INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **49**:1305–1327
Published online 8 August 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1024

A finite element strategy for the solution of interface tracking problems

C. Devals[1], M. Heniche[1], F. Bertrand[1,*,†], R. E. Hayes[2] and P. A. Tanguy[1]

[1]*Department of Chemical Engineering, Ecole Polytechnique, Montreal, QC, Canada H3C 3A7*
[2]*Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada T6G 2G6*

SUMMARY

A finite element-based numerical strategy for interface tracking is developed for the simulation of two-phase flows. The method is based on the solution of an advection equation for the so-called 'pseudo-concentration' of one of the phases. To obtain an accurate description of the interface, a streamline upwind Petrov–Galerkin (SUPG) scheme is combined with an automatic mesh refinement procedure and a filtering technique, making it possible to generate an oscillation-free pseudo-concentration field. The performance of the proposed approach is successfully tested on four classical two-dimensional benchmark problems: the advection skew to the mesh, the transport of a square shape in a constant velocity flow field, the transport of a cut-out cylinder in a rotating flow field and the transport of a disc in a shear flow. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: interface tracking; finite element method; SUPG; adaptive meshing; filtering technique; unstructured grid

1. INTRODUCTION

Interface flows are encountered in chemical engineering applications that involve multiphase flows. For example, they occur in paper coating flows in the nip of a high speed roll coater in the presence of air bubbles [1], and in liquid–liquid dispersions and emulsions in rotor-stator devices [2]. The simulation of these flow problems remains extremely challenging, although significant progress has recently been made for the tracking of inter-phase boundaries. From a numerical perspective, the main challenge is the accurate determination of the location of the interface, which is *a priori* unknown. Many methods have been proposed for the solution of

*Correspondence to: F. Bertrand, Department of Chemical Engineering, Ecole Polytechnique, Montreal, QC, Canada H3C 3A7.
†E-mail: francois.bertrand@polymtl.ca

Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada (NSERC)

Received 18 March 2005
Revised 10 May 2005
Accepted 12 May 2005

two-phase flow problems, which can be broadly classified into three categories; Lagrangian, Eulerian and the arbitrary Lagrangian–Eulerian (ALE) methods.

The three most popular examples of the Eulerian approach are the volume of fluid (VOF) method [3], the level set method [4] and the pseudo-concentration method [5]. Many example applications of these methods can be found in the literature, as recently reviewed in Reference [6]. In the VOF method, a colour function F indicates the fractional volume of the fluid in a cell. $F=1$ indicates that the cell is full of fluid '1', $F=0$ corresponds to a cell full of fluid '2', and $0<F<1$ is the interface between fluid '1' and '2'. The colour function is advected in some manner [7]. It can be moved algebraically with the flux-corrected transport scheme (FCT), or geometrically using the simple line interface calculation procedure (SLIC), the SOLA-VOF method, or the piecewise linear interface calculation procedure (PLIC). The main advantage of the VOF method is that mass conservation is ensured, although the interface reconstruction is time-consuming and its extension to 3D is complex, especially in the context of unstructured grids.

In the level set approach, the interface is defined by a zero level set of a smooth function F representing the distance from the interface: $F=0$ at the interface, $F>0$ outside the initial surface and $F<0$ inside. The function F is transported using the advection equation

$$\frac{\partial F}{\partial t} + \mathbf{u}.\mathbf{grad}\, F = 0 \tag{1}$$

where $\mathbf{u}$ is the fluid velocity obtained from the solution of the momentum and continuity equations. This function is used to evaluate the physical properties φ of the fluid, such as the viscosity or the density, based on a weighted average:

$$\varphi = \varphi_1 F + (1-F)\varphi_2 \tag{2}$$

where φ_1 and φ_2 denote the properties of the two phases 1 and 2, respectively. A drawback of the level set method is that it does not ensure rigorously mass conservation.

The pseudo-concentration is based on the value of F ($F=0.5$ corresponds to the position of the interface) to determine the distribution in the domain of the two fluids '1' and '2'. The value of F ranges between 0 and 1, and F is also transported by Equation (1). This method was developed because of its relative ease of implementation in 2D and in 3D, even if it is necessary to remesh the computational domain to obtain the necessary solution accuracy [5].

The objective of this paper is to present a finite element strategy for the solution of interface tracking problems based on the pseudo-concentration method. With the finite element method (FEM), it is well-known that the use of the Galerkin formulation for the solution of Equation (1) may give non-physical oscillations. These spurious phenomena can be controlled, but rarely fully eliminated, by adding some numerical diffusion to the original problem using, for example, the streamline upwind Petrov–Galerkin (SUPG) method [8], the Galerkin-least-square (GLS) method [9] or the discontinuous Galerkin method [10, 11]. A review of these methods can be found in Reference [12]. Numerical diffusion tends to smooth out steep gradients, and mesh refinement techniques are needed to minimize this effect. Several authors have proposed the use of adaptive dynamic grids based on hierarchical methods, such as quadtrees for rectangular grids [13–15] and tri-trees for triangular grids [16] in the two-dimensional case. Unfortunately, although the combination of Petrov–Galerkin methods and mesh refinement yield better results, small oscillations remain in the solution in the sharp gradient region.

In the present work, a new filtering technique is developed to help remove these oscillations. This filtering technique is based on a change of variable that forces the bounds of the colour function to be exactly 0 and 1. The solution of the colour function advection is obtained with the SUPG method due to its proven efficiency in a wide variety of situations and its ease of implementation. An automatic mesh refinement procedure developed earlier for the simulation of fluid flow in twin-screw extruders [17], and for interface tracking [18] is also used to capture the shock-like behaviour, so as to reduce diffusion and keep the interface sharp and smooth as in References [5, 19, 20]. In the forthcoming, the automatic mesh refinement technique will be first presented and tested in the case of standard academic benchmark problems [21]. It will be shown that the oscillations at the sharp front cannot be fully eliminated. The filtering technique will then be fully explained and employed to show how it can alleviate the oscillation problem. Next, the overall strategy (mesh refinement and filtering) will be applied in the case of the finite element solution of standard two-dimensional benchmark problems. Finally, the extension of the proposal strategy to two-phase flows in chemical engineering processes will be discussed.

2. MESH REFINEMENT TECHNIQUE

2.1. Methodology

The mesh refinement technique relies on one reference (generally coarse) mesh that is repeatedly refined locally throughout a simulation, when needed, using the values of the colour function F as a remeshing indicator. The procedure comprises two refinement steps:

(1) a tessellation of the targeted finite elements into four elements in such a way that the regularity of the child elements is preserved, see Figure 1(a).
(2) a further refinement step in some neighbouring elements owing to the presence of hanging nodes (the dot in Figure 1(a), for instance) to satisfy continuity.

In the latter step, the neighbouring element identified is split into either two, three or four child elements as illustrated in Figure 1(b), (c) and (d), respectively. This procedure, also referred to as the red–green refinement procedure, is discussed by Verfurth [22], Secretan [23] and Johnson [24]. A similar approach was also used in sheet metal forming processes [25],

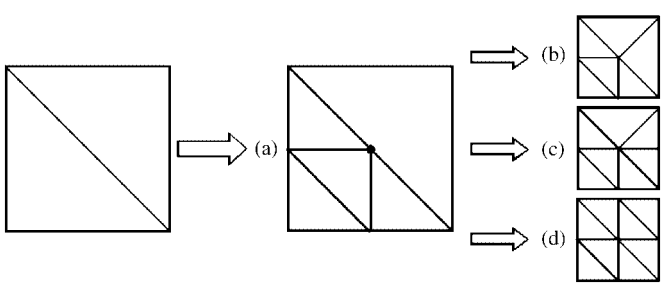

Figure 1. Decomposition into (a) four elements; subsequent decomposition of the neighbouring element into two (b), three (c) and four (d) child elements, to satisfy continuity.

where large plastic deformations generate large element distortions and require remeshing to capture the geometrical details of the resulting surface.

