Journal of Computational Science and Technology

Vol.3, No.2, 2009

Moving Computational Domain Method and Its Application to Flow Around a High-Speed Car Passing Through a Hairpin Curve*

Koji WATANABE** and Kenichi MATSUNO**

** Department of Mechanical and System Engineering, Kyoto Institute of Technology,
Matsugasaki, Sakyo-ku, Kyoto, 606-8585, JAPAN
E-mail: matsuno@kit.ac.jp

Abstract

This paper presents a new method for simulating flows driven by a body traveling with neither restriction on motion nor a limit of a region size. In the present method named 'Moving Computational Domain Method', the whole of the computational domain including bodies inside moves in the physical space without the limit of region size. Since the whole of the grid of the computational domain moves according to the movement of the body, a flow solver of the method has to be constructed on the moving grid system and it is important for the flow solver to satisfy physical and geometric conservation laws simultaneously on moving grid. For this issue, the Moving-Grid Finite-Volume Method is employed as the flow solver. The present Moving Computational Domain Method makes it possible to simulate flow driven by any kind of motion of the body in any size of the region with satisfying physical and geometric conservation laws simultaneously. In this paper, the method is applied to the flow around a high-speed car passing through a hairpin curve. The distinctive flow field driven by the car at the hairpin curve has been demonstrated in detail. The results show the promising feature of the method.

Key words : Numerical Method, Finite-Volume Method, Moving Grid, Unsteady Flow, Car Aerodynamics, Moving Boundary,

1. Introduction

In recent years, vehicle aerodynamics have gained increased attention for yielding performance improvements(1). The present main interest on car aerodynamics is in general the estimation and generation of downforce, and thus many simulations have been performed to estimate the downforce. In order to evaluate aerodynamics of the car, simulations of flows around the car in a uniform flow have been usually performed. This situation corresponds to a case that the car runs on a straight-line road. These simulations might give various information about forces acting on the car and are important for design of the high performance car. However, from the view point of total performance of the race car, the aerodynamics at various scenarios should be considered because the car has to run not only on straight-line road but also through hairpin curves. The estimation of aerodynamic forces acting on the car at the hairpin curve or in a S-shape run will be important for not only a next generation race car but also a sedan. As stated in Reference(1), the improved cornering due to the use of aerodynamic downforce led to the dramatic increase in cornering speed.

One of the focuses of this paper is on methods of simulation, which can estimate aerodynamic performance of the car passing through the hairpin curves. Simulation by Computational Fluid Dynamics is a powerful tool for estimation of car aerodynamics as widely recognized. The simulation of the car running on the straight-line road is easy and widely performed. However, the simulation of the car running at the hairpin curve is not found in a literature as far as the authors know. In order to simulate the flow around the car passing

*Received 22 June, 2009 (No. 09-0284)
[DOI: 10.1299/jcst.3.449]

through the hairpin curve, two approaches might be considered. The one is to use a non-inertial coordinate system or moving coordinate system, in which the governing equations are transformed and thus the governing equations include extra terms of apparent forces, such as the Coriolis force, centrifugal force. In the case of the car in the S-shape run, the governing equations become further complex. Therefore, this approach is not considered in this paper. The other approach, which might be more popular and general, is to directly compute the car moving in the computational domain. Since the car moves in the fluid in this approach, this is so called 'moving boundary problem', where the boundary moves and flow is driven due to the movement of the boundary. One of the most popular methods for such a moving boundary problem is an overset grid method(2), where sub-grid placed around a moving-body moves on the main-grid placed in computational domain with interchanging the data between the sub-grid and the main-grid by simple tri-linear interpolation procedure. The overset method, unlike body-fitted single grid system, has the advantage of a high degree of freedom of the body motion and ease of calculation. The overset method, however, includes inherent unphysical procedure of interpolation of flow variables from main-grid to sub-grid and vice versa without consideration of physical laws. Thus the overset method breaks conservation laws at the interpolation points and might be a source of inaccuracy. The other drawback of the overset method is, as is any other current methods, the limit of computational regions, in other words, the sub-grid which includes the body can move only inside the main-grid prepared beforehand.

