

Vehicle Aerodynamics Simulation for the Next Generation on the K Computer: Part 1 Development of the Framework for Fully Unstructured Grids Using up to 10 Billion Numerical Elements

Makoto Tsubokura and Andrew Kerr
Hokkaido Univ.

Keiji Onishi
RIKEN AICS

Yoshimitsu Hashizume
Suzuki Motor Corp.

ABSTRACT

A simulation framework for vehicle aerodynamics using up to 10 billion fully unstructured cells has been developed on a world-fastest class supercomputer, called the K computer, in Kobe, Japan. The simulation software FrontFlow/red-Aero was fully optimized on the K computer to utilize up to 10,000 processors with tens of thousands of cores. A hybrid parallelization method using MPI among processors and OpenMP among cores inside each processor was adopted. The code was specially tuned for unsteady aerodynamic simulation including large-eddy simulation, and low Mach number approximation was adopted to avoid excessive iterations usually required for the fully incompressible algorithm.

The automated mesh refining system was developed to generate unstructured meshes of up to 10 billion cells. In the system, users only generate unstructured meshes in the order of tens of millions of cells directly using commercial preprocessing software. Then the original elements are refined in the preconditioning process using the K computer, and billions of cells are automatically generated. In the same way, the CFD results of the billions of elements are mapped on the original coarse grid points during post processing, and the users can transport the data from the K computer without any difficulty visualizing the results using conventional post-processing software.

The simulation framework was validated on vehicles with different rear slant angles, and dependence of the aerodynamic forces on the unstructured-mesh resolution as well as the grid topology near the body surface was discussed. It was demonstrated that the simulation of the multi-billion cell mesh on the K computer agreed with wind-tunnel measurements within a small percentage of error, and validity of the fully unstructured finite volume method using multi-billion cells was confirmed.

CITATION: Tsubokura, M., Kerr, A., Onishi, K., and Hashizume, Y., "Vehicle Aerodynamics Simulation for the Next Generation on the K Computer: Part 1 Development of the Framework for Fully Unstructured Grids Using up to 10 Billion Numerical Elements," *SAE Int. J. Passeng. Cars - Mech. Syst.* 7(3):2014, doi:10.4271/2014-01-0621.

INTRODUCTION

Computational Fluid Dynamics (CFD) has been widely used to develop vehicle aerodynamics and has now replaced most of the aerodynamic assessment processes once measured in wind tunnels. Undoubtedly, CFD contributes a lot to reducing man-hours and the overall cost of vehicle development. The two main reasons for this permeation of CFD in industrial processes are usability and multifunctionality of commercial CFD software and rapid advancement in computer hardware in recent decades. As far as we know, conventional CFD

simulations of vehicle aerodynamics in the development process use commercial software based on the unstructured finite volume (FVM) or lattice Boltzmann method (LBM). For the hardware, general-purpose Linux cluster computers are commonly used that consist of thousands of cores with a peak performance of some tens of TFLOPS (FLOPS: floating-point operations per second). In fact, the current CFD framework for vehicle aerodynamics including the grid topology/resolution, the numerical methods, and the turbulence model have been determined by considering the following factors: the cost of hardware resources and software licensing fees, the required

simulation turnover time (typically and hopefully less than five days including the pre- and post-processes); and its accuracy (less than five percent for the drag coefficient against the wind-tunnel data) in the development process.

In the case of the unstructured FVM, the surface of vehicles including their engine compartment and complicated underbody geometry are reproduced with the resolution of typically around 5 to 10mm using triangular elements. The flow field is divided mostly by tetrahedral elements, while spatial resolution near the wall does not satisfy the requirement of the low-Reynolds number turbulence model. Thus, the surface is often covered by prism layers, which usually requires additional time for mesh generation. The total cell number for these simulations is from around tens of millions up to one or two hundred million. Due to the restriction of the turnover time for simulations, Reynolds Averaged Navier-Stokes (RANS) is commonly used in this simulation framework. While LBM (e.g. [1]) requires a relatively finer resolution of 1mm to 5mm for their voxel type grids, the total number of cells becomes relatively similar to that of FVM (tens of millions) because of the advantages of the voxel mesh. Note that the merit of the voxel type LBM is its lower requirements for surface cleanup and repair of the dirty CAD data than FVM, which requires mesh points to be fitted on the surface. In addition, unlike RANS, LBM can treat transient phenomena of flow. In both cases, cores numbering from hundreds up to a thousand are typically utilized for these simulations to obtain the results in the course of a night.

