Proceedings of the 9th WSEAS International Conference on Applied Mathematics, Istanbul, Turkey, May 27-29, 2006 (pp25-28)

Free Surface Flow with Moving Rigid Bodies. Part 2. On the Development of a Numerical Simulation Tool

LARISA A. MIRONOVA
Memorial University of Newfoundland
Department of Mathematics and Statistics
St. John's, NL, A1C 5S7
CANADA

OLEG I. GUBANOV
Memorial University of Newfoundland
Computational Science Program
St. John's, NL, A1C 5S7
CANADA

SERPIL KOCABIYIK
Memorial University of Newfoundland
Department of Mathematics and Statistics
St. John's, NL, A1C 5S7
CANADA

Abstract: This paper describes the design of a numerical simulation tool for the solution of the incompressible viscous flow past a rigid cylinder beneath a free surface. The integral form of governing equations for the two-dimensional viscous incompressible flows is solved subject to no-slip conditions at the solid body surface and nonlinear conditions at the free surface. Well-posed boundary conditions are enforced at the inflow and outflow boundaries since they ensure correct physical development of the flow near the computational domain boundaries. The fractional area-volume obstacle representation method and volume of fluid method are used to track the solid body and free surface interfaces, respectively. The numerical algorithm is verified by applying it to the special cases of stationary and oscillating cylinders.

Key–Words: Incompressible, Viscous, Unsteady, Free Surface, Cylinder, Oscillation, Computation

1 Introduction

The numerical simulation of free surface flows with moving bodies is a complex problem. Two types of moving boundaries are present, namely free surface and the boundary of the moving rigid body. The major technical issue encountered in the solution of free surface problems is the appearance of numerical instabilities, that arise due to the description of the mesh movement to track the moving boundaries, nonlinearity of the governing equations and boundary condition implementation at the free surface and rigid body surface. Since the integral form of governing equations for the incompressible viscous flow was derived and discretized in a prior paper [6], the present study essentially focuses on developing a numerical algorithm for the solution of the incompressible viscous flow past a cylinder beneath a free surface under appropriate boundary conditions. The volume of fluid (VOF) method [7] is used to track moving free surface interface. A piecewise-linear interface reconstruction (PLIC) algorithm [3] is used at each time step for determining the position of both the free surface and fluid-body interfaces. The reconstructed free surface is then advected using computed local velocity field based on a geometrical area-preserving VOF advection algorithm.

In the present work, we shall use the same governing equations (FAVOR$^{\text{TM}}$ equations) as those derived in Part 1 [6] in the case of two dimensional free surface flows of a viscous incompressible Newtonian fluid with moving rigid bodies:

$$\frac{d}{dt}\int\limits_{\mathbb{V}} \mathrm{d}V + \int\limits_{\mathbb{A}} (\underline{n}\cdot\underline{u})\,\mathrm{d}V = 0, \tag{1}$$

$$\frac{d}{dt}\int\limits_{\mathbb{V}} \underline{u}\,\mathrm{d}V + \int\limits_{\mathbb{A}} (\underline{n}\cdot\underline{u})\cdot\underline{u}\,\mathrm{d}V = \int\limits_{\mathbb{A}\cup\mathbb{I}} \underline{\underline{\sigma}}\cdot\underline{n}\,\mathrm{d}S + \int\limits_{\mathbb{V}} \underline{g}\,\mathrm{d}V \tag{2}$$

where $\underline{u} = (u, v)$ is a velocity vector, $\underline{n} = (n_1, n_2)$ is an outward normal, $\underline{g} = (0, g)$ is a gravity vector, $\underline{\underline{\sigma}}$ is a Newtonian stress tensor, $\mathbb{A}$ and $\mathbb{V}$ are the corresponding fractional area and volume open to flow, $\mathbb{I}$ is the length of fluid-body interface.

Boundary conditions are the nonlinear kine-

Proceedings of the 9th WSEAS International Conference on Applied Mathematics, Istanbul, Turkey, May 27-29, 2006 (pp25-28)

matic and dynamical conditions:

$$\mu \left(\frac{\partial \underline{u}_\tau}{\partial \underline{x}_n} + \frac{\partial \underline{u}_n}{\partial \underline{x}_\tau} \right) = 0, \tag{3}$$

$$p = 2\mu \frac{\partial \underline{u}_n}{\partial \underline{x}_n} \tag{4}$$

at the free surface and the no-slip condition:

$$\underline{u} = \underline{u}_b \tag{5}$$

on the rigid body surface. Here (3) and (4) are given in the normal-tangential frame of reference, μ is the fluid viscosity, $\underline{u}_b$ is the body velocity. Flow velocity is set to the velocity of inflow at the inlet boundary. A well posed hydrostatic boundary condition:

$$\mu \frac{\partial \underline{u}}{\partial \underline{n}} - p\underline{n} = p_h \tag{6}$$

is used at the outlet boundary. Here, p_h is the hydrostatic pressure and $\underline{n}$ is the outward normal to the boundary of the computational domain.

