Immersed Boundary Method for Boltzmann and Navier-Stokes Solvers with Adaptive Cartesian Mesh

Robert R. Arslanbekov[a], Vladimir I. Kolobov[a], and Anna A. Frolova[b]

[a]*CFD Research Corporation,*
215 Wynn Dr, Huntsville, AL, 35803, USA
[b]*Dorodnicyn Computing Centre of the Russian Academy of Sciences,*
Vavilova Str., 40, Moscow, 119333, Russia

Abstract. Adaptive Cartesian mesh methods have demonstrated unique abilities for automated mesh generation and dynamic mesh adaptation to flow solution and moving boundaries. However Navier-Stokes (NS) solvers with Cartesian mesh often produce large fluctuations of surface quantities (pressure, skin friction, and heat flux) at solid boundaries. We show that the Immersed Boundary Method (IBM) with adaptive octree Cartesian mesh allows one to eliminate unphysical fluctuations of skin friction and heat flux at solid boundaries for viscous flow simulations. Implementation of IBM for deterministic Boltzmann solvers with adaptive Cartesian mesh is described.

Keywords: Rarefied Gas Dynamics, direct Boltzmann solver, Immersed Boundary Method, octree Cartesian mesh.
PACS: 51.10.+y, 52.25.Dg, 52.30.-q

INTRODUCTION

Adaptive Cartesian mesh offers unique abilities for automated mesh generation and dynamic mesh adaptation to flow solution and moving boundaries. However, Navier-Stokes (NS) solvers with Cartesian mesh often produce large fluctuations of surface quantities (pressure, skin friction, and heat flux) at solid boundaries [1]. Heat fluxes are particularly sensitive to boundary conditions being gradient quantities of flow properties. Several methods are being developed to mitigate or eliminate this deficiency.

In this paper, we compare embedded cut-cell methods and Immersed Boundary Methods (IBM) for Navier Stokes and Boltzmann solvers for subsonic and supersonic flows over a cylinder at different *Kn* numbers. In IBM approach, ghost cells are introduced inside solids to satisfy boundary conditions on the surface [2]. We show that the IBM is capable of greatly reducing the heat flux fluctuations for both gas-kinetic NS solvers and "traditional" NS solvers with octree Cartesian mesh. We show that IBM allows one to eliminate unphysical fluctuations of skin friction and heat flux at solid boundaries for viscous flow simulations. The implementation of IBM for the direct Boltzmann solver is discussed.

IMMERSED BOUNDARY METHOD

Both cut cells and IBM methods have been implemented in the Unified Flow Solver (UFS) [3]. UFS uses Adaptive Mesh and Algorithm Refinement (AMAR) procedure for multi-scale simulations of rarefied and continuum flows in a wide range of Mach and Knudsen numbers. Figure 1 compares the Embedded Boundary Method (EBM) and the Immersed Boundary Method (IBM). The first method produces cut-cells of smaller size, which are merged with larger neighboring cells. In IBM approach, ghost cells are introduced inside solid boundaries to allow treatment of the boundaries without cut cells [2],[4]. The cut-cell approach is known to introduce numerical fluctuations in the predicted surface quantities, which do not disappear with increasing grid resolution [1]. The IBM is increasingly being used for simulations of viscous compressible flows with complex steady and moving boundaries (see [5],[6] for further references).

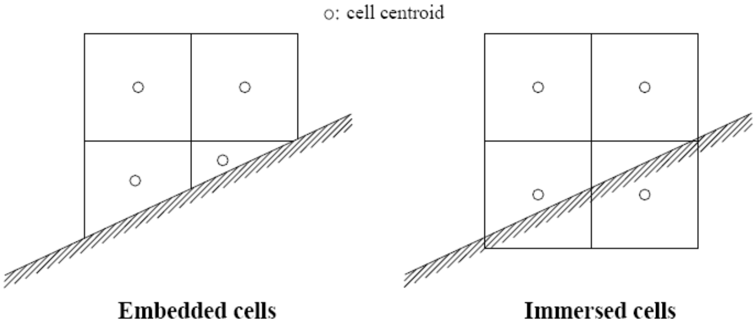

FIGURE 1. Illustration of embedded and immersed boundary approach

Figure 2 illustrates details of the IBM implementation in UFS. Instead of directly specifying boundary conditions on a wall, primitive variables are enforced on ghost cells such as 15 and 16 which act as wall boundaries. Cut cells are then considered as full cells and gradients of variables are calculated using values of *these* variables in ghost cells.

