14th Australasian Fluid Mechanics Conference
Adelaide University, Adelaide, Australia
10-14 December 2001

A Numerical Method For Modelling Of Two-Fluid Flow – Solid Interaction

A.V. Bui and M. Rudman

CSIRO Thermal & Fluids Engineering
PO Box 56, Highett, Victoria, 3190 AUSTRALIA

Abstract

A numerical method has been developed to simulate the interaction of two-phase flow with embedded internal boundaries. Movement of the interface between fluid phases is captured by the level-set front tracking method, which is coupled with a high-resolution Navier-Stokes solver (employing the Cubic-Interpolated-Propagation or CIP method) to provide an accurate prediction of flow with free surfaces. This model of multiphase flow has been successfully validated in the past and applied to simulate flows with gas bubble or high-density liquid drop. In this work, the multiphase flow solver has been modified to model the embedded internal boundaries (or solids). The non-slip boundary conditions are imposed by assigning virtual body forces over surfaces which need not coincide with grid lines. This approach readily allows simulation of flows in complex geometries using simple computational meshes and without a need of coordinate transformation or domain partition.

The proposed numerical method was successfully validated for single-phase flow in simple or cross-section varying channels. Simulations of flow around a cylinder and fluid dynamics during the collision of a liquid droplet on a substrate were performed to illustrate the method capability in handling single/two-phase flow-solid interaction. The computational results were compared with available experimental data and with simulation results obtained by using other computational methods.

Introduction

Multiphase flows comprising multiple distinct, immiscible fluids and bounded by topologically complex geometries are prevalent in many natural and industrial processes. However, analysis of such fluid dynamics problems is a challenging task due to the presence of fluid-fluid interfaces and the limitation of the analytical and numerical methods in dealing with complex geometries.

As scientific computing and computer power have greatly advanced in recent years, numerical methods for multiphase flows, in general, and for interfacial problems, in particular, have been developed. However, accurate simulation of multiphase flows with complex and sharp interfaces presents a problem of considerable difficulty, which is caused by complex interfacial topology and steep changes of fluid properties across the interface. Numerical methods developed to “capture” behaviour of fluid-fluid interfaces are generally classified into either ‘Lagrangian’, ‘Eulerian’ or ‘hybrid’ groups. The 'Eulerian' methods of dealing with interfacial problems employ a volume-discretization of the whole Eulerian computational domain, while modifying the numerical approximation to minimize numerical diffusion at the interfaces. The numerical diffusion can be excessive if the numerical method is of low order. At the same time, high-order numerical schemes may cause numerical oscillations around the interface, thus, leading to unphysical solutions. In difference from the Eulerian approach, the direct-tracking approach explicitly introduces additional computational elements to track the front. This approach is known as a 'Lagrangian' approach, when the equations of motion are transformed to equations for the interface itself. In general, this decreases considerably the amount of resolution necessary to provide front sharpness, and eliminates numerical diffusion altogether. In practice, direct tracking is best suited for well-defined fronts that are easily identified by the initial conditions.

Hybrid 'Eulerian-Lagrangian' methods have also been developed. These can be divided into ‘volume-tracking’ and ‘front-tracking’. The ‘volume-tracking’ methods are relatively simple and they are different in the ways of reconstructing the interface from the volume information, which usually makes the coding complex and may lead to incorrect results. Moreover, the volume-tracking methods have difficulties in dealing with complex three-dimensional surface topology. The ‘front-tracking’ methods are based on the use of a combination of a stationary grid, on which the Navier-Stokes equations are solved, coupled with a ‘front tracker’, which provides information about the interface location and topology. In the case of the immersed boundary method [6,8], this ‘tracker’ is an additional moving grid representing the interface. Another interface tracking technique is based on the use of a level function, which is smooth and is convected by underlying flow (on the Eulerian stationary grid) (see [1,5]). This technique retains most of the advantages of the volume-tracking approach, while being simple enough for realization.