Considering these facts, the advantage of unstructured FVM is its highly accurate reproduction of complicated geometry by directly fitting the numerical grid to the surface, while its disadvantages are relatively higher computational load and parallel simulation performance than LBM. These disadvantages force us to use RANS instead of unsteady simulation like LES as well as additional ad hoc prism layers on the surface.

On the other hand, the history of the state of the art supercomputer indicates that its peak performance becomes about one thousand times faster for every decade, and now the performances of the world fastest class supercomputers have reached tens of PFLOPS. From the fact that the performance of conventional computers used in car companies for CFD so far is around tens of TFLOPS, we can say that most car companies will obtain computers that have peak performance in the order of PFLOPS within ten years for their next-generation CFD research. From this viewpoint, we can give an account of the CFD techniques for vehicle aerodynamics that will be used in the next decade by using the world's current fastest class of supercomputer. In fact, it is expected that High-Performance Computing (HPC) using these next-generation supercomputers will solve the problems of unstructured FVM mentioned above.

Accordingly, the objective of this study is to develop a simulation framework for vehicle aerodynamics for the next generation using a world-fastest class supercomputer, called K-computer, in Kobe, Japan. The K computer was installed in 2012 at the RIKEN Advanced Institute for Computational Science (AICS), consists of over 80,000 processors of a distributed memory architecture with its peak performance of about 10 PFLOPS, and is ranked as the world's fourth fastest computer as of November, 2013. In this study, we focus on the unstructured FVM adopted in the simulation software FrontFlow/red-Aero developed by our group ([2][3][4]). In fact, this research is a follow-up to our previous challenge on a previous world-fastest Earth Simulator (40TFLOPS in 2002), in which we modified Frontflow/red for vector processors and realized an aerodynamics simulation of a formula car using 120 million fully unstructured cells [5].

The software was fully optimized on the K computer to utilize up to 10,000 processors with tens of thousands of cores. A hybrid parallelization method using Message Passing Interface (MPI) among multiple processors and OpenMP among cores inside the processor was adopted. The software was optimized for unsteady aerodynamic simulation, thus Large-Eddy Simulation (LES) was adopted for the turbulence simulation. Note that, depending on the resolution for the boundary layer on the body surface, we utilize wall models. To solve the pressure field and its coupling with the velocity field, an incompressible flow algorithm based on simplified marker and cell (SMAC) was originally adopted. However, it was recognized in this study that such a massively large number of elements requires an excessive amount of iterations for convergence of the pressure Poisson solver. Thus, we have recently adopted a weakly compressible algorithm [6] to reduce the computational load required to solve the pressure field.

In addition, the challenging to realize a massively parallel simulation for vehicle aerodynamics is how not only to tune the simulation software itself but also to generate the billions of fully unstructured numerical elements. Thus, we have developed an automated mesh refining system. In the system, users only generate unstructured grids in the order of tens of millions of elements by hand using commercial preprocessing software. Then the original elements are refined in the preconditioning process using the K computer, billions of elements are automatically generated, and the simulation is subsequently conducted. In the same way, if a user so requires, the results of the billions of elements are mapped on the original coarse grid during post processing and the user can transport the data from the K computer without difficulty. The results can then be visualized using conventional software. That is to say, the user can conduct the massively parallel simulation of billions of elements, without the consciousness of doing so, in a conventional CAE environment.

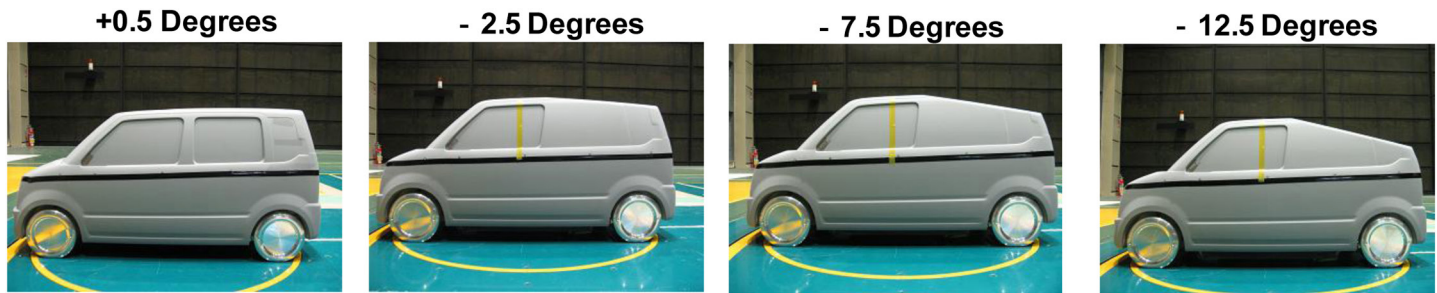


Figure 1. 33% scale wind-tunnel models of different rear slant angles.