The second focus of this paper is to perform simulations without a limit of the physical region size. There often exist the cases that the size of the computational region is not known beforehand. For example, when the movement of the body is strongly influenced by disturbances or some kind of interaction, the movement of the body can not be determined a priori. Conventional simulation methods have to estimate the enough size of the computational region and prepare the grid covering the whole of the region. Moreover, if we simulate a full course of racing track the huge computational region and related huge amount of grid point will be necessary. Hence, if we perform the simulation of the flow around the car in various types of motion, such as straight-line run or S-shape run, without the restriction of the computational region size, we should look for another new approach.

The purpose of this paper is to propose the new method that does not have such restrictions mentioned above and can simulate any kind of motion without the limit of the region size. The present method is not the one such that the body moves in the fixed computational domain but the method such that the whole of the computational domain with the body inside moves according to the movement of the body in the considering physical space. We call this new method 'Moving Computational Domain Method', in short, MCD method in this paper. Thus the MCD method can consider the region without limit and thus it might be said that at the extreme the computational region of the MCD method is infinite. The only necessary assumption is that the conditions just in front of the computational domain should be known a priori, such as, stationary fluid state or uniform flow and so on. In this paper, the MCD method is presented and applied to the simulation of the flow around a high-speed car passing through a hairpin curve.

2. Moving Computational Domain Method

2.1. Moving Computational Domain Method

The basic coordinate system of the MCD method is the general, fixed, stationary (x, y, z) Cartesian coordinate system. The computational domain itself, including the body inside, moves in the fixed (x, y, z)-space, as illustrated in Fig.1. The flow around the body is calculated as the moving boundary problem. Unknown flow variables, such as density ρ, x-directional momentum ρu and so on, are defined at each grid cell center in the computational domain. The motion of the computational domain according to the motion of the body in the physical space is arbitrary, and thus the any kind of the motion of the body, such as straight-line motion,

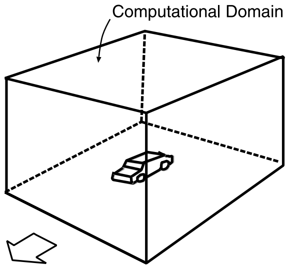


Fig. 1 Image of the Moving Computational Domain method

turning motion and their combination, can be simulated by the MCD method. The flow field driven by the body is calculated in the computational domain in which the body fitted grid system is used. Since the computational domain itself moves in the physical (x, y, z) space time-dependently and thus the grid system of the computational domain also moves in the (x, y, z) space, the flow solver has to be constructed for the moving grid system. In the present MCD method, the Moving-Grid Finite-Volume method, which has been proposed and developed by the authors(3)(4)(5), is adopted. Only necessary and essential assumption is that the condition in front of the moving computational domain has to be known because it is necessary as a boundary condition of the flow solver. This assumption is not severe in general. The natural assumption may be the stationary fluid condition in front of the moving computational domain.

2.2. Governing Equations

The three-dimensional Reynolds-averaged Navier-Stokes equations for compressible flow are written in the conservation law form in the x-, y-, z-Cartesian coordinate system as follows:

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}(E - E_v) + \frac{\partial}{\partial y}(F - F_v) + \frac{\partial}{\partial z}(G - G_v) = 0, \qquad (1)$$

where $Q = [\rho, \rho u, \rho v, \rho w, e]^T$ is a vector of conserved variables, E, F and G are convective flux vectors, E_v, F_v, and G_v are the viscous flux vectors. Density is denoted by ρ and u, v and w are velocity components in the x, y and z directions, respectively, e is the total energy. Pressure p is obtained by the equation of state for a perfect gas:

$$p = (\gamma - 1)\left[e - \frac{1}{2}(u^2 + v^2 + w^2)\right], \qquad (2)$$

where γ is the specific heat ratio. For viscous flows with high Reynolds number, Spalart-Almaras one-equation model(9), which is standard for high speed-flows and has been successfully applied to various type of flows, is employed.

