INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **45**:421–439 (DOI: 10.1002/fld.706)

Simulation of sharp gas–liquid interface using VOF method and adaptive grid local refinement around the interface

A. Theodorakakos[1,‡] and G. Bergeles[2,*,†,§]

[1]*Fluid Research Co, Greece*
[2]*Department of Mechanical Engineering, Laboratory of Aerodynamics, National Technical University of Athens, 15773 Athens, Greece*

SUMMARY

The volume of fluid (VOF) method is used to perform two-phase simulations (gas–liquid). The governing Navier–Stokes conservation equations of the flow field are numerically solved on two-dimensional axisymmetric or three-dimensional unstructured grids, using Cartesian velocity components, following the finite volume approximation and a pressure correction method. A new method of adaptive grid local refinement is developed in order to enhance the accuracy of the predictions, to capture the sharp gas–liquid interface and to speed up the calculations. Results are compared with experimental measurements in order to assess the efficiency of the method. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: VOF; droplet impact; adaptive grid local refinement

INTRODUCTION

There are mainly two different approaches for the simulation of free surfaces and fluid interfaces [1]: surface methods and volume methods.

In the surface method, the interface is tracked explicitly either by marking it with marker points or by attaching it to a mesh that follows the movement of the interface. The advantage of this approach is that the exact position of the interface is always known during the simulation, and that the interface remains sharp as it moves through the mesh. Several methods of marking the interface have been used. The most obvious way is to represent the interface with a sequence of massless points which are transported in a Langrangian way (i.e. Reference [2]). That idea has been extended by relating these marker points with a height function which

*Correspondence to: G. Bergeles, Laboratory of Aerodynamics, National Technical University of Athens, 9 Heroon Polytehneiou, 15773 Athens, Greece.
†E-mail: bergeles@fluid.mech.ntua.gr
‡Research engineer.
§Professor.

Contract/grant sponsor: European Community; contract/grant number: ENK6-CT2000-00051

Received 18 November 2002
Revised 15 November 2003

calculates the distance of the points from a reference plane or point (i.e. Reference [3]). Another variation of that method is the level set method, where an equation is solved for the whole domain, representing the minimum distance from each point to the interface (i.e. Reference [4]). The values of this function lie between $(-\infty)$ to $(+\infty)$ and the interface is considered to lie in the region where α has the value of 0. Another variation of the surface method is the surface fitted method, where deformable numerical mesh is used, which follows the shape of the liquid droplet or the interface. Following this approach the outer boundaries of the mesh are moved and the mesh is deformed in every time step in a Langrangian manner thus following the liquid movement. Most of the times a fully unstructured mesh is used. This technique has proven quite successful in predicting small droplets impact on walls, with possible solidification of the liquid phase (i.e. Reference [5]). The disadvantages of the method is that it should be extremely difficult to be applied in cases of very sharp and deformed interfaces, with possible splashing or break-up and secondary droplets generation. For the case of relative simple liquid interfaces (interfacial waves of moderate height) a similar technique has been presented, with the use of two different grids, for the two liquid phases with the two grids sharing a common boundary edge (i.e. Reference [6]). Both grids are deformed in time.

In the second approach (volume method), the fluid in the whole computational domain (not only the interface) is marked. In most of these methods, the continuum surface force model (CSF [7]) is used to calculate the surface tension forces. The first approach to this method is to use particle in fluids to identify the presence of liquid inside the gas phase. The second approach, and probably most widely used today, is the volume of fraction (VOF) method. The method is based on the solution of a transport equation for variable 'α' (often also referred as indicator or colour function) for the liquid phase. Variable α takes value 0 in the region of pure gas and value of 1 in the region of pure liquid. Although this method is very simple, flexible and economical and has proven reliable in simulating a wide variety of problems, it has the inherited disadvantage that the transitional region between the liquid and the gas phase (which in reality should be close to zero or at least in the order of magnitude of the distance of the molecules) at the best case is equal to the grid distance (without considering any possible artificial diffusion problems of the transport variable α). This lowers the accuracy of the predictions concerning the shape of the interface. In order to maintain a well defined and sharp interface several methods have been presented. For example, line techniques (such as simple line interface calculation, SLIC, [8]) have been used in order to approximate the interface in each mesh cell. Another category of proposed solutions to that problem involves the use of more accurate differencing scheme for the solution of the indicator function. These are either higher-order differencing scheme (i.e. STOIC [9], a second- and third-order interpolation for convection, based on the normalized variable diagram (NVD) [10]), or differencing schemes that fall in the category of donor–acceptor scheme (i.e. [11]), where the volume fraction value of the downwind cell of a cell face is used to predict the level of the volume fraction transported.

In this work the VOF method is used. For the indicator function a differencing scheme based on the donor–acceptor method is used. The continuum surface force model (CSF) is used to calculate the surface tension forces.

The method proposed here for the further enhancement of the accuracy of the predicted interfaces and the speed-up of the calculations is based on the adaptive grid local refinement around the gas–liquid interface.

In general, the major categories of mesh adaptation are the following:

- *h-refinement*: where computational points are inserted inside the computational domain (i.e. References [12–17]).
- *r-refinement*: where the number of computational points inside the computational domain is kept constant, but their location is rearranged (i.e. References [18, 19]).
- *p-refinement*: where the local order of descritization is adjusted (mainly used in FEM).
- Hybrids of the above methods.

The criterion for the mesh refinement is usually the solution accuracy and/or the gradient of a variable. h-refinement is usually simpler and more widely used than r-refinement technique. The work here also falls in the category of h-refinement.

Many works about mesh refinement have been published. Only few of them will be mentioned here.

Chang and Haworth [14] presented a methodology for local solution-adaptive mesh refinement using cell-level and global kinetic energy balances. It is demonstrated that local kinetic energy imbalance correlates with local solution accuracy, and thus is it used as a criterion for halting the mesh refinement, lowering in that way the computational time needed for the simulation, while retaining the same level of accuracy, when compared with the results obtained by using a uniform finer mesh. Results are presented for two-dimensional steady incompressible laminar benchmark problems.

Chen *et al.* [15] presented a multi-level flow-adaptive mesh refinement strategy which has been formulated and implemented within a structured grid. Although the locally refined irregular grid portions are organized as block structured patches, a special data structure allows high memory storage economy, to be achieved. The effectiveness of this approach, especially in terms of CPU and memory storage resources, is examined by reference to several test cases simulated.

Papadakis and Bergeles [16] presented a local refinement method, which they applied to three-dimensional turbulent recirculating flow. In their method, the computational domain is covered by block-structured subgrids of different refinement levels, while the variables arrangement on those grids is staggered. They reported a 62% gain in computer time compared with a case with single fine mesh, while the results accuracy is significantly improved when compared with a case of single coarse mesh.

Jasak and Gosman [17] presented an automatic adaptive h-type mesh refinement and coarsening procedure with directional sensitivity, based on the estimated error and solution gradient. Mesh adaptation operates on hexahedra cells by hierarchically splitting cells into two, four or eight new cells. The method is tested on two test cases in 2D to examine its error reduction rate and the quality of the refined meshes.

