INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **32**: 201–217

An adaptive front tracking technique for three-dimensional transient flows

O. S. Galaktionov, P. D. Anderson, G. W. M. Peters and F. N. Van de Vosse*

Department of Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, Netherlands

SUMMARY

An adaptive technique, based on both surface stretching and surface curvature analysis for tracking strongly deforming fluid volumes in three-dimensional flows is presented. The efficiency and accuracy of the technique are demonstrated for two- and three-dimensional flow simulations. For the two-dimensional test example, the results are compared with results obtained using a different tracking approach based on the advection of a passive scalar. Although for both techniques roughly the same structures are found, the resolution for the front tracking technique is much higher. In the three-dimensional test example, a spherical blob is tracked in a chaotic mixing flow. For this problem, the accuracy of the adaptive tracking is demonstrated by the volume conservation for the advected blob. Adaptive front tracking is suitable for simulation of the initial stages of fluid mixing, where the interfacial area can grow exponentially with time. The efficiency of the algorithm significantly benefits from parallelization of the code. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: adaptive front tracking; three-dimensional; transient flows

1. INTRODUCTION

In a variety of hydromechanical problems the tracking of fluid volumes and their interfaces is of great importance, especially for multi-fluid flows. In this paper an adaptive front tracking technique is presented, which can be used to track strongly deforming fluid volumes in two- and three-dimensional flows.

The tracking of material interfaces in fluid flows is widely addressed in the literature. Typically, the surface to be tracked represents the interface between two fluids with different mechanical properties (density, viscosity, etc.). Several techniques developed for tracking were reviewed by Beris [1], Hyman [2], Rudman [3] and Unverdi [4]. Without repeating these reviews, it should be mentioned that, in general, there are two basic approaches: *front*

* Correspondence to: Department of Mechanical Engineering, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, Netherlands.

CCC 0271–2091/2000/020201–18$17.50

Received 5 June 1998
Revised 2 February 1999

capturing (also referred as 'volume of fluid' (VOF) methods or, sometimes, a little misleading as 'volume tracking') and *front tracking*.

Several *front capturing* techniques for finite volume and finite differences computations have been proposed in the literature. Well-known methods are the simplified line interface calculation (SLIC) [5] method, the volume of fluid method (VOF) [6], or the method of Young [7]. Recently, Rudman introduced another advection technique, namely FCT-VOF [3] ('FCT' denotes 'flux-corrected transport'). In these techniques a fractional volume or 'colour' function c is defined, which indicates the fraction of a mesh cell that covers the particular type of fluid. Algorithms for volume tracking have been designed to solve the passive scalar advection equation:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) = 0,$$

where $\mathbf{u}$ is the velocity and t is the time.

In this case, special techniques are used to restore ('capture') the shape of the interface, using the computed values of the colour function c. The advantages of this approach are the relatively simple formulation for the advection of the colour function and the ability to deal with the interaction of interfaces with relative ease. The drawback is the complexity of the interface shape restoration techniques, which frequently (especially in the three-dimensional case) have difficulties with restoring a smooth and continuous surface, see e.g. [3,8].

Within a *front tracking* approach, a separate moving mesh is used to describe the interface. Front tracking methods do not suffer from disadvantages, mentioned above, that result in an erroneously non-smooth or discontinuous surface. In principle, this gives a better possibility to account for surface-related forces like surface tension. Originally, front tracking ('shock fitting') was proposed by Richtmyer and Morton [9]. This technique was further developed by Glimm *et al.* (see [10,11] and references therein), who used unstructured triangular meshes to describe the moving discontinuity, and applied it to a variety of problems ranging from the flow in porous media to shock waves interaction. A similar approach was taken by Unverdi and Tryggvason [4,12,13], who combined a moving grid for the description of the interface with flow computations on a fixed three-dimensional grid. Both approaches use the size and the aspect ratio of the surface mesh triangles to control local remeshing.

In this article, an adaptive front tracking technique is presented, which is efficient for describing strongly deforming interfaces, arising, for example, in fluid mixing. With this technique, a surface is described by an additional two-dimensional unstructured triangular mesh, similar to what was done by Glimm *et al.* and by Unverdi and Tryggvason, while the velocity field is computed using a fixed three-dimensional mesh. In addition to their techniques, surface curvature is here *implicitly* used to decide about any necessary refinement of the surface mesh. It provides the possibility of using small surface elements in strongly curved zones of the surface (near sharp edges, for example) and much larger elements in relatively planar zones, keeping the number of elements as low as possible and, thus, significantly reducing the amount of computational work for tracking the moving vertices. The technique is used in the numerical simulation of laminar distributive mixing, where repeated stretching and folding of material volumes and their interfaces constitute the underlying kinematic

mechanism of efficient mixing [14]. Additional mesh refinement in strongly curved zones of the surface ensures that the forming of sharp folds, typical for mixing flows, is properly resolved. Another distinctive feature of the present technique is that the history of the surface deformation is used for adding new markers during mesh refinement. New markers are added on a less deformed, earlier configuration of the surface (this allows the use of simple linear interpolation) and then tracked to their current position. The main disadvantage of the presented adaptive front tracking technique is that no topological changes of the surface are allowed. Thus, in its current variant, it cannot be used to describe, for example, the processes of break-up and coalescence (dispersive mixing).

In the current study only kinematic aspects of front tracking are addressed and not the specific computation of the velocity field (for this, the reader is referred to [15]). The outline of this paper is as follows. Section 2 presents the computational techniques to represent the fluid volumes, dynamic restructuring of the mesh and the use of parallel algorithms. The results of the adaptive tracking technique are discussed in Section 3. Finally, in Section 4 conclusions are drawn.

2. COMPUTATIONAL TRACKING METHOD

2.1. Representation of three-dimensional surfaces

The task to be accomplished can be described as follows. At an initial moment, a material surface is defined in a three-dimensional fluid domain. Typically this surface represents the interface between fluids that differ in mechanical, physical or chemical properties. The simplest case is when the fluids differ in colour only and the interface is passive. The surface is convected and deformed due to flow. Knowing the velocity field, the objective is to give a correct representation of the shape of the surface as a function of time given an initial shape.

For straightforward surface tracking, the initial surface is discretized by covering it with a grid of markers defining a mesh. The markers, which represent the nodal points of the mesh, are tracked in the flow by means of integrating the dynamical system $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}, t)$. The discretization of the surface should be such that certain imposed criteria (discussed later in this paper) on the accuracy of description are met. During tracking, the deformed mesh has to be refined, when necessary, to keep fulfilling these criteria.

