Computation of flow problems with the Mixed Interface-Tracking/Interface-Capturing Technique (MITICT)

J. Ed Akin *, Tayfun E. Tezduyar, Mehmet Ungor

Mechanical Engineering, Rice University, MS 321, 6100 Main Street, Houston, TX 77005, USA

Received 8 July 2005; accepted 19 July 2005

Abstract

In computation of flow problems with fluid–solid interfaces, an interface-tracking technique, where the fluid mesh moves to track the interface, would allow us to have full control of the resolution of the fluid mesh in the boundary layers. With an interface-capturing technique (or an interface locator technique in the more general case), on the other hand, independent of how accurately the interface geometry is represented, the resolution of the fluid mesh in the boundary layer will be limited by the resolution of the fluid mesh at the interface. In computation of flow problems with fluid–fluid interfaces where the interface is too complex or unsteady to track while keeping the remeshing frequency under control, interface-capturing techniques, with enhanced-discretization as needed, could be used as more flexible alternatives. Sometimes we may need to solve flow problems with both fluid–solid interfaces and complex or unsteady fluid–fluid interfaces. The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) was introduced for computation of flow problems that involve both interfaces that can be accurately tracked with a moving mesh method and interfaces that are too complex or unsteady to be tracked and therefore require an interface-capturing technique. As the interface-tracking technique, we use the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation. The interface-capturing technique rides on this, and is based on solving over a moving mesh, in addition to the Navier–Stokes equations, the advection equation governing the time-evolution of the interface function. For the computations reported in this paper, as interface-capturing technique we are using one of the versions of the Edge-Tracked Interface Locator Technique (ETILT).

© 2005 Elsevier Ltd. All rights reserved.

1. Introduction

In computation of flow problems involving moving boundaries and interfaces, depending on the nature of the problem, one typically uses either an interface-tracking or interface-capturing technique. An interface-tracking technique requires meshes that move to “track” the interfaces. In an interface-capturing technique for two-fluid flows, the computations are based on fixed spatial domains, where an interface function, marking the location of the interface, needs to be computed to “capture” the interface. The interface is captured within the resolution of the finite element mesh covering the area where the interface is.

The Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation [1–4] is an interface-tracking technique, where the finite element formulation of the problem is written over its space–time domain. As the spatial domain occupied by the fluid changes its shape in time, the mesh needs to be updated. In general, we accomplish this with an automatic mesh-moving method [5,4]. The motion of the nodes is governed by the equations of elasticity, with full or partial remeshing (i.e., generating a new set of elements, and sometimes also a new set of nodes) as needed. The stabilized space–time formulations were used earlier by other researchers to solve problems with fixed spatial domains (see for example [6]). The DSD/SST formulation is
based on stabilized formulations. The stabilized methods are the streamline-upwind/Petrov–Galerkin (SUPG) [7–10] and pressure-stabilizing/Petrov–Galerkin (PSPG) [1,11] formulations. An earlier version of the pressure-stabilizing formulation for Stokes flows was reported in [12].

In computation of two-fluid flows with interface-tracking techniques, sometimes the interface might be too complex or unsteady to track while keeping the frequency of remeshing at an acceptable level. In such cases, interface-capturing techniques, which do not normally require costly mesh update steps, could be used with the understanding that the interface will not be represented as accurately as we would have with an interface-tracking technique. Because they do not require mesh update, the interface-capturing techniques are more flexible than the interface-tracking techniques. However, for comparable levels of spatial discretization, interface-capturing methods yield less accurate representation of the interface. These methods can be used as practical alternatives in carrying out the simulations when compromising the accurate representation of the interfaces becomes less of a concern than facing major difficulties in updating the mesh to track such interfaces.

The Edge-Tracking Interface Locator Technique (ETILT) [4,13–16] was introduced to have an interface-capturing technique with better volume conservation properties and sharper representation of the interfaces. In the ETILT, the underlying method is a basic stabilized finite element interface-capturing technique, where the Navier–Stokes equations of incompressible flows are solved over a non-moving mesh, together with the advection equation governing the evolution of the interface function. In addition to the nodal representation of the interface function, the ETILT involves an edge-based representation, with the interfaces represented as collection of positions along element edges crossed by those interfaces. While the advection equation is solved for the node-based representation of the interface function, the volume conservation is enforced using the edge-based representation. The edge-based representation is reconstructed from the edge-based representation after every correction for the volume conservation. A number of 2D test computations with the ETILT were reported in [17,18].