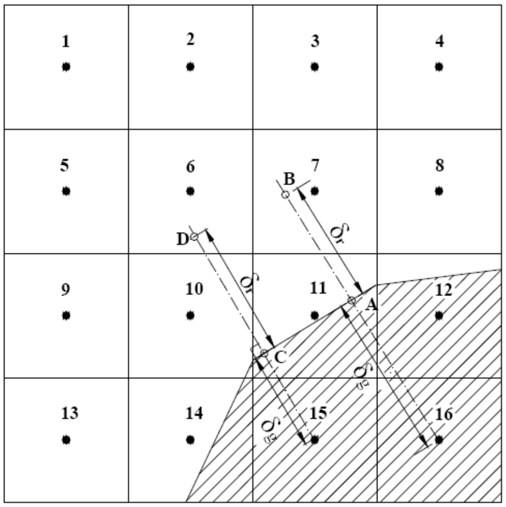

FIGURE 2. Implementation of IBM in UFS

NAVIER STOKES SOLVERS

We have developed two NS solvers in UFS for viscous flow simulations. The first solver is a gas-kinetic NS solver, which facilitates coupling to the Boltzmann solver in UFS. The gas kinetic solver employs embedded cut-cell approach for the treatment of solid boundaries. Cut cells of small size are merged with larger neighboring cells. The second solver is the “traditional” NS solver using either a Roe approximate Riemann solver with MUSCL reconstruction or an exact Riemann solver with a Godunov-type scheme. This solver was implemented following the methods of the NASCART-GT code (see [7]) and complement the previously developed gas-kinetic NS solver. The traditional solver provides better capabilities for highly viscous flows where low dissipation schemes are required for modeling turbulence effects. The traditional solver uses an IBM approach with ghost cells introduced inside solid boundaries without cut cells.

The numerical scheme used for the kinetic NS solver in UFS was described in detail in several publications.[8] For the gas-kinetic NS solver, the boundary conditions specified in terms of the velocity distribution function are translated into boundary conditions for the primitive variables. Our analysis has shown that the IBM approach can be directly used for these variables. The VOF approach was implemented by introducing a tracer function (e.g., named here as “T”), which is equal to 1 in the flow domain, equal to 0 in the solid domain and has values between 0 and 1 in the boundary cells. We then followed the NASCART-GT methodology to find the ghost and reference cells.

Below, we compare results obtained with the traditional NS solvers with IBM for gas flow around cylinder. Figure 3 shows the surface quantities (pressure coefficient, surface friction, and heat flux) for the M = 0.6, Re = 40 case. One can see that using IBM, the predicted surface quantities are smooth along the surface even on relatively coarse grids used for these simulations. The predictions using the cut-cell approach are noisy for such coarse grids, and their quality can only be improved by increasing the grid resolution around the solid surface.