In the present work, the simplicity of the original VOF method is retained, while the use of a generalized fluid solver capable of handling unstructured meshes which has numerical cells with an arbitrary number of faces permits the handling of locally refined regions without any modifications.

The governing Navier–Stokes conservation equations of the flow field are numerically solved on an unstructured grid, using Cartesian velocity components, following the finite volume approximation and a pressure correction method.

In the next paragraphs, the solution method and the adaptive grid local refinement technique will be presented. Following that, results are presented for the convection and rotation of a rectangular bubble, as well as results for a real case micrometric droplet impingement on walls (2D and 3D) with comparison to experimental measurements.

THE SOLUTION METHOD

The governing equations

The equations for the mass, momentum and the indicator function α, for the time-dependent problem, are expressed for an arbitrary co-ordinate system and for Cartesian velocity components. These can be written as follows:

Mass conservation equation. Conservative form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0$$

Non-conservative form for incompressible fluid is $\nabla \cdot \mathbf{u} = 0$ where ρ is the density, t the time, $\mathbf{u}$ the velocity vector.

Momentum conservation equation.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - \bar{T}) = \mathbf{S}_u \quad \bar{T} = -\left(P + \frac{2}{3}\mu \nabla \cdot \mathbf{u}\right)\bar{I} + \mu(\nabla \otimes \mathbf{u} + (\nabla \otimes \mathbf{u})^{\mathrm{T}})$$

where $\bar{T}$ is the stress tensor, P the pressure, μ the dynamic viscosity of the fluid, $\bar{I}$ is the unit tensor and $\mathbf{S}_u$ is any added source terms.

Conservation equation for any scalar variable φ.

$$\frac{\partial \rho \varphi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \varphi - \mathbf{q}) = S_\varphi \quad \mathbf{q} = \Gamma_\varphi \nabla \varphi$$

where S_φ is any added source terms, $\mathbf{q}$ is the diffusion flux vector and Γ_φ is the diffusion coefficient.

Indicator function α conservation equation (incompressible fluid).

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot \alpha = 0$$

Description of the descritized procedure

The VOF method is well established and documented, and here only the key features of the method shall be discussed very briefly. According to our experience, the main requirements for employing the VOF method are the following:

- The non-conservative form of the equation for the mass should be used for deriving the pressure equation.

- For the indicator equation, α a special differencing scheme should be used that can handle the density changes across the interface and reduce numerical diffusion in these areas. In this study the CICSAM scheme is used [20].
- A second-order time differencing scheme should be used for all variables. The Crank–Nicholson implicit time differencing is used in the present study.
- Time step should be adjusted so that local courant number does not exceed a value between 0.2 and 0.4.
- Local radii of curvature of the interface should be calculated and used to derive the surface tension forces, which are added to the momentum equations. Moderate smoothing of the radii of curvature maybe needed. The surface tension is calculated according to the method of the continuum surface force model (CSF [7]).

The transport equations are integrated and discretized over the common control volume following the finite volume method.

The grid that is used for the cases simulated in this work, although looks like a local refined Cartesian structured grid, it is actually treated as an unstructured mesh, where every cell can have an arbitrary number of faces and neighbouring cells. In this way, no special treatment is needed when applying local refinement in a region (i.e. a two-dimensional cell neighbouring to a 'splitt' cell is treated as a cell with five faces and thus having five neighbouring cells instead of 4). The grid arrangement is collocated, where all the unknown variables are stored in the centre of the computational cell. In order to avoid pressure–velocity decoupling problems, arising from the fact that pressure and velocities are calculated in the same location, the convective flux through each cell face is calculated using the modification first proposed by Rhie and Chow [21] for Cartesian grid and extended here for generalized curvilinear co-ordinates. The key feature of this approach is that the velocity used to calculate the convective flux through a cell face, is not calculated by a linear interpolation of the adjacent cells velocities, but is modified to be directly linked to the two adjacent pressure nodes. Following this procedure, a pressure prediction–correction method (resembling the well-known SIMPLE algorithm [22]) is used in order to derive the pressure equation from the continuity equation.

The convective and normal diffusion terms are discretized using the BSOU scheme (Bounded Second Order Upwind, [23]). The cross diffusion terms and the second-order derivatives are discretized using standard central difference scheme. These terms are moved to the right-hand side of the conservation equation, and are treated explicitly in the iterative procedure.

The set of the linear equations that result after the discretization of the conservation equations, are solved iteratively using a preconditioned conjugate gradient method solver.

Description of the local grid refinement method

The grid refinement technique is as follows:

- *Step* 1: First decide if a new locally refined grid has to be created. Depending on the time step used, usually in the cases that shall be presented afterwards, a new local refined mesh is created every 10–20 time steps. The desired requirement is that the interface always lies in the densest grid region. Ideally the procedure should be performed in every time step. But because it is a slightly time consuming procedure, it is preferred alternatively to extend the refined region (within a given cell distance around the interface, as explained in step 4 below) and perform the new grid generation at every 10–20 time steps.

- *Step* 2: The solution of the previous time step is remapped in the coarse level grid. The method of variables remapping from one numerical mesh to another shall be explained later.
- *Step* 3: The cells of the coarse grid where the α indicator function variable lies between $\alpha_{\min}$ and $\alpha_{\max}$ are marked. Values used for those limits are $\alpha_{\min} = 0.2$ and $\alpha_{\max} = 0.8$.
- *Step* 4: The cells that are marked, plus some additional cells around them (neighbouring cells within a given cell distance, usually including 3–4 cell layers around the actual interface) are locally refined by a factor of 2 (i.e. in 3D case a cell is splitt into eight cells). In that way, a new grid with 1 level of local refinement, is created.
- *Step* 5: Remap the solution of the previous time step in this new mesh (grid with 1 level of local refinement).
- *Step* 6: Again the cells of this grid where the α indicator function variable lies between $\alpha_{\min}$ and $\alpha_{\max}$ are marked.
- *Step* 7: The cells that are marked (plus some additional cells around them) are locally refined by a factor of 2. In that way a new grid with 2 levels of local refinement, is created.
- *Step* 8: The above procedure can be repeated as many times as desired, with the limitation of the available computer resources.
- *Step* 9: The solution of the previous time step is remapped in the new local refined mesh. The remapping of the variables is done by interpolating the values of the variables in the cells of the grid of the previous time step closest to the cell of the new grid. If the cells of the 2 grids coincide, the exact value of the variables are transferred (this is the case for the cells in the vicinity of the interface, where the new grid remains refined to the highest refinement level). For example, and in reference with Figure 1, for creating a local refined grid (3 levels of refinement) around a droplet, the following steps are performed:
- *Step* 10: Grid 'old' is the grid and the solution for variable α for the previous time step (or the beginning of the calculations). The values of variable α are remapped from 'old