In computation of flow problems with fluid–solid interfaces, an interface-tracking technique, where the fluid mesh moves to track the interface, would allow us to have full control of the resolution of the fluid mesh in the boundary layers. With an interface-capturing technique (or an interface locator technique in the more general case), on the other hand, independent of how accurately the interface is located, the resolution of the fluid mesh in the boundary layer will be limited by the resolution of the fluid mesh where the interface is. Then again, in computation of flow problems with fluid–fluid interfaces where the interface is too complex or unsteady to track by a mesh-moving method, using an interface-capturing method becomes rather inevitable. But sometimes we may be faced with solving flow problems with both fluid–solid interfaces and complex or unsteady fluid–fluid interfaces.

The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) [4,13–16] was introduced primarily for fluid–object interactions with multiple fluids. The class of applications targeted were fluid–particle–gas interactions and free-surface flow of fluid–particle mixtures. However, the MITICT can be applied to a larger class of problems, where it is more effective to use an interface-tracking technique to track the solid–fluid interfaces and an interface-capturing technique to capture the fluid–fluid interfaces. The interface-tracking technique used in MITICT is the DSD/SST formulation but could as well be the Arbitrary Lagrangian–Eulerian method or other moving-mesh methods. The interface-capturing technique rides on this, and is based on solving over a moving mesh, in addition to the Navier–Stokes equations, the advection equation governing the time-evolution of the interface function.

The governing equations are described in Section 2. The semi-discrete stabilized finite element formulation of the Navier–Stokes equations is given in Section 3. The interface-tracking and interface-capturing techniques are described in Sections 4 and 5. The MITICT is highlighted in Section 6. Numerical examples are presented in Section 7 and the concluding remarks in Section 8.

2. Governing equations

2.1. Navier–Stokes equations of incompressible flows

Let \( \Omega, \subset \mathbb{R}^{3d} \) be the spatial flow domain with boundary \( \Gamma_t \) at time \( t \in (0, T) \), where \( n_{sd} \) is the number of space dimensions. The subscript \( t \) indicates the time-dependence of the domain. The Navier–Stokes equations of incompressible flows are written on \( \Omega_t \) and \( \forall t \in (0, T) \) as

\[
\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{u} \mathbf{u} - \mathbf{f} = \mathbf{0},
\]

\[
\nabla \cdot \mathbf{u} = 0,
\]

where \( \rho, \mathbf{u} \) and \( \mathbf{f} \) are the density, velocity and the external force, respectively. The stress tensor \( \sigma \) is defined as

\[
\sigma(p, \mathbf{u}) = -p \mathbf{I} + 2\mu \mathbf{u}, \quad \varepsilon(\mathbf{u}) = \frac{1}{2} \left( (\nabla \mathbf{u}) + (\nabla \mathbf{u})^T \right).
\]

Here \( p \) is the pressure, \( \mathbf{I} \) is the identity tensor, \( \mu = \rho \nu \) is the viscosity, \( \nu \) is the kinematic viscosity, and \( \varepsilon(\mathbf{u}) \) is the...
strain-rate tensor. The essential and natural boundary conditions for Eq. (1) are represented as
\[ u = g \quad \text{on } (T_i)_g, \quad \mathbf{n} \cdot \sigma = h \quad \text{on } (T_i)_h, \tag{4} \]
where \((T_i)_g\) and \((T_i)_h\) are complementary subsets of the boundary \(T_i\), \(\mathbf{n}\) is the unit normal vector, and \(g\) and \(h\) are given functions. A divergence-free velocity field \(u_d(x)\) is specified as the initial condition.

### 2.2. Advection equation of interface motion

Modeling of flows with moving boundaries and interfaces is not always based on defining the spatial domain associated with each fluid to be the part of the space occupied by that fluid. For example, fluid-fluid interfaces can also be modeled by using Eqs. (1) and (2) over a fixed spatial domain \(\Omega\) by assuming that the domain is occupied by two immiscible fluids, A and B, with densities \(\rho_A\) and \(\rho_B\) and viscosities \(\mu_A\) and \(\mu_B\). A free-surface problem can be modeled as a special case where Fluid A is irrelevant and assigned a sufficiently low density. An initial condition.