Equation (1) can be represented in a divergence form using extended divergent operator $\tilde{\nabla} = (\partial/\partial x, \partial/\partial y, \partial/\partial z, \partial/\partial t)$ and flux tensor $\tilde{\mathbf{F}} = (E - E_v, F - F_v, G - G_v, Q)$ as follows :

$$\tilde{\nabla}\tilde{\mathbf{F}} = 0 \qquad (3)$$

Equation (3) means that the governing equations are divergence-free in (x, y, z, t) space-time unified four-dimensional space.

The flow solver of the present MCD method is the Moving-Grid Finite-Volume Method, and is based on this divergence-free form, Eq.(3), in the (x, y, z, t) unified four dimensional space, which is briefly described in next sub-section.

2.3. Moving-Grid Finite-Volume Method

For the present MCD method, since the computational grid itself moves with the computational domain in the physical space, the numerical flow solver on the moving grid system is necessary. For simulations of compressible flows on the moving grid system, it is crucial for the numerical scheme to satisfy both the physical conservation law and the geometric conservation law simultaneously. The Moving-Grid Finite-Volume method is incorporated in the present MCD method. The Moving-Grid Finite-Volume method is constructed based on the divergence form of the governing equation, Eq.(3), and the control volume in (x, y, z, t) space-time unified four-dimensional space for three-dimensional flow problems. The flow variables, $Q^n_{i,j,k}$ at time t^n, are defined at the center of the grid-cell in three-dimensional space as usual. Since the grid is moving in the (x, y, z) physical space, the grid-cell defining flow variables is also moving according to the grid motion. The control volume of the Moving-Grid Finite-Volume method, Ω, stands in unified four-dimensional space-time (x, y, z, t) -domain and is uniquely defined as the octahedron lain between $t = t^n$ and $t = t^{n+1}$ planes in the four-dimensional space.

Now let Ω be the octahedral control volume in the (x, y, z, t) space and $\partial\Omega$ be its boundary surface. Figure 2(left) schematically illustrates the octahedral control volume in space-time unified domain and Fig.2(right) illustrates the one of the surfaces of the octahedron. By integrating Eq.(3) over the control volume Ω and applying Gauss' divergence theorem, we obtain the following integral equation,

$$\int_{\Omega} \tilde{\nabla}\tilde{\mathbf{F}} dV = \int_{\partial\Omega} \tilde{\mathbf{F}} \cdot \tilde{\mathbf{k}} dS = 0, \tag{4}$$

where $\tilde{\mathbf{k}}$ is a unit vector outwardly normal to the octahedral control volume surface. In discrete form, Eq.(4) becomes, with $\tilde{\mathbf{k}}dS = \tilde{\mathbf{n}}$,

$$\sum_{l=1}^{8} \tilde{\mathbf{F}}_l \cdot \tilde{\mathbf{n}}_l = 0, \tag{5}$$

where, $\tilde{\mathbf{n}}_l = (n_x, n_y, n_z, n_t)_l$ $(l = 1, 2, \cdots 8)$ is the normal vector of the control volume surface in the space-time unified domain, and the length of the vector equals to the area of the four-dimensional boundary surface. Since the surfaces $l = 7$ and 8 are perpendicular to the t-axis, $\tilde{\mathbf{n}}_7$ and $\tilde{\mathbf{n}}_8$ have only n_t component and correspond to the cell volumes, $V^n_{i,j,k}$ and $V^{n+1}_{i,j,k}$, in the (x, y, z) space at n and $(n + 1)$-time steps respectively. Thus, Eq.(5) becomes as follows:

$$Q^{n+1}(n_t)_8 + Q^n(n_t)_7 + \sum_{l=1}^{6} \left(\tilde{\mathbf{F}}^{n+1/2}_l \cdot \tilde{\mathbf{n}}\right)_l = 0, \tag{6}$$

or

$$Q^{n+1}_{i,j,k} V^{n+1}_{i,j,k} + Q^n_{i,j,k} V^n_{i,j,k} + \sum_{l=1}^{6} \left[(E - E_v)n_x + (F - F_v)n_y + (G - G_v)n_z + Qn_t\right]^{n+1/2}_l = 0. \tag{7}$$