Both structured and unstructured meshes can be used to describe the surface (see examples in Figure 1). Unstructured meshes composed of triangular cells are the most suitable for an adaptive front tracking technique. Their nodal points and mesh cells are, in general, numbered randomly and, therefore, should be addressed in an indirect way. This complication, however, is largely compensated by the fact that an unstructured mesh can easily be locally refined.

The mesh data that needs to be stored consists of the list of nodal points with their co-ordinates and the information about their connectivity. Thus, both closed and non-closed surfaces and surfaces consisting of separate segments can be described as one single object. The elementary objects used by the algorithm to describe the surface are cells (triangles), edges (sides of triangles) and nodes. Thus, three lists are kept. For all nodal points their co-ordinates are stored. The integer arrays are used to establish relations between basic objects. The first

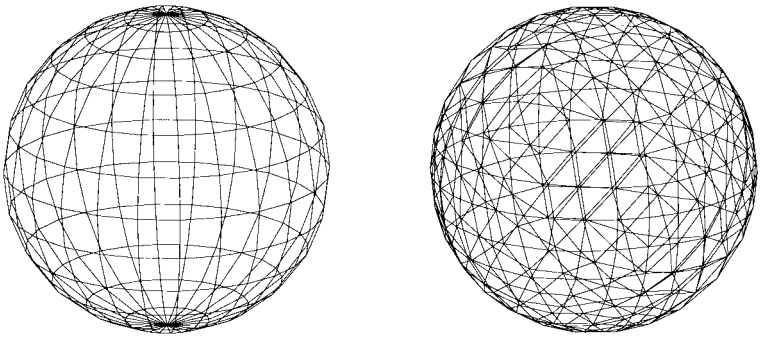


Figure 1. Examples of a structured (a) and an unstructured (b) mesh covering a sphere (266 and 258 nodes respectively).

array establishes relations between the cells and nodes, providing the pointers for every triangle to its vertices (nodes). Another array links cells to their edges (three for each cell). Edges, in turn, are referred to their two end points. This scheme of reference is redundant, as, for example, the reference to the vertices of a triangle can be recovered from the references to its edges and linking them to nodes. Such redundancy, however, simplifies the algorithms of mesh refinement and restructuring. As any edge belongs to one or two triangles, an additional array refers the edges to the cells they belong to. If only one pointer to a triangle is stored for a certain edge, it serves as an indication that this edge belongs to the contour (boundary) of the surface. This is used during mesh restructuring, preventing the contour shape from being distorted. The last array, referring edges to triangles, is easily recreated in a single loop over all triangles, using previously mentioned reference arrays. It is updated after mesh restructuring and refinement. Its main usage is finding the neighbouring triangular cells sharing common edge.

Remarks:

1. The information about the surface shape allows to compute the area of the surface and, for closed surfaces, also the enclosed volume can be calculated. The computation of the surface area is straightforward: it is the sum of the areas of all triangular cells. The enclosed volume can be computed as a surface integral using the divergence theorem

$$V = \int_{\Omega} \mathrm{d}v = \frac{1}{3} \int_{\Omega} \nabla \cdot \mathbf{x} \, \mathrm{d}v = \frac{1}{3} \oint_{\partial\Omega} \mathbf{x} \cdot \mathbf{n} \, \mathrm{d}s, \tag{1}$$

where Ω is the domain enclosed by the surface $\partial\Omega$, $\mathrm{d}s$ is the element of the surface and $\mathbf{n}$ is the local outer normal to this surface. When the surface is approximated by flat triangular cells, the surface integral (1) can be replaced by the sum over these triangles. When the vertices of all triangles are numbered such that orientation is preserved, the volume can be expressed by

a compact, simple formula. For clockwise numbering (looking from the side of outer normal) the volume is then given by

$$V = -\frac{1}{6}\sum_{n=1}^{N}\sum_{i,j,k=1}^{3}\varepsilon_{ijk}x_i^{(n)}y_j^{(n)}z_k^{(n)}. \tag{2}$$

The first sum is over all triangular cells (elements) of the surface, where $x_i^{(n)}$, $y_j^{(n)}$ and $z_k^{(n)}$ are the co-ordinates of the vertices numbered respectively i, j and k in the cell number n. In Equation (2), ε_{ijk} are the components of the well-known Levi–Civita tensor, defined as

$$\varepsilon_{ijk} = \begin{cases} 0 & \text{if } i=j \vee j=k \vee i=k, \\ +1 & \text{if } (i,j,k) \text{ is an even permutation of } (1,2,3), \\ -1 & \text{if } (i,j,k) \text{ is an odd permutation of } (1,2,3). \end{cases} \tag{3}$$

Formula (2) is exact for a volume enclosed by a surface composed of triangles. It can also be derived directly.

2. Given an arbitrary point (x_0, y_0, z_0) one can also determine if this point is inside the closed surface or not. This is of particular importance when the surface represents the interface between two different fluids. The following simple algorithm can be used for this purpose. First, an auxiliary point (x^*, y^*, z^*) is chosen, which is for granted located outside the surface. Then, this point is connected by a straight line with the point (x_0, y_0, z_0), and the intersections of this line with a surface are determined and counted. If the total number of intersections is even, the point (x_0, y_0, z_0) is outside the closed surface. If the total number of intersections is odd, it is inside. Another approach to detect if a point is inside or outside a closed surface is by computing a solid angle via the surface integral. The latter technique, however, involves more arithmetic operations, thus the former is preferred.

3. In the two-dimensional case, the boundaries to be tracked are lines instead of surfaces. This simplifies the data structure, eliminating its highest level cells.

2.2. Tracking

The evolution of material volumes in two- and three-dimensional flows is described by the dynamical system

$$\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}, t),$$

where $\mathbf{x}$ denotes the position vector and $\mathbf{u}(\mathbf{x}, t)$ the local velocity. This differential equation is hyperbolic in type and requires a stable time integration method. Methods widely used for this purpose are a second-order Heun predictor–corrector method [16] or a fourth-order adaptive Runge–Kutta–Cash–Karp method [17]. Here the latter method is used.

Since the velocity field can be either steady or transient, a distinction is made between these two possibilities. For transient flows it is hardly possible to keep a complete record of the flow

field evolution. Instead, the data from a few last time steps are kept and used. Then, mesh refinement decisions are made using the data from the last time level, but actual insertion of new nodal points is performed at the earliest time level still available. The new nodes are then tracked to the current time level, using the known history of the flow. If necessary, this refinement cycle is repeated. The advantage of this approach is a reasonably accurate representation of the surface (the mesh is refined at the earliest time level where it was less deformed) without storing the complete history of the flow and deforming surface. In the case of a steady flow, only one velocity field is used and it is possible to place the new nodal points at the initial configuration (where an exact description of the surface shape is available) and then track them to the current time level. In this way an accumulation of interpolation and time integration errors can be avoided.