In dealing with flows in complex geometries, one usual approach is using conventional structured-grid finite-difference/finite-volume methods with a body fitted co-ordinate system. This approach usually requires coordinate transformation and division of the computational domain into blocks of simple geometry. The main advantages of this approach are that imposition of boundary conditions is easy and the solver can be simply designed to maintain conservation properties. However, the complexity of the governing equations resulting from coordinate transformation may cause difficulty in maintaining an accurate numerical solution and adversely impact upon the stability and convergence of the solver. Alternatively, there is a different approach to simulate flows in complex geometries, which is still based on the finite-volume/finite-difference discretisation and employing Cartesian structured grids [3,7]. By representing the internal boundaries via a forcing term added to the governing equations, the method allows simulation of flows in complex geometries using simple computational meshes and without a need of coordinate transformation or domain partition.

In this study, a numerical method has been developed to simulate the interaction of two-dimensional, two-phase flow in complex geometries (or boundaries). Movement of the interface between fluid phases is captured by the level-set front tracking method, which is coupled with a high-resolution Navier-Stokes solver (employing the Cubic-Interpolated-Propagation or CIP method [9,10]) to provide an accurate prediction of the interface movement. The non-slip conditions on the internal boundaries are imposed by assigning virtual body forces over surfaces which need not coincide with grid lines. The method is validated for some cases of single-phase flows in irregular geometries. Some test cases of two-fluid flow interacting with a virtual boundary are also studied.

Modelling Approach

As noted before, the modelling approach is based on: i) the level-set front-capturing technique to model immiscible multiphase flows with distinctive interfaces; (ii) a 'forcing' method to simulate the complex internal/external boundaries; (iii) an explicit high-resolution Navier-Stokes solver. With such a modelling approach, simple Cartesian structured computational meshes can be applied to simulate various flows with interfaces in different geometries.

The Level-Set Front-Capturing Technique

Essentially, the level-set technique employs a *smooth* level function, ϕ, to describe the interface separating two immiscible fluids [1,5]. This level function is chosen as a signed distance function with the zero level set defining the interface location. The level function is positive in one fluid region and negative in the other one and its absolute value indicates the distance to the interface. The level function is convected by flow field $\vec{u}$ as follows:

$$\phi_t + (\vec{u} \cdot \nabla)\phi = 0 \tag{1}$$

Since this function is smooth across the interface (unlike the fluid properties), the above convection equation can easily be solved with high-order accuracy and without introducing numerical oscillations. Using the level function, the steep changes of fluid properties across the interface can be smoothed out to mitigate numerical oscillations in the solution of Navier-Stokes equations. For instance, fluid density and viscosity can be determined as:

$$\begin{aligned} \rho_\varepsilon &= \rho_1 + (\rho_2 - \rho_1) \cdot H_\varepsilon(\phi), \\ \mu_\varepsilon &= \mu_1 + (\mu_2 - \mu_1) \cdot H_\varepsilon(\phi), \end{aligned} \tag{2}$$

where H_ε is a regularization Heaviside function and ε is a regularization parameter:

$$H_\varepsilon(d) = \begin{cases} 0 & \text{if } d < \varepsilon, \\ (d+\varepsilon)/(2\varepsilon) + \sin(\pi d/\varepsilon)/(2\pi) & \text{if } |d| \le \varepsilon, \\ 1 & \text{if } d > \varepsilon. \end{cases} \tag{3}$$

One of the major advantages of using the level function is that the geometric properties of the interface can also be easily determined:

Normal vector: $\vec{\mathrm{n}} = \dfrac{\nabla\phi}{|\nabla\phi|}$,

(4)

Surface curvature: $\kappa = \nabla\vec{\mathrm{n}} = \nabla \cdot \dfrac{\nabla\phi}{|\nabla\phi|}$.

(5)

Then, the surface tension force appearing in the momentum equations can be fully defined in terms of the level function as:

$$\vec{\mathrm{F}}_\mathrm{s} = \sigma\kappa\delta(\phi)\vec{\mathrm{n}} = \sigma\delta_\varepsilon(\phi) \cdot \nabla\phi \cdot \nabla\left(\frac{\nabla\phi}{|\nabla\phi|}\right), \tag{6}$$

where σ is the surface tension coefficient and $\delta_\varepsilon(\phi)$ is a delta function corresponding to H_ε:

$$\delta_\varepsilon(\phi) = \begin{cases} 0 & \text{if } |\phi| > \varepsilon, \\ \frac{1}{2}(1 + \cos(\frac{\pi\phi}{\varepsilon}))/\varepsilon & \text{if } |\phi| \le \varepsilon. \end{cases} \tag{7}$$

The level set method can handle complex topologic changes of the interface (breaks, merges, etc.) and its formulation is general for two- and three-dimensional problems.

