

Vehicle Aerodynamics Simulation for the Next Generation on the K Computer: Part 2 Use of Dirty CAD Data with Modified Cartesian Grid Approach

Keiji Onishi and Makoto Tsubokura
RIKEN

Shigeru Obayashi
Tohoku Univ.

Kazuhiro Nakahashi
JAXA

ABSTRACT

The applicability of high-performance computing (HPC) to vehicle aerodynamics is presented using a Cartesian grid approach of computational fluid dynamics. Methodology that allows the user to avoid a large amount of manual work in preparing geometry is indispensable in HPC simulation whereas conventional methodologies require much manual work. The new frame work allowing a solver to treat 'dirty' computer-aided-design data directly was developed with a modified immersed boundary method. The efficiency of the calculation of the vehicle aerodynamics using HPC is discussed.

The validation case of flow with a high Reynolds number around a sphere is presented. The preparation time for the calculation is approximately 10 minutes. The calculation time for flow computation is approximately one-tenth of that of conventional unstructured code. Results of large eddy simulation with a coarse grid differ greatly from experimental results, but there is an improvement in the prediction of the drag coefficient prediction when using 23 billion cells.

A vehicle aerodynamics simulation was conducted using dirty computer-aided-design data and approximately 19 billion cells. The preparation for the calculation can be completed within a couple of hours. The calculation time for flow computation is approximately one-fifth of that of conventional unstructured code. Reasonable flow results around a vehicle were observed, and there is an improvement in the prediction of the drag coefficient prediction when using 19 billion cells. The possibility of the proposed methodology being an innovative scheme in computational fluid dynamics is shown.

CITATION: Onishi, K., Tsubokura, M., Obayashi, S., and Nakahashi, K., "Vehicle Aerodynamics Simulation for the Next Generation on the K Computer: Part 2 Use of Dirty CAD Data with Modified Cartesian Grid Approach," *SAE Int. J. Passeng. Cars - Mech. Syst.* 7(2):2014, doi:10.4271/2014-01-0580.

INTRODUCTION

When conducting simulations based on computational fluid dynamics (CFD) in the field of automotive aerodynamics, there remains a major problem that the user must carry out much work in generating a high-quality grid, irrespective of whether the grid is unstructured or Cartesian. Recently, commercial software has been widely used in the industrial field. Additionally, the applied technology of CFD, which includes grid generation software that is easy to use, has made remarkable progress in the past 20-30 years [1]. For example, surface wrapping [2] and hex-dominant structured grids [3] allow users to generate a grid around complicated geometry easily. However, it still usually takes several days to weeks [4]

to prepare the geometry so that grid-generation software generates volume grids without failure. This difficulty in grid generation is mainly due to gaps and overlaps that are generated during data conversion by computer-aided-design (CAD) software or due to the many surfaces that have zero thickness, unnecessary surfaces inside parts, and tiny parts that have size less than the cell size. Such data are called a 'dirty' dataset. The user needs to remove or reshape the data with manual operations. This heavy workload of geometry preparation strongly hinders the broadening of applications in the automotive CFD field, and as such, it is an important issue to resolve.

Recent CFD research has reinstated the use of Cartesian grid approaches, especially in the case of the immersed boundary method (IBM) [5, 6]. The use of a Cartesian grid is advantageous in the treatment of complex geometries, but the treatment of a wall remains difficult. Mittal et al. [7] proposed a wall boundary model that can handle sharp geometry, but the model is still limited in that the geometry cannot have gaps, overlaps, or thin walls. The Lattice Boltzmann Method generates grids similar to Cartesian grids but it requires the cleaning up of geometries that have gaps and overwraps using surface wrapping for complex geometries such as engine compartments and underfloor parts [8]. Akasaka et al. [9] proposed a method that can handle a 'non-watertight' geometry. It has disadvantages when there is a narrow gap within geometries, and the calculation becomes unstable because it fails to find the point of intersection between a wall and grid lines passing through the geometries when it defines the wall boundary condition. Therefore, a method that uses arbitrary boundary representation and a dummy-cell technique based on a Cartesian grid and modified IBM was proposed by the author [10].

The accuracy of modeling near a wall is not only governed by the geometry representation but also the turbulent boundary layer. Tessicini et al. [11] compared analytical results with results obtained in an IBM-large eddy simulation (LES) using a logarithmic wall function and discussed the deviations with results obtained in a fully resolved LES. Capizzano et al. [12] proposed a wall diffusion model that can generate inner boundary layer points using a sub-grid approach in a Reynolds-averaged Navier-Stokes (RANS) simulation to get an accurate flow profile within the boundary layer, but they have not yet reported an extension of the model to include a convective term or to handle three-dimensional problems.