For a strongly deforming fluid volume in a flow, a large varying amount of markers need to be tracked. This tracking makes parallel processing attractive because for every point these computations are independent. For the implementation of the parallel algorithm, a master–slave scheme with the dynamical workload distribution over slave processes in a Parallel Virtual Machine (PVM) network [18] was chosen. The master program performs all kinds of mesh transformations and distributes the extensive computational work (time integration) among slaves. One more auxiliary program is used to handle the velocity field data, broadcasting them to slaves on request of the master program. More details about the parallel algorithm and its efficiency can be found in [19].

2.3. Techniques to add and remove mesh cells

The mesh can be significantly distorted during flow and should be adapted where necessary. The most important aspect of the mesh adaptation is formed by criteria based on which cells should be added or removed. The mesh refinement algorithm proposed here is controlled by three parameters:

- $h_{\max}$—the maximum edge length,
- $\alpha_{\max}$—the critical angle between cells (the angle between neighbouring cells is defined as the angle between their outer normals (see Figure 2), and
- $h_{\mathrm{c,max}}$—the maximum length of edge in a cell that forms the angles larger than $\alpha_{\max}$ with its neighbours.

Thus, the adaptation algorithm *indirectly* takes into account the curvature of the surface—via the angle between neighbouring cells in combination with the length of their edges. While evaluating the curvature directly would be computationally expensive, imposing a stronger restriction on the maximum length of the edges that belong to cells that form too large angles with their neighbours offers a reasonable alternative. In the more curved regions of the surface, where angles between the normals of the neighbouring triangles are larger, the maximum allowed edge is restricted to $h_{\mathrm{c,max}}$ (in the presented examples $h_{\mathrm{c,max}} = 0.5h_{\max}$, $\alpha_{\max}$ ranges from 30° to 45°). This presents a possibility to cover strongly curved parts of the surface by small cells, while relatively flat zones of the surface are covered by large cells. This allows for minimizing the amount of markers, and thus the computational expenses. The technique described here is used for studying laminar distributive mixing, where stretching and folding

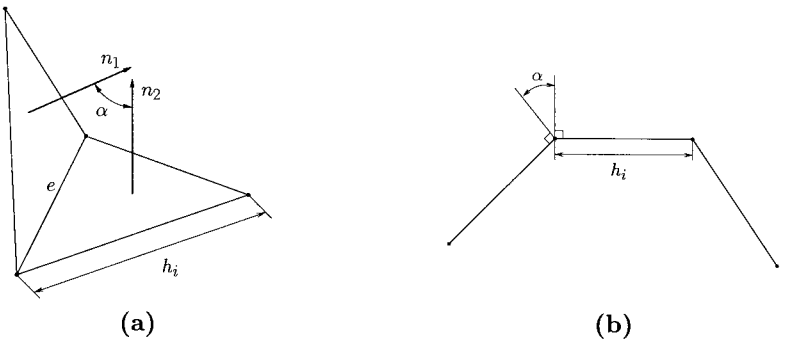


Figure 2. (a) The angle α formed by two adjacent triangles sharing common edge e is determined as the angle between their outer normals. The value of α then dictates the maximum limit that is imposed on the lengths h_i of the edges of these triangles. (b) The angle between neighbouring edges in the two-dimensional case is defined in a similar way.

are the main kinematic mechanisms [14]. Extra mesh refinement in curved zones ensures that growing sharp folds of the surface will not be missed.

The mesh edges that are tagged for refinement are split into two edges by adding new nodal points. Figure 3 illustrates how a single triangular cell is transformed when one, two or three of its edges were tagged for refinement.

When the surface is locally rapidly contracting, deletion of excessive nodal points in these regions can be beneficial. An excessive amount of nodal points can also result from mesh refinement and restructuring when parts of the surface undergo nearly uniaxial stretching (see below). In this case, an algorithm marking too short edges for deletion (using a specified minimum length as criterion), similar to how it is done for refinement, can be applied. The removal of a tagged edge (by merging two nodal points) results in the collapse of cells adjacent to this edge. Edges that become duplicated are removed.

2.4. Dynamic restructuring of the interface grid

After the mesh adaptation is completed, local changes of the mesh topology (reconnection of the nodes) is performed optionally. Highly stretched cells are avoided if possible. They are especially undesirable in finite element calculations, when physical quantities are associated

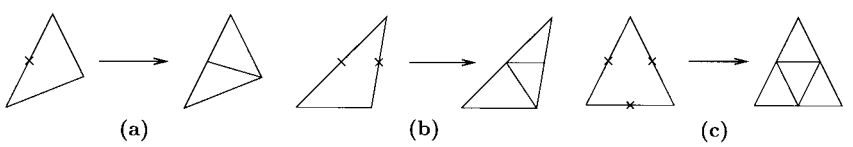


Figure 3. Scheme of transformation of the mesh cell with 1, 2 or 3 edges tagged for refinement.

with the cells. The scalar value, proportional to the ratio of the cell area to the square of its linear size,

$$q = 12\sqrt{3}\,\frac{s}{p^2}, \tag{4}$$

is used to characterize the quality of the cell, where s is the area and p is the circumference of the triangle. This gives the maximum value for a perfect triangle, and goes to zero in the limiting case when the triangle is collapsing into a single line. The value $q = 1$ represents a perfect triangle and $0 < q < 1$ otherwise (see Figure 4(a)).

An elementary transformation involves two adjacent cells. Their common edge is removed and a new one is created, connecting the other two opposite vertices (see Figure 4(b)). This 'flipping' transformation is performed when the minimum quality of the two cells before flipping is smaller than q_{crit} (for the presented results $q_{\text{crit}} = 0.6$ was specified) and the minimum quality of two new cells after transformation is larger than that before it ($\min(q_3, q_4) > \min(q_1, q_1)$ according to Figure 4(b)).