Here, the flux $F^{n+1/2}_l$, for example, can be evaluated as $\left(F^n_l + F^{n+1}_l\right)/2$ of second order accuracy. In this paper, however, the backward-Euler type estimation, $F^{n+1/2}_l = F^{n+1}_l$, is used for the higher-stability for high-Reynolds number turbulent flows. The convective flux vectors, E, F and G are evaluated using the Roe flux difference splitting scheme(7) with the second order ENO scheme of Harten and Osher(8), and the viscous flux vectors E_v, F_v and G_v are centrally differenced. For the turbulence modeling, we employ the Spalart-Allmaras one-equation turbulence model(9). In order to solve the nonlinear system, Eq.(7), the present scheme employs

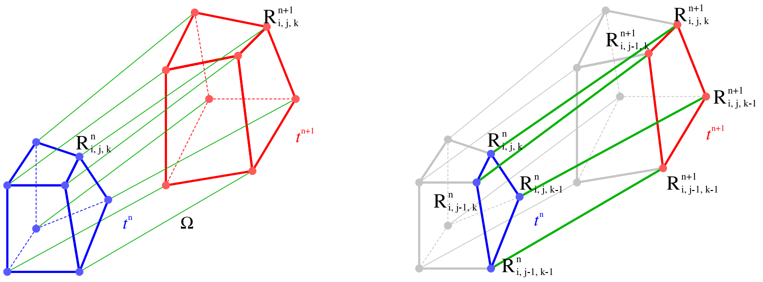


Fig. 2 Schematic view of control volume Ω for Moving-Grid FVM

sub-iteration procedure based on a pseudo-time approach(10) with LU-SGS implicit factorization(11) to assure the time accuracy at every time step.

2.4. Boundary Conditions

The boundary conditions for the present MCD method are not different from conventional methods. The boundary conditions at the time $t^{n+1/2}$ have to be given to the boundary grid cells. Since the computational domain moves in the physical space, inflow, outflow or in/out mixed boundary appears even if the fluid is stationary. In general, the inflow boundary condition is applied to the front side of the moving-computational domain. On the other hand, the outflow boundary condition is applied to the back side of the domain. In/out boundaries appear generally at the left, right and upper sides of the computational domain. The type of the boundary condition depends on how to move the computational domain. In general, since the inflow or outflow is not always known a priori, the characteristic boundary method such as the use of the Riemann invariant(12) is the standard boundary condition because the inflow or outflow boundary condition is determined by the characteristics. If the inflow or outflow side is known beforehand the usual subsonic boundary condition is also applicable and efficient.

It should be noted that the necessary assumption of the MCD method is that the state of circumstances just in front of the computational domain should be known a priori. The stationary fluid or uniform flow is usually assumed and given as the boundary conditions of the method. For the lower side of the domain, which corresponds to the ground in this paper, non-slip and adiabatic boundary conditions are given for the Navier-Stokes simulations. Non-slip and adiabatic wall conditions are imposed on the body surface for the Navier-Stokes computations. For the Euler flow simulations, the tangential velocity conditions are given for the solid wall and the ground surface.