Considering the future development of computers, modeling of the boundary layer using high-performance computing (HPC) technology will become a viable alternative [13, 14]. Directly using a finer grid near the wall, the physical modeling can be simplified compared with the RANS approach. In other words, the use of a faster computer presents the possibility of improving the accuracy of the boundary layer automatically. However, there will be no automatic improvement in the accuracy of shape representation because greater care will need to be taken with the detail of the geometry. In addition, on next-generation computers using many cores on a chip, conventional methodologies, especially those employing an unstructured grid, will have limited performance because of the complexity of calculation procedures.

Therefore, in this study, a new framework is developed to allow users to conduct a vehicle aerodynamics simulation in a massively parallel environment adopting the modified IBM. Results for flow with a high Reynolds number around a sphere and a vehicle are shown, and the efficiency of work flow compared with that of the conventional method is discussed.

NUMERICAL METHOD

The modified IBM is a technique based on the IBM that allows a solver to treat 'dirty' CAD data of a product as they are. Additionally, this study employs the building cube method (BCM) [15, 16] as the basis of the Cartesian grid approach.

The BCM is a computational method with which to calculate the flow field using uniform Cartesian grids in the spatial domain. The space is discretized with cubic units called cubes, as shown in Fig. 1. Cubes are regularly divided so that neighboring cubes are of the same size or have lengths that differ by a factor of 2; the smallest unit is called a cell. The algorithm of the computation is quite simple, and it maintains the same number of elements (cells) within each cube. As such, the technique has high efficiency in large-scale parallel computation. To keep the problem simple, a wall boundary represented by a stair shape is usually used. The IBM is used to establish the boundary condition in this study.

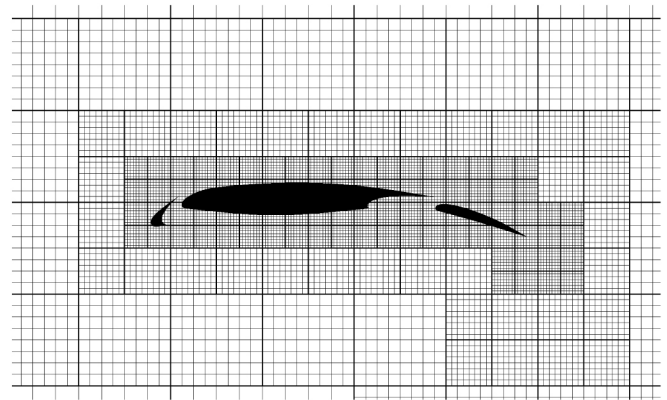


Figure 1. Example of a BCM grid structure. Bold lines indicate cubes and thin lines indicate cells.

A spatially filtered incompressible Navier-Stokes equation with the IBM forcing term expressed in Eq. (2) and the continuity equation expressed in Eq. (1) are employed as governing equations.

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \quad (1)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{2}{Re} \frac{\partial}{\partial x_j} (1 + \nu_{SGS}) \bar{S}_{ij} + \bar{f}_i, \quad (2)$$

where u_i and p are the velocity component of each axis, and pressure is defined on the cell center in the collocated grid, and U_i is a contravariant velocity component that is defined on the face center. Re is the Reynolds number of the flow, and f_i is a forcing term. The over-bar denotes spatially filtered quantities. The standard Smagorinsky model [17] is used to model the subgrid-scale (SGS) eddy viscosity ν_{SGS} , and S_{ij} is the strain rate tensor:

$$v_{SGS} = (C_s \Delta f_\mu)^2 (\overline{S_{ij} S_{ij}})^{1/2}, \quad (3)$$

$$\overline{S_{ij}} = \frac{1}{2} \left(\frac{\partial \overline{u_j}}{\partial x_i} + \frac{\partial \overline{u_i}}{\partial x_j} \right), \quad (4)$$

$$f_\mu = 1 - \exp(-y^+/26), \quad (5)$$

where Δ is the width of the spatial filter determined by the cube root of the volume of the cell. A model coefficient C_s of 0.15 is used; is the coefficient was validated by channel flow analysis and is vital in defining the effect of eddy viscosity. As for the damping of the effect of v_{SGS} in the vicinity of a solid boundary, the van Driest damping function f_μ is used, where y^+ is the wall distance.