The simulation framework was validated on vehicles with different rear slant angles, and dependence of the aerodynamic forces on the unstructured-grid resolution as well as the grid topology near the surface was investigated. In this study, two topologies of the mesh near the wall were compared: one is a full tetrahedral mesh (an original mesh in the order of tens of millions of cells and then refined billions of cells) and the other is the hybrid tetra/prism mesh (original mesh in the order of tens of millions of cells). The advantage of the full tetra mesh is its ease of mesh generation, while the troublesome prism meshes on the surface are always necessary in the conventional CAE environment under the restriction of the computational resources with the total cell count of less than one hundred million. In this study, we have confirmed that, in the simulation using the multi-billion cells, auto-refined tetra mesh on the K computer agrees with wind-tunnel measurements within a small percentage of error. Finally, some issues regarding surface modification from so-called dirty CAD data for fully unstructured meshes are discussed.

TARGET MODELS

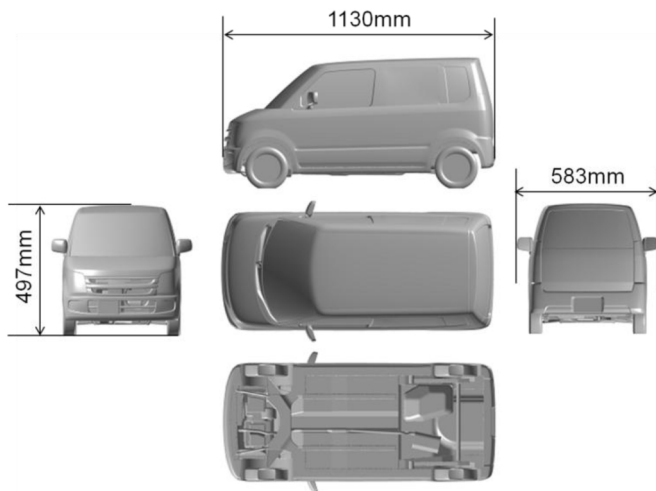


Figure 2. Overview of the model (slant of 0.5 deg.) used for simulations

The models used for validation in this study are based on a production vehicle developed and supplied by the Suzuki Motor Corporation. From the original model with the slant angle of +0.5 degrees, three other models with different angles were made for studying their dependence on the aerodynamics

forces, as shown in Fig. 1. These models share the same simplified geometry of 33% scale as that of the models used to conduct experimental wind tunnel research and include simplified surfaces and parts such as the engine, suspension components, and underbody. Figure 2 shows an overview of the geometry and dimensions of the model.

NUMERICAL METHODS AND CONDITIONS

Governing Equations

The governing equations being solved in the LES are spatially filtered continuity and Navier-Stokes equations:

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

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{\partial \bar{P}}{\partial x_i} + 2 \frac{\partial}{\partial x_j} (\nu + \nu_{sgs}) \bar{S}_{ij} = 0 \quad (2)$$

$$\bar{P} = \bar{p}/\rho + (\bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j)/3 \quad (3)$$

where u_i , p , ρ , and ν are the i -th velocity component, pressure, density, and kinematic viscosity of the fluid. The over-bar denotes spatially filtered quantities. The strain rate tensors S_{ij} were defined as

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j} \right) \quad (4)$$

The standard Smagorinsky model [7] was used to model the subgrid-scale (SGS) eddy viscosity ν_{sgs} in Eq. (2):

$$\nu_{sgs} = (C_s f \Delta)^2 \sqrt{2 \bar{S}_{ij} \bar{S}_{ij}} \quad (5)$$

where Δ is the width of the spatial filter, which was determined by the volume of numerical elements. A model coefficient C_s of 0.15 was used. As for the damping of the effect of ν_{sgs} in the vicinity of the solid boundary, Van Driest damping function f_d was used:

$$f_d = 1 - e^{-y^+ / 25} \quad (6)$$

where y^+ is the distance from the solid surface in wall coordinate.