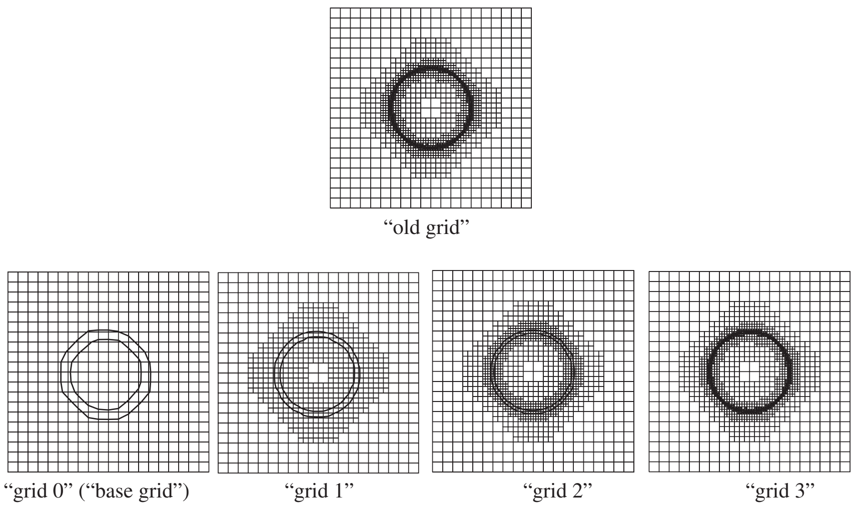

Figure 1. Grid creation sequence.

grid' to 'grid 0' ('base grid'). It is evident that the field of variable α in 'grid 0' is smoothed, and the interface is much thicker than in the 'old grid'. In all the plots of Figure 1, contours are plotted for $\alpha = 0.2$ and 0.8.

- *Step* 11: Cells that are inside or close to the region of the contours (i.e. near the interface) are divided, and 'grid 1' is constructed.
- *Step* 12: The values of variable α are remapped again from 'old grid' to 'grid 1'.
- *Step* 13: Cells that are inside or close to the region of the contours are divided, and 'grid 2' is constructed.
- *Step* 14: The values of variable α are remapped again from 'old grid' to 'grid 2'.
- *Step* 15: Cells that are inside or close to the region of the contours are divided, and 'grid 3' is constructed.
- *Step* 16: Finally, the values of variable α are remapped again from 'old grid' to 'grid 3', which is the final grid that will be used for the calculations of the new time step.

Figure 2 shows the procedure of successive local refinements around the interface.

The remapping or transfer of the variables from one grid to another is done by interpolating the values of the closest cells, using the inverse distance between the cell centres. For example, and in reference with Figure 3, when the value at a cell of the new grid is to be calculated

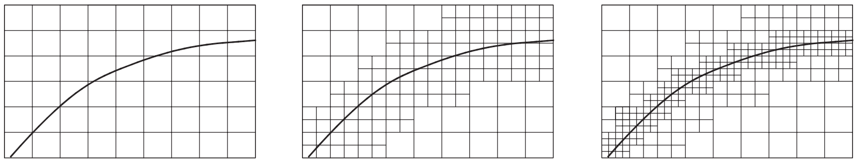
Figure 2. Successive local refinement around the interface.

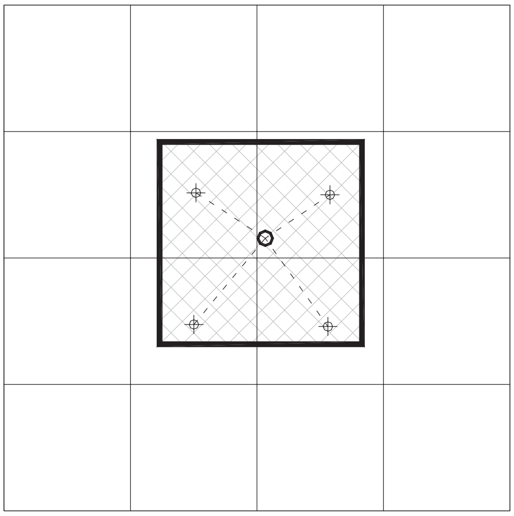
Figure 3. Transfer of variables from one grid to another.

(cell with thick lines with cross hatch), the cells of the old grid which have centres that surround the centre of the cell of the new grid are found. The value 'φ' of a variable at the centre of the new cell is estimated as

$$\varphi_p = \frac{\sum_i \varphi_i/\ell_{ip}}{\sum_i 1/\ell_{ip}}$$

where 'i' all the surrounding cells of the old grid, 'p' the cell of the new grid and ℓ_{ip} the distance of the centres of cells 'i' and 'p'. When the centres 'i' and 'p' coincide, then $\varphi_p = \varphi_i$. This method is not conservative, but it is fast, especially when using unstructured meshes. Moreover, and because the grid in the vicinity of the interface does not change when creating the grid for the new time step (because it is always refined up to a distance around the interface), the variables in the region of interest are transferred without modifications.

Although this procedure is triggered only by the presence of the interface inside the flow field, it could as well be used for capturing i.e. sharp velocity gradients (i.e. in step 3 of the above procedure, cells are marked for refinement using a different criterion). Or both mechanisms could as well be combined.

RESULTS AND DISCUSSION

The first 2 test cases presented involve the simulation of a square bubble convection and a rectangular bubble rotation and are typical for examining the accuracy of the VOF method and its ability to capture sharp interfaces. For both cases the simulation is 2D, the velocity field is diagonal to the grid lines and is supposed to be steady and is not solved, and the surface tension forces are not simulated (surface tension is zero).

For the first case, a square bubble is convected across a velocity field, in a diagonal direction. The initial and final position of the bubble inside the fluid field is shown in Figure 4. The test cases simulated are shown in Table I. The final positions of the bubble for all cases simulated are shown in Figure 5. In Figure 6, the grid for test case A5 for a time step

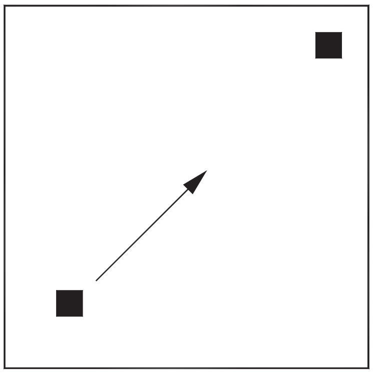

Figure 4. Initial and final bubble position for square bubble convection.

Table I. Test cases for square bubble convection.