'Forcing Field' Approach To Simulate Complex Boundaries

As described in [2,3,6,7], by introducing a body-force field, complex geometries and boundary motion can be handled using regular computational meshes in the Cartesian coordinate system. Essentially, the body force is introduced in the fluid surrounding the boundaries, so that the chosen speed of boundary movement can be achieved. In practice, there are two methods to compute the boundary forces based on the concepts of either (i) feedback forcing [3,7] or (ii) direct forcing [2]. In this work, the method of feedback forcing is employed, in which the forcing term assumes the expression:

$$\begin{aligned} \vec{\mathrm{f}}(\vec{x}_b, t) = \alpha_f & \int_0^t [\vec{u}(\vec{x}_b, t') - \vec{V}(\vec{x}_b, t')]dt' \\ & + \beta_f \cdot [\vec{u}(\vec{x}_b, t) - \vec{V}(\vec{x}_b, t)], \end{aligned} \tag{8}$$

where α_f and β_f are negative constants, $\vec{u}$ is the instantaneous velocity of the boundary, and $\vec{V}$ is the required velocity of the boundary. As noted in [3], such a formulation of force corresponds to the introduction of a feedback control loop, which enforces the condition $\vec{u} = \vec{V}$ at the boundary. It must be noted that the boundary location $\vec{x}_b$ is generally not coincident with the Eulerian (fixed) grid nodes and, therefore, the force must be interpolated from the actual boundary location to the fixed grid points. The body force at grid nodes is then determined as:

$$\vec{\mathrm{F}}_\mathrm{b}(\vec{x}) = \sum_{ib} \vec{\mathrm{f}}_{ib} \cdot \delta_h(\vec{x} - \vec{x}_{ib}) \cdot \Delta l_{ib}, \tag{9}$$

where *ib* is the index of boundary nodes, Δl is the length of a boundary segment, and δ is the discrete delta function defined as:

$$\begin{aligned} &\delta_h(\vec{x}) = \prod_i \delta_{h_i}(x_i), \quad i = 1..\text{Ndim} \\ &(\text{Ndim - number of the space dimensions}), \end{aligned} \tag{10}$$

and

$$\delta_h(r) = \begin{cases} \frac{1}{8h}(3 - 2q + \sqrt{1 + 4q - 4q^2}) & \text{if } q(= |r|/h) \le h, \\ \frac{1}{8h}(5 - 2q - \sqrt{-7 + 12q - 4q^2}) & \text{if } h < q \le 2h, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

In order to compute the boundary force, the boundary velocity must be determined by interpolating the fixed-grid fluid velocities to the boundary nodes as follows:

$$\vec{u}(\vec{x}_b) = \sum_{\vec{x}} \vec{u}(\vec{x}) \cdot \delta_h(\vec{x} - \vec{x}_b) \cdot \prod_i h_i, \quad i = 1..\text{Ndim}. \tag{12}$$

In this work, a second-order accurate Adams-Bashforth scheme is used for the time marching of the forcing term.

In application to simulation of flow in complex geometries, this 'forcing field' approach can significantly simplify numerical procedure and reduce computational cost and difficulties associated with coordinate transformation and grid regeneration.

Numerical solver

To solve the equation of level function convection and the system of Navier-Stokes equations, a high-resolution numerical procedure, which can minimize numerical diffusion and oscillations, is needed. In most finite-difference high-order numerical methods, high accuracy is achieved by increasing the number of discretization points used in the discrete approximation of the derivative terms (stencil). Such an increase of the stencil is often undesirable because of many reasons (loss of the tridiagonal nature of matrices, difficulties in dealing with boundary conditions, etc.). In this work, an explicit numerical procedure, called Cubic-Interpolated Propagation (CIP) [9,10], is applied to solve the system of differential equations. The method was coupled with the level set front-capturing method and has been successfully applied to simulate various multiphase problems with interfaces [1]. With the CIP method, high-order accuracy of the numerical solution can be achieved on the normal finite-difference stencil (5 points for two-dimensional problems). The method, however, must update the spatial derivatives of all variables at every time step and requires extra memory to store these derivatives. Even though the CIP method is an explicit solver, it still requires the pressure field to be updated iteratively. In this work, a multigrid method has been employed to accelerate the iterative solution of the pressure correction equation.