### 3. Stabilized semi-discrete formulation

Solution of Eqs. (1) and (2) over a fixed spatial domain \(\Omega\) is accomplished with a stabilized semi-discrete finite element formulation. For that, we form our suitably-defined finite-dimensional trial solution and test function spaces for velocity and pressure: \(S_u^h\), \(V_u^h\), \(S_p^h\) and \(V_p^h\). The stabilized finite element formulation of Eqs. (1) and (2) can be written as follows: find \(u^h \in S_u^h\) and \(p^h \in S_p^h\) such that \(\forall \mathbf{w}^h \in V_u^h\) and \(\forall q^h \in V_p^h\):

\[
\int_{\Omega} \mathbf{w}^h \cdot \rho \left( \frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \mathbf{V} \mathbf{u}^h - f^h \right) \, d\Omega + \int_{\Omega} \mathbf{w}^h \cdot \mathbf{V} \mathbf{u}^h \, d\Omega - \int_{T_h} \mathbf{w}^h \cdot \mathbf{h} \, d\Gamma + \int_{\Omega} q^h \cdot \mathbf{V} \cdot \mathbf{u}^h \, d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \frac{1}{\rho} \left[ \tau_{\text{SUPG}} \mathbf{u}^h \cdot \mathbf{V} \mathbf{w}^h + \tau_{\text{PSPG}} \mathbf{V} q^h \right] \cdot \left[ L(p^h, \mathbf{u}^h) - \rho f^h \right] \, d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \gamma_{\text{LSIC}} \mathbf{V} \cdot \mathbf{w}^h \rho \mathbf{V} \cdot \mathbf{u}^h \, d\Omega = 0, \tag{6} \]

where

\[
L(q^h, \mathbf{w}^h) = \rho \left( \frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \mathbf{V} \mathbf{w}^h - \mathbf{f}^h \right) - \mathbf{V} \cdot \sigma(q^h, \mathbf{w}^h). \tag{7} \]

Here \(n_{el}\) is the number of elements, \(\Omega\) is the domain for element \(e\), and \(\tau_{\text{SUPG}}, \tau_{\text{PSPG}}\) and \(\gamma_{\text{LSIC}}\) are the SUPG, PSPG and LSIC (least-squares on incompressibility constraint) stabilization parameters. For various ways of calculating \(\tau_{\text{SUPG}}, \tau_{\text{PSPG}}\) and \(\gamma_{\text{LSIC}},\) see [19,20,14].

### 4. Interface-tracking technique: Deforming-Spatial-Domain/Stabilized Space–Time (DS/SST) formulation

In the DS/SST method [1–3], the finite element formulation of Eqs. (1) and (2) is written over a sequence of \(N\) space–time slabs \(Q_n\), where \(Q_n\) is the slice of the space–time domain between the time levels \(t_n\) and \(t_{n+1}\). At each time step the integrations are performed over \(Q_n\). The space–time finite element interpolation functions are continuous within a space–time slab, but discontinuous from one space–time slab to another. The notation \(\left( \cdot \right)_n\) and \(\left( \cdot \right)_{\frac{n}{2}}\) denotes the function values at \(t_n\) as approached from below and above. Each \(Q_n\) is decomposed into elements \(Q_e\), where \(e = 1,2,\ldots,n_{el}\). The subscript \(n\) used with \(n_{el}\) is for the general case in which the number of space–time elements may change from one space–time slab to another. The essential and natural boundary conditions are enforced over \((P_n)_g\) and \((P_n)_h\), the complementary subsets of the lateral boundary of the space–time slab. The finite element trial function spaces \((S_u^h)_n\) for velocity and \((S_p^h)_n\) for pressure, and the test function spaces \((V_u^h)_n\) and \((V_p^h)_n\) are defined by using, over \(Q_n\), first-order polynomials in both space and time. The DS/SST formulation is written as follows: given \((\mathbf{u}^h)_n\), find \(\mathbf{u}^h \in (S_u^h)_n\) and \(p^h \in (S_p^h)_n\) such that \(\forall \mathbf{w}^h \in (V_u^h)_n\) and \(\forall q^h \in (V_p^h)_n\):