The simulation strategy is based on an interface tracking method that is now described for the steady-state case. First, for a given velocity field, the colour function F is computed by solving Equation (1) using the finite element method on the reference mesh. Once the values of F have been obtained, a new mesh can be generated. Figure 2 illustrates the four-step approach in the case of the advection skew to the mesh problem (see Section 2.2.1). First, the elements to be refined on the reference mesh of computational domain Ω (Figure 2(a)) are determined. They correspond here to those elements for which F is between preset values $F = 0.5 \pm \varepsilon$, where ε is chosen by the user (Figure 2(b)). The targeted elements are next marked with control points arbitrarily located at their centre (Figure 2(c)). The mesh is then adapted around the control points (Figure 2(d)) and a new simulation is performed. These control points are only used to estimate the position of the interface and guide the remeshing step. They do not represent the accurate position of the front. The procedure can be repeated until

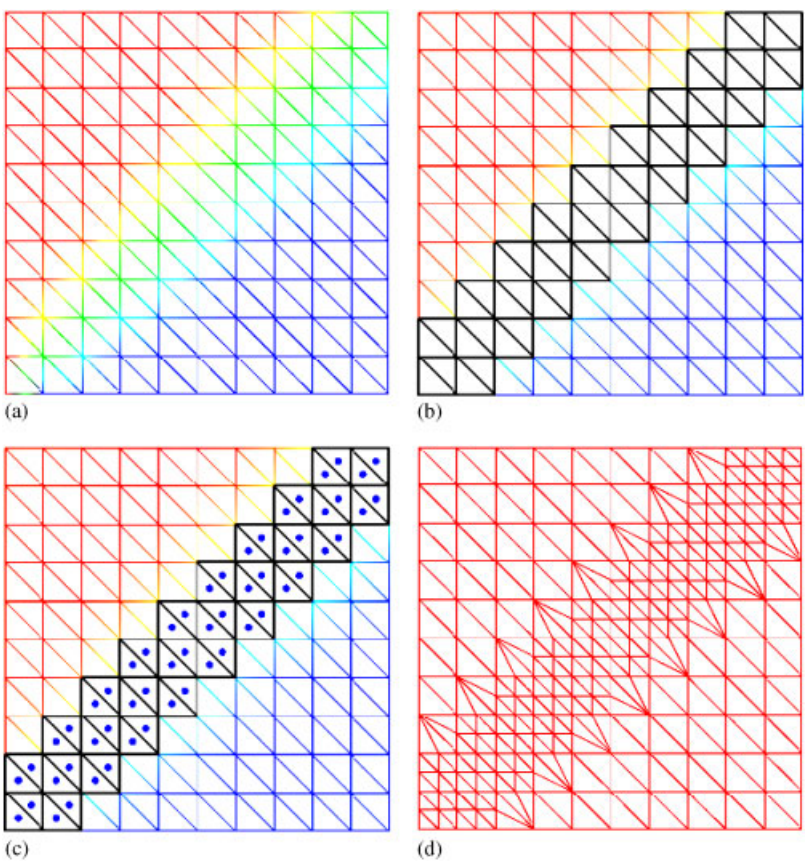

Figure 2. Mesh refinement technique: (a) reference mesh; (b) selection of elements to be refined; (c) creation of a control point for each element to be refined; and (d) generation of new mesh.

an accurate solution is obtained. The algorithm is as follows:

1. Start the problem with a reference mesh
2. For $i = 1$ to n passes, do
   2.1. Flow calculation
   2.2. Determination of the interface: $F = 0.5 \pm \varepsilon$
   2.3. Creation of control points
   2.4. Adaptive mesh refinement based on reference mesh and control points

At present, in our implementation, the number of mesh refinements or iterations is set by the user, but a criterion based on the variation of F between two successive iterations could be used for more flexibility, as for example:

$$\|F^{n+1} - F^n\|_{\infty,\Omega} = \max_i \|F_i^{n+1} - F_i^n\| < \text{tol} \tag{3}$$

where i stands for the node number and *tol* for some tolerance value. In this study, the maximum number of iterations was arbitrarily fixed to 7, unless stated otherwise.

When dealing with unsteady–state problems, initial conditions are needed to start the simulation. One time step is performed with the finite element solver and the control points are set according to the position of the fluid interface. At this stage, the remeshing technique described above is applied. Control points are added in any element for which $F = 0.5 \pm \varepsilon$ in order to estimate the position of the interface, and the remeshing procedure is performed for the elements lying within a preset distance from these control points. The reference mesh needs to be adapted at each time step. Once the mesh has been locally adapted by means of one or more iterations, the solution is projected onto this new mesh and the next time iteration can be computed. The algorithm for the unsteady–state case is as follows:

1. Start the problem with initial conditions: $t = 0$; $F(0) = F_0$
2. Do
   2.1. Flow calculation
   2.2. Determination of the interface: $F = 0.5 \pm \varepsilon$
   2.3. Creation of control points
   2.4. For $j = 1$ to n passes, do
      2.4.1. Adaptive mesh refinement based on reference mesh and control points
   2.5. Projection of the solution onto the new mesh
   2.6. $t = t + \Delta t$

2.2. Benchmark problems

The mesh refinement strategy discussed above was tested on two benchmark problems widely used in the literature: advection skew to the mesh [8] and the transport of a square in a steady-state flow field [13]. In the first problem, the domain is a unit square and the reference mesh comprises $2 \times 10 \times 10$ regular triangular elements. The velocity field $\mathbf{u}$ is unidirectional and constant. Its magnitude is set to 1 and its direction θ is 15, 30, 45, 60° with respect to the x-axis for the steady-state problem and 45° for the transient case. In the second problem, a square shape, on which F is set to 1, is transported by the velocity field $\mathbf{u} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$. For this case, the reference mesh is minimal, with only $2 \times 1 \times 1$ elements.

For all problems, the solution of the advection equation (1) was obtained using the SUPG method with P_1 linear finite elements. A very small diffusive term was added to the right-hand side of (1), yielding the following hyperbolic problem:

$$\frac{\partial F}{\partial t} + \mathbf{u}.\mathbf{grad}\, F = k\nabla^2 F \tag{4}$$

The value of k was set to 10^{-9}, which corresponds to a high Peclet number problem.

To evaluate the accuracy of the computed finite element solution F_h versus the exact solution F_e, the following error terms, used in the literature [7], are introduced:

$$\Delta E_i = |F_{h_i} - F_{e_i}| \tag{5}$$

$$E_0 = \frac{\sum_{i=1}^{n} (\Delta E_i)^2}{n} \tag{6}$$

$$E_\infty = \max_{i=1,\ldots,n} (\Delta E_i) \tag{7}$$

$$E_{\text{tot}} = \frac{\sum_{i=1}^{n} \Delta E_i}{\sum_{i=1}^{n} F_{e_i}^0} \tag{8}$$

where i stands for the node number ($i = 1, \ldots, n$) and $F_{e_i}^0$ the exact solution on node i at $t = 0$. All the computations were performed on an IBM P630 workstation.

2.2.1. Steady-state case. For the steady-state advection skew to the mesh problem, F is set to 1 along the y-axis and 0 along the x-axis. The minimum and maximum values of F obtained on the reference mesh as well as on the refined meshes are given in Figures 3(a) and (b), respectively.

The results show that both these values are within 3% from the expected values of 0 and 1, and that the discrepancies decrease below 0.7% as successive mesh refinements are applied. Figures 4, 5 and 6 show the elevated surface of F obtained with the reference mesh and the

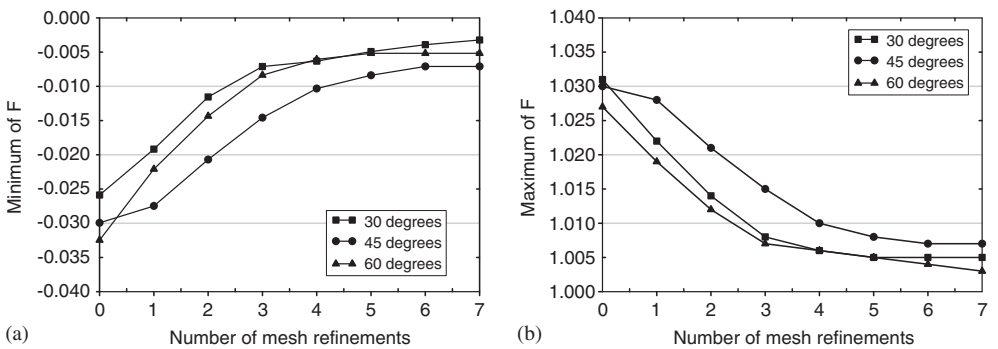

Figure 3. Minimum (a) and maximum (b) values of the colour function F for the steady-state advection skew to the mesh problem.

