COMPUTATIONAL FLUID AND SOLID MECHANICS

K.J. Bathe, editor

Proceedings
First MIT Conference on
Computational Fluid and Solid Mechanics
June 12-15, 2001

Elsevier

Interface-tracking and interface-capturing techniques for computation of two-fluid flows

Tayfun E. Tezduyar *

Team for Advanced Flow Simulation and Modeling [1], Mechanical Engineering and Materials Science, Rice University – MS 321, 6100 Main Street, Houston, TX 77005, USA

Abstract

We describe the interface-tracking and interface-capturing methods developed for computation of flow problems with two-fluid interfaces. The interface-tracking methods are based on the Deforming-Spatial-Domain/Stabilized Space-Time formulation, where the mesh moves to track the interface. The interface-capturing methods are based on the stabilized formulation, over non-moving meshes, of both the flow equations and the advection equation governing the time-evolution of an interface function marking the location of the interface. In this category, when it becomes necessary to increase the accuracy in representing the interface, the Enhanced-Discretization Interface-Capturing Technique (EDICT) can be used to accomplish that goal. We also describe some of the additional ideas and methods developed to increase the scope and accuracy of these two classes of methods.

Keywords: Two-fluid flows; Interface-tracking; Interface-capturing; EDICT; ETILT; MITICT

1. Introduction

In computation of flow problems with two-fluid interfaces, depending on the nature of the problem, we can use an interface-tracking or interface-capturing method. An interface-tracking method requires meshes that "track" the interfaces. The mesh needs to be updated as the flow evolves. In an interface-capturing method, 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] is an interface-tracking method, where the finite element formulation of the problem is written over its space-time domain. This automatically takes into account the motion of the interfaces. At each time step the locations of the interfaces are calculated as part of the overall solution.

* Tel.: +1 (713) 348-6051; Fax: +1 (713) 348-5423; E-mail: tezduyar@rice.edu

[1] http://www.mems.rice.edu/TAFSM/

Computational Fluid and Solid Mechanics
K.J. Bathe (Editor)

2. DSD/SST formulation

In the DSD/SST method, the finite element formulation is written over a sequence of N space-time slabs Q_n, where Q_n is the slice of the space-time domain between t_n and t_{n+1}. At each time step, the integrations involved in the finite element formulation are performed over Q_n. The interpolation functions are discontinuous across the space-time slabs. We use first-order polynomials. The notation $(\cdot)_n^-$ and $(\cdot)_n^+$ denotes the function values at t_n as approached from below and above respectively. Each Q_n is decomposed into space-time elements Q_n^e, where $e = 1, 2, \dots, (n_{el})_n$. The subscript n used with n_{el} is to account for the general case in which the number of elements may change from one slab to another.

For incompressible flows, the trial function spaces for velocity and pressure will be $(\hat{\mathcal{S}}_u^h)_n$ and $(\hat{\mathcal{S}}_p^h)_n$. The weighting function spaces for momentum equation and incompressibility constraint will be $(\hat{\mathcal{V}}_u^h)_n$ and $(\hat{\mathcal{V}}_p^h)_n$ $(= (\hat{\mathcal{S}}_p^h)_n)$. The formulation can then be written as follows: given $(u^h)_n^-$, find $u^h \in (\hat{\mathcal{S}}_u^h)_n$ and $p^h \in (\hat{\mathcal{S}}_p^h)_n$ such that $\forall w^h \in (\hat{\mathcal{V}}_u^h)_n$ and $\forall q^h \in (\hat{\mathcal{V}}_p^h)_n$:

$$\int_{Q_n} w^h \cdot \rho \left(\frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(w^h) : \boldsymbol{\sigma}(p^h, u^h)\,\mathrm{d}Q + \int_{Q_n} q^h \nabla \cdot u^h\,\mathrm{d}Q + \int_{\Omega_n} (w^h)_n^+ \cdot \rho \left((u^h)_n^+ - (u^h)_n^- \right) \mathrm{d}\Omega + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{LSME}} \frac{1}{\rho} \left[\rho \left(\frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h \right) - \nabla \cdot \boldsymbol{\sigma}(q^h, w^h) \right] \cdot \left[\rho \left(\frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) - \nabla \cdot \boldsymbol{\sigma}(p^h, u^h) \right] \mathrm{d}Q + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{LSIC}} \nabla \cdot w^h \rho \nabla \cdot u^h\,\mathrm{d}Q = \int_{P_n} w^h \cdot h^h\,\mathrm{d}P. \quad (1)$$

Here h^h represents the Neumann-type boundary condition associated with the momentum equation, and τ_{LSME} and τ_{LSIC} are the stabilization parameters. The solution to Eq. (1) is obtained sequentially for $Q_0, Q_1, Q_2, \ldots, Q_{N-1}$, and the computations start with $(u^h)_0^- = u_0^h$.