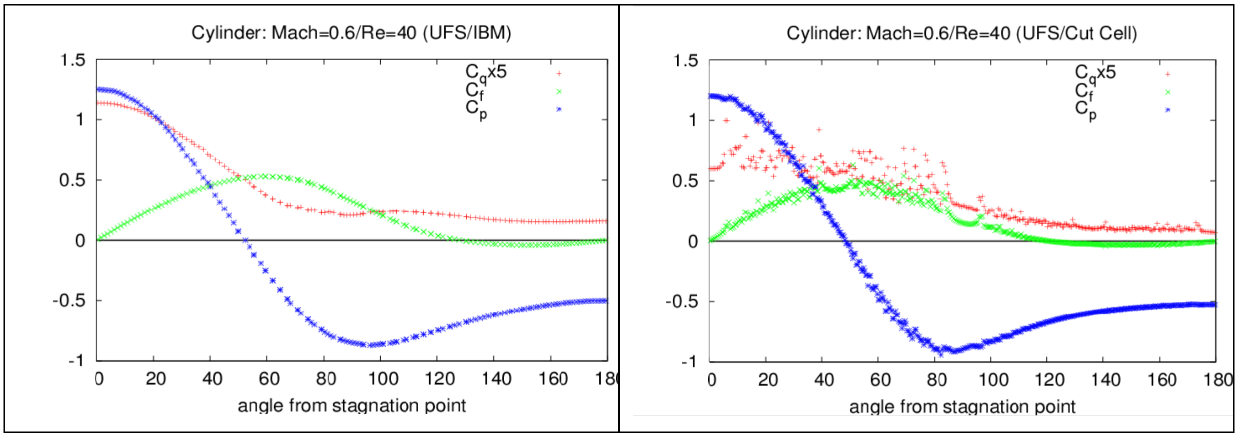

FIGURE 3. Results of calculations with IBM (left) and cut-cell (right) for M_∞=0.6, Re_∞=40.

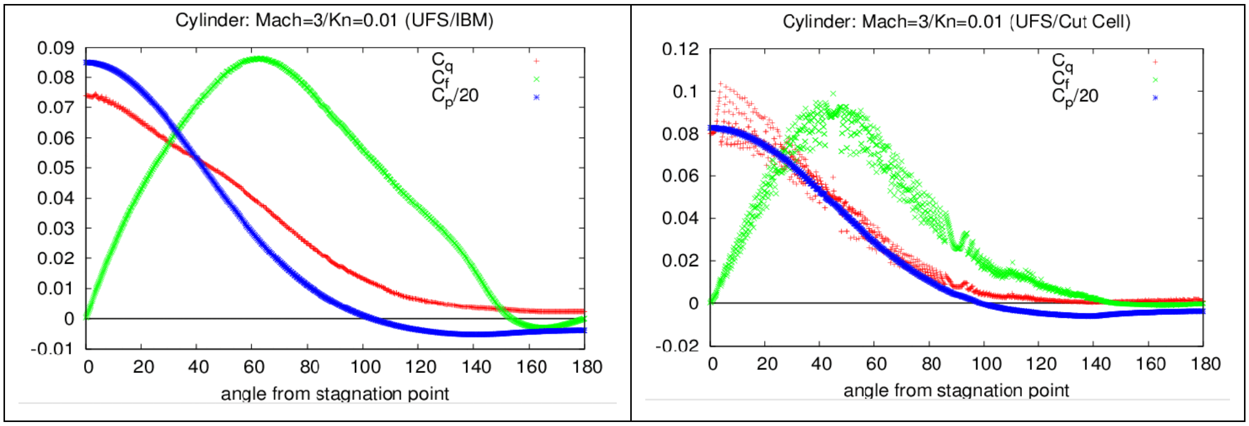

FIGURE 4. Results of calculations with IBM (left) and cut-cell (right) for M_∞=3, Kn_∞=0.01.