Remark:
A disadvantage, associated with the flipping transformation is that it can distort the local surface shape (influencing surrounding elements): if a pair of cells represents a concave piece of the surface, it becomes convex and vice versa. Therefore, flipping is applied only in relatively flat zones (where the angle between adjacent cells outer normals is less than some prescribed value β). Also the requirement is imposed that both cells to be flipped should have only sharp angles adjacent to the common edge (otherwise artificial folds of surface can be created). Extra precautions are taken to avoid the creation of duplicated cells as a result of flipping (when one of new cells accidentally coincides with one of its neighbours). For

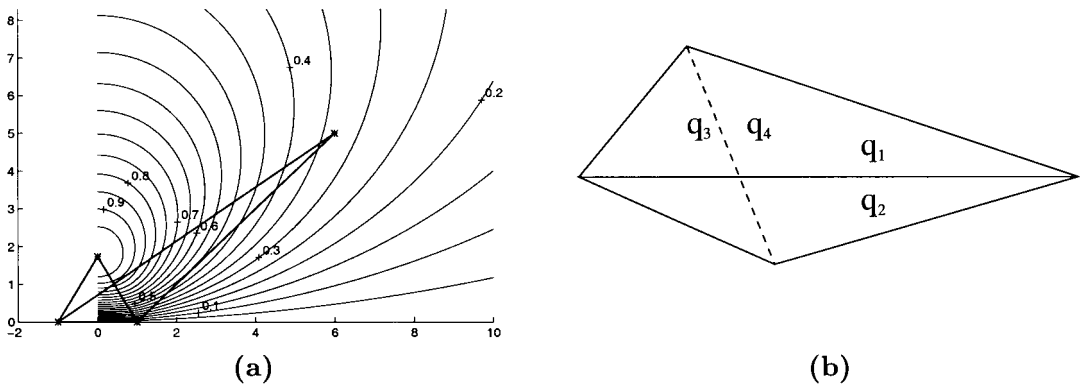


Figure 4. (a) Contour plot of quality q of triangular cell. Two vertices of the triangle are fixed at points $(-1, 0)$ and $(1, 0)$. The value of q is the function of the position of the third vertex of the triangle. (b) 'Flipping' of two adjacent cells in order to increase the minimum quality.

applications that are very sensitive to the local changes of curvature, like flows strongly affected by surface tension, the flipping algorithm should be applied with care. But for certain problems, especially when the surface is subjected to almost unidirectional stretching, as it will be shown by an example in the next subsection, such local transformations allow the computational expenses to be reduced considerably.

3. RESULTS

The adaptive front tracking technique is applied to three problems. The first (three-dimensional) problem is used to demonstrate the importance of mesh restructuring. The second problem is a two-dimensional mixing shear layer flow, which is an example of an essentially transient flow. For this problem, the results are compared with results obtained with a front capturing technique [20]. The third problem under study is the tracking of volumes in a three-dimensional time periodic mixing flow. It is an example of a three-dimensional flow in a very simple flow domain but, nevertheless, possessing chaotic mixing properties. In this case, the conservation of the volume enclosed by strongly deformed surface offers a good possibility of checking the accuracy of the adaptive front tracking technique.

The flow field in the domain Ω is described by the time-dependent Navier–Stokes equations

$$\begin{cases} St\dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = \dfrac{1}{Re}\nabla^2\mathbf{u} - \nabla p & \text{in } \Omega, \\ \nabla\cdot\mathbf{u} = 0 & \text{in } \bar{\Omega}, \end{cases} \tag{5}$$

together with the imposed boundary and initial conditions, where $\bar{\Omega}$ denotes the closure of the open domain Ω. The velocity and pressure are represented by $\mathbf{u}$ and p respectively, and St and Re denote the Strouhal and Reynolds number respectively. Since no analytical solutions are present for the velocity fields in the three-dimensional cavity and the two-dimensional mixing shear layer flow, numerical discretization is required to obtain an approximation of the velocity.

A large class of numerical techniques can be used for this purpose. For example, finite differences, finite elements or a boundary integral method can be applied. Mainly because of prior experiences, an approximate projection method for time discretization and a spectral element method for spatial discretization are applied here. Details on this numerical solution strategy and its accuracy can be found in References [15,21]. This projection scheme is second-order accurate in time and the spectral method ensures an accurate spatial resolution. During tracking of material points, a consistent interpolation scheme is used, since interpolation is done using the spectral basis functions. The interpolation error is, therefore, of the same order as the approximation error.

3.1. Performance of mesh restructuring

The influence of mesh restructuring ('flipping' of adjacent cells and removal of too short edges) is illustrated with the following example. Figure 5 shows the result of the tracking of an initially flat surface in a Stokes flow in a rectangular box. All walls, except the front and back ones, remain fixed. The flow was induced by first moving the back wall to the right, with total displacement equal to 1.75 of the cavity length L_x, and then moving the front wall to the left with the same displacement. The initial surface was a rectangle $-0.98 < y < 0.98$, $0.05 < z < 0.95$ in the plane $x = 0$.

The surface is covered with relatively large triangles to make the wire frame plot, showing all cells clear enough. The maximum allowed edge length was set to $h_{\max} = 0.25$ of the cavity height L_z, while the maximum edge length in curved parts of the surface is twice smaller ($h_{c,\max} = 0.125$), the critical angle is $\alpha_{\max} = 45°$. Figure 5(a) shows the results of tracking without mesh restructuring. It is rather clear that strong almost unidirectional stretching in some parts of the surface results in cells of very low quality q and the mesh contains a rather excessive amount of nodal points (3084). If mesh restructuring is allowed, the mesh shown in Figure 5(b) is obtained. It contains about twice less nodal points (1611).

To show how mesh restructuring in this example influences the relative number of cells with different quality q, all cells were subdivided into 50 groups according to their quality, thus each group representing the quality range $\Delta q = 0.02$ ($0 \le q \le 1$). In Figure 6(a) these numbers are plotted versus the cell quality q of the group. The thin line corresponds to the case when mesh refinement was not performed (see Figure 5(a)), while the thick line shows the distribution when mesh restructuring was done during tracking (the dotted line shows the intermediate case when only flipping was performed but no cells were deleted). As it is clear from the plot, mesh restructuring shifts the balance towards the cells of higher quality. In the given example, cells with $q > 0.6$ were considered 'good enough' and did not trigger mesh restructuring.

