AN IMMERSED INTERFACE METHOD FOR HIGH MACH NUMBER FLOWS

V. H. AURICHIO¹, A. CUCCHIERI¹, M. L. B. DE OLIVEIRA²

¹ Universidade de São Paulo, Instituto de Física de São Carlos
² Universidade de São Paulo, Instituto de Ciências Matemáticas e da Computação

Contact email: vinicius.aurichio@usp.br

ABSTRACT – We present an immersed interface method (IIM) for solving the 2D compressible viscous Navier-Stokes equations in the supersonic limit. We employ an IIM based on the ghost-point technique combined with a shock-detection method in order to handle discontinuities in the flow variables. The shock-detection step allows us to use a slower, more dissipative method near the shocks, while a faster, non-dissipative method is used away from the shock discontinuities. We present the results for a supersonic flow around a cylinder.

1. INTRODUCTION

When using a computational fluid dynamic solver, one is often interested in simulating flows involving complex domain geometries. Usually this involves creating a mesh that aligns exactly with the domain boundaries, and then proceed to use a numerical solver using the mesh as an input. Yet, producing high-quality meshes is a challenge on its own, and, for systems involving multiple immersed bodies, multi-block meshes or unstructured meshes must be employed. Immersed interface methods (IIMs) allow us to separate the computational grid from the internal boundaries. This means we can use a simple, fixed Cartesian grid to perform the calculations, irrespective of how complex the boundaries geometry might be.

For static geometries, the main difference among IIMs is how the boundary presence is accounted for, and they can be broadly divided in sharp-interface methods and diffuse-interface methods. As their names indicate, the former force the fluid to respect the boundary conditions at the interface exactly, while the latter distribute the effect of the boundary presence to a region around it. In this work we employ a method representative of the sharp-interface variety, called the ghost-point method, described in section 2.1.

As we are simulating high Mach number flows another type of discontinuity must be addressed: shock waves. Various types of shock waves can form in supersonic flows creating discontinuities in the flow properties. If not handled properly, the solver generates unphysical oscillations in the solution due to the discontinuities. Shock-capturing methods can handle those discontinuities by locally introducing more dissipation in regions containing large gradients, effectively smearing them until they are no longer true discontinuities. Most shock-capturing methods employ either a flux splitting or a flux-limiter scheme, and both are more computationally intensive than a central-difference scheme. This motivates us to use a hybrid method that switches from a fourth-order central finite-difference method to the WENO method (Shu (1997))—which is a shock-capturing method—in regions where discontinuities have been detected. The shock detector used
is presented in section 2.2. Finally, central-difference schemes have zero dissipation, but still introduce dispersion errors. Left alone, those dispersion errors build up during the simulation and generate a checkerboard-patterned oscillation. To mitigate this, we employ high-order filtering. Specifically, we employ a filter defined in Vasilyev et al. (1998).

It is important to mention the shorter runtime of a simulation with shock-detection and filtering steps in comparison with the use of a shock-capturing method only.

2. METHODOLOGY

We briefly describe the ghost-point method and the shock-detection step used to perform the simulation. We refer the reader to the literature for a more in depth description of the methods. Concluding this section, we show how these methods are combined in our solver.

2.1. Ghost-point based immersed interface method

![Illustration of the ghost-point construct.](image)

Figure 1 – Illustration of the ghost-point construct. The grid points are located at the intersection of the grid lines, and the body interface is shown in dark blue. Ghost points are shown in red, their associated boundary points are shown in black and the corresponding image points are shown in light blue. The values at the image points are computed using a bilinear interpolation constructed from the corners of the hatched areas.

We use a variation of the method proposed by Ghias et al. (2007), where points inside the body that have at least one of its eight immediate neighbors in the fluid become ghost points. As shown in figure 1, we create a normal projection of each ghost point onto the boundary and prolong the line for an equal distance into the fluid. This associates two points with each ghost-point: a point on the boundary, and a point in the fluid called image point. We then compute the image points values from a bilinear interpolation using known values, either from fluid points, or from boundary points. We can then employ a linear extrapolation using the computed image point values and the boundary conditions we want to enforce, in order to determine the ghost-point values.
2.2. Shock detector

For shock-detection we use the method introduced by de Oliveira e Pires (2016), i.e. we compute the ratio between sums of derivatives defined in a fine and a coarse grid. If a function is smooth enough, computing the derivatives in either grid will lead to very similar results, while if the function has a discontinuity in one of its derivatives, the results will differ greatly. The ratio between the computations on the fine and coarse grids gives a measure of the function continuity. We consider the use of the shock-capturing method to be necessary only if the density or its first derivative is discontinuous. This shock detector is defined for one spatial dimension; thus, in order to apply it in a two dimensional problem, we use it to detect shocks both line-wise and column-wise on the grid, and then we merge the results.

2.3. Solution steps

To perform the simulation we employ a total variation diminishing (TVD) third-order Runge-Kutta scheme. Before each time-step, the grid is filtered and the shocks are detected. At each Runge-Kutta substep we have to update the ghost-point values preparing the grid for the next substep.

3. RESULTS

Figure 2 – Density field near a cylinder immersed in a Mach 2.5 flow. The full domain is defined in the rectangle $[0, 15] \times [-5, 5]$ and has $601 \times 401$ equally spaced grid points. The cylinder has a diameter of 1 and is centered at $x = 4, y = 0$.

In figure 2 we present the result of applying the described IIM to a supersonic flow. We simulated a left-to-right flow at Mach 2.5 and Reynolds number of 500 around a
cylinder. We use characteristic boundary conditions for all the domain boundaries, being
the left domain boundary a supersonic inlet and the other three boundaries supersonic
outlets. We see that the bow shock is well resolved and unphysical oscillations are absent.
Keeping all other parameters fixed, a simulation using the hybrid method presented here
runs about four times faster than a simulation using the WENO method in the entire
domain.

4. CONCLUSIONS

We showed that it is possible to create an immersed interface method to solve the 2D
compressible viscous Navier-Stokes equations in high Mach number flows. The embedded
shock-detector is enough for the solver to be capable of handling discontinuities, while
avoiding the overhead of using the WENO scheme everywhere in the domain.

REFERENCES

DE OLIVEIRA, M. L. B.; PIRES, V. A. Analysis of a high-order finite difference detector

GIHAS, R.; MITTAL, R.; DONG, H. A sharp interface immersed boundary method for

SHU, C.-W. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory

VASYLYEV, O. V.; LUND, T. S.; MOIN, P. A General Class of Commutative Filters for