Numerical Applications

In this section, some simulations are performed to validate the developed modelling approach and demonstrate its capability. In all simulations, the force-field constants α_f and β_f (see equation 8) are chosen to be –400,000 and -600, respectively. From some preliminary numerical tests, such a choice of α_f and β_f provides sufficiently fast response of the force field, so that correct boundary conditions can be enforced. With this choice of constants, the stable time step should be smaller than 1.5ms [3].

Single-phase flow over a backward-facing step

Flow over a backward-facing step is one of the simplest flow problems where flow separation and reattachment occurs. The flow configuration is illustrated in figure 1. The problem prescribes a parabolic velocity profile at the inlet. At the outlet, the following boundary condition is applied:

$$u_t + \overline{u}_o u_x = 0 \quad \text{and} \quad v_t + \overline{u}_o v_x = 0, \qquad (13)$$

where $\overline{u}_0$ is the mean velocity at the outlet.

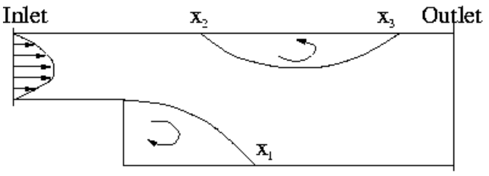

Figure 1. Illustration of the flow over a backward-facing step, showing a main reattachment region, which attaches at x_1, an upper recirculation region, which separates at x_2 and reattaches at x_3.

The step height and the size of the inlet channel are the same. The Reynolds number, based on the inlet height, is set to *Re*=400. The computational domain is 30 step heights long in the main channel and 6 step heights long for the inlet channel. The simulation is carried out on a 40 by 480 uniform grid. Boundaries of the step shown as discrete points in figure 2 are where the boundary forces are specified.

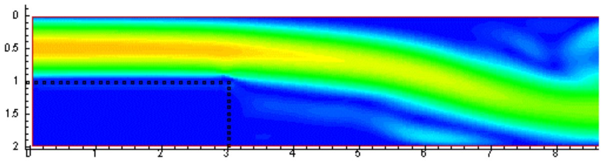

Figure 2. Speed of the flow in the channel with a backward-facing step.

Figure 2 shows the prediction result for the flow velocity in the channel. The predicted length of reattachment region (x_1) is approximately equal to 8.2 step heights, which is in good agreement with experimental data and numerical results reported in other studies [4].

Single-phase flow around a cylinder

First, start-up flow around an impulsively cylinder is considered. The calculation is performed on a square domain of size 2. The cylinder diameter is 0.44 and the cylinder is located at the centre of the computational domain. The grid is uniform and the grid resolution is 256 by 256. To create a start-up flow, a uniform velocity field is initialised in the domain and, at *t*=0, the force field on the cylinder perimeter is turned on. The Reynolds number of the flow is 550.

Figure 3 shows the streamlines and the vorticity isolines of the flow at times $\tau=t^*U_\infty/R$=2.12 and 5.12. At τ=5.12, the size of the primary vortex is comparable to that of the cylinder. Some secondary vortices are also seen to develop on the side of the cylinder. The pictures and the calculated drag coefficient (equal to 1.2 at τ=5.12) are consistent with those obtained by Goldstein et al. [3] using a spectral solver and the 'forcing field' method.

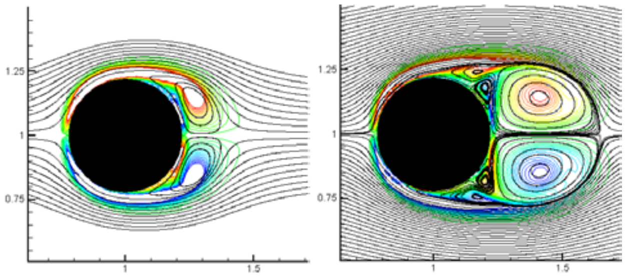

Figure 3. Streamlines and vorticity isolines around the cylinder at time t^*U_∞/R=2.12 and 5.12 (*R*-cylinder radius).