Case	Grid (coarse level)	Level of local refinement	Total Courant of cells	number number	Cells in droplet	Time steps	Total CPU time (s)
A0	28×28	0	784	0.2	2×2	235	46.4
A1	28×28	1	900–950	0.2	4×4	432	92.1
A2	28×28	2	1150–1250	0.2	8×8	828	256.4
A3	28×28	3	1850–2000	0.2	16×16	1624	886.8
A4	28×28	4	3200–3500	0.2	32×32	3220	3537.4
A5	28×28	5	6100–6800	0.2	64×64	6416	13970.2
B0	56×56	0	3136	0.2	4×4	432	394.6
C0	112×112	0	12544	0.2	8×8	828	3805.6
D0	224×224	0	50176	0.2	16×16	1624	29963.3
A4b	28×28	4	3200–3500	0.4	32×32	1624	2483.9
A4c	28×28	4	3200–3500	0.6	32×32	1093	1857.1

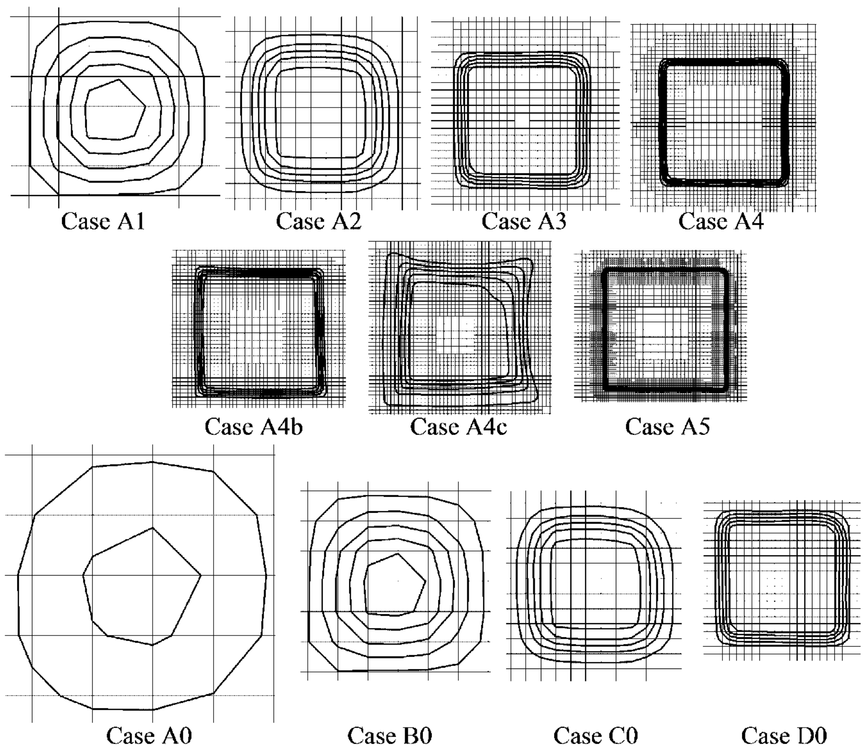

Figure 5. Results for square bubble convection with uniform flow field (isolines are plotted for values of $\alpha = 0.1$, 0.3, 0.5, 0.7, 0.9).

approximately at the middle of the simulation is shown. It is worth mentioning that the results for cases with the same grid refinement around the interface are almost identical (i.e. case B0 with A1, C0 with A2, D0 with A3).

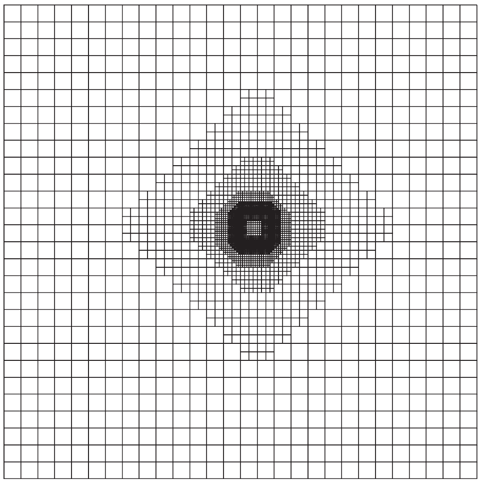
Figure 6. Mesh and predicted bubble shape for case A5, approximately at the middle of its trajectory.

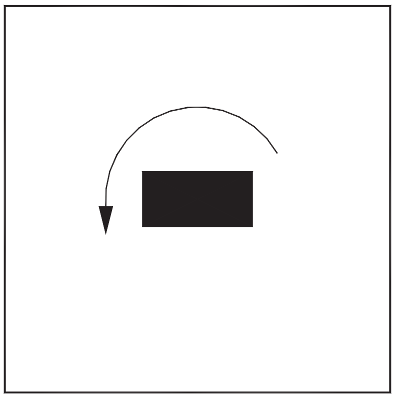
Figure 7. Initial bubble position for rectangle bubble rotation.

For the second case, a rectangular bubble is rotated inside a solid body rotation velocity field. The initial position and the direction of rotation of the bubble is shown in Figure 7. The test cases simulated are shown in Table II. The position of the bubble after a complete rotation for all cases simulated are shown in Figure 8. In Figure 9, the grid for test case A5r for rotation $7\pi/4$ is shown. From the observation of the results of each case is concluded that local refinement does not introduce any thickening or distortion in the shape of the interface, when compared to the results obtained by using a very fine uniform mesh. Further to that, the computational time needed is significantly lower when using local refinement. From another point of view, by using an adaptive local grid refinement technique, for the same computational resources, a substantial increase in the accuracy of the predicted interface, can be achieved, without using any surface reconstruction method.

Table II. Test cases for rectangle bubble rotation.

Case	Grid (coarse level)	Level of local refinement	Total number of cells	Courant number	Cells in droplet	Time steps	Total CPU time for a complete rotation (s)
A0r	28×28	0	784	0.2	8×4	223	35.4
A1r	28×28	1	1100–1200	0.2	16×8	406	117.2
A2r	28×28	2	1950–2200	0.2	32×16	786	446.2
A3r	28×28	3	3600–4400	0.2	64×32	1550	1979.5
A4r	28×28	4	7200–9900	0.2	128×64	3085	9699.2
A5r	28×28	5	11000–14000	0.2	256×128	6163	27199.9
B0r	56×56	0	3136	0.2	16×8	405	317.0
C0r	112×112	0	12544	0.2	32×16	786	2803.5
D0r	224×224	0	50176	0.2	64×32	1550	24624.5