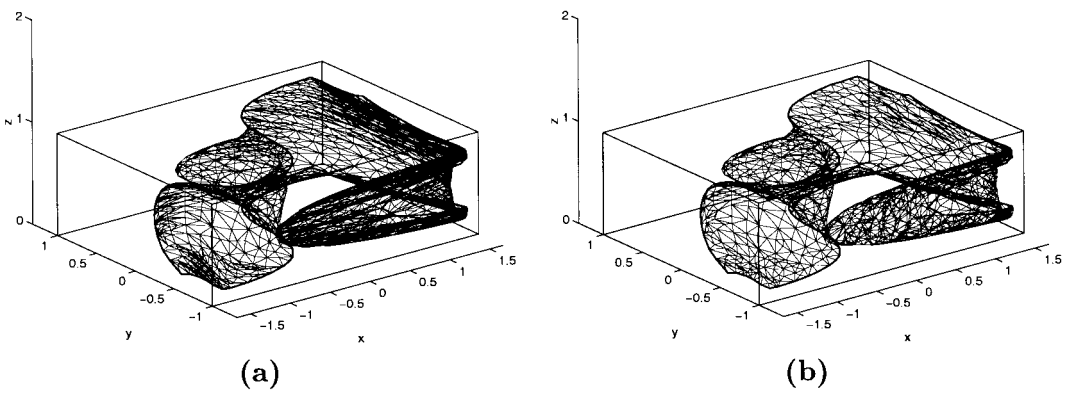


Figure 5. The example of a strongly deformed surface tracked with and without mesh restructuring: (a) no restructuring was done, mesh contains 3084 nodes and 6056 cells; (b) with dynamic restructuring, the mesh contains only 1611 nodes and 3110 cells. In both cases, the wire frame showing all mesh cells is plotted while a thicker line is used to indicate the surface edge.

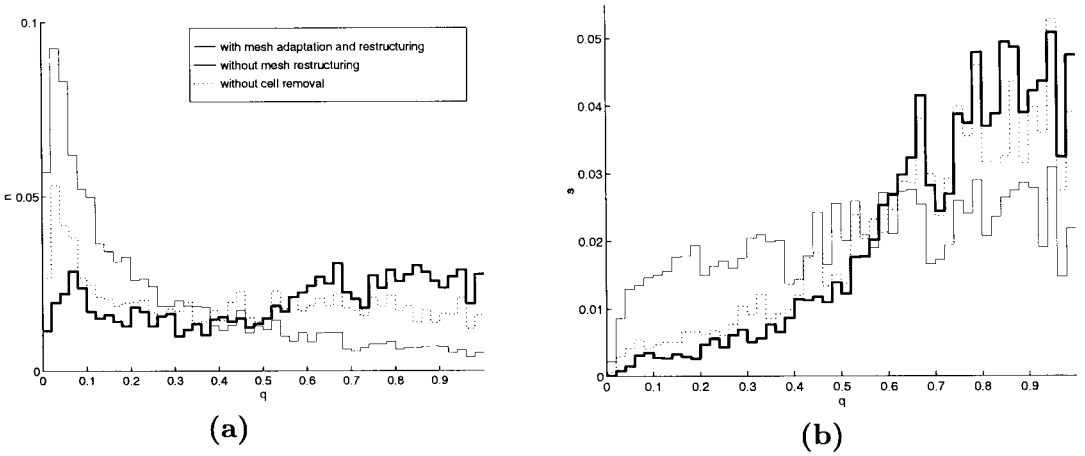


Figure 6. Relative number of cells (a) and relative area (b) covered by the mesh cells of the quality q. The range $0 \le q \le 1$ is subdivided into 50 intervals ($\Delta q = 0.02$). The thin solid line corresponds to the mesh, shown in Figure 5(a) (no mesh restructuring done). The thick line corresponds to results with dynamic mesh restructuring (Figure 5(b)). The dotted line presents the intermediate case, when only 'flipping' of adjacent cells was allowed but no removal of cells was done.

Figure 6(a) gives information only about the *number* of mesh cells with a different quality. Figure 6(b) shows the relative area covered by the cells belonging to different quality groups (the same as in the previous figure). It is rather clear that the mesh restructuring also efficiently increases the relative area covered by the cells with a high q.

3.2. Application to a shear layer flow; a comparison

Consider a mixing shear layer problem in the two-dimensional domain Ω: $0 \le x \le 8$, $-0.5 \le y \le 0.5$. At $t = 0$ the velocity and pressure fields are set to zero. The boundary conditions for the velocity $\mathbf{u} = (u, v)$ at the inlet top and bottom boundaries read

$$u(y, t) = \hat{u}(t)\left(1 + 0.5 \tanh \frac{y}{\delta}\right), \qquad v = 0, \tag{6}$$

with $\delta = 0.005$. In order to ensure a smooth start-up, the velocity $\hat{u}(t)$ is smoothly increased for $0 \le t \le 1$,

$$\hat{u}(t) = \begin{cases} 0.5(1 - \cos(\pi t)), & 0 \le t \le 1 \\ 1, & t > 1 \end{cases}. \tag{7}$$

At the outlet, stress-free outflow boundary conditions are prescribed. The viscosity is taken such that the Reynolds number, based on the distance between the upper and lower walls,

equals $Re = 1000$. The spectral element mesh that was used to compute the velocity field consisted of 16×8 elements of eighth degree and is displayed in Figure 7. The mesh is refined around the line $y = 0$ and at the inlet. The time step was chosen as $t = 0.01$.

During the initial transience ($t \leq 1$), the shear layer flow spontaneously sheds a travelling wave, which is amplified and convected through the whole domain, showing interactions with the lower and upper walls. In order to get a clear picture of the flow, it is visualized by convecting the scalar field c. In the flow domain the initial scalar field c is set to y, and the boundary condition $c = y$ at $x = 0$ is imposed. This representation of the flow can be seen in Plate 1 (top figures) where plots of the scalar field c are given at time levels $t = 2$, 4, 6, 8.

The results of the passive scalar advection technique are compared with the adaptive front tracking results. For this, the fluid inside the flow domain is subdivided into ten strips of equal width at the initial moment of time, and the boundaries between these strips are tracked using the adaptive algorithm described above. New markers are added at the entrance of the domain to separate the liquid strips in the inflow region. The colours are prescribed to the strips in a way analogous to that done for passive scalar distribution. In Plate 1, the upper plot in each pair represents the result of passive scalar convection technique; the lower plot shows the results of adaptive contour tracking. The patterns in the shear layer flow obtained using the adaptive tracking method confirm the results obtained by passive scalar advection technique. Although for both techniques roughly the same structures are found, the resolution for the newly introduced adaptive technique is much higher.