Discretization

The governing equations were discretized in space using a vertex-centered unstructured finite volume method. In this method, each dependent variable on the vertex of the numerical elements is defined and a virtual control volume around the vertex is re-constructed from the originally generated mesh. Figure 3 shows a simplified two dimensional graphical illustration of a vertex-centered control volume. The governing equations were integrated over the volume. The second-order central differences were applied for the spatial derivative, blended with the first-order upwind scheme for the convective term in the Navier-Stokes equations to avoid excessive numerical oscillation appearing at coarse tetrahedral cells. Note that the dissipation property of upwind schemes is desirable to a certain extent for engineering applications of LES on unstructured meshes. As a compromise, the contribution of the upwind discretization to the convective fluxes was set to be as low as 5%. The diffusive fluxes on the volume surface were treated on the basis of the deferred correction formula suggested by Muzafferija [8] to avoid checkerboard oscillations. The monotone upwind scheme for conservation laws (MUSCL) by van Leer [9] was also adopted for the convective term at the region away from the vehicle where upwind dissipation does not affect the wake or near-wall turbulence of the target vehicle.

The time marching was based on the fractional step method by Kim and Moin [10], in which the third-order Adams-Moulton scheme or the implicit Euler scheme was adopted for the velocity prediction step. For the framework of incompressible CFD, the coupling of the velocity and pressure fields to obtain the pressure and to correct the velocity was based on the simplified marker and cell (SMAC) method by Amsden and Harlow [11]. The flow rate on the control-volume surface was estimated in accordance with the method proposed by Rhie and Chow [12] to stop checkerboard pressure oscillations. The pressure Poisson equation was solved using the incomplete Cholesky conjugate gradient (ICCG) method. However, we have realized that this incompressible algorithm is not suitable for the simulation of billions of unstructured elements because of the excessively large number of iterations required for the pressure Poisson to satisfy the fully incompressible condition. Thus, we have also adopted a numerical scheme [6] based on the weakly compressible condition for low Mach number flow. The derived equations are essentially the same as the incompressible Navier-Stokes as indicated in Eqs. (1) and (2), except for additional terms for the weakly compressible effect. Thus, we can apply almost the same algorithm adopted for the incompressible flow (pressure-based method) mentioned above.

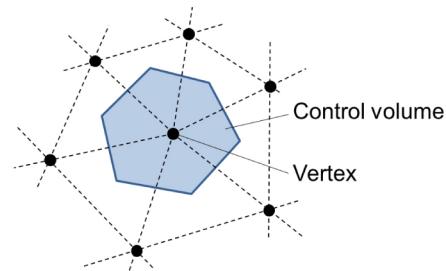


Figure 3. Vertex-centered control volume

Computational Domain and Boundary Conditions

The vehicle models were mounted on the floor of a rectangular numerical domain that was 25.0m, 10.0m, and 5.0m (the blockage ratio of the vehicle's projection area and the cross section of the domain is less than 0.5%) for streamwise, spanwise, and normal-wall directions, respectively. The center line of the vehicle is aligned to that of the rectangular domain, and the leading edge of the vehicle is set 5m downstream from the inlet plane. At the inlet plane, uniform velocity of 200km/h is imposed, while a simple gradient-free condition is used on the outlet plane. For the side and ceiling of the domain, a free-slip velocity boundary condition is imposed. The corresponding Reynolds number with respect to the vehicle height of 0.50m is 2.68×10^6 . On the surface of the vehicle and the floor, a universal velocity profile of the fully developed turbulent boundary layer is assumed, and the surface friction stress on the surface is estimated and given as a Neumann type boundary condition for the velocity field.

Software "FrontFlow/Red"

The computational code FrontFlow/red (FFR) adopted in the present study was originally developed in a project called Frontier Simulation Software for Industrial Science [13]. The project started in 2002 as an IT research program sponsored by the Ministry of Education, Culture, Sport, Science and Technology and was successfully concluded in 2005. The code was intensively optimized by the authors for LES applications under the subsequent IT project, Revolutionary Simulation Software (RSS21) [14], and parallel and vector efficiency as high as 96% and 99%, respectively, were achieved in simulations performed on 100node/800cpu. This allowed LES calculations of flow around a formula car with complex geometry to be completed in about 120 hours on the Earth Simulator in 2006 [5]. In the simulation, the computational meshes adopted numbered up to 120 million cells with a typical memory usage of more than 500 GB, also including data storage for the calculation of time-averaging turbulence statistics. The code was then optimized for unsteady vehicle aerodynamics simulations by coupling vehicle motion equations with CFD [3] under the project supported by an Industrial Technology Research Grant Program in 2007 from the New Energy and Industrial Technology Development Organization (NEDO) of Japan and was named FrontFlow/red-Aero. The purpose of this study is to optimize the