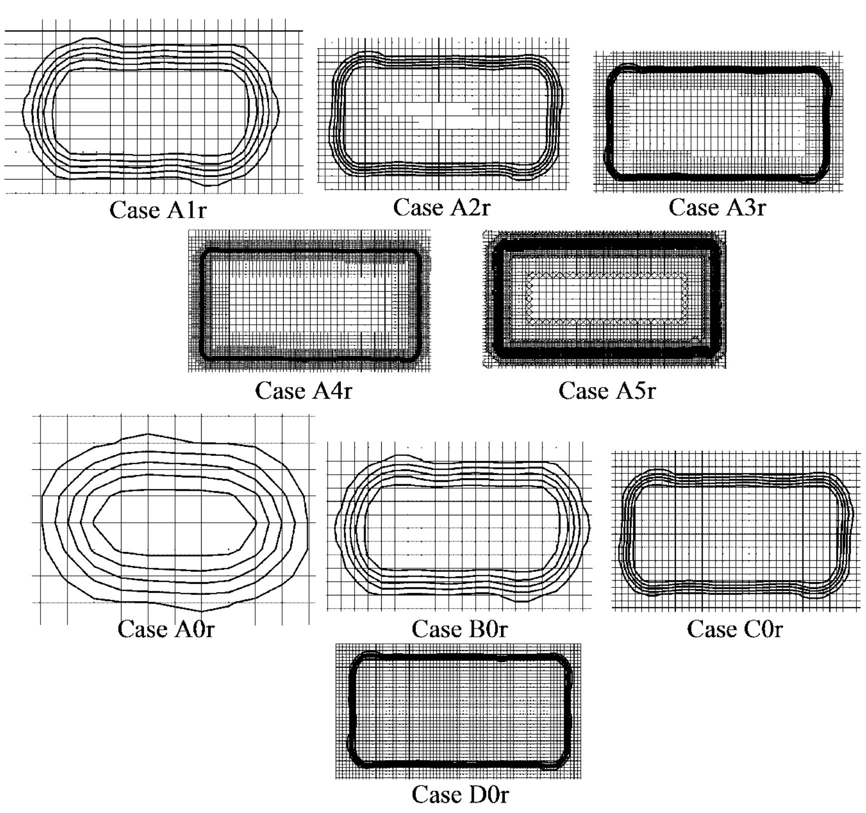

Figure 8. Results for rectangle bubble rotation (isolines are plotted for values of $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$).

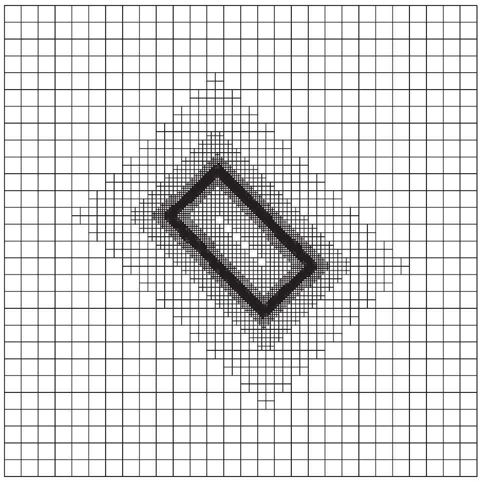

Figure 9. Mesh and predicted bubble shape for case A5r, at rotation angle $7\pi/4$.

The use of successive local refinements around the interface does increase the total number of cells of the computational domain, but still it is far more economical in terms of computational resources, than using a uniform mesh of the same refinement. It must be said though, that the increase of the computational time required for a simulation when using more than 4–5 levels of refinement around the interface does not come from the increase of computational points, but mainly from the subsequent decrease of the time step that should be used, in order to keep the local courant number in the prescribed region. As happened with the previous case, the results for cases with the same grid refinement around the interface are almost identical (i.e. case B0r with A1r, C0r with A2r, D0r with A3r).

In the first 2 cases considered the relative position of arbitrary pair of points in the flow remains unchanged, and thus the simulation was made in order to make sure that the shape of the rectangular bubble is not affected by the technique that was employed.

The third case examined however, is also made with predefined velocity, but this time the flow contains shear. A circle (radius 0.15) is centred at a location $x = 0.50$, $y = 0.75$, in a unit square computational domain. A single vortex is imposed with a velocity field defined by the stream function

$$\Psi = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y), \quad u = -\frac{\partial \Psi}{\partial y}, \quad v = \frac{\partial \Psi}{\partial x}$$

When the circular fluid body is placed in this field, it stretches and spirals about the centre of the domain. This is a rather stricter test case for this kind of calculations (i.e. References [24, 25]).

In Figure 10, the time evolution of the exact solution is shown. This is obtained by initially placing 10^5 marker points inside the circle in random positions. Those marker points are transported in a Lagrangian manner using a quite small time step.

In Table III, the test cases that were simulated are shown. For all simulations the time step was kept constant (keeping the Courant number approximately at a value of 0.2) and

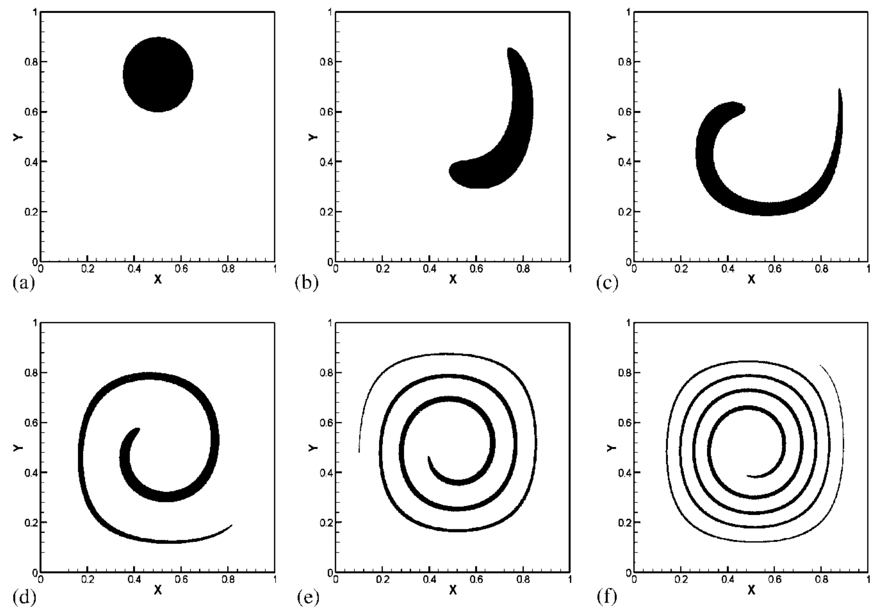

Figure 10. 'Exact' solution for the single vortex flow field case, for: (a) $t=0$; (b) $t=0.5$; (c) $t=1.0$; (d) $t=2.0$; (e) $t=4.0$; and (f) $t=6.0$.

Table III. Test cases for the single vortex flow field case.

Case	Grid (coarse level)	Level of local refinement	Total number of cells	Courant number	Time steps	Total CPU time (s)
A0s	32 × 32	0	1024	0.2	1200	469
A1s	32 × 32	1	1450–2650	0.2	2400	2309
A2s	32 × 32	2	2550–9250	0.2	4800	15897
A3s	32 × 32	3	4850–28500	0.2	9600	96359
B0s	64 × 64	0	4096	0.2	2400	4449
C0s	128 × 128	0	16384	0.2	4800	44350

calculation were performed until $t=6$. Three cases were considered using fixed grids (32×32, 64×64 and 128×128). Another three cases with adaptive grid local refinement were also simulated (1, 2 and 3 levels of local refinement, starting from the coarse 32×32 grid). Table IV presents also information about the number of cells and the corresponding CPU time for the complete simulation. It must be noted that the CPU time reported for each of the three cases (bubble convection, bubble rotation and single vortex flow field) are not comparable to each other, as different types of PC computers were used.