3.3. Application to a three-dimensional mixing problem

The adaptive front tracking technique described above is now applied to a three-dimensional fluid mixing simulation. The time periodic flow of viscous fluid in cubic cavity $-1 \leq x \leq 1$, $-1 \leq y \leq 1$, $-1 \leq z \leq 1$ is examined, which is a three-dimensional extension of the two-dimensional lid-driven cavity flow, extensively studied by many authors (see, for example [14,22]) for experimental and theoretical study of mixing mechanisms. The periodic flow is induced by successive motion of two opposite walls in perpendicular directions (see the walls motion protocol in Figure 8). In the example given here, the displacement of the moving wall on each step was 2.5 times larger then the cube edge length. The fluid motion was calculated using the Stokes approximation, the spectral element projection scheme was used to obtain the velocity field for a single wall motion. Figure 8 shows the spectral mesh used to discretize the cubic cavity. The mesh consists of 64 elements, each of sixth-order in all three directions and is

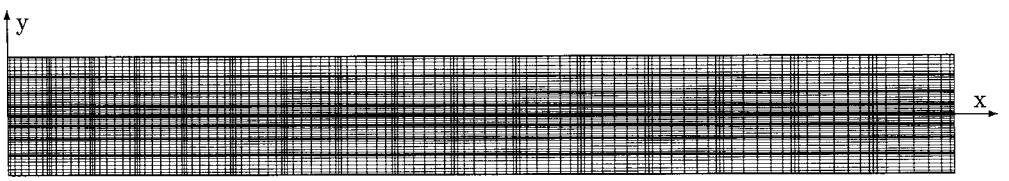


Figure 7. Finite element representation of the 16×8 spectral element mesh of degree 8.

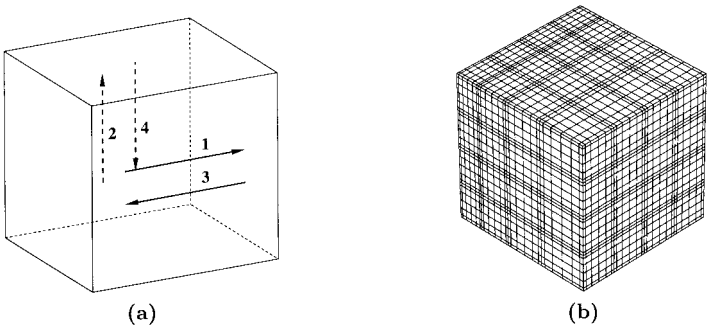


Figure 8. (a) The four-step wall motion protocol in a cubic cavity. (b) The spectral element mesh used to resolve the velocity field for a single (back) wall motion.

refined near the moving wall. In total, the mesh consists of 15625 nodal points and the flow problem is approximated using 52125 degrees of freedom. The velocity induced by the other wall motion is obtained from a simple rotation of the computed field. This flow is a convenient prototype flow for studying mixing phenomena: the simple geometry yields accurate numerical solutions for the velocity field, while the fluid motion is essentially three-dimensional.

The mixing abilities of the flow under study were evaluated by determining and diagnosing numerically its periodic points (points that return to their original position after an integer number of flow periods) [23]. To test the front tracking technique, an initially spherical blob was centred round a first-order unstable (hyperbolic) periodic point. The diameter of the blob in the given example is equal to 10% of the cavity size. The material in the vicinity of such a point undergoes locally strong deformation over a period: stretching in one direction and contraction in another. The local stretching ratio over one period at the blob location exceeds 10.0. This ensures that the blob will be strongly deformed in the flow, giving a severe test of the adaptive tracking technique. The exponential increase in interfacial area of the deforming blob with time makes the tracking problem rather difficult.

The unstructured mesh used to cover the initial blob contains 258 nodal points (see Figure 1(b)). Computations were conducted to simulate the motion of the selected blob up to three periods of the flow. The evolution of the blob shape is shown in Figures 9–12. For visualization purposes, the simulations were performed with relatively low requirements for the quality of surface shape description: the maximum allowed edge length $h_{\max}$ was equal to half of the radius of the initial blob, $h_{\max} = 0.05$, while the cell size in strongly curved regions of the surface was twice smaller, with the critical angle being 45°. The prehistory of the flow, used for mesh refinement, constitutes 0.25 of the flow period. The surface area of the blob after three full periods of motion is enlarged approximately 48.5 times. This gives a rough estimate of the average linear stretching factor (taking into account that the blob is strongly flattened) of about 100. The final shape of the blob after three periods was described by 48192 markers, which results in a good description of the shape and not too high computational expenses, but the volume conservation for the given set of parameters is relatively poor: the error reaches 14% of initial blob volume after three periods. Estimations show that, to achieve the same

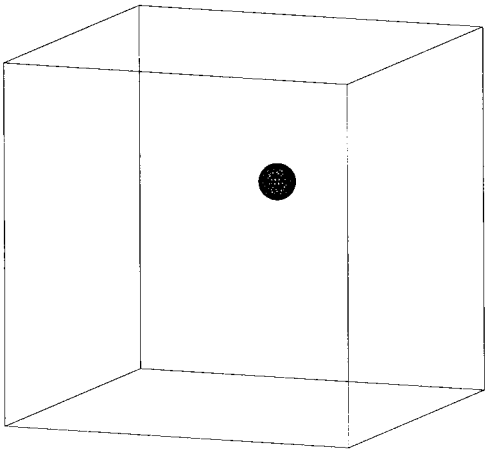


Figure 9. Initial blob around unstable periodic point.

quality of surface shape description in the most stretched and deformed parts of surface, using a non-adaptive tracking algorithm (specifying initial very fine mesh and not performing any mesh transformations) would require about 5×10^5 markers. This is caused mainly by the fact that the surface is being stretched very non-uniformly.

Some applications require better volume (mass) conservation than was achieved in this test. To achieve this, it is necessary to make the maximum allowed cell size smaller. Table I demonstrates the dependence of the volume, enclosed by the computed blob surface, on the maximum edge length $h_{\max}$. Maximum cell size in curved regions $h_{\mathrm{c,max}}$ was always twice smaller

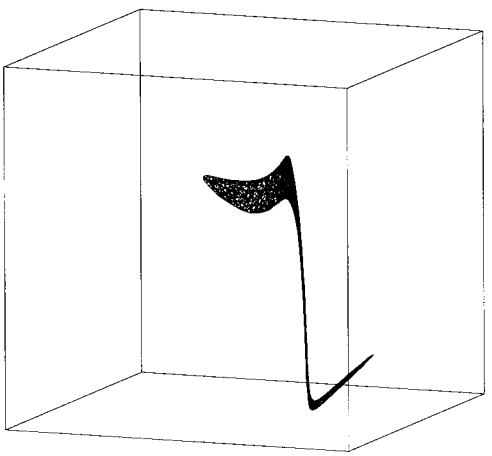


Figure 10. Blob after one period of motion.