With the DSD/SST formulation an automatic mesh moving technique is used for updating the mesh as the spatial domain occupied by the fluid changes its shape. This method is based on moving the nodal points as governed by the equations of linear elasticity.

3. EDICT

The EDICT was first introduced in [2] to increase accuracy in representing an interface. We start with an interface-capturing technique such as the volume of fluid method [3]. The Navier–Stokes equations are solved over a non-moving mesh with an interface function ϕ serving as a marker identifying the two fluids. The evolution of ϕ is governed by a time-dependent advection equation. The trial function spaces for velocity, pressure and interface function are $(\mathcal{S}_u^h)_n$, $(\mathcal{S}_p^h)_n$, and $(\mathcal{S}_\phi^h)_n$. The weighting function spaces for the momentum equation, incompressibility constraint and advection equation are $(\mathcal{V}_u^h)_n$, $(\mathcal{V}_p^h)_n$ $(= (\mathcal{S}_p^h)_n)$, and $(\mathcal{V}_\phi^h)_n$.

The stabilized formulations of the flow and advection equations can be written as follows: given u_n^h and ϕ_n^h, find $u_{n+1}^h \in (\mathcal{S}_u^h)_{n+1}$, $p_{n+1}^h \in (\mathcal{S}_p^h)_{n+1}$, and $\phi_{n+1}^h \in (\mathcal{S}_\phi^h)_{n+1}$, such that, $\forall w_{n+1}^h \in (\mathcal{V}_u^h)_{n+1}$, $\forall q_{n+1}^h \in (\mathcal{V}_p^h)_{n+1}$, and $\forall \psi_{n+1}^h \in (\mathcal{V}_\phi^h)_{n+1}$:

$$\int_{\Omega} w_{n+1}^h \cdot \rho \left(\frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) \mathrm{d}\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(w_{n+1}^h) : \boldsymbol{\sigma}(p^h, u^h)\,\mathrm{d}\Omega + \int_{\Omega} q_{n+1}^h \nabla \cdot u^h\,\mathrm{d}\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \left(\tau_{\mathrm{SUPG}}\, u^h \cdot \nabla w_{n+1}^h + \frac{\tau_{\mathrm{PSPG}}}{\rho} \nabla q_{n+1}^h \right) \cdot \left[\rho \left(\frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h - f \right) - \nabla \cdot \boldsymbol{\sigma}(p^h, u^h) \right] \mathrm{d}\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_{\mathrm{LSIC}} \nabla \cdot w_{n+1}^h \rho \nabla \cdot u^h\,\mathrm{d}\Omega = \int_{\Gamma_h} w_{n+1}^h \cdot h^h\,\mathrm{d}\Gamma, \quad (2)$$

$$\int_{\Omega} \psi_{n+1}^h \left(\frac{\partial \phi^h}{\partial t} + u^h \cdot \nabla \phi^h \right) \mathrm{d}\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \tau_\phi u^h \cdot \nabla \psi_{n+1}^h \left(\frac{\partial \phi^h}{\partial t} + u^h \cdot \nabla \phi^h \right) \mathrm{d}\Omega = 0, \quad (3)$$

where τ_{SUPG}, τ_{PSPG}, τ_{LSIC} and τ_ϕ are the stabilization parameters.

To increase the accuracy, we use function spaces corresponding to enhanced discretization at and near the interface. A subset of the elements in the base mesh, Mesh-1, are identified as those at and near the interface. A more refined mesh, Mesh-2, is constructed by patching together second-level meshes generated over each element in this subset. The interpolation functions for velocity and pressure will all have two components each: one coming from Mesh-1 and the second one coming from Mesh-2. To further increase the accuracy, we construct a third-level mesh, Mesh-3, for the interface function only. The construction of Mesh-3 from Mesh-2 is very similar to the construction of Mesh-2 from Mesh-1. The interpolation functions for the interface function will have three components, each coming from one of these three meshes. We re-define the subsets over which we build Mesh-2 and Mesh-3 not every time step but with sufficient frequency to keep the interface enveloped in.