In Figures 11 and 12, the results of the test cases at selected times are shown. It is evident the improvement that the adaptive local refinement technique offers to the accuracy of the predictions, as well as the saving in computational time. It is also noted that the introduction

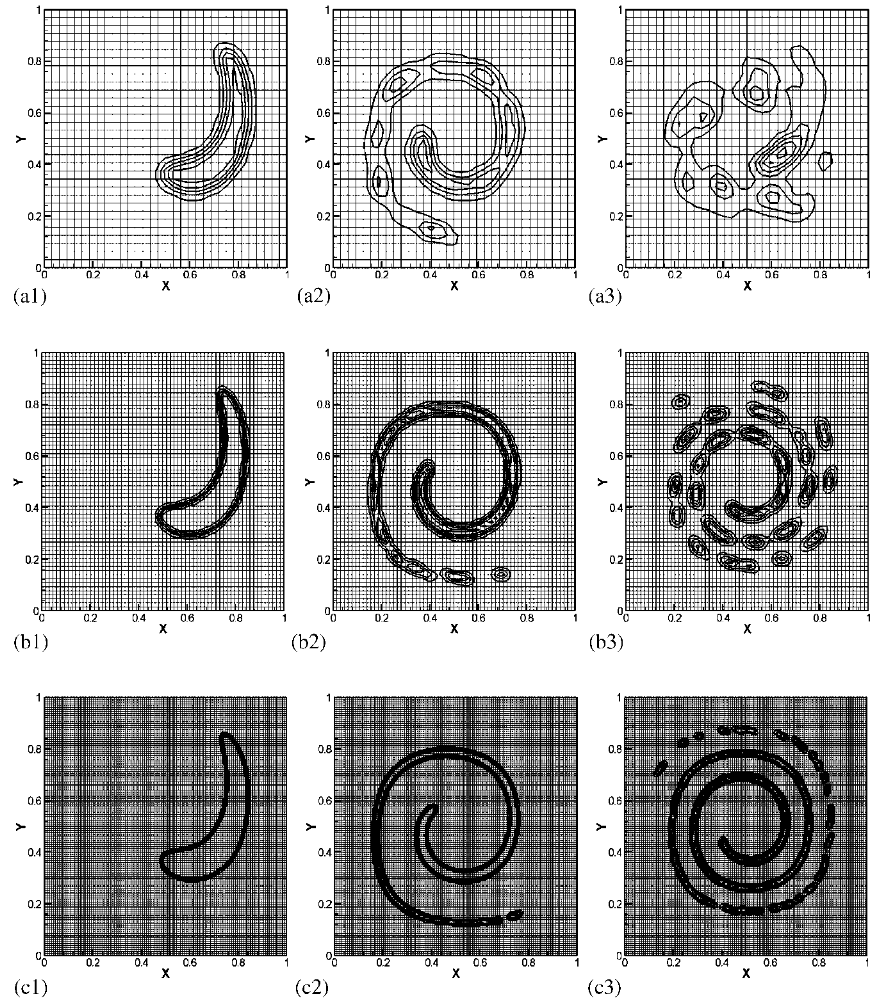

Figure 11. Results for the single vortex flow field case, for: (a1, a2, a3) Case A0s, $t = 0.5$, 2.0, 4.0; (b1, b2, b3) Case B0s, $t = 0.5$, 2.0, 4.0; and (c1, c2, c3) Case C0s, $t = 0.5$, 2.0, 4.0. (Isolines are plotted for values of $\alpha = 0.1$, 0.3, 0.5, 0.7, 0.9.)

of local refinement does not introduce any additional distortion in the shape of the predicted shape (i.e. comparing the results of cases A1s–B0s and A2s–C0s, which have the same cell size around the interface, it is evident that the resulting shapes are almost the same).

The third case examined was the water droplet impingement on pyrex glass when the surrounding gas is air. The main parameters of the impact of a single droplet onto a solid wall taken into account are droplet diameter D_0, initial impact velocity U_0 surface tension σ and dynamic contact angles, representing wettability i.e. advancing and receding. For the examined case where $D_0 = 3.76\text{mm}$, $U_0 = 1.5\text{m/s}$ and $\sigma = 0.076\,\text{Nt/m}$, the Reynolds and Weber numbers are $Re = 3010$, $We = 58.4$. This case was investigated experimentally by Fukai *et al.* [26]. The advancing and receding angles are reported to be 60 and 22°, respectively.

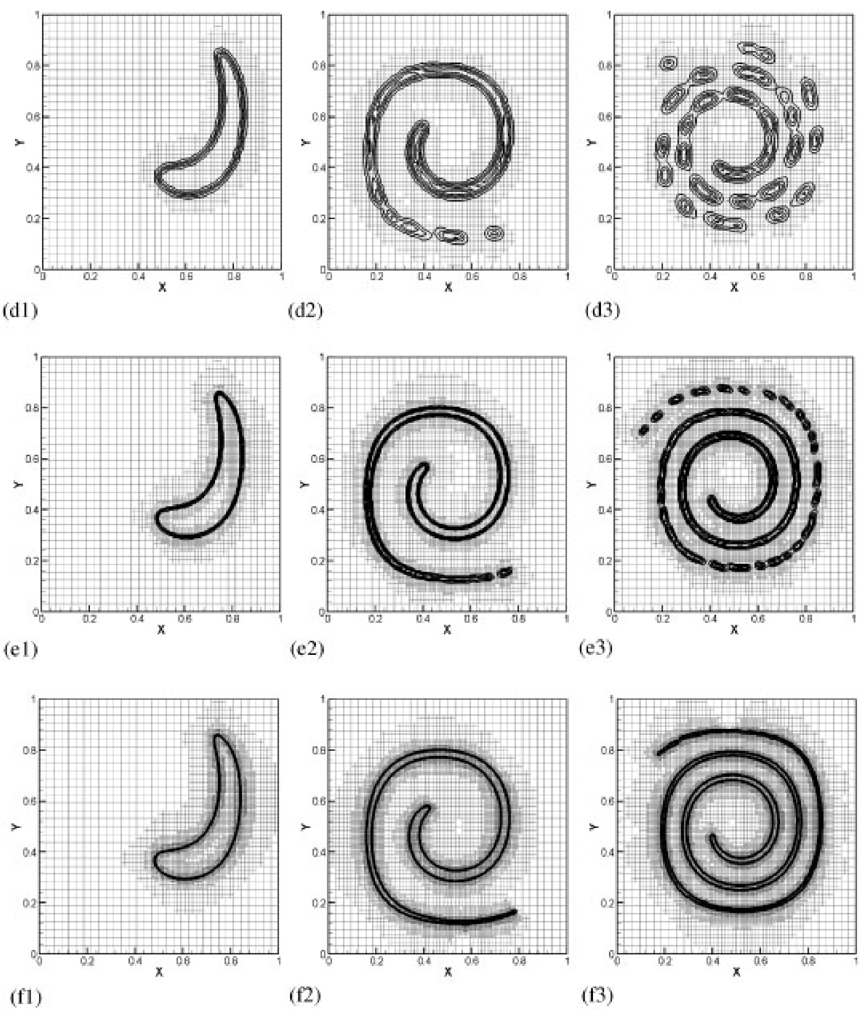

Figure 12. Results for the single vortex flow field case, for: (d1, d2, d3) Case A1s, $t = 0.5$, 2.0, 4.0; (e1, e2, e3) Case A2s, $t = 0.5$, 2.0, 4.0; and (f1, f2, f3) Case A3s, $t = 0.5$, 2.0, 4.0. (Isolines are plotted for values of $\alpha = 0.1$, 0.3, 0.5, 0.7, 0.9.)