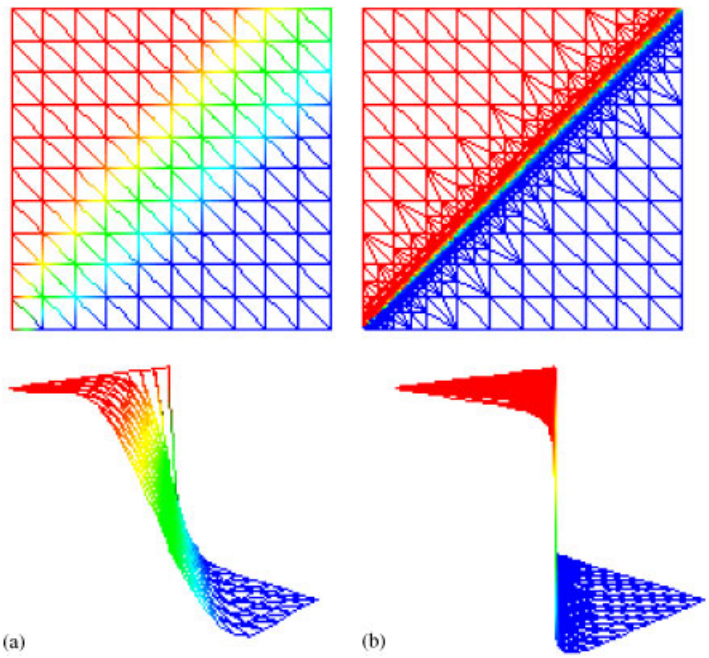

Figure 4. Colour function F for the $\theta = 45^\circ$ steady-state advection skew to the mesh problem: (a) with the reference mesh; and (b) with the refined mesh after 7 iterations.

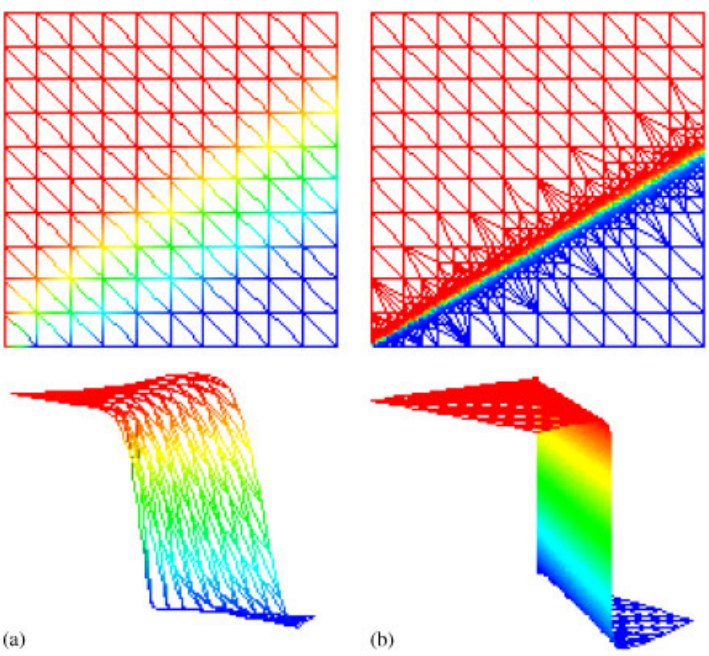

Figure 5. Colour function F for the $\theta = 30^\circ$ steady-state advection skew to the mesh problem: (a) with the reference mesh; and (b) with the refined mesh after 7 iterations.

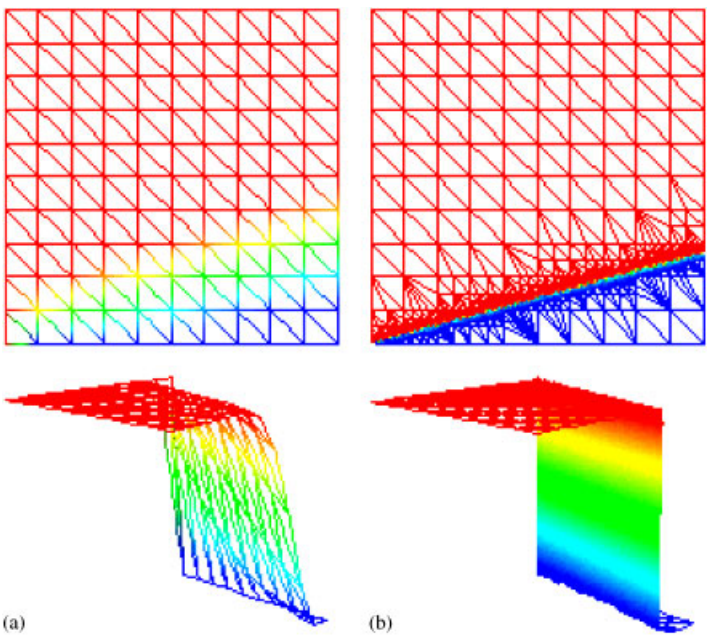

Figure 6. Colour function F for the $\theta = 15^\circ$ steady-state advection skew to the mesh problem: (a) with the reference mesh; and (b) with the refined mesh after 7 iterations.

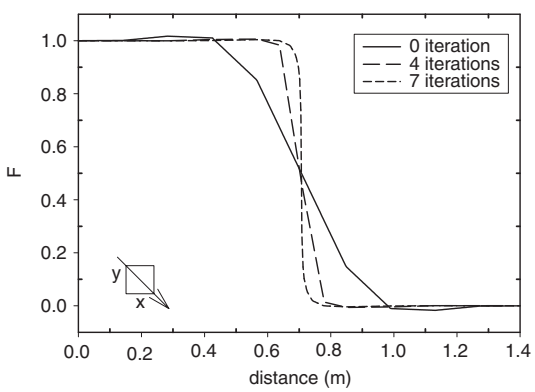

Figure 7. Steady-state advection skew to the mesh problem with $\theta = 45^\circ$. Colour function F along a line normal to the flow, as indicated by the arrow.

refined mesh after 7 refinement iterations, for the 45°, 30° and 15° cases, respectively. On the reference mesh, diffusion and oscillations can be readily noticed. For these three cases, when the mesh refinement technique is applied, diffusion becomes limited to a very small area, as expected, and oscillations nearly vanish. This is illustrated in Figure 7 for the 45° case. Figures 8(a), (b) and (c) show the evolution of the surface area with the number of

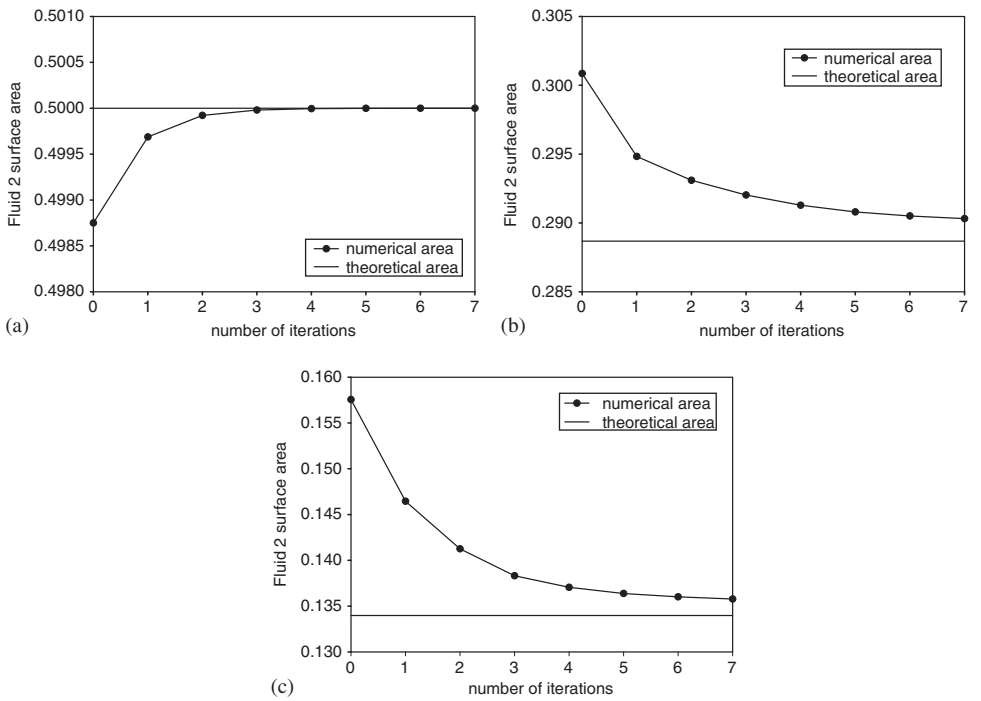

Figure 8. Evolution of the colour function surface area for the: (a) $\theta = 45°$; (b) $\theta = 30°$; and (c) $\theta = 15°$, steady-state advection skew to the mesh problem.

refinement iterations for $\theta = 45°$, 30° and 15°, respectively. This surface is the blue area in Figures 4–6 (between the $F = 0.5$ contour and the x-axis). As the mesh is refined, this area tends towards the analytical solution $A_{\text{theo}} = 1/2L^2 \tan\theta$, where L is the length of the domain in the x-direction ($L = 1$ m in that case) and θ the angle as already defined. For the 45° case, the nodes of the elements at the interface are aligned with respect to the velocity field, and literally split the domain into two parts because of the use of a structured mesh. The situation is different for the other two cases. This explains why the numerical solution is better in Figure 8(a) for the 45° case than in Figures 8(b) and (c) for the 30° and 15° cases, respectively.