\[
\int_{Q_n} \mathbf{w}^h \cdot \rho \left( \frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \mathbf{V} \mathbf{u}^h - \mathbf{f}^h \right) \, dQ + \int_{Q_n} \mathbf{w}^h \cdot \mathbf{V} \mathbf{u}^h \, dP + \int_{Q_n} q^h \mathbf{V} \cdot \mathbf{u}^h \, dQ + \int_{Q_n} (\mathbf{w}^h)_n \cdot \rho (\mathbf{u}^h)_n - (\mathbf{u}^h)_n \, d\Omega + \sum_{e=1}^{n_{el}} \int_{Q_e} \frac{1}{\rho} \left[ \tau_{\text{SUPG}} \mathbf{u}^h \cdot \mathbf{V} \mathbf{w}^h + \tau_{\text{PSPG}} \mathbf{V} q^h \right] \cdot \left[ L(p^h, \mathbf{u}^h) - \rho f^h \right] \, dQ + \sum_{e=1}^{n_{el}} \int_{Q_e} \gamma_{\text{LSIC}} \mathbf{V} \cdot \mathbf{w}^h \rho \mathbf{V} \cdot \mathbf{u}^h \, dQ = 0. \tag{8} \]
This formulation is applied to all space–time slabs $Q_0, Q_1, Q_2, \ldots, Q_{N-1}$, starting with $(u^h)^0 = u_0$. For an earlier, detailed reference on the formulation see [1–3].

How the mesh is updated as the spatial domain occupied by the fluid changes in time depends on several factors. These factors include the complexity of the interface and overall geometry, how unsteady the interface is, and how the starting mesh was generated. Detailed descriptions of various mesh update techniques developed in conjunction with the DSD/SST formulation can be found in [4,14]. These include an automatic mesh-moving method, techniques to reduce the frequency of remeshing, and a mesh update method for handling solid objects (or surfaces) in fast linear or rotational relative motion. Also included are the techniques for handling the structured layers of elements generated around solid or deformable solid objects (to fully control the mesh resolution near solid objects and have more accurate representation of the boundary layers).

5. Interface-capturing technique: Edge-Tracking Interface Locator Technique (ETILT)

In the basic stabilized finite element interface-capturing technique, the semi-discrete formulation given by Eq. (6) is complemented with a stabilized formulation of Eq. (5). For that, let us first assume that we have constructed some suitably-defined finite-dimensional trial solution and test function spaces $V_h^1$ and $V_h^0$. The stabilized finite element formulation of Eq. (5) can then be written as follows: find $\phi^h \in V_h^1$, such that $\forall w^h \in V_h^0$:

$$
\int_{\Omega} w^h \left( \frac{\partial \phi^h}{\partial t} + u \cdot \nabla \phi^h \right) \, d\Omega + \sum_{i=1}^{n_i} \int_{\Gamma} T_{\text{SUPG}} u^h \cdot \nabla w^h \, d\Gamma = 0. \quad (9)
$$

The ETILT [4,13–16] was introduced to design an interface-capturing technique with better volume conservation properties and sharper representation of the interfaces. This is based on first defining a second finite-dimensional representation of the interface function, namely $\phi^{he}$. The added superscript ‘e’ indicates that this is an edge-based representation. With $\phi^{he}$, interfaces are represented as collection of positions along element edges crossed by the interfaces (i.e., along the “interface edges”). Nodes belong to “chunks” of Fluid A or Fluid B. An edge either belongs to a chunk of Fluid A or Fluid B or is an interface edge. Each element is either filled fully by a chunk of Fluid A or Fluid B, or is shared by a chunk of Fluid A and a chunk of Fluid B. If an element is shared like that, the shares are determined by the position of the interface along the edges of that element.

At each time step, given $u^n$ and $\phi^{he}_n$, we determine $u^{n+1}$, $p^{n+1}$, and $\phi^{he}_{n+1}$. The definitions of $\rho$ and $\mu$ are modified to use the edge-based representation of the interface function: $\rho^e = \phi^{he} \rho_A + (1 - \phi^{he}) \rho_B$, $\mu^e = \phi^{he} \mu_A + (1 - \phi^{he}) \mu_B$. In marching from time level $n$ to $n+1$, we first calculate $\phi^h$ from $\phi^{he}$ by a least-squares projection:

$$
\int_{\Omega} \psi_h \left( \phi^h_n - \phi^{he}_n \right) \, d\Omega = 0.
$$

To calculate $\phi^{he}_{n+1}$, we use Eq. (9). From $\phi^h_{n+1}$, we calculate $\phi^{he}_{n+1}$ by a combination of a least-squares projection:

$$
\int_{\Omega} \left( \phi^{he}_{n+1} \right)_n \left( \phi^h_{n+1} - \phi^{he}_{n+1} \right) \, d\Omega = 0,
$$

and corrections to enforce volume conservation for all chunks of Fluid A and Fluid B, taking into account the mergers between the chunks and the split of chunks. This volume conservation condition can symbolically be written as $\text{VOL}(\phi^{he}_{n+1}) = \text{VOL}(\phi^{he}_n)$. Here the subscript $P$...
is used for representing the intermediate values following the projection, but prior to the corrections for volume conservation. It can be shown that the projection given by Eq. (11) can be interpreted as locating the interface along the interface edges at positions where $\phi^{n+1}_n = 1/2$.