The flow domain is axisymmetric (along the x-axis) and extends three droplet diameters in all directions around the impingement point. Two cases were considered: for the first case the numerical grid used was 100×100 cells without the use of local refinement, while for the second case the grid size was 50×50 cells with two levels of local refinement around the interface. For the second case the saving in computational time, compared to the first case without local refinement, reached approximately 70%. On the other hand, the accuracy by which the shape of the interface was captured, was enhanced. Figure 13 shows the predicted droplet shape for the second case, for various non-dimensional times T. It is also shown a magnification of the tip of the droplet for non-dimensional time $T = 8$. It is evident that the indicator function numerical diffusion is kept minimum, and the transitional region

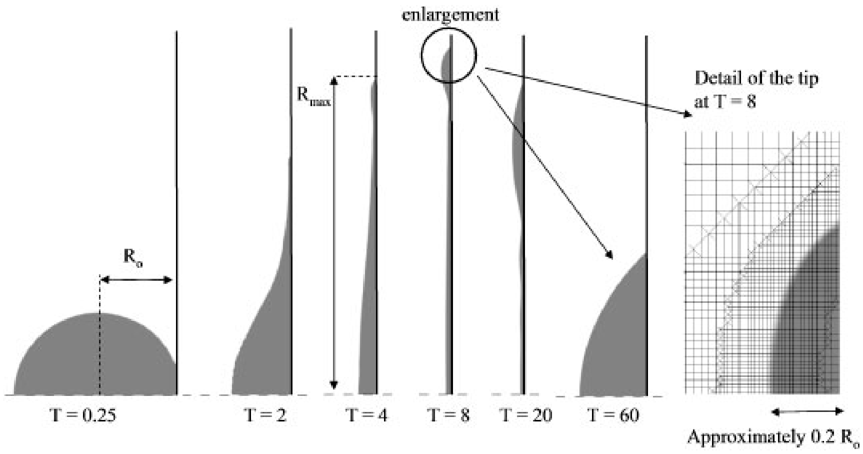

Figure 13. Time evolution of predictions for a droplet splashing on wall.

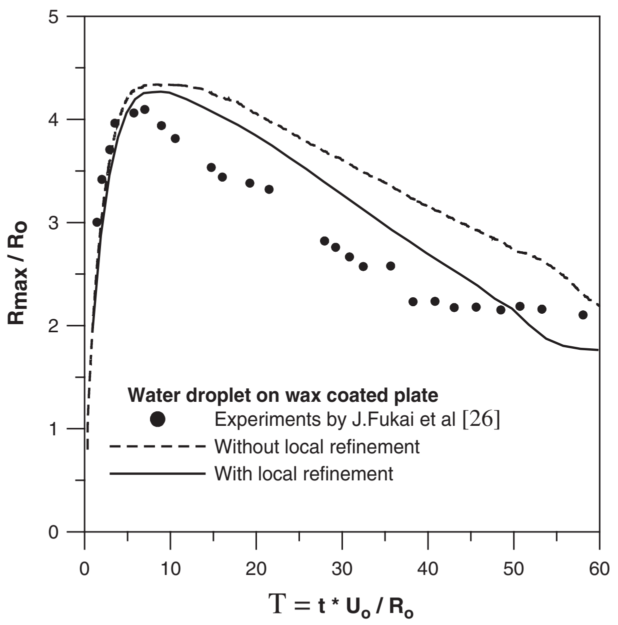

Figure 14. Comparison of droplet spreading radius.

from the cells where $\alpha \approx 1$ to the cells where $\alpha \approx 0$ is maybe 1–2 cells. In Figure 14, the predicted splashing non-dimensional radius versus non-dimensional time is shown. Wettabilitty advancing and receding angles were supposed constant during the evolution of the phenomenon. Without entering to the details of the simulation and about the influence of the wettabilitty angles for the simulation of similar cases (which falls outside the scopes of

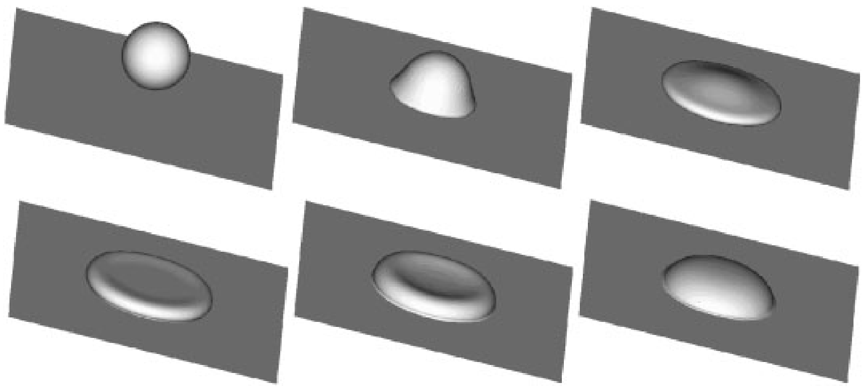

Figure 15. 3D view of the simulated droplet impinging on wall.

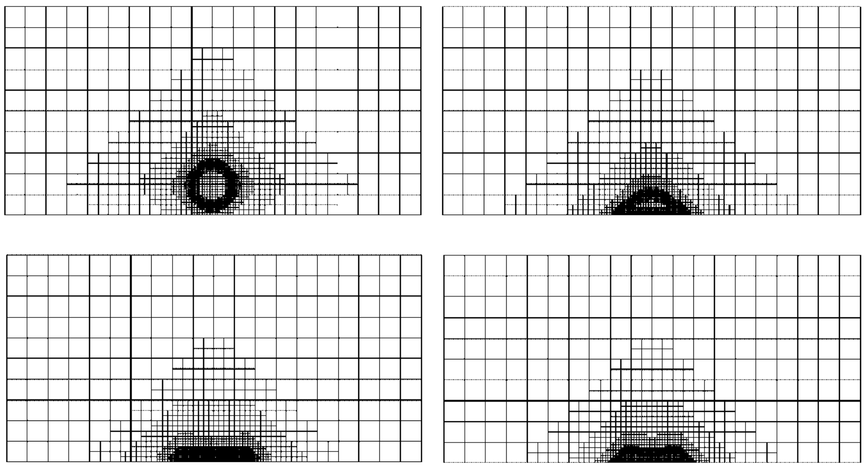

Figure 16. Slice of the 3D grid used for the simulation at 4 time steps.

this paper), it is evident the improvement that local refinement offers to the accuracy of the predictions.