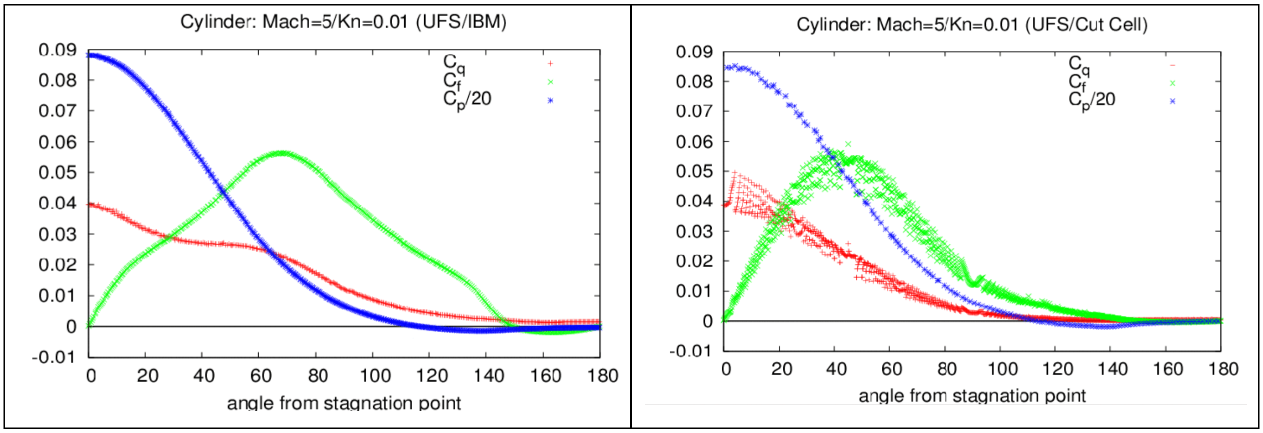

FIGURE 5. Results of calculations with IBM (left) and cut-cell (right) for M_∞=5, Kn_∞=0.01.

The IBM results shown above have been compared with the results of an in-house NS solver with a body-fitted grid shown in Figure 6. A very good agreement was observed.

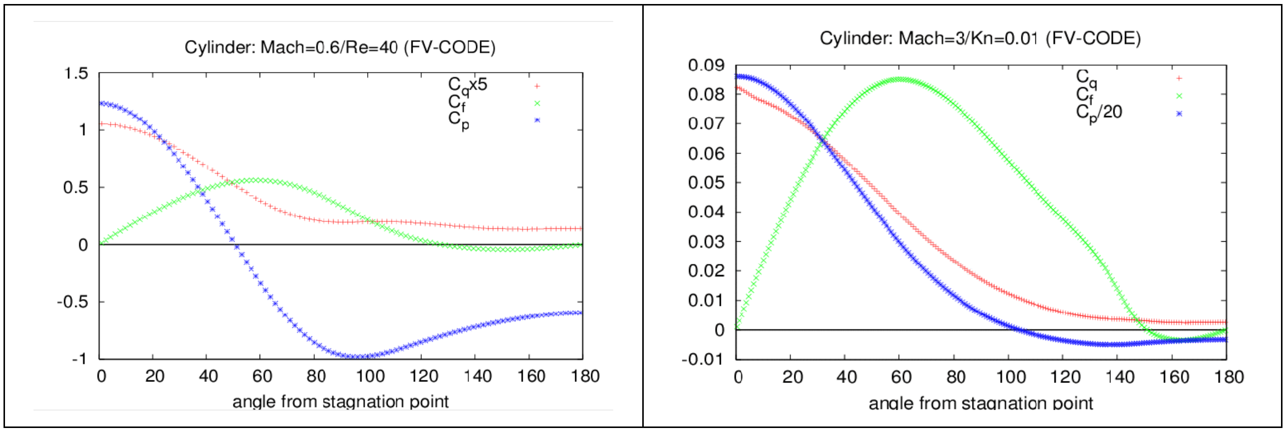

FIGURE 6. Results of calculations using an in-house NS code with a body-fitted grid at M_∞=0.6, Re_∞=40 (left) and M_∞=3, Kn_∞=0.01 (right).

Summing up, Immersed Boundary Method with adaptive octree Cartesian mesh allows one to eliminate unphysical fluctuations of skin friction and heat flux at solid boundaries for viscous flow simulations.

BOLTZMANN SOLVER

The numerical solution of the Boltzmann equation in UFS is obtained by splitting into advection in physical space

$$\frac{\partial f_i}{\partial t} + \nabla_r \cdot (\xi_i f_i) = 0 \qquad (1)$$

and relaxation stage due to collisions

$$\frac{\partial f_i}{\partial t} = S(f_i) \qquad (2)$$

Here ξ is the particle velocity, and S is the collision integral. We used a tree-based dynamically adaptive Cartesian grid to solve Eq (1) by traditional finite volume CFD techniques. Thus, implementation of IBM for the advection equation (1) follows the similar methodology as for the NS equation described above.