2.2.2. Transient case. A second-order implicit Crank–Nicholson time scheme with a time step of 5×10^{-4} s was used for the transport of a square in a steady-state flow field. An enlarged view of the position of the square at $t = 0$ and 0.75 s is shown as F contours in Figure 9. The shape of the square remains constant except at the corners, which are slightly rounded because of the small numerical diffusion.

The mesh evolution from $t = 0$ to 0.75 s and the F contours are presented in Figure 10. As many as 10 refinement iterations were used in each case. The results obtained show the accuracy that can be expected with the proposed mesh refinement procedure. It also provides evidence for its robustness since the reference mesh was comprised of only two finite elements.

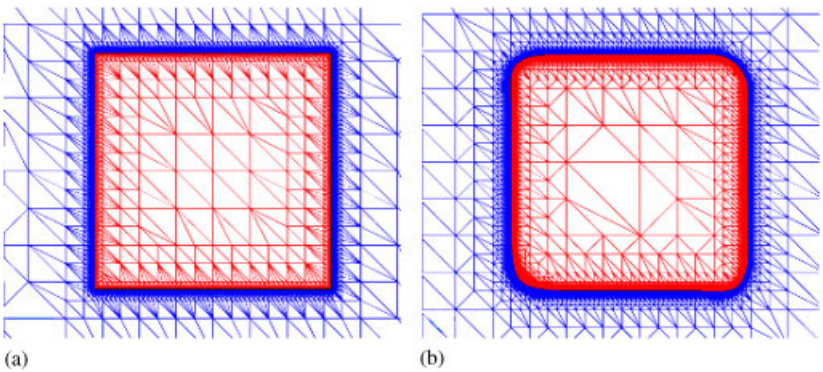

Figure 9. Transport of a square in a steady-state flow field: F contours at: (a) $t = 0$ s; and (b) $t = 0.75$ s.

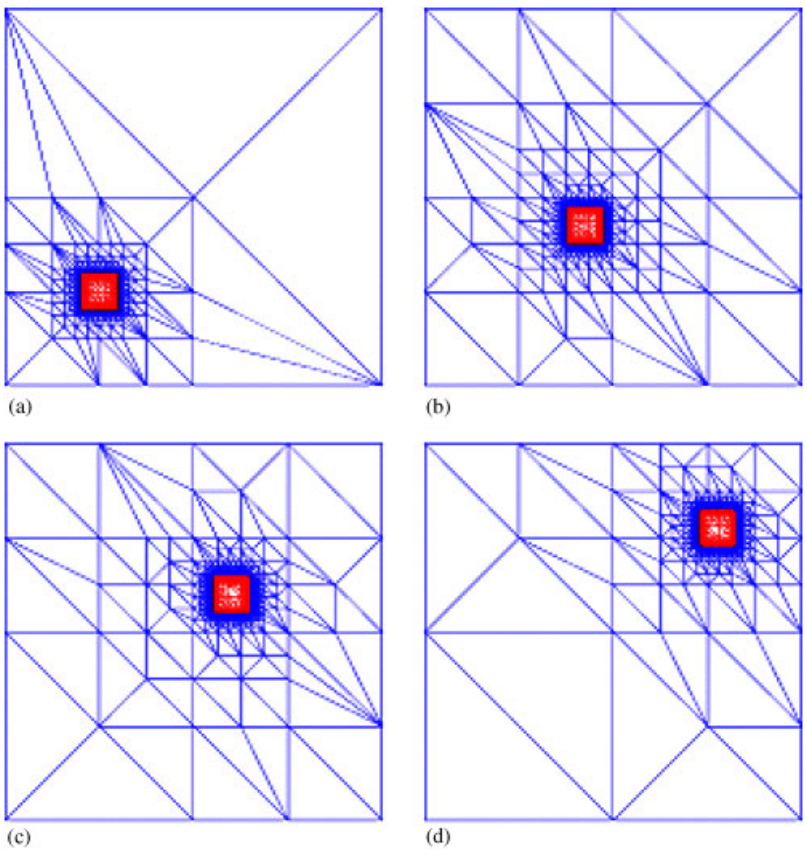

Figure 10. Transport of a square in a steady-state flow field: mesh evolution and F contours at: (a) $t = 0$ s; (b) $t = 0.15$ s; (c) $t = 0.45$ s; and (d) $t = 0.75$ s.

In summary, the mesh refinement procedure is capable of capturing a sharp front but some non-physical wiggles still remain in the steep front region. To make the overall interface tracking strategy more accurate, a procedure is required to filter out the residual oscillations that may remain in the solution. The construction of such a filter is next described.

3. FILTERING TECHNIQUE

3.1. Methodology

Let us first introduce some notations. $\tilde{F}$ represents the oscillatory solution of the colour function, the values of which are obtained after solving Equation (1). $\bar{F}$ is the oscillation-free solution of the colour function obtained after application of the filter.

We start from the mathematical function

$$\tanh(\tilde{F}) = \frac{\exp(\tilde{F}) - \exp(-\tilde{F})}{\exp(\tilde{F}) + \exp(-\tilde{F})} \tag{9}$$

which ranges from -1 to 1 for $\tilde{F} \in]-\infty, +\infty[$, as shown in Figure 11(a). Because we want $\bar{F}$ to range strictly between 0 and 1, a translation is applied:

$$\bar{F} = \frac{\tanh(\tilde{F}) + 1}{2} \tag{10}$$

or, using (9),

$$\bar{F} = \frac{1}{1 + \exp(-2\tilde{F})} \tag{11}$$

This function is shown in Figure 11(b).

In practice, as $\tilde{F}$ is the solution of Equation (1) obtained with the SUPG finite element method described above, its values lie in the vicinity of 0 and 1, more precisely between $[-\beta_1, 1 + \beta_2]$, where β_1 and β_2 are generally small positive numbers. Consequently, two additional transformations are applied to (11) in order to take into account explicitly the domain of $\tilde{F}$. A scaling factor α is first introduced such that $\tilde{F}$ is replaced by $\alpha\tilde{F}$, where $\alpha \gg 1$ and the values of $\tilde{F}$ are mapped from $]-\infty, +\infty[$ to $[-1, +1]$. More details on the value of α will be given in Section 3.2. Equation (11) then becomes:

$$\bar{F} = \frac{1}{1 + \exp(-2\alpha\tilde{F})} \tag{12}$$

A second transformation is required to map the values of $\tilde{F}$ from $[-1, +1]$ to $[0, +1]$, which gives

$$\bar{F} = \frac{1}{1 + \exp(-2\alpha(2\tilde{F} - 1))} = g(\tilde{F}) \tag{13}$$

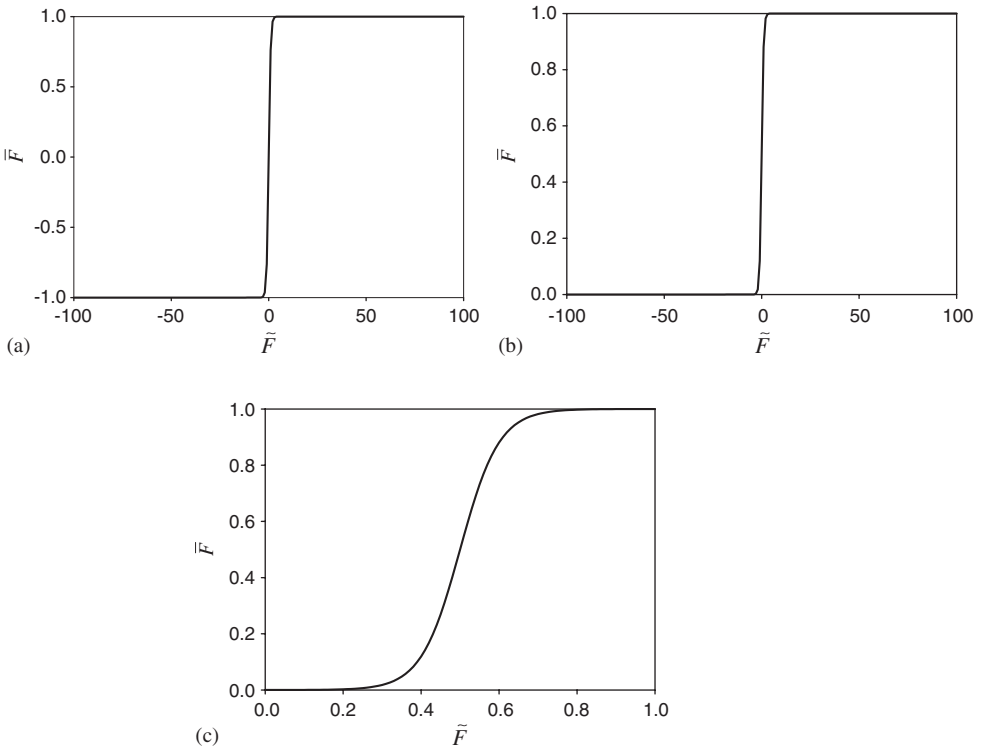

Figure 11. Different steps in the construction of the filter: $\bar{F}$ with respect to $\tilde{F}$: (a) from Equation (9); (b) from Equation (11); and (c) from Equation (13).