A longer simulation of flow around a cylinder is also performed. In this simulation, the size of the computational domain is [5x20] and a much coarse grid ([64x256]) is used. The cylinder size is 1 and its centre is located at (5,2.5). The flow Reynolds number is 200. Figure 4 shows the flow streamlines and vorticity field around the cylinder, in which the Von Karman alternating eddies are clearly observed in the wake of the obstacle. The predicted drag and lift coefficients as well as the frequency of the vortex shedding (figure 5) are in good agreement with the results and data reported in [4].

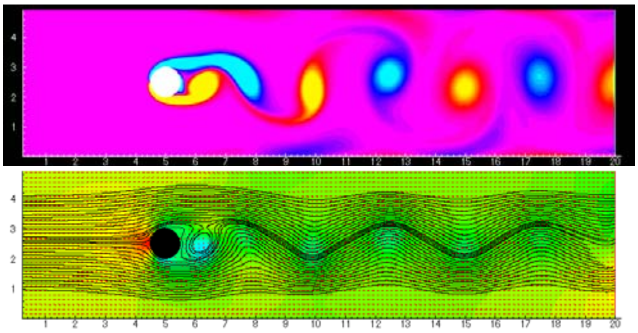

Figure 4. Vorticity field and streamlines of the flow around a cylinder.

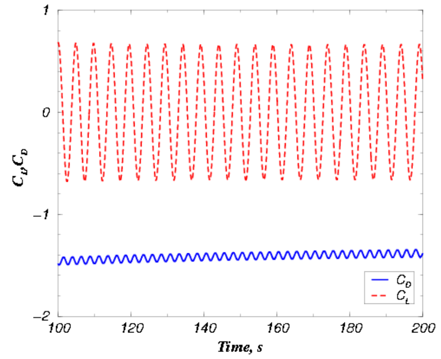

Figure 5. Variation of lift and drag coefficients with time.

Collision of a liquid droplet on a substrate

To demonstrate the capabilities of the developed modelling approach in dealing with multiphase problems, the model is used to simulate the collision of a liquid drop with a substrate. The drop is initially circular in shape and located at some distance from a horizontal substrate. Under gravitational force, the drop will gain speed before hitting the substrate. In this simulation, the interface between the liquid drop and its surrounding medium is represented by the zero level set and the substrate surface is modelled using the 'forcing field' technique. In this particular test case, the substrate surface is not a complex geometry and the simulation is only for a demonstrative purpose.

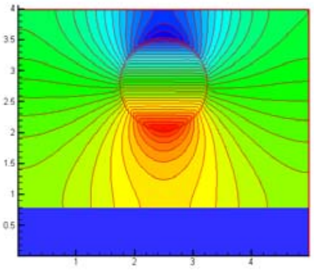

Figure 6. Pressure field at the beginning of drop fall.

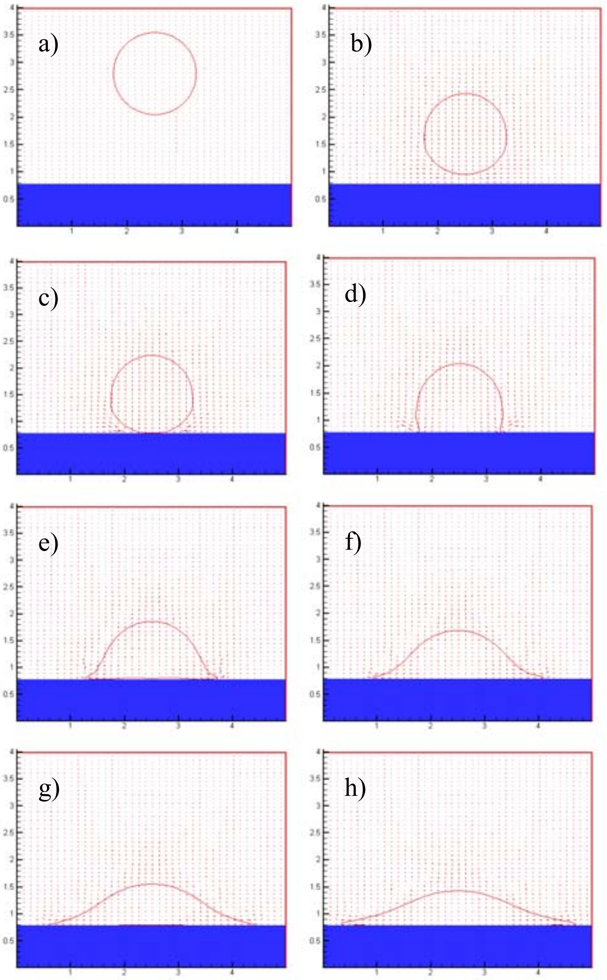

Figure 7. Deformation of a drop hitting a substrate. Except for the first picture (7a), the time advancement in the subsequent pictures is 0.1s.