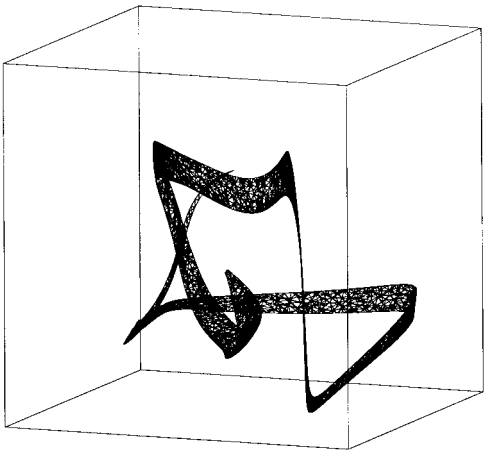


Figure 11. Blob after two periods of motion.

than $h_{\max}$ and $\alpha_{\max} = 45°$ in all cases. Table I also presents the number of markers required to describe the surface with the prescribed quality. In this way, the computational expenses for the different cases can be compared.

Another possibility is not to perform flipping transformation of the mesh. It gives better volume conservation but becomes strikingly inefficient when the surface is strongly stretched. Moreover, the cells become extremely stretched and flattened with their quality q tending to zero. When the width of such flattened cells become comparable with the inevitable numerical integration errors, continuation of simulations becomes impossible. By increasing the integration precision

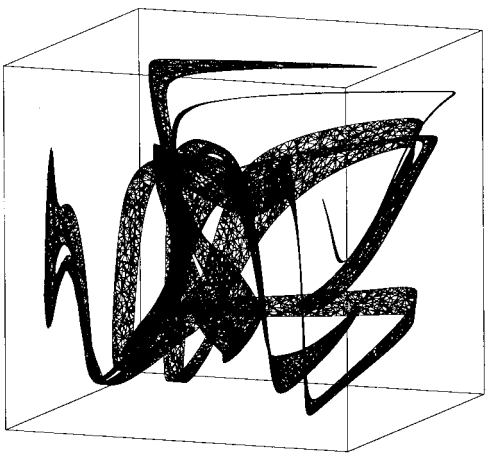


Figure 12. Blob after three periods of motion.

Table I. Dependence of the relative volume of deformed blob and the number of markers necessary to describe its surface on the maximum allowed edge length in the mesh

t/T	s/s_0	$h_{max}/r = 1.0$[a]		$h_{max}/r = 1.0$		$h_{max}/r = 0.5$		$h_{max}/r = 0.3$	
		n	V/V_0	n	V/V_0	n	V/V_0	n	V/V_0
0.5	1.7	270	0.97	229	0.96	679	0.97	1573	0.98
1.0	4.9	1804	0.94	1198	0.89	3165	0.94	9128	0.97
1.5	10.1	7392	0.92	3786	0.84	7516	0.91	22067	0.95
2.0	17.0	15326	0.90	8126	0.81	14577	0.90	41281	0.95

Here t/T is the number of periods; n, the number of markers; r, the radius of the initial spherical blob; V/V_0, the relative volume (ratio of the volume of deformed blob to the initial; s/s_0, the relative area (approximately equal in all cases). The shape of the initial blob was described with 258 markers.

[a] Without removal of too short edges.

and, necessarily, the precision of the velocity field, computations could shift the limit for such simulations to a later stage of deformation, but would result in significant computational expenses.

4. CONCLUSIONS

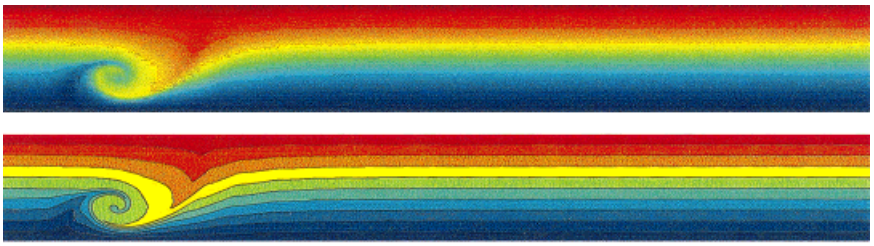
An adaptive front tracking technique for the tracking of strongly deforming fluid volumes is presented. The performance of the technique was demonstrated and examples of application of the technique to two- and three-dimensional flows are given. The distinctive features of this technique are a simple curvature-related criterion for deciding upon mesh refinement and the use of history of the surface deformation for inserting new nodal points. The adaptive nature of the described algorithm makes it suitable and efficient for complex flows. A typical example is the numerical simulation of the initial stages of (distributive) fluid mixing. As in this case the interfacial area typically grows exponentially, the usage of a non-adaptive approach (as done in other studies, e.g. [16,24]) for tracking such strongly deforming fluid volumes makes the adequate restoration of the surface shape very complicated or impossible, even in two-dimensional problems, and is rarely attempted in three-dimensional case. The fact that the surface is being stretched and bent very non-uniformly strongly favours the usage of the introduced adaptive technique, since a non-adaptive approach would lead to excessive computational expenses. The positive effect of mesh restructuring on the average quality of the mesh elements and number of required nodal points (which is directly linked to computational expenses) was demonstrated. A comparison of the adaptive front tracking technique with a passive scalar advection technique clearly shows that much finer details of the flow structure and the interfaces can be revealed. Moreover, an explicit description of the interface enables a straightforward algorithm to distinguish between interior and exterior of the closed surface, which is important for the simulation of multi-fluid flow. Future research involves the application of the adaptive technique to the mixing of rheologically different fluids. The presented algorithm allows a straightforward parallelization using a message passing paradigm [18] and the current results are obtained using such an approach.

ACKNOWLEDGMENTS

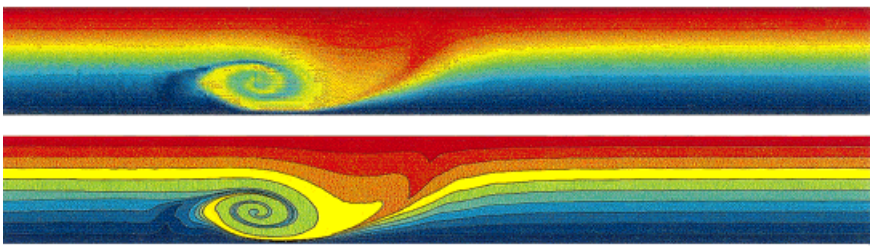
The authors would like to acknowledge support by the Dutch Foundation of Technology (STW), Grant No. EWT44.3453.