This function is the proposed filter, the graph of which is presented in Figure 11(c). Finally, we have to ensure that $\bar{F}$ satisfies Equation (1):

$$\frac{\partial \bar{F}}{\partial t} + \mathbf{u}.\mathbf{grad}\,\bar{F} = 0 \tag{14}$$

Upon using

$$\frac{\partial \bar{F}}{\partial t} = g'(\tilde{F})\frac{\partial \tilde{F}}{\partial t} \tag{15}$$

and

$$\frac{\partial \bar{F}}{\partial x_i} = g'(\tilde{F})\frac{\partial \tilde{F}}{\partial x_i} \tag{16}$$

where

$$g'(\tilde{F}) = \frac{\partial \bar{F}}{\partial \tilde{F}} \tag{17}$$

it follows that

$$\frac{\partial \bar{F}}{\partial t} + \mathbf{u}.\mathbf{grad}\,\bar{F} = g'(\tilde{F})\left(\frac{\partial \tilde{F}}{\partial t} + \mathbf{u}.\mathbf{grad}\,\tilde{F}\right) \tag{18}$$

and, since

$$\frac{\partial \tilde{F}}{\partial t} + \mathbf{u}.\mathbf{grad}\,\tilde{F} = 0 \tag{19}$$

we get

$$\frac{\partial \bar{F}}{\partial t} + \mathbf{u}.\mathbf{grad}\,\bar{F} = 0 \tag{20}$$

As already said, the SUPG method [8] was used in this work to solve Equation (19). Alternatively, Equation (19) could be solved by any efficient hyperbolic solver. One takes as initial and boundary conditions for $\tilde{F}$ those for $\bar{F}$, as they remain the same. $\bar{F}$ can be recovered *a posteriori* by applying transformation (13). For steady-state problems, the filtering technique is applied after the solution has been obtained for $\tilde{F}$ to generate the filtered values of $\bar{F}$. For transient problems, the values of $\bar{F}$ can be updated at any chosen time steps.

3.2. Validation

3.2.1. Steady-state advection skew to the mesh problem. The domain in this case is the unit square meshed with $2 \times 64 \times 64$ regular triangular elements. The velocity field $\mathbf{u}$ is unidirectional and constant. Its magnitude is set to 1 and its direction θ is 30° with respect to the x-axis. The value of F is set to 0 along the x-axis and 1 along the y-axis.

The choice $\alpha = 5$ in Equation (13) gives good results in practice, because it keeps the numerical diffusion small. Indeed, such a choice yields $\bar{F} = [\bar{F}_{\min}, \bar{F}_{\max}] =]\varepsilon; 1-\varepsilon[$ with $\varepsilon = 4.5 \times 10^{-5}$. A smaller value of ε could be obtained by choosing a larger value of α (Table I). Higher values of α reduce the amount of diffusion but the interface then tends to become stair-like, as shown in Figure 12. On the other hand a too small value of ε results in a non-minimal amount of numerical diffusion for a given mesh. A trade-off value is required for optimal results.

Figure 13 displays the elevated surface of the colour function obtained without (Figure 13(a)) and with (Figure 13(b)) the filtering procedure. The results clearly show that the filtering procedure can reduce the numerical oscillations very efficiently. A more quantitative assessment is given in Table II, which summarizes the errors obtained with and without the filtering procedure. As expected, the errors E_0 and E_{tot} decrease with filtering. Moreover, one can see that the maximum value of the error E_∞ is larger when filtering is used, a fact that can

Table I. Values of $\bar{F}_{\min}$ and $\bar{F}_{\max}$ for different values of α.

α	1	2	3	4	5	10	20
$\bar{F}_{\min}$	0.12	0.02	0.003	3.0×10^{-4}	4.5×10^{-5}	2.1×10^{-9}	4.2×10^{-18}
$\bar{F}_{\max}$	0.88	0.98	0.997	~ 1	~ 1	~ 1	~ 1

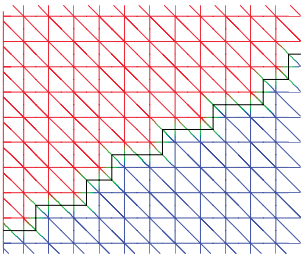

Figure 12. Steady-state advection skew to the mesh problem: $\bar{F} = 0.5$ contour line when $\alpha = 10$.

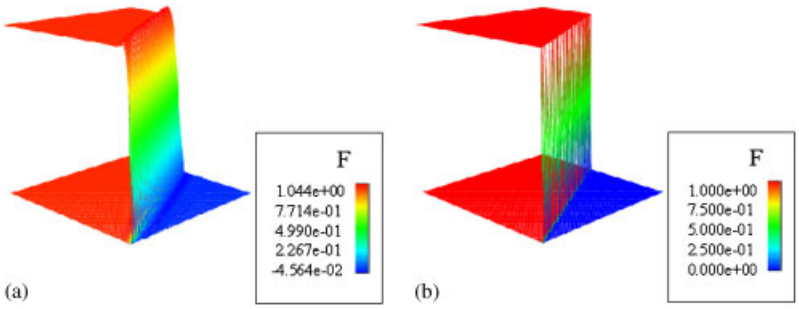

Figure 13. Steady-state advection skew to the mesh problem: elevated surface of F without (a) and with (b) the filtering procedure.

Table II. Errors obtained with and without the filtering procedure in the case of the steady-state advection skew to the mesh problem.

	E_0	E_∞	E_{tot}
Without filtering	0.13×10^{-2}	0.52	0.36×10^{-1}
Filtering	0.63×10^{-3}	0.59	0.80×10^{-2}

be surprising *a priori*. This behaviour is local and results from the more important smearing effect of the colour function at the interface in the non-filtered case.

Figure 14 illustrates a contour plot of the error field ΔE_i (from Equation (5)) obtained with and without filtering. For both cases, one can readily see that the error diffuses on both sides of the interface due to the presence of oscillations. However, with the filtering technique, the error decreases much more rapidly as we move away from the interface, which explains the behaviour of E_0 and E_{tot} in Table II.

3.2.2. Zalesak's problem. The Zalesak's 2D problem [21] is commonly used to test the quality of advection solution methods. This problem consists of a solid body rotating in a 4×4 square domain. The solid body is a cut-out cylinder of height 1 (the value of the colour function) advected by a counterclockwise rotating flow field (Figure 15). The velocity field

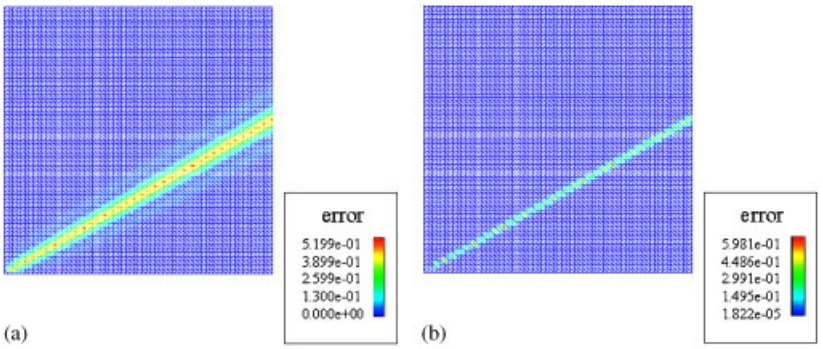

Figure 14. Steady-state advection skew to the mesh problem: contour plot of ΔE_i: (a) without; and (b) with the filtering procedure.