The computational domain is 4 by 5 in size and the centre of the drop is initially located at (2.5,2.8). The substrate surface is set at a distance of 0.75 above the bottom of the computational domain. The grid resolution is [160x200].

In this simulation, a small viscosity ($1 \cdot 10^{-5}$) is assigned to the surrounding medium and the ratio between the liquid-drop density and the surrounding density is 200 to 1. A surface tension coefficient of 0.01 is specified for the drop. The gravity acceleration is assumed to be 9.8 m/s^2. The calculated speed of the drop before hitting the substrate is about 4.8 m/s.

In figure 6, the calculated result of pressure field is presented. As can be seen from the picture, our numerical solver can provide a good prediction of pressure distribution inside the drop and near the drop interface even with such a steep change of density. The prediction of drop deformation in time is presented in figure 7. Even with the small grid used, the thin edge of drop spreading front is still resolved. The simulation stopped when the edge of the drop became close to the sides of computational domain.

Conclusions

In this work, a numerical method has been developed to simulate interaction between two-fluid flows with interfaces and complex solid boundaries. The method is based on the front-capturing level set method, used to deal with fluid-fluid interfaces, and the 'forcing field' technique, which is efficient in dealing with complex boundaries. Coupled with a high-accuracy numerical solver, the modelling method is shown to be able to simulate complex flows using simple Cartesian regular grids. The proposed numerical method was successfully validated for single-phase flow in cross-section varying channels. Simulations of flow around a cylinder and fluid dynamics during the collision of a liquid drop on a substrate were performed to illustrate the method capability in handling single/two-phase flow-solid interaction.

Future improvements of the modelling method may include (i) adaptive grid refinement near the fluid-fluid and fluid-solid interfaces to reduce computational cost and (ii) a method to simulate elastic deformation of the solid boundaries under fluid-solid interaction.

References

[1] Bui, V.A., Dinh, T.N., and Sehgal, B.R., Numerical Simulation of Interface Phenomena Using CIP and the Level Set Front-Capturing Method, *Comp. Fluid Dynamics J.*, **8**, No.1, 1999, 103-112.

[2] Fadlun, E., Verzicco, R., Orlandi, P., and Mohd-Yusof, J., Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations, *J. Comp. Phys.*, **161**, 2000, 35-60.

[3] Goldstein, D., Handler, R., and Sirovich, L., Modeling a No-Slip Flow Boundary with an External Force Field, *J. Comp. Phys.*, **105**, 1993, 354-366.

[4] Khadra, K., Angot, P., Parneix, S., and Caltagirone, J.P., Fictitious Domain Approach For Numerical Modelling of Navier-Stokes Equations, *Int. J. Num. Meth. Fluids*, **38**, 2000, 651-684.

[5] Osher, S., and Sethian, J.A., Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *J. Comp. Phys*., **79**, 1988, 12-49.

[6] Peskin, C.S., Numerical Analysis of Blood Flow in The Heart, *J. Comp. Phys.,* **25**, 1977, 220-252.

[7] Saiki, E.M. & Biringen, S., Numerical Simulation of a Cylinder in Uniform Flow: Application of a Virtual Boundary Method, *J. Comp. Phys.* **123**, 1996, 450-465.

[8] Unverdi, S.O, and Tryggvason, G., A Front-Tracking Method for Viscous, Incompressible, Multi-Fluid Flows, *J. Comp. Phys*., **100**, 1992, 25-37.

[9] Yabe, T., Ishikawa, T., Kadota, Y., and Ikeda, F., A Multi-dimensional Cubic-Interpolated Pseudoparticle (CIP) Method without Time Splitting Technique for Hyperbolic Equations, *J. Phys. Soc. Jpn*, **59**, 1990, 2301-2304.

[10] Yabe, T., & Wang, P.Y., Unified Numerical Procedure for Compressible and Incompressible Fluid, *J. Phys. Soc. Jpn*, **60**, 1991, 2105-2108.