DIRECT NUMERICAL SIMULATIONS OF COMPLEX FLOWS

G. Tryggvason*, D. Juric*, J. Lu*, S. Mortazavi*, A. Fernandez* and A. Esmaeeli****
*Worcester Polytechnic Institute, Worcester, MA, USA
** Southern Illinois University, Carbondale, IL, USA
*** Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur (LIMSI), CNRS-UPR 3251, BP133, 91403 Orsay, France

1. Introduction

Direct Numerical Simulations (DNS) of multiphase flows, where all continuum scales of the flow are fully resolved, have progressed enormously in the last few years. Increases in computer power and new algorithms now make it possible to follow the unsteady motion of several hundred particles (drops, bubbles and solids) for long enough times so that meaningful averages for the fluid mixture can be calculated. However, most progress has so far been made for disperse flow of two-fluid systems. For a wide range of engineering applications it is, however, necessary to consider systems with a more complex physics.

One of the more successful approach to DNS of multiphase flows has been the finite volume/front tracking method introduced by Unverdi and Tryggvason (1992). This method has been used to study a large number of multiphase systems, but the majority of applications has been to disperse bubbly flows. Here we will examine three extension of this method. The first is the development of an algorithm to handle topology changes in simulations where the interface is described by connected marker points, the second is the inclusion of electric fields and studies of electrohydrodynamics of multiphase systems and the last example is simulations of nucleate boiling.

2. Topology Change

While several numerical approaches for DNS of multiphase flows are possible, the use of the "one-fluid" formulation of the governing equations, where one set of conservation equations is used to describe the complete flow field, including all the different materials, is by far the most popular one. The "one-fluid" formulation makes it possible to use a regular structured grid for the simulations, thus resulting in relatively simple and efficient codes. The main challenge in the use of the "one-fluid" formulation is the advection of a marker function identifying the different fluids. Advecting the marker function can be done in two ways: either by advecting the marker function directly or by advecting points marking the interface. In the latter case the marker function must then be constructed from the location of the interface. One of the main practical differences between advecting a marker function and an advection of the points marking the interface is the handling of coalescence and breakup. A marker function is represented by its value at fixed grid points, and an interface location is indirectly given by places where the marker function changes from one value to the other. If two patches of the same fluid come together in such a way that there is no grid point between them where the marker function has a different value, the interface disappears and the fluid patches merge. A tracked interface, on the other hand, always keeps its identity. If two patches of the same fluid come together, each patch is still identified by its surface markers

and while the interfaces may coincide, the markers are still there. Put simply, the default behavior of advected marker functions is always to merge and the default behavior of tracked interfaces is never to merge. For two-dimensional flows, where a tracked interface is simply a string of connected marker particles, changes of the topology of the interface are relatively easy, although identifying efficiently which parts of the interface to merge can be cumbersome. For three-dimensional flows the challenges are considerable greater and the complexity of including topology changes has often been listed as the main drawback of explicit tracking of the interface. Such modifications are, however, possible. Nobari and Tryggvason (1996) did, for example, develop an algorithm to coalesce two colliding drops, but the method was not very general and did not include the possibility of 'pinching' thin threads. A more general topology change algorithm was discussed by Du et al. (2006) and implemented in the FronTier code. A different strategy was introduced by Shin and Juric (2002), who choose to abandon any effort to modify the interface grid itself and instead recreated the grid periodically from the marker function. Thus, they generated a marker function from the interface markers (as in the method of Unverdi and Tryggvason, 1992) and the regenerated the interface markers as a level surface of the grid marker function.

We have recently developed a general algorithm to change the topology of traced interfaces. The algorithm can handle both thin films that rupture and thin threads that snap. It consists of two steps, a search procedure to find those parts of the front where topology change should take place and a procedure to actually perform the change. The first step can be done in several different ways and we have experimented with both an efficient search algorithm where the interface elements are grouped together by location so that the element pairs that have to be examined can be reduced and an algorithm where a marker function constructed from the front is used to aid in the identification of close element pairs. The second part, the actual topology change, is based on an incremental global strategy where each step of the topology change is performed globally on all the front objects taking part in the change, before moving on to the next step. This contrast fundamentally with a more local approach, such as the one we have used successfully for two-dimensional flow and the one used by Nobari and Tryggvason (1996), where the change was completed locally before we moved on to other places where topology change might take place.

We are currently using this algorithm in DNS of channel flows where we area studying flow regime transition due to coalescence in gas liquid bubbly flows. Figure 1 shows bubbles moving upward in a channel between two vertical walls. The computational domain is periodic in the spanwise and the streamwise direction. In addition to the bubbles moving upward due to buoyancy, we impose a pressure gradient forcing the liquid to flow upward as well. In the first frame the original eight bubbles have moved close to the walls, as expected. Those bubbles undergo pairwise coalescence, forming first four larger bubbles (frame two), then two very deformable bubbles and eventually one large Taylor bubble filling the channel. To quantify the change in both the total area as well as the structure of the interface, we have monitored the area tensor (Wetzel and C. Tucker, 1999; Kachanov and Sevostianov, 2005) defined by

$$\mathbf{A} = \frac{1}{Vol}\int_S \mathbf{nn}\, da \tag{1}$$

The evolution of the components of the tensor is plotted versus time in figure 2. The projections in *x* and y (horizontal) remain approximately the same, as one would expect, but the projection in *z* (vertical) is initially larger as the bubbles become flatter. The coalescence