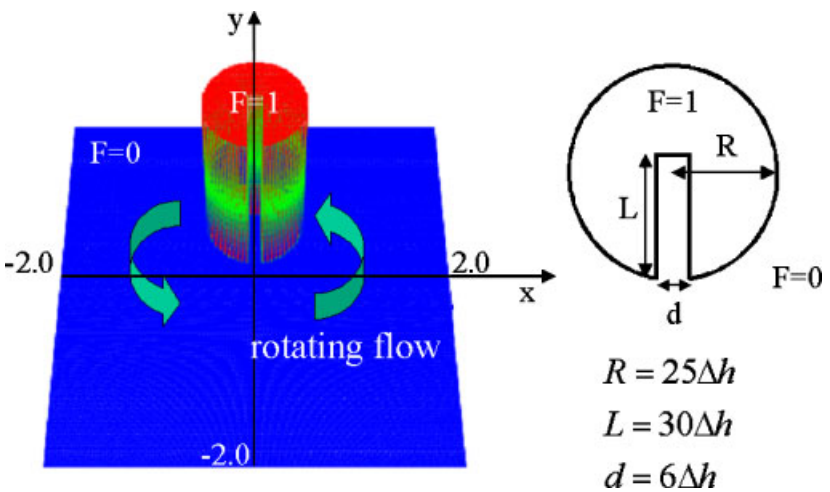

Figure 15. Zalesak's problem: transport of a cut-out cylinder in a rotating flow.

is given by $\mathbf{u} = (-y/2, x/2)$. The centre of the cylinder is initially located at (0.0, 0.75). The computational grid contains $2 \times 200 \times 200$ regular triangular elements, so that the mesh size is $\Delta h = 0.02$. The diameter of the cylinder is $50\Delta h$, the slot width is $6\Delta h$ and the slot length is $30\Delta h$, as shown in Figure 15.

Two simulations were performed: one with the SUPG method alone and the other with the SUPG method coupled with the filtering technique. In each case, a second-order implicit Crank–Nicholson time scheme was used. The time step Δt was set to 5×10^{-3} s and 2524 time iterations were needed for the cut-out cylinder to perform one complete revolution around the centre of the domain (0.0, 0.0).

Figures 15, 16(a) and (b) display, respectively, the elevated contours of F at $t = 0$ s and after one complete revolution, without and with the filtering procedure. As can be seen in Figure 16(a), the SUPG method suffers from diffusion and does not eliminate all the oscil-

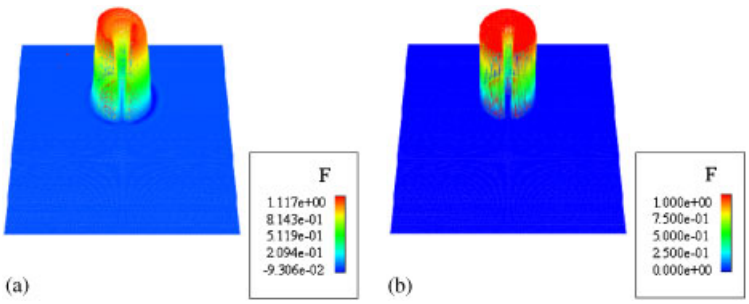

Figure 16. Zalesak's problem: elevated surface of F after one revolution: (a) without; and (b) with the filtering procedure.

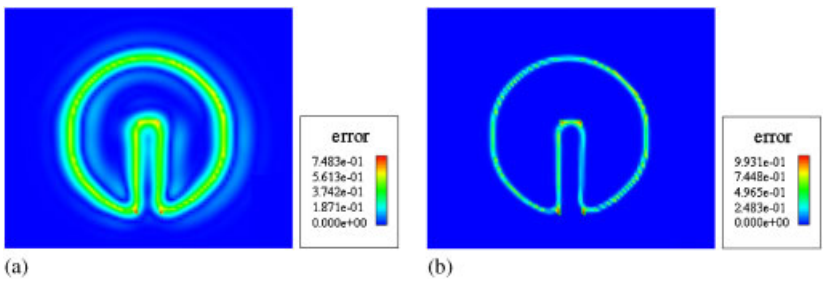

Figure 17. Zalesak's problem: contour plot of ΔE_i after one revolution: (a) without filtering; and (b) with the filtering procedure.

lations. Combined with the filtering technique, it yields a sharp interface (Figure 16(b)) and removes the spurious oscillations in a satisfactory manner.

Figures 17(a) and (b) show a contour plot ΔE_i for the non-filtered and the filtered cases, respectively. After one complete revolution, the largest errors are located at the corners of the cut-out cylinder as these initially sharp corners have rounded out. Overall, the error is actually smaller when SUPG method is combined with the filtering technique, as shown by the values of E_0 and E_{tot} in Table III.

A comparison of the results obtained with the volume-tracking methods (SLIC, Hirt and Nichols, FCT–VOF and Youngs' methods) of Reference [7] and the filtering strategy of this work is presented in Figure 18 and Table IV. One can conclude from these results that the present filtering strategy outperforms the standard SUPG method as well as the SLIC and Hirt–Nichols VOF methods, but is less accurate than the FCT and Youngs' VOF methods. As we shall see next, one way to reduce the slightly more important diffusion obtained with the filtering technique in comparison with, for example, Youngs' method would be to combine it with the mesh refinement procedure presented in Section 2.

4. COMBINATION OF THE MESH REFINEMENT AND FILTERING TECHNIQUES

To assess the efficiency of the strategy that consists of combining the mesh refinement and filtering techniques discussed above, two benchmark problems are considered.

Table III. Errors obtained with and without the filtering procedure in the case of Zalesak's problem.

	E_0	E_∞	E_{tot}
Without filtering	0.28×10^{-3}	0.74	0.25
Filtering	0.18×10^{-3}	0.99	0.69×10^{-1}

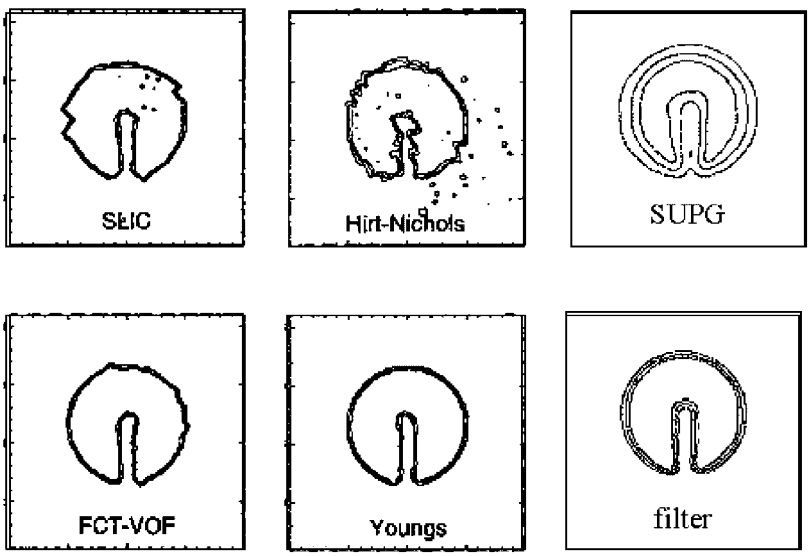

Figure 18. Comparison of advection methods for the Zalesak's problem, after one revolution (from Reference [7] and present work).

Table IV. Comparison of advection methods for Zalesak's problem, after one revolution (from Reference [7] and present work).

Method	SLIC	Hirt–Nichols	FCT–VOF
E_{tot}	8.38×10^{-2}	9.62×10^{-2}	3.29×10^{-2}
Method	Youngs	SUPG	Filter
E_{tot}	1.09×10^{-2}	2.53×10^{-1}	6.99×10^{-2}

4.1. Zalesak's problem

The test in Section 3.2.2 is taken up again for the same conditions. Seven iterations of the mesh refinement technique were performed, leading to Δh varying from 0.003 to 0.4. The elevated surfaces and $F = 0.5$ contour line at $t = 0\,\text{s}$ are shown in Figures 19(a) and (b), respectively, whereas Figures 20(a) and (b) display the elevated surfaces of F after one complete revolution in the case where the mesh refinement technique was used without and

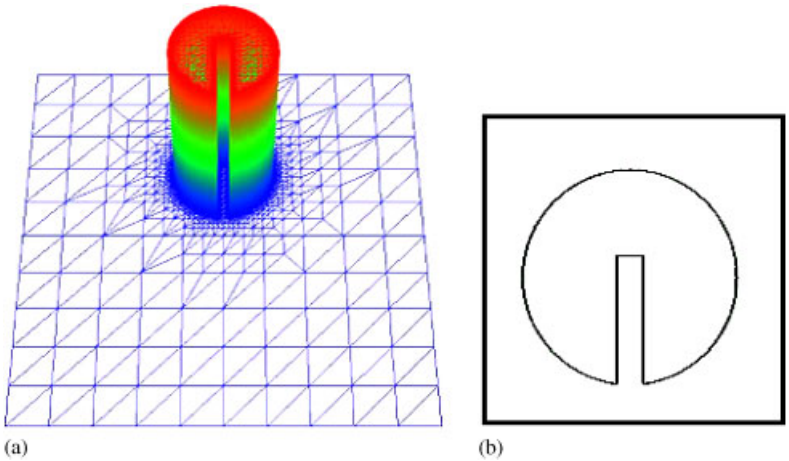

Figure 19. Zalesak's problem: (a) elevated surface of F; and (b) $F = 0.5$ contour line at $t = 0$ s.