4. MITICT

The Mixed Interface-Tracking/Interface-Capturing Technique (MITICT) was introduced in [4], primarily for computation of fluid-object interactions with multiple fluids. In particular, the class of application we are targeting are fluid-particle-gas interaction problems 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 is the 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. The additional DSD/SST formulation is for this advection equation:

$$\int_{Q_n} \psi^h \left(\frac{\partial \phi^h}{\partial t} + u^h \cdot \nabla \phi^h \right) \mathrm{d}Q + \int_{\Omega_n} (\psi^h)_n^+ \left((\phi^h)_n^+ - (\phi^h)_n^- \right) \mathrm{d}\Omega + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_\phi \left(\frac{\partial \psi^h}{\partial t} + u^h \cdot \nabla \psi^h \right) \left(\frac{\partial \phi^h}{\partial t} + u^h \cdot \nabla \phi^h \right) \mathrm{d}Q = 0. \quad (4)$$

This equation, together with Eq. (1), 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.

5. ETILT

The Edge-Tracked Interface Locator Technique (ETILT) was introduced in [4], to have an interface-capturing technique with better volume conservation properties and sharper representation of the interfaces. To this end, we first define a second finite-dimensional representation of the interface function, namely ϕ^{he}. With ϕ^{he}, interfaces are represented as collection of positions along element edges crossed by the interfaces. 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. The base finite element formulation is essentially the one described by Eqs. (2) and (3). Although the ETILT can be used in combination with the EDICT, we assume that we are working here with the plain, non-EDICT versions of Eqs. (2) and (3).

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

$$\int_{\Omega} \psi^h \left(\phi_n^h - \phi_n^{he} \right) \mathrm{d}\Omega = 0. \quad (5)$$

To calculate ϕ_{n+1}^h, we use Eq. (3). From ϕ_{n+1}^h, we calculate ϕ_{n+1}^{he} by a combination of a least-squares projection:

$$\int_{\Omega} (\psi_{n+1}^{he})_P \left((\phi_{n+1}^{he})_P - \phi_{n+1}^h \right) \mathrm{d}\Omega = 0, \quad (6)$$

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 $VOL(\phi_{n+1}^{he}) = VOL(\phi_n^{he})$. Here the subscript P is used for representing the intermediate values following the projection, but prior to the corrections for volume conservation. These projections and volume corrections are embedded in the iterative solution technique, and are carried out at each iteration (see [4]).

6. Examples

6.1. Free-surface flow past a bridge support

The free-stream Reynolds and Froude numbers are 10 million and 0.564. The mesh consists of 230,480 prism-based space-time elements. The DSD/SST formulation is

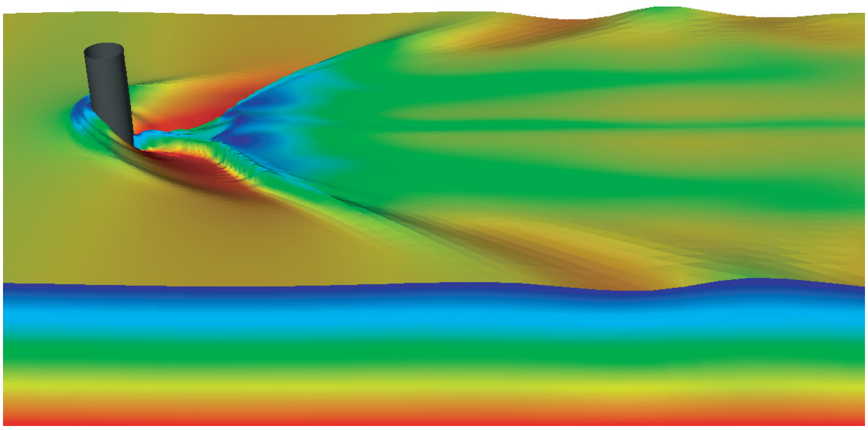

Fig. 1. Free-surface flow past a bridge support. The bridge support and the free-surface color-coded with the velocity magnitude.