REFERENCES

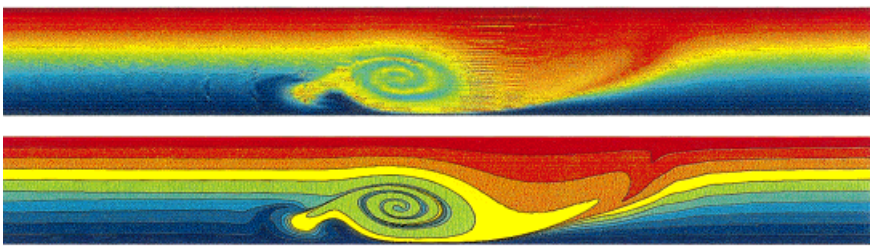
1. A.N. Beris, J.R. Richard and A.M. Lenhoff, 'A volume of fluid method applied to liquid–liquid jet break-up and drop dynamics', in Y.Y. Renardy and A.V. Coward (eds.), *Proceedings in Applied Mathematics*, SIAM, Philadelphia, PA, 1996, pp. 349–367.
2. J.M. Hyman, 'Numerical methods for tracking interfaces', *Physica D*, **12**, 396–407 (1984).
3. M. Rudman, 'Volume-tracking methods for interfacial flow calculations', *Int. J. Numer. Heat Fluid Flow*, **24**, 671–691 (1997).
4. S.O. Unverdi and G. Tryggvason, 'Computations of multi-fluid flows', *Phys. Fluids D*, **60**, 70–83 (1992).
5. W.F. Noh and P. Woodward, 'SLIC (simple line interface calculation)', in A.I. Dooren and P.J. Zandbergen (eds.), *Fifth International Conference on Numerical Methods in Fluid Dynamics*, vol. 59 of *Lecture Notes in Physics*, Springer, Berlin, 1976, pp. 330–340.
6. C.V. Hirt and B.D. Nichols, 'Volume of fluid (VOF) methods for the dynamics of free boundaries', *J. Comput. Phys.*, **39**, 201–225 (1981).
7. S.S. Young, J.J. White, E.S. Clark and Y. Oyanagi, 'A basic experimental study of sandwich injection moulding with sequential injection', *Pol. Eng. Sci.*, **20**, 798–804 (1980).
8. X. Qu and X. Li, 'A 3D surface tracking algorithm', *Comput. Vision Graphics Image Process.*, **64**, 147–156 (1996).
9. R.D. Richtmyer and K.W. Morton, *Difference Methods for Initial-Value Problems*, Inter-science, New York, 1967.
10. J. Glimm, J.W. Grove, X.L. Li, K.-M. Shyue, Q. Zhang and Y. Zeng, 'Three-dimensional front tracking', *SIAM J. Sci. Comput.*, **19**, 703–727 (1998).
11. J. Glimm, J.W. Grove, X.L. Li, R. Young, Q. Zhang and Y. Zeng, 'Front tracking: a parallel approach for internal boundaries and interfaces', in J. Dongarra, K. Masden and J. Wasniniewski (eds.), *Applied Parallel Computing*, vol. 1041 of *Lecture Notes in Computer Science*, Springer, Berlin, 1996, pp. 257–266.
12. G. Tryggvason and S.O. Unverdi, 'Computations of three-dimensional Rayleigh–Taylor instability', *Phys. Fluids A*, **5**, 656–659 (1990).
13. S.O. Unverdi and G. Tryggvason, 'A front-tracking method for viscous, incompressible multi-fluid flows', *J. Comput. Phys.*, **100**, 25–37 (1992).
14. J.M. Ottino, *The Kinematics of Mixing: Stretching, Chaos and Transport*, Cambridge Texts in Applied Mathematics, Cambridge University Press, Cambridge, 1989.
15. L.J.P. Timmermans, P.D. Minev and F.N. van de Vosse, 'An approximate projection scheme for incompressible flow using spectral elements', *Int. J. Numer. Methods Fluids*, **22**, 673–688 (1996).
16. G.F. Carey and Y. Chen, 'Simulation of fluid mixing using least-squares finite elements and particle tracing', *Int. J. Numer. Heat Fluid Flow*, **5**, 549–573 (1995).
17. W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannerly, *Numerical Recipes in Fortran*, 2nd edn., Cambridge University Press, Cambridge, 1992.
18. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderman, *PVM: Parallel Virtual Machine. A User's Guide and Tutorial for Networked Parallel Computing*, MIT Press, Cambridge, MA, 1994.
19. A.S. Galaktionov, P.D. Anderson and G.W.M. Peters, 'Mixing simulations: tracking strongly deforming fluid volumes in 3D flows', in M. Bubak, J. Dongarra and J. Wasniniewski (eds.), *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, vol. 1332 of *Lecture Notes in Computer Science*, Springer, Berlin, 1997, pp. 463–469.
20. F.N. van de Vosse, P.D. Minev and L.J.P. Timmermans, 'A spectral element projection scheme for incompressible flow with application to shear-layer stability studies', *Proc. 3rd Int. Conf. on Spectral and High Order Methods*, 1995, pp. 295–303.
21. P.D. Minev, F.N. van de Vosse, L.J.P. Timmermans and A.A. van Steenhoven, 'A second-order splitting algorithm for thermally-driven flow problems', *Int. J. Numer. Heat Fluid Flow*, **6**, 51–60 (1995).
22. C.W. Leong and J.M. Ottino, 'Experiments on mixing due to chaotic advection in a cavity', *J. Fluid Mech.*, **209**, 463–499 (1989).
23. P.D. Anderson, O.S. Galaktionov, G.W.M. Peters, F.N. van de Vosse and H.E.H. Meijer, 'Analysis of mixing in three-dimensional time-periodic cavity flows', *J. Fluid Mech.*, **386**, 149–166 (1999).
24. T. Nishimura and K. Kunitsuga, 'Fluid mixing and mass transfer in two-dimensional cavities with time-periodic lid velocity', *Int. J. Heat Fluid Flow*, **18**, 497–506 (1997).



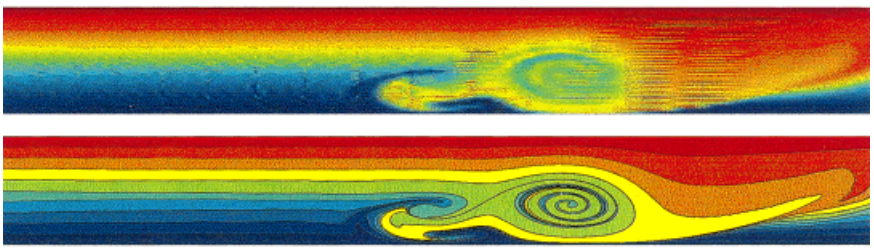
(a) $t = 2.0$



(b) $t = 4.0$



(c) $t = 6.0$



(d) $t = 8.0$

Plate 1. Mixing patterns in shear layer flow, shown at time levels $t = 2$, 4, 6 and 8. Upper plot in each pair represents the result of passive scalar advection technique; the lower plot represents the results of adaptive contour tracking.