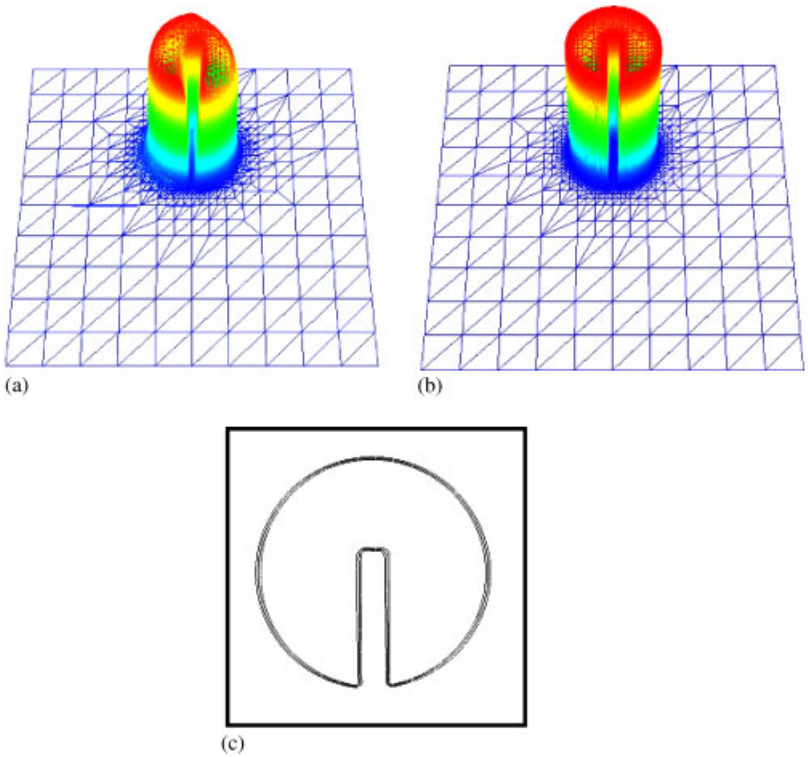

Figure 20. Zalesak's problem: elevated surface of F after one revolution obtained with the adaptive mesh refinement technique: (a) without; and (b) with the filtering procedure; (c) $F = 0.025$, $F = 0.5$ and $F = 0.975$ contour lines after one revolution.

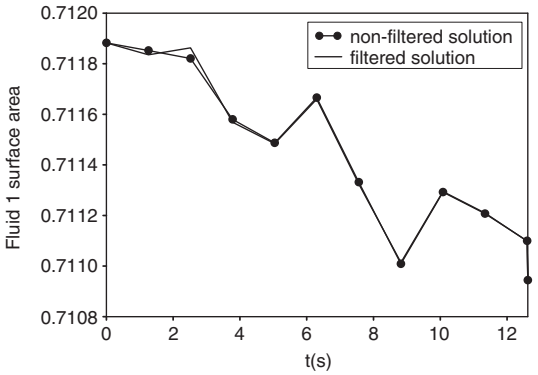

Figure 21. Zalesak's problem: surface area of the cut-out cylinder during its advection.

with the filtering technique, respectively. One can readily see in Figure 20(b) that the solution is really close to the initial solution (Figure 19(a)). The $F = 0.025$, 0.5 and 0.975 contour lines of Figure 20(c) can also be compared to those of Figure 18. The comparison is a clear indication of the efficiency of the method. It is also interesting to compare Figure 20(c) with the $F = 0.5$ contour line at $t = 0$ s of Figure 19(b); the two results are nearly the same, which shows that the smearing effects after one revolution are very small and barely visible. In fact, as in the case of the advection of the square presented in Section 2.2.2, the shape of the cut-out cylinder remains unchanged except at the corners, which are slightly rounded because of the small numerical diffusion. Finally, the evolution with time of the area of the cut-out cylinder is displayed in Figure 21. One may notice that, with and without filtering, the variations are very small (of about 0.1%) after one revolution.

4.2. Shear flow problem

The shear flow problem, a test widely used in the literature [7, 14, 26] was also considered. This test consists of the deformation of a disc in a shear flow. More precisely, a disc inside of which $F = 1$ and outside of which $F = 0$ is initially centred at $x = 1.57$ and $y = 0.8$ in a square domain of length π. This disc is then deformed by a single vortex described by the stream function

$$\psi = \sin^2(x)\sin^2(y) \tag{21}$$

and the velocity field

$$\begin{aligned} u &= +\frac{\partial \psi}{\partial y} = +\sin^2(x)\sin(2y) \\ v &= -\frac{\partial \psi}{\partial x} = -\sin(2x)\sin^2(y) \end{aligned} \tag{22}$$

Figure 22(I) shows the $F = 0.5$ contour lines corresponding to the analytical solution at t values of 0, 1.5, 3, 4.5 and 6 s. Figure 22(II) shows the contour lines obtained with the mesh refinement technique alone, and Figure 22(III) those obtained when the mesh refinement technique is combined with the filtering technique. One may readily observe that there is

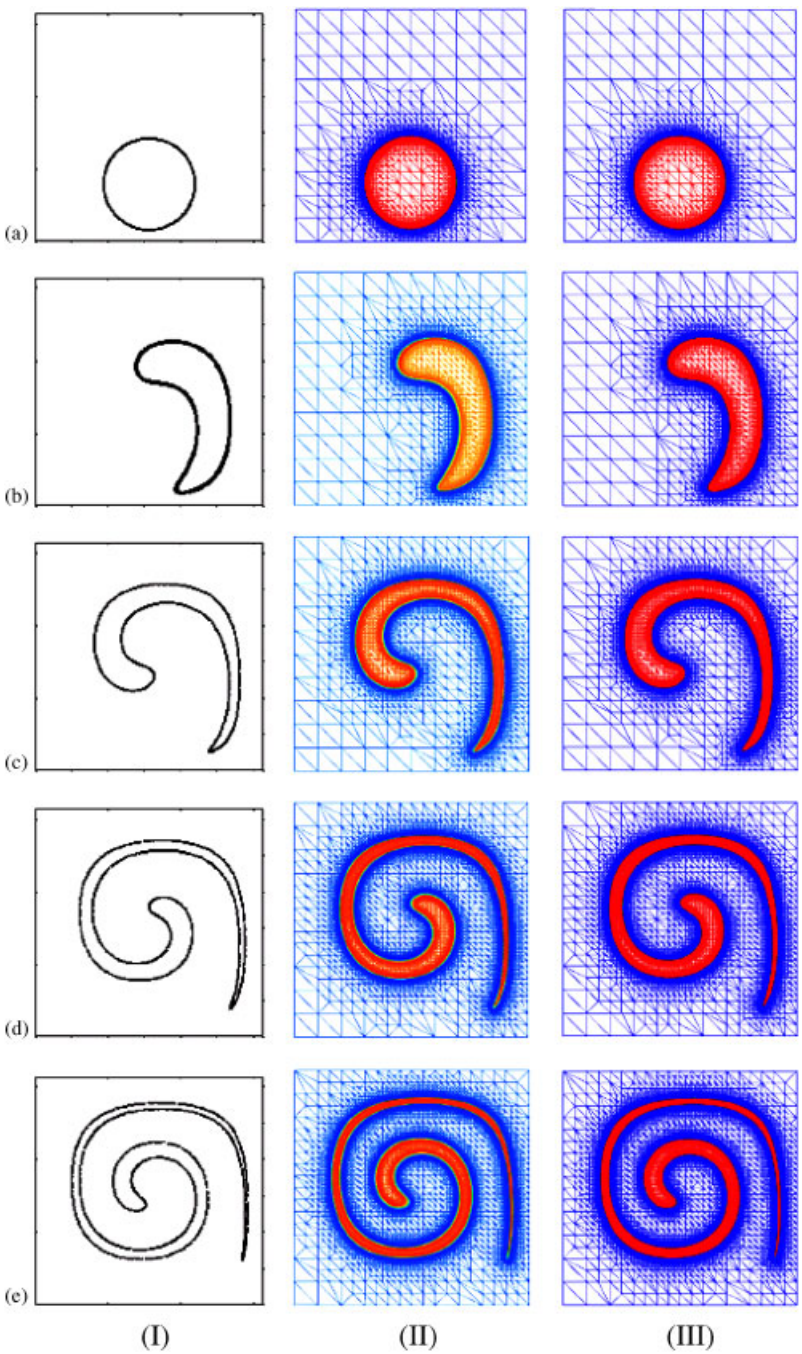

Figure 22. Shear flow problem: $F = 0.5$ contour lines corresponding to (I) the analytical solution; (II) the numerical solution obtained without filtering and (III) the numerical solution obtained with filtering at: (a) $t = 0$ s; (b) $t = 1.5$ s; (c) $t = 3$ s; (d) $t = 4.5$ s; and (e) $t = 6$ s.