2 Implementation of boundary conditions

Following the work of Gerrits and Veldman [4], every pressure cell in staggered grid is given a corresponding label based on the value of fractional volume open to flow ($0 \leqslant \mathbb{V} \leqslant 1$) in this cell. In

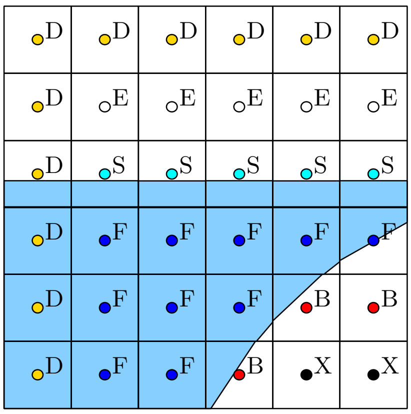

Fig. 1: Pressure cell labeling.

Fig. 1 we distinguish cells with $\mathbb{V} \geqslant \frac{1}{2}$ as

Air(E)	:	Cells which contain no fluid
AirFluid(S)	:	Cells which border at least one E cell
Fluid(F)	:	All remaining cells

and cells with $\mathbb{V} < \frac{1}{2}$ are labeled as

FluidSolid(B)	:	Cells which border at least one of the E, S or F cells
Solid(X)	:	All remaining cells

Domain boundary cells are labeled as DomainBoundary(D). It should be noted that all cell labels, except D cells, are time-dependent. As demonstrated in [4], this classification is advantageous for stability of numerical method since it eliminates occurrence of very small cells and thus no severe time-step restrictions are expected.

In terrestrial flows, viscous effects on the free surface are negligible compared to other terms and thus can be neglected. Consequently, pressure is determined in every S cell via linear interpolation between the pressure $p_f = 0$ and the known value of pressure in the neighbour F cell. In Fig. 2 pressure in S cell becomes:

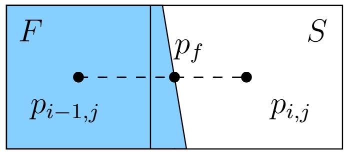

Fig. 2: Pressure near free surface.

$$p_{i,j} = LINT(0, p_{i-1,j}), \tag{7}$$

where $LINT$ is a linear interpolation/ extrapolation function written in C++ programming language.

In the approximation of momentum equations EE velocities are needed. Since over the time step EE velocity may become SE velocity, it is reasonable to assume that its value is nonzero. This velocity can be calculated from equation (3) which is discretized by finite differences ([3], [4]). SE velocities are computed by constant extrapolation from fluid cell [2].

To account for a rigid body in numerical model, equation (5) is used. Three points are considered: B, N and I. B denotes a base velocity knot which is labeled as BF, BS or BB. N is the neihbour fluid knot located in the direction normal to the estimated fluid-body interface. I is the interface location reconstructed in the vicinity of knots B and N. There are three principal locations of neighbour, interface and base knots with respect to each other (see Fig. 3). If $|K_B - K_I| < |K_N - K_I|$ the $LINT$ function is executed, otherwise the value of velocity in the base knot is set to the velocity of the body in that knot. Fig. 3b illustrates the case when no $LINT$ function is applied. In Fig. 3a and 3c we use $LINT$

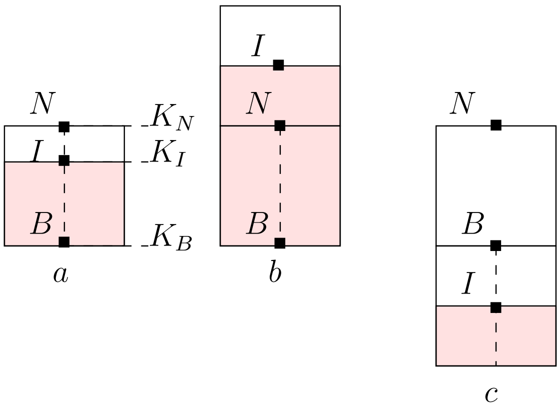

Fig. 3: Locations of base, interface and neighbour knots in y direction.

function to extrapolate and interpolate the value of velocity in B knot from the velocities in N and I knots, correspondingly. The same procedure is applied for velocities in x direction.

At the inlet boundary we prescribe the inflow velocity. At the outlet boundaries, equation (6) is solved. Unlike some commonly used outlet boundary conditions, equation (6) couples pressure and velocity. This well-posed form of boundary conditions ensures that flow properties do not change when flow leaves the computational domain (see [5]).