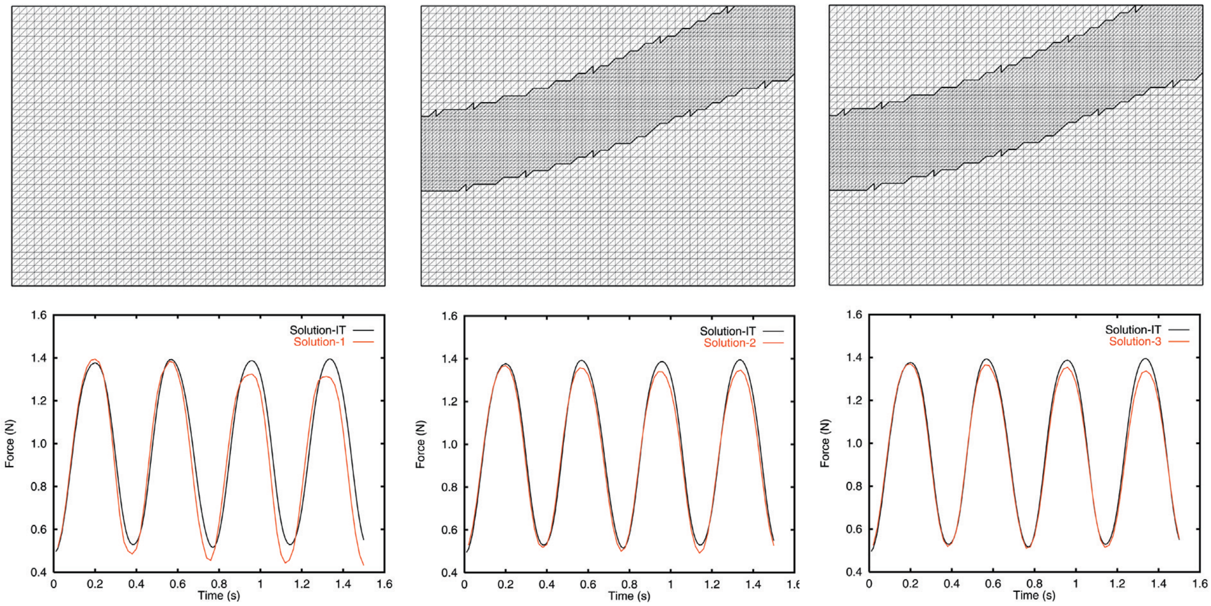

Fig. 2. 2D sloshing in a container. Pictures shows, at the top, at t = 0.2 s, Mesh-1 together with Mesh-2 and Mesh-3 (both shown on top of Mesh-1); and at the bottom, the time histories of the horizontal forces exerted on the container.

used with an algebraic mesh update method. Fig. 1 shows, at an instant, the cylinder together with the free-surface color-coded with the velocity magnitude. For more on this simulation see [5].

6.2. 2D sloshing in a container

A container partially filled with water is subjected to a horizontal acceleration of $0.2g$. We first compute with the DSD/SST formulation, with 6,000 quadrilateral elements. We refer to this as Solution-IT. Solution-1 is obtained by using Mesh-1, with 30,000 triangular elements. Solution-2 is obtained with the EDICT, where all functions come from Mesh-1 $\oplus$ Mesh-2. Solution-3 is obtained with the functions for velocity and pressure coming from Mesh-1 $\oplus$ Mesh-2, and for the interface function from Mesh-1 $\oplus$ Mesh-2 $\oplus$ Mesh-3. Fig. 2 shows the results. For more details see [6].

7. Concluding remarks

In this paper, we provided an overview of the stabilized finite element interface-tracking and interface-capturing methods developed for computation of flow problems with two-fluid interfaces. The EDICT increases the accuracy in representing the interface. The MITICT was developed for the classes of 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. The ETILT was developed to have an interface-capturing technique with better volume conservation properties and sharper representation of the interfaces.

References

[1] Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 1991;28:1–44.

[2] Tezduyar TE, Aliabadi S, Behr M. Enhanced-Discretization Interface-Capturing Technique. In: Matsumoto Y, Prosperetti A (Eds}, Proceedings of the ISAC '97 High Performance Computing on Multiphase Flows, 1–6. Japan Society of Mechanical Engineers, 1997.

[3] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. J Comput Phys 1981;39:201–225.

[4] Tezduyar TE. Finite element methods for flow problems with moving boundaries and interfaces. To appear in Arch Comput Methods Eng, 2001.

[5] Güler I, Behr M, Tezduyar TE. Parallel finite element computation of free-surface flows, Comput Mech 1999;23:117–123.

[6] Tezduyar TE, Aliabadi S. EDICT for computation of unsteady flows with interfaces. In: Atluri S, O'Donoghue P (Eds), Modeling and Simulation Based Engineering. Proceedings of International Conference on Computational Engineering Science, Atlanta, GA, 1998.