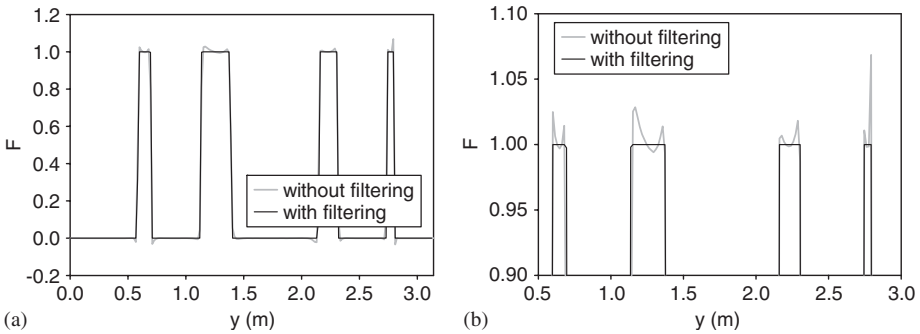

Figure 23. Colour function F along a line at $x = 0$ and at $t = 6\,\mathrm{s}$ (regular view (a) and zoom (b)).

a close agreement between the analytical and the computed solutions obtained with both methods. The filtered solution is more accurate than the unfiltered. It reduces significantly the oscillations and yields values of the colour function F that are strictly between 0 and 1. This fact is evidenced in Figure 23, which represents a section of the colour function along a line at $x = 0$ for the non-filtered and the filtered cases.

5. CONCLUSION

In this work, an efficient finite element strategy for the solution of interface tracking problems was presented. First, a mesh refinement technique based on one single reference mesh was proposed and was shown to significantly improve the accuracy of the SUPG method. A filtering procedure was then introduced to remove residual oscillations. The filter was derived in such a way that the colour function strictly lies between 0 and 1, an important feature when this colour function is used to evaluate physical properties such as density or viscosity. This filtering method does not degrade the solution; for instance, the area of the cut-out cylinder was observed to remain constant with and without filtering. The filter that was developed is low-cost and easy to implement. Finally, the overall strategy proved robust and accurate on a few benchmark problems, which shows its potential for two-phase flow problems.

In future work, the proposed strategy will be extended to the case where the velocity is given by the solution of the Navier–Stokes equations, for instance to simulate the hydrodynamics in the application nip of high speed roll coaters in paper coating applications. Results will be compared to experimental data [1] and also to numerical data computed with a free surface method [27]. More work is also required concerning the control of the regularity of the meshes that are generated by the refinement technique. In particular, special care is needed if the mesh refinement technique is to be applied recursively because it can then lead to the deterioration of the element aspect ratio. To alleviate this difficulty, a non-standard mesh refinement technique developed for the Navier–Stokes equations [18] will be extended to the solution of interface tracking problems. Future work will also deal with the influence of time integration schemes such as those proposed by Gresho and Sani [28], on numerical diffusion.

ACKNOWLEDGEMENTS

The financial contribution of the Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

REFERENCES

1. Ascanio G. Misting in the transfer nip of a film coater. *Ph.D. Thesis*, Ecole Polytechnique de Montréal, 2003.
2. Brocart B, Tanguy PA, Magnin C, Bousquet J. Design of in-line emulsification processes for water-in-oil emulsions. *Journal of Dispersion Science and Technology* 2002; **23**:45–53.
3. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
4. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
5. Dufour S, Pelletier D. Computations of multiphase flows with surface tension using an adaptive finite element method. *Numerical Heat Transfer*, *Part A* 2001; **40**:335–362.
6. Prosperetti A, Tryggvason G. Appendix 3: report of study group on computational physics. *International Journal of Multiphase Flow* 2003; **29**:1089–1099.
7. Rudman M. Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids* 1997; **24**:671–691.
8. Brooks AN, Hughes TJR. Streamline upwind Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
9. Hughes TJR, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:173–189.
10. Zienkiewicz OC, Taylor RL, Sherwin SJ, Peiró J. On discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering* 2003; **58**:1119–1148.
11. Lesaint P, Raviart PA. On a finite element method for solving the neutron transport equation. In *Mathematical Aspects of Finite Elements in Partial Differential Equations*, de Boor (ed.). Academic Press: New York, 1974.
12. Ilinca F, Hetu JF, Pelletier D. On stabilized finite element formulation for incompressible advective–diffusive transport and fluid flow problems. *Computer Methods in Applied Mechanics and Engineering* 2000; **188**: 235–255.
13. Greaves D. A quadtree adaptive method for simulating fluid flows with moving interfaces. *Journal of Computational Physics* 2004; **194**:35–56.
14. Wang JP, Borthwick AGL, Eatock Taylor R. Finite-volume-type VOF method on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids* 2004; **45**:485–508.
15. Theodorakakos A, Bergeles G. Simulation of sharp gas-liquid interface using VOF method and adaptive grid local refinement around the interface. *International Journal for Numerical Methods in Fluids* 2004; **45**: 421–439.
16. Hu ZZ, Greaves DM, Wu GX. Numerical simulation of fluids flows using an unstructured finite volume method with adaptive tri-tree grids. *International Journal for Numerical Methods in Fluids* 2002; **39**:403–440.
17. Bertrand F, Thibault F, Delamare L, Tanguy PA. Adaptive finite element simulations of fluid flow in twin-screw extruders. *Computers and Chemical Engineering* 2003; **27**:491–500.
18. Devals C, Heniche M, Bertrand F, Tanguy PA, Hayes RE. A finite element solver for the advection equation applied to interface tracking. *Proceedings of Numiform 2004*, *Materials Processing and Design*: *Modeling*, *Simulation and Applications*, Ghosh S, Castro JM, Lee JK (eds), *AIP Conference*, *Proceedings 712*, Columbus, Ohio, U.S.A., 13–17 June 2004.
19. Usmani AS. An h-adaptative SUPG-FEM solution of the pure advection equation. *Applied Numerical Mathematics* 1998; **26**:193–202.
20. Hayes RE, Tanguy PA. An adaptive domain decomposition method for simulation of transport in porous media. *Transport in Porous Media* 1988; **3**:549–562.
21. Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics* 1979; **31**:335–362.
22. Verfurth R. *A Review of a Posteriori Error Estimation and Adaptative Mesh Refinement Techniques*. Wiley-Teubner: New York, 1996.
23. Secretan YF. Un schéma élément fini simple et adaptatif pour les écoulements de Navier–Stokes compressibles. *Revue Européenne des Éléments Finis* 1992; **1**(1):31–50.
24. Johnson C. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press: New York, 1987; 199–204.

25. Giraud-Moreau L, Borouchaki H, Cherouat A. Sheet metal forming using adaptive remeshing. *Proceedings of Numiform 2004*, *Materials Processing and Design*: *Modeling*, *Simulation and Applications*. Ghosh S, Castro JM, Lee JK (eds), *AIP Conference*, *Proceedings 712*, Columbus, Ohio, U.S.A., 13–17 June 2004.
26. Rider WJ, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
27. Mmbaga JP, Hayes RE, Bertrand FH, Tanguy PA. Flow simulation in the nip of a rigid roll coater. *International Journal for Numerical Methods in Fluids* 2005, **48**:1041–1066.
28. Gresho PM, Sani RL. Incompressible flow and the finite element method. *Advection–Diffusion*, vol. 1. Wiley: Chichester, 2000.