As an alternative way for computing $\phi^{n+1}_n$ from $\phi^{he}_n$, it was proposed in [14–16] to solve the equation

$$\int_{\Delta\Omega} \left( \psi^{b}_n \phi^{n}_n - \phi^{he}_n \right) d\Omega = 0,$$

$$+ \sum_{k=1}^{nE} \psi^{b}_n(x_k) \lambda_{PEN} (\phi^{b}_n(x_k) - 1/2) = 0,$$

(12)

where $nE$ is the number of interface edges, $x_k$ is the coordinate of the interface location along the $k$th interface edge, $\lambda_{PEN}$ is a penalty parameter, and $\Delta\Omega$ is the solution domain. This domain is the union of all the elements containing at least one node where the value of $\phi^{n}_n$ is unknown. We can assume $\phi^{h}_n$ to be unknown only at the nodes of the interface edges, with known values $\phi^{h}_n = 1$ (for Fluid A) and $\phi^{h}_n = 0$ (for Fluid B) at all other nodes. We can also augment the number of nodes where $\phi^{h}_n$ is unknown and thus enlarge the solution domain. This can be done all the way to the point where $\Omega_{\Delta\Omega} = \Omega$. As another alternative proposed in [14–16], in Eq. (12) the least-squares projection term can replaced with a slope-minimization term:

$$\int_{\Delta\Omega} \nabla \phi^{b}_n \cdot \nabla \phi^{b}_n d\Omega = 0 + \sum_{k=1}^{nE} \psi^{b}_n(x_k) \lambda_{PEN} (\phi^{b}_n(x_k) - 1/2) = 0.$$  

(13)

A 1D version of the way of computing $\phi^{b}_n$ from $\phi^{he}_n$ was proposed in [14–16] by minimizing $(\phi^{h}_n - \phi^{he}_n)^2$ along “chains” of interface edges:

$$\int_{S_{\Delta\Omega}} \psi^{b}_n (\phi^{n}_n - \phi^{he}_n) ds = 0,$$

where $S_{\Delta\Omega}$ is the collection of all chains of interface edges, and $s$ is the integration coordinate along the interface edges. This is, of course, a simpler formulation, and
much of the equations for the unknown nodal values will be uncoupled.

These projections and volume corrections are embedded in the iterative solution technique, and are carried out at each iteration. The iterative solution technique, which is based on the Newton–Raphson method, addresses both the nonlinear and coupled nature of the set of equations that need to be solved at each time step. More explanation of how the projections and volume corrections would be handled at a nonlinear iteration step can be found in [4,14].

6. Mixed Interface-Tracking/Interface-Capturing Technique (MITICT)

Solution of Eq. (5) over a moving mesh can be accomplished by applying the DSD/SST formulation
to that equation. For that, first we assume that we have constructed some suitably-defined finite-dimensional trial solution and test function spaces \((S_h^0)^n\) and \((V_h^0)^n\). The DSD/SST formulation of Eq. (5) can then be written as follows: given \((\phi^h)^n\), find \(\phi^h\) in \((S_h^0)^n\) such that \(\forall w^h \in (V_h^0)^n\):

\[
\int_{Q^n} w^h \left( \frac{\partial \phi^h}{\partial t} + u \cdot \nabla \phi^h \right) dQ
+ \int_{\Omega^n} (w^n)^+ \left( (\phi^h)^+ - (\phi^h)^- \right) d\Omega
\]
This equation, together with Eq. (8), constitute a mixed interface-tracking/interface-capturing technique that would track the solid–fluid interfaces and capture the fluid–fluid interfaces that would be too complex or unsteady to track with a moving mesh.

7. Numerical examples

A number of 2D and 3D test computations with the ETILT and MITICT were carried out in [21], all using a 3D solver, and with linear tetrahedral or trilinear hexahedral elements. Here we present two of those test problems: first a 3D problem to evaluate the performance of the ETILT as an interface-capturing technique, and then a 2D problem to show how the MITICT works in computation of fluid–object interactions with free surfaces. In both problems, as an interface-capturing technique, we test three of the methods described in Section 5: the basic stabilized finite element interface-capturing technique, and the versions of the ETILT given by Eqs. (12) and (13). Here, the version of the ETILT given by Eq. (12) will be called ETILT-LSP, and the version given Eq. (13) ETILT-SMP. At every nonlinear iteration of each time step, the linear equation systems involved are solved with an iterative technique with diagonal preconditioner and GMRES update method [22]. Computations are carried in a parallel computing environment, using a cluster of PCs.