The fourth case examined was the three-dimensional simulation of a micrometric water droplet impinging normally on a dry wall. The diameter of the droplet is 34.7 μm, its initial velocity 21 m/s, while the surface tension of the liquid is 0.00965 Nt/m and the density is 597 kg/m^3; the corresponding Reynolds and Weber numbers are $Re = 157.6$ and $We = 473.4$, respectively. The surrounding gas is air of ambient conditions. In Figure 15, the predicted evolution of the phenomenon is shown. In Figure 16, slices of the numerical mesh along the symmetry plane, for different time steps are shown. The numerical grid that was used was $20 \times 20 \times 10$ with five levels of local refinement around the gas–liquid interface. The total number of cells inside the calculation domain varied during simulation between 110 000 and 180 000 cells. That number of cells is realistic for performing three-dimensional time dependent simulations of cases of practical interest in low cost personal workstations. Although for this case the predictions are not compared with experimental measurements, the results for the droplet's shape seem reasonable. The use of a very refined mesh around the interface helps to minimize arithmetic diffusion of the indicator function, keeping the transitional region within 1–2 cells and retaining the predicted interface sharp.

CONCLUSIONS

An adaptive grid local refinement method has been presented. This technique is used to refine the region around the interface in two-phase simulations, using the well-known and established VOF method. It is shown that this approach helps to reduce the computational time needed for the simulation, while achieving a very good prediction for the gas–liquid interface, with minimal transitional region (with values of the indicator function between 0 and 1) and numerical diffusion.

NOMENCLATURE

D_0	initial droplet diameter
$\bar{I}$	unit tensor
ℓ_{ip}	the distance of the centres of cells 'i' and 'p'
P	pressure
$\mathbf{q}$	diffusion flux vector for a general scalar variable φ
R_0	initial droplet's radius
Re	Reynolds number $Re = \rho R_0 U_0/\mu$
$\mathbf{S}_u$	source term for the momentum equation
S_φ	source term for the conservation equation of a general scalar variable φ
t	time
$\bar{T}$	stress tensor
T	non-dimensional time $T = tU_0/R_0$
$\mathbf{u}$	velocity vector
u, v	velocity components
U	velocity
U_0	initial droplet impact velocity
We	Weber number $We = \rho R_0 U_0^2/\sigma$

Greek symbols

α	indicator function
Γ_φ	diffusion coefficient of a general scalar variable φ
μ	viscosity
ρ	density
σ	liquid surface tension
φ	general scalar variable
Ψ	stream function

ACKNOWLEDGEMENTS

The work presented in this paper was funded in part by the European Community under the 5th FWP, Program title: 'Droplet-wall-interaction Phenomena Of Relevance To Direct Injection Gasoline Engines (DWDIE)', Contract No. ENK6-CT2000-00051.

The authors are solely responsible for the work presented in this paper which does not necessarily represent the opinion of the European Community. The Community is not responsible for any use that might be made of data or information appearing here.

REFERENCES

1. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics* (Springer edn). Springer: Berlin, 1996.
2. Daly BJ. A technique for including surface tension effects in hydrodynamic calculations. *Journal of Computational Physics* 1969; **4**:97–117.
3. Nichols BD, Hirt CW. Calculating three-dimensional free surface flows in the vicinity of submerged and exposed structures. *Journal of Computational Physics* 1973; **12**:234–246.
4. Osher S, Sethian JA. Fronts propagating with curvature-dependant speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics* 1988; **79**:12–49.
5. Fukai J, Zhao Z, Poulikakos D, Megaridis CM, Miyatake O. Modeling of the deformation of a liquid droplet impinging upon a flat surface. *Physics of Fluids* 1993; **A5**:2588–2599.
6. Theodorakakos A, Bergeles G. Numerical investigation of the interface in a continuous steel casting mold water model. *Metallurgical and Materials Transactions B* 1998; **B29**:1321–1327.
7. Brackbill JU, Kothe DB, Zemach CA. Continuum method for modelling surface tension. *Journal of Computational Physics* 1992; **100**:335–354.
8. Noh WF, Woodward P. SLIC (Simple Line Interface Calculations). *Lecture Notes in Physics* 1976; **59**: 330–340.
9. Darwish MS. A new high-resolution scheme based on the normalized variable formulation. *Numerical Neat Transfer* 1993; **B24**:353–371.
10. Leonard BP. The ultimate conservative difference scheme applied to unsteady one-dimensional advection. *Computer Methods in Applied Mechanics and Engineering* 1991; **88**:17–74.
11. Ramshaw JD, Trapp JA. A numerical technique for low-speed homogeneous two-phase flow with sharp interfaces. *Journal of Computational Physics* 1976; **21**:438–453.
12. Berger MJ, Oliger J. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics* 1984; **53**:484–512.
13. Coelho P, Pereira JCF, Carvalho MG. Calculation of laminar recirculating flows using a local non-staggered grid refinement system. *International Journal for Numerical Methods in Fluids* 1991; **12**:535–557.
14. Chang S, Haworth DC. Adaptive grid refinement using cell-level and global imbalances. *International Journal for Numerical Methods in Fluids* 1997; **24**:375–392.
15. Chen WL, Lien FS, Leschziner MA. Local mesh refinement within a multi-block structured-grid scheme for general flows. *Computer Methods in Applied Mechanics and Engineering* 1997; **144**:327–369.
16. Papadakis G, Bergeles G. A local grid refinement method for three-dimensional turbulent recirculating flows. *International Journal for Numerical Methods in Fluids* 1999; **31**:1157–1172.
17. Jasak H, Gosman AD. Automatic resolution control for the finite-volume method, Part 2: Adaptive mesh refinement and coarsening. *Numerical Heat Transfer* 2000; **B38**:257–271.
18. Ramakrishnan R. Structured and unstructured grid adaptation schemes for numerical modelling of field problems. *Applied Numerical Mathematics* 1994; **14**:285–310.
19. Dandekar HW, Hlavacek V, Degreve J. An explicit 3D finite-volume method for simulation of reactive flows using a hybrid moving adaptive grid. *Numerical Heat Transfer B* 1993; **24**:1–29.
20. Ubbink O, Issa R. A method for capturing sharp fluid interfaces on arbitrary meshes. *Journal of Computational Physics* 1999; **1**:26–50.
21. Rhie MC, Chow LW. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA*-82-0998 1982.
22. Patankar SV, Spalding BD. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer* 1972; **15**:1787–1806.
23. Papadakis G, Bergeles G. A locally modified second order upwind scheme for convection terms discretisation. *International Journal of Numerical Methods in Heat and Fluid Flow* 1995; **5**:49–62.
24. Rider JW, Kothe DB. Reconstructing volume tracking. *Journal of Computational Physics* 1998; **141**:112–152.
25. Cerne G, Petelin S, Tiselj I. Numerical errors of the volume-of-fluid interface tracking algorithm. *International Journal for Numerical Methods in Fluids* 2002; **38**:329–350.
26. Fukai J, Shiiba Y, Yamamoto T, Miyatake O, Poulikakos D, Megaridis CM, Zhao Z. Wetting effects on the spreading of a liquid droplet colliding with a flat surface: experiment and modelling. *Physics of Fluids* 1995; **A7**:236–247.