3. Validation

3.1. Geometric Conservation Law

It is essential requirement for a moving grid method to satisfy the geometric conservation laws(6). The geometric conservation laws mean that even if the grid moves the flow is not affected by the movement of the grid. The present MCD method is just the method such that the whole grid moves in physical space, and it is crucial for the method to satisfy the geometric conservation laws. A simple validation test has been performed as a numerical check. As illustrated in Fig.3, a cubic computational domain of 21 × 21 × 21 uniform grid rotates in a stationary fluid. When the numerical method satisfies the geometric conservation laws, the stationary fluid state is never disturbed by the grid movement. The error is defined as:

$$Error = \sqrt{\frac{\Sigma_{i,j,k}(\rho_{i,j,k} - \rho_\infty)^2}{(imax - 1)(jmax - 1)(kmax - 1)}} \tag{8}$$

Here, *imax*, *jmax* and *kmax* are the number of grid point in *i*-, *j*- and *k*-direction respectively. Figure 4 shows the time history of the error. The error keeps the order of 10^{-15}, or machine zero, and therefore it is shown that the MCD method satisfies the geometric conservation

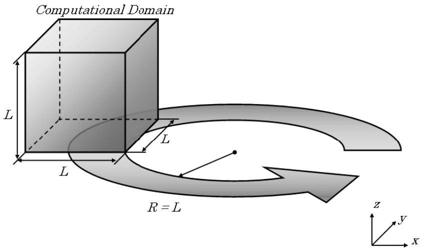


Fig. 3 Rotation of the cubic computational domain in a stationary fluid

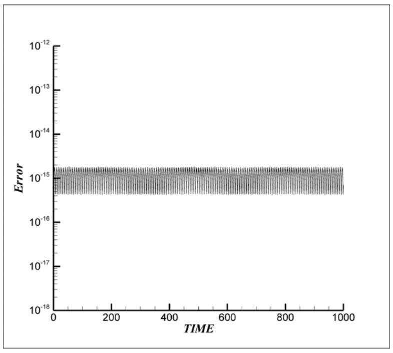
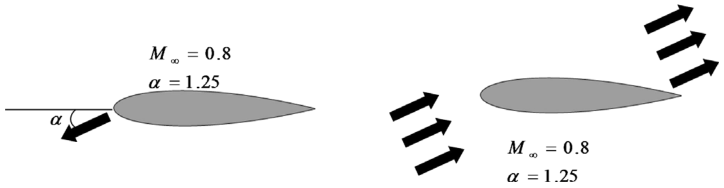


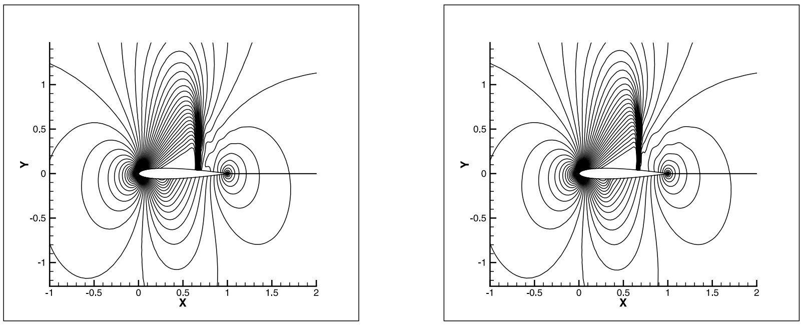
Fig. 4 History of error of density

laws perfectly. Thus it is confirmed that the flow is not affected by the movement of the computational domain with the MCD method.



(a) Condition 1 - Moivng airfoil in a stationary fluid (b) Condition 2 - Airfoil in a uniform flow

Fig. 5 Two types of condition of flow around a NACA0012 airfoil



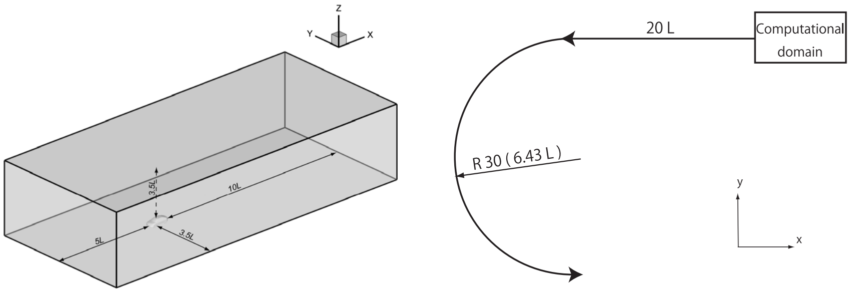
(a) Present Method (Condition 1) (b) Reference solution (Condition 2)

