The art of moving a surface

"What is the best way to numerically represent and advect a complex fluid interface such as the air-water interface of a breaking ocean wave?"

During the past two years, I have tried to answer this question in a postdoc project entitled "Breaking the Code of Breaking Waves"[1]. In this talk, I will present the (still ripening) fruit of my work: A new numerical scheme dubbed *IsoAdvector* for passive advection of a sharp interface across an unstructured finite volume mesh.

The new algorithm is based on the *volume of fluid* (VOF) idea of calculating the volume of one of the fluids transported across the mesh faces during a time step. The novelty of the isoAdvector concept is twofold: First, we exploit an *isosurface* concept for modelling the interface inside cells in a geometric surface reconstruction step. Second, from the reconstructed surface, we model the motion of the *face-interface intersection line* for a general polygonal mesh face to obtain the time evolution *within* a time step of the submerged face area. Integrating this submerged area over the time step leads to an accurate estimate for the total volume of fluid transported across the face.

In the talk, this fairly simple procedure should become much clearer with ample use of illustrations.

I will also show the results of simple 2D and 3D interface advection problems involving motion of discs and spheres on both structured and unstructured meshes. Compared to other interface advection schemes such as MULES, HRIC and CICSAM the results are remarkably good, not only in terms of volume conservation, but also in terms of boundedness, surface sharpness, efficiency, and insensitivity to the mesh quality.

"Why are we doing this?"

At DHI we have a decade-long tradition for developing and using Computational Fluid Dynamics (CFD) codes for simulating free surface ocean waves. We use these simulations to assess realistic wave loads on projected coastal and offshore structures such as break waters, ships, wind turbines, and oil platforms. In recent years, we have mainly used the open source C++ based code, OpenFOAM, for our CFD work. This code has the major advantage of being maintained and developed by a foundation dedicated to open source and by an ever-growing worldwide user community. Also, being a well-structured object-oriented open source code, we can quickly modify it and customize it to the particular task at hand in our research and commercial projects.

The OpenFOAM interfacial flow solver, interFoam, for two immiscible, incompressible fluids is thus the starting point of our work. The applicability of this solver for detailed ocean wave simulations is well-documented in the literature. However, in our work with the solver, we have come across a number of shortcomings. One such shortcoming is its sensitivity to cell shapes in spite of the OpenFOAM package supporting arbitrary polyhedral cells. This makes meshing cumbersome and time consuming, and deteriorates solution quality when dealing with flows around complex geometries where "ugly" cells are unavoidable. Another issue with the solver is its apparently inconsistent handling of the momentum balance at the fluid interface – a problem which is particularly delicate and pronounced when simulating an air-water interface with its density ratio of almost 1:1000. A third problem - maybe of a more academic nature - is the lack of documentation of the MULES interface advection method currently implemented in interFoam. While the exact implementation may of course be read from the source code, the *ideas* (e.g. the artificial interface compression concept) behind the method seem rather heuristic and are by no means clear. In contrast, the steps and assumptions going into the derivation of the the IsoAdvector method are much more straightforward and are documented in a paper recently submitted to Royal Society Open Science[2].

The IsoAdvector method is implemented as an OpenFOAM extension module and will be released as open source in the hope that it will be used, tested, and further developed in the multitude of application areas that may benefit from accurate interfacial flow simulations.

[1] www.roenby.com/postdoc

[2] http://arxiv.org/abs/1601.05392