The following algorithm is used to calculate the distribution function in ghost cells. Instead of directly specifying boundary conditions on a wall, distribution functions are enforced on ghost cells which act as wall boundaries. Linear extrapolation is used from a reference point and a wall point to the ghost cell

$$f_{\mathrm{ghost}}(\xi_x, \xi_y, \xi_z) = (f_{\mathrm{wall}} - f_{\mathrm{ref}})\delta_{\mathrm{ghost}}/\delta_{\mathrm{ref}} + f_{\mathrm{wall}}, \qquad (3)$$

where $f_{\mathrm{wall}}(\xi_x, \xi_y, \xi_z)$ is the distribution function at wall and $f_{\mathrm{ref}}(\xi_x, \xi_y, \xi_z)$ is the distribution function at the reference point, and δ_{ghost} is the distance from the wall panel to the ghost cell center and δ_{ref} is the distance from the wall panel to the reference point. The reflected distribution function f_{wall} at the wall is computed based on the type of reflection: diffusive (with some given wall temperature), mirror or mixed (with some given accomodation coefficient α). In particular, the boundary conditions at the surface of solid objects define the distribution function of the reflected particles as a sum of diffuse and specular reflections with an accommodation coefficient α

$$f_{wall}(\xi) = \alpha f_M(\xi) + (1-\alpha) f_r(\xi) \quad . \qquad (4)$$

The specular reflection term is $f_r = f(\xi_r)$ where ξ_r is the velocity of an incoming molecule towards the boundary, which after specular reflection transforms into $\boldsymbol{\xi}_r = \boldsymbol{\xi} - 2(\boldsymbol{\xi}\cdot\mathbf{n})\mathbf{n}$ where **n** is a unit vector normal to the wall panel. The diffuse reflection term contains Maxwellian distribution $f_M(\boldsymbol{\xi})$ with a zero mean velocity, depending the temperature of the boundary, T_w, and the gas density n_w calculated to ensure zero particle flux at the boundary at a given point. At the boundaries of computational domain, for many problems, the distribution function can be assumed as Maxwellian $f_M(\boldsymbol{\xi})$ with a mean velocity **U** for $(\boldsymbol{\xi}\cdot\mathbf{n}) > 0$. For parts of the boundary with $(\boldsymbol{\xi}\cdot\mathbf{n}) < 0$, the distribution function is found from the solution.

The distribution function at the reference point, f_{ref}, is then computed using bi-linear interepolation based on values of *f* in neighboring cells. The cut cell is then considered as a full, flow cell and all fluxes across those cell faces between flow and ghost cells are computed taking into account the values of the distribution function in ghost cells computed from Eq. (3).

Results obtained with the Boltzmann solver with IBM will be added during the preparation of final paper.

ACKNOWLEDGMENTS

This work is partially supported by an AFOSR STTR project monitored by Dr John Schmisseur. We would like to thank Dr S. Ruffin for help in implementing IBM methods in UFS.

REFERENCES

[1] W.J.Courier, "An Adaptively Refined, *Cartesian Cell-Based Scheme for the Euler and Navier-Stokes Equations*, PhD Thesis, University of Michigan, Ann Arbor (1993)

[2] C.S.Peskin, "The Immersed Boundary Method", Acta Numerica (2002) 1

[3] V.I. Kolobov, R.R. Arslanbekov, V.V. Aristov, A.A. Frolova, and S.A. Zabelok, *Unified Solver for Rarefied and Continuum Flows with Adaptive Mesh and Algorithm Refinement*, J. Comp. Phys. 223 (2007) 589-608

[4] R.Mittal and G.Iaccarino, "Immersed Boundary Methods," Annu. Rev. Fluid Mech. 37 (2005) 239

[5] M.D.de Tullio, et al, "An immersed boundary method for compressible flows using local grid refinement", J. Comput. Phys. 225 (2007) 2098

[6] R.Ghias, R.Mittal, H.Dong, "A sharp interface immersed boundary method for compressible viscous flows", J. Comput. Phys. 225 (2007) 528

[7] Lee J.D.: Development of an Efficient Viscous Approach in a Cartesian Grid Framework And Application To Rotor-Fuselage Interaction. PhD Thesis, Georgia Tech (2006)

[8] K. Xu, "A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method," J. Comp. Phys. 171 (2001) 289.