top right), $t = 37$ (bottom left) and $t = 40$ (bottom right).

function $f(d)$ for each nodal point that belongs to levels $l_i < 0$ and $l_i > 1$ as

$$f_i^{n-1/2} := f_i^{n-1} + \lambda \delta f_i^{n-1} \tag{15}$$

$$f_i^{n} := f_i^{n-1/2} + \mu \delta f_i^{n-1/2} \quad \text{for } n = 1, \ldots, n_{\text{pass}} \tag{16}$$

where n_{pass} is the number of smoothing passes, λ and μ are constants with optimal values of 0.6313983 and −0.6739515, respectively (see Reference [21]), and

$$\delta f_i = \sum_{j \in \mathcal{N}_i} w_{ij}(f_j - f_i), \quad w_{ij} = \left(d_{ij} \sum_{j \in \mathcal{N}_i} \frac{1}{d_{ij}} \right)^{-1} \tag{17}$$

where $\mathcal{N}_i$ is the neighbourhood of node i, that is, the set of nodes j connected to the node i, and d_{ij} is the distance between nodes i and j.

Finally, the smoothed pseudo-concentration is defined by (12), taking $f(d)$ for the nodal point i equal to $f_i^{n_{\text{pass}}}$. The number of passes to smooth f was designed in a heuristic manner as $n_{\text{pass}} = \sqrt{n_{\text{p}}/2}$, where n_{p} is the total number of grid points. As a final remark, note that f_i for the points of the levels $l_i = 0$ and 1 is not smoothed to avoid variations in the free surface position; $f_i = d_i$ for such points. In fact, the smoothing could be applied only to the points neighbouring the free surface, and not to all the levels.

4. NUMERICAL EXAMPLE

In this section, we present the numerical simulation of the filling of a mould. In this case, the two fluids involved are molten metal and air. The finite element mesh employed in the analysis consists of approximately 40 000 linear tetrahedra. The molten metal occupies a reservoir at the top of which the pressure is constant, and falls down to the mould through a pipe due to gravity. The material properties for the molten metal used in this example are (SI units) $\rho = 6700$ and $\mu = 10^{-3}$.

The results of the simulation are shown in Figure 1. The magenta colour corresponds to the molten material, on top of which the isopressure contours have been plotted. As it can be observed there, the fluid front and the pressure contours are perfectly smooth.

REFERENCES

1. Codina R. Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:1579–1599.
2. Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering* 2002; at press.
3. Yang 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. Iafrati A, Di Mascio A, Campana EF. A level set technique applied to unsteady free surface flows. *International Journal for Numerical Methods in Fluids* 2001; **35**:281–297.
11. Pichelin E, Coupez T. A Taylor discontinuous Galerkin method for the thermal solution in 3d mold filling. *Computer Methods in Applied Mechanics and Engineering* 1999; **178**:153–169.
12. Ilinca F, Hétu JF. Finite element solution of three-dimensional turbulent flows applied to mould-filling problems. *International Journal for Numerical Methods in Fluids* 2000; **34**:729–750.
13. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: Berlin, 1991.
14. Bell JB, Colella P, Glaz HM. A second-order projection method for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1989; **85**:257–283.
15. Shen J. On error estimates for some higher order projection and penalty-projection methods for Navier–Stokes equations. *Numerische Mathematik* 1992; **62**:49–73.

16. van Kan J. A second-order accurate pressure correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:870–891.
17. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2001; **170**:112–140.
18. Franca LP, Frey SL. Stabilized finite element methods: II. The incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:209–233.
19. Codina R. A stabilized finite element method for generalized stationary incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2681–2706.
20. Codina R, Blasco J, Buscaglia GC, Huerta A. Implementation of a stabilized finite element formulation for the incompressible Navier–Stokes equations based on a pressure gradient projection. *International Journal for Numerical Methods in Fluids* 2001; **37**:419–444.
21. Taubin G. A signal processing approach to fair surface design. *IBM Research Report*, 1995.