7.1. Collapse of a cylindrical water column

With this test problem we evaluate the performance of the ETILT as an interface-capturing technique. The initial radius of the water column is 5.715 cm, and the height is twice that. As density and viscosity values, we use $1.0 \times 10^3$ kg/m$^3$ and $1.0 \times 10^{-3}$ kg/m/s for the water, and $1.0$ kg/m$^3$ and $1.0 \times 10^{-5}$ kg/m/s for the surrounding fluid. We compute with one quadrant of the water column in a computational domain with all three dimensions equal to 4 times the radius of the water column. We use a uniform mesh with $39 \times 39 \times 39$ hexahedral elements. The time-step size is $0.005$ s.

Fig. 1 shows the time-evolution of the water column, computed with the ETILT-LSP. Figs. 2 and 3 show the time-histories of the water front location and column height, obtained with the basic stabilized interface-capturing technique, ETILT-SMP and ETILT-LSP. Fig. 2 also shows the experimental data reported in [23]. Fig. 4 shows the time-history of the error in volume conservation for the water. The ETILT-LSP and ETILT-SMP both show superior volume conservation performance compared to the basic stabilized interface-capturing technique.

7.2. Fluid–object interactions with an oscillating cylinder and free surface

With this 2D test problem we show how the MITICT works in computation of fluid–object interactions involving a vertically-oscillating cylinder and a free surface. The cylinder has a diameter of 1.0 m and an initial vertical position of 5.0 m. The water surface level is at 4.0 m. The downward displacement of the cylinder is given as $(1 - \cos \frac{\pi t}{T})$ m, where $T = 2$ s. As density and viscosity values, we use $1.0 \times 10^3$ kg/m$^3$ and $1.788 \times 10^{-3}$ kg/m/s for the water, and $1.229$ kg/m$^3$ and $1.71 \times 10^{-5}$ kg/m/s for the surrounding fluid.

We compute this 2D problem with a 3D solver, with 2D domain dimensions 10.0 m $\times$ 10.0 m. We use a mesh with 4800 hexahedral elements in the 2D domain and with one element in the third dimension. The time-step size is $0.0125$ s.

Figs. 5 and 6 show the time-history of the cylinder–water interactions computed with the MITICT, where the interface-capturing techniques are: the basic stabilized interface-capturing technique, ETILT-LSP and ETILT-SMP. Fig. 7 shows the time-history of the error in volume conservation for the water. Again, the ETILT-LSP and ETILT-SMP both show superior volume conservation performance compared to the basic stabilized interface-capturing technique.
8. Concluding remarks

The MITICT combines the desirable features of the interface-tracking and interface-capturing techniques in computation of flow problems involving both fluid–solid and fluid–fluid interfaces. In the MITICT, the overriding method is an interface-tracking technique in computation of the fluid–solid interfaces and an interface-capturing technique in computation of the fluid–fluid interfaces. An interface-tracking technique, where the fluid mesh moves to track the interface, allows us to have full control of the resolution of the fluid mesh in the boundary layers of the solid surface. This results in better accuracy compared to using an interface-capturing technique. This is because, independent of how accurately it can represent the geometry of the fluid–solid interface, the interface-capturing technique will be disadvantaged by the fact that the resolution of the fluid mesh in the boundary layer will be limited by the resolution of the fluid mesh at the interface. An interface-capturing technique, where the interfaces are computed over non-moving meshes, on the other hand, allows us to carry out the computations for the fluid–fluid interfaces even when they are too complex or unsteady for a mesh-moving method. The interface-tracking and interface-capturing techniques used in this paper are the DSD/SST formulation and the ETILT. The DSD/SST formulation, which was applied over the years to a wide class of flow problems involving moving boundaries and interfaces, has a proven record of good accuracy at the interfaces. The ETILT, as it was demonstrated with the test problems reported in this paper, has good volume conservation properties. As it was also demonstrated in this paper, combining these two techniques, which are well-rated in their own categories, in the framework of the MITICT gives us a powerful method with enhanced scope and accuracy.

References