3 Fluid interface reconstruction and advection

Since moving fluid interface position changes continuously its location must be defined at every time step. In our numerical algorithm, the fluid interface is tracked by means of VOF method. We approximate all fluid interfaces using PLIC algorithm. In this method for every pressure cell with the volume of fluid $(0 \leqslant V \leqslant 1)$ we define an orientation and then a distance of a straight-line segment:

$$\underline{n}\underline{x} + d = 0, \tag{8}$$

which cuts amount of fluid equal to V in cell ([3], [9]). Here, unit normal is $\underline{n}$ pointed out of fluid and d is a line distance from the origin. An approximation of the fluid interface normal:

$$\underline{n} = \frac{\nabla V}{|\nabla V|} \tag{9}$$

is obtained by standard finite difference discretization in every pressure cell with respect to its neighbours in 3×3 block of cells. We note that fluid-body interface normal is determined based on the geometry of the body. The calculation of the line distance is the most difficult task because the value of distance is constrained by volume conservation. The distance d in equation (8) is found from equation:

$$V(d) - V = e, \tag{10}$$

where $V(d)$ is the volume bounded by cell area and line (8) and e is a value of tolerance. Following Rider and Kothe [9], we use a Brent's root-finding algorithm to find zeros of the function (10). A root of function (10) is a value of distance which equals volumes $V(d)$ and V to within the desired tolerance and corresponding line reconstructs fluid interface segment in cell.

While a rigid boundary is very easy to locate in time, free surface interface requires a suitable advection algorithm to model its motion. In the present work, geometrical area-preserving VOF advection algorithm is used to advect the free surface in time. In this method for every pressure cell we introduce two independent one dimensional linear mappings in x and y directions which define pre-images of vertical and horizontal edges of this pressure cell, correspondingly. Over the time Δt the velocities located at the edges of considered pressure cell may add some fluid into either cell itself or into its neighbour cell. The sum of all contributed volumes into cell itself corresponds to a new volume of fluid for the new time step. We note that the work of Aulisa [1] demonstrated that this advection algorithm conserves the area (or mass) exactly for incompressible fluids.

4 Validation

The numerical algorithm is verified by applying it to the special case of uniform flow past a stationary horizontal circular cylinder beneath a free surface. The flow is calculated at Reynolds number $R = 180$ when the depth of cylinder submergence, h/d, is 0.55 (d is the cylinder diameter). A non-uniform, 160×70, staggered grid is used. Figure 4 shows a comparison between the equivorticity plots obtained in the present study and those obtained by Reichl [8] using the commercial fluid CFD software package FLUENT 5.

Bearing in mind that Reichl [8] did not use the same boundary conditions as those used in the present work, it may be noted that, even so, there is a good qualitative agreement between the results.

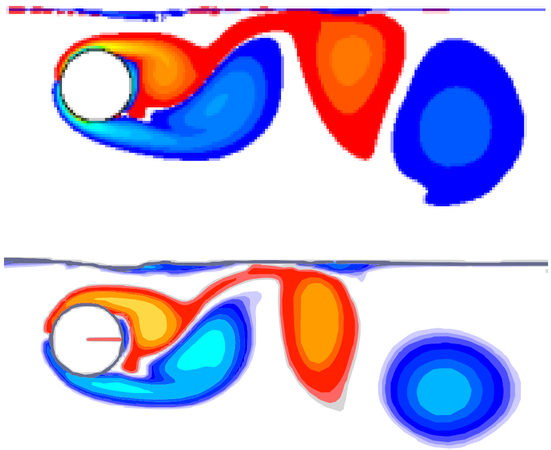

Fig. 4: Patterns of equi-vorticity lines for $h/d = 0.55$ and $R = 180$. Comparison between numerical results of Reichl [8] (top) and the present computed results (bottom).

5 Conclusion

In this paper, the description of a numerical algorithm is given for solving the incompressible viscous flow past an oscillating cylinder beneath a free surface. Particular attention is focused upon numerical implementation of boundary conditions. The numerical algorithm is verified by applying it to the special cases of stationary and oscillating (not shown here) cylinders.

Acknowledgments: We acknowledge with thanks to Natural Sciences and Engineering Research Council of Canada.

References:

[1] E. Aulisa, S. Manservisi, R. Scardovelli and S. Zaleski, A geometrical area-preserving volume-of-fluid advection method, *J. Comput. Phys.* Vol.192, No.1, 2003, pp. 355–364.

[2] G. Fekken, *Numerical simulation of free-surface flow with moving rigid bodies*, PhD thesis, Rijksuniversiteit Groningen 2004

[3] J. Gerrits, *Dynamics of liquid-filled spacecraft*, PhD thesis, Rijksuniversiteit Groningen 2001

[4] J. Gerrits and A. Veldman, Numerical simulation of coupled liquid-solid dynamics, *Proc. of the European Congress on Computational Methods in Applied Sciences and Engineering*, Barcelona 2000

[5] P. M. Gresho and R. L. Sani, *Incompressible flow and the finite element method*, John Wiley and Sons Ltd., 1998

[6] O. I. Gubanov, L. A. Mironova and S. Kocabiyik, Free surface flow with moving rigid bodies. Part 1. Computational flow model, *Proc. of the* 13^{th} *Annual Conference of the Fluid Dynamics Society of Canada*, St. John's, Newfoundland, Canada 2005

[7] C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* Vol.39, 1981, pp. 201–225.

[8] P. J. Reichl, *Flow past a cylinder close to a free surface*, PhD thesis, Monash University, Clayton, Australia 2001

[9] W. J. Rider and D. B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* Vol.141, No.2, 1998, pp. 112–152.