The governing equations are discretized with a second-order central-difference scheme. To stabilize the flow field, especially on the interfaces of each cube, the results of a first-order upwind scheme are blended by approximately 5% for the advection term. In the case of vehicle aerodynamics, additional viscosity is required in the advection term, and the quadratic upstream interpolation for the convective kinematics scheme (QUICK) is thus used to stabilize the flow. Rhie-Chow interpolation [18] is adopted to eliminate the unphysical oscillation on the collocated grid. To solve the time integration, a second-order Adams-Bashforth explicit scheme is used in the case of a sphere, and a second-order Crank-Nicolson implicit scheme is used to solve the advection-diffusion term in the case of a vehicle; the scheme is based on a fractional step using an intermediate velocity that is corrected with the pressure gradient obtained from a Poisson equation:

$$\begin{aligned} \frac{u_i^* - u_i^n}{\Delta t} + \frac{3}{2} U_j^n \frac{\partial u_i^n}{\partial x_j} - \frac{1}{2} U_j^{n-1} \frac{\partial u_i^{n-1}}{\partial x_j} \\ = \frac{2}{Re} \frac{\partial}{\partial x_j} (1 + v_{SGS}) S_{ij} + f_i^n, \end{aligned} \quad (6)$$

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = - \frac{\partial p^{n+1}}{\partial x_i}, \quad (7)$$

$$\frac{\partial}{\partial x_j} \frac{\partial p^{n+1}}{\partial x_j} = \frac{1}{\Delta t} \left(\frac{\partial u_j^*}{\partial x_j} \right). \quad (8)$$

The Poisson equation and the implicit time integration are solved using the red/black ordered successive over-relaxation method, which can be parallelized to solve a linear equation system on multi-processors.

To define a boundary condition on a wall, first, the equation of the plane is determined from the characteristic face center and face normal for each cell, as shown in Fig. 2. A circle with

diagonal lines indicates the characteristic face gravity center, and a bold arrow indicates the characteristic normal of a wall within a cell. Complicated surfaces are divided into sub-polygons by splitting surface elements with the faces of the cell. The characteristic plane equation is obtained by summing all face centers and normals of subdivided polygons within each cell. The plane will be treated as the characteristic wall in the cell. Next, the points of intersection between this characteristic plane and the grid lines that connect the cell centers are calculated. In Figs. 2 and 3, a white circle indicates an intersecting point.

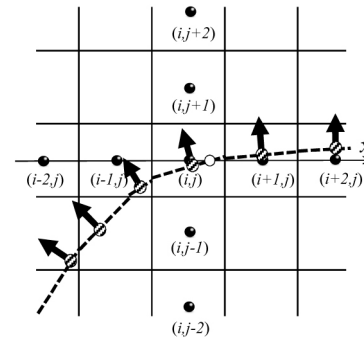


Figure 2. Two-dimensional expression of grid stencil lines with characteristic wall information.

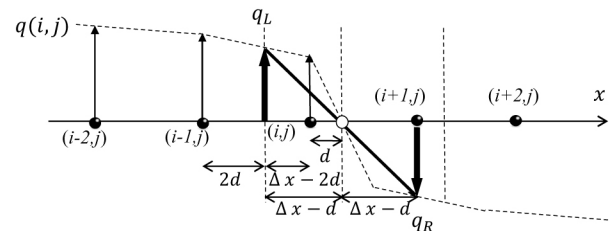


Figure 3. Definition of direct forcing for the ghost cell on the axis line around the wall intersecting point.

In the same way as in the 'ghost cell' approach of Mittal et al. [7], an imaginary point is defined by the distance from the cell center towards the opposite side of the intruding fluid region. The fluid velocity at the imaginary point, expressed as q_L in Eq. (9), is then calculated through linear interpolation using the distance from the intersecting point, as shown in Fig. 3. The sign-inverted velocity, q_R in Eq. (10), is imposed directly onto the ghost cell on the solid side.

$$q_L = \frac{\Delta x - 2d}{\Delta x} q(i-1, j) + \frac{2d}{\Delta x} q(i, j), \quad (9)$$

$$q_R = -q_L. \quad (10)$$

This is the approach of direct forcing of the IBM proposed by Mohd-Yusofz [19]. The forcing term f_i is obtained directly from the imposed velocity satisfying a boundary condition of the next time step $u^{l+1} = V_i^{l+1}$, using the discrete form of Eq. (6) and discrete time l :