Fig. 6 Pressure distribution around a NACA0012 airfoil

3.2. NACA0012 Airfoil

The second test for the validation of the MCD method is a simulation of an inviscid transonic flow around NACA0012 airfoil. Two dynamically equivalent flows were simulated and compared each other. The one is the flow driven by the NACA0012 airfoil moving in the stationary fluid with the constant speed of Mach 0.8 and the angle of attack 1.25, illustrated in Fig.5 (a)(left, condition 1). The other one is the flow around NACA0012 airfoil in the uniform flow of Mach 0.8 with the angle of attack 1.25, illustrated in Fig. 5(b) (right, condition 2). These two conditions have to result the dynamically equivalent solutions by the Galilean transformation. The completely same size of the computational domain and grid was used for both computations. The condition 1 was simulated by the present MCD method. The condition 2 was simulated by the conventional manner with the same flux estimation as the MCD method.

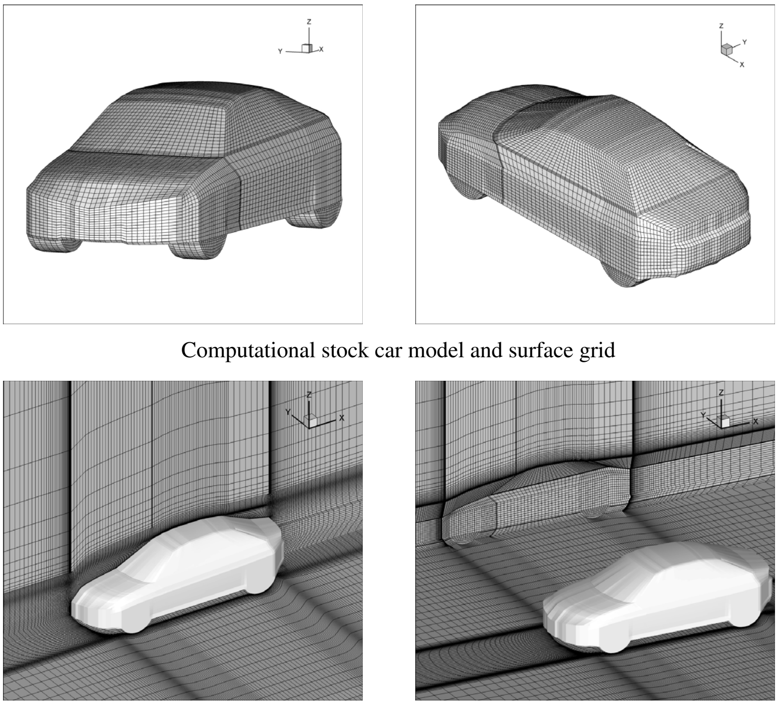
The boundary conditions are also given in an equivalent way. The Riemann boundary method is applied with the stationary fluid state for the condition 1 and the uniform flow state for the condition 2. Although the present test case is two-dimensional flow, the three-dimensional code was applied with three grid points along the third direction. The number of grid points are 301 (180 on airfoil) × 81 × 3 of C-type grid and the size of computational region is 15ℓ (ℓ: chord length) in forward, backward, upper and lower directions respectively. Figure 6 shows the comparison of the results. The averaged difference of pressure contours between two conditions was less than 0.1%, showing excellent agreement. Thus the MCD method works well and shows the applicability to the flow around the body.



(a) Computational Domain

(b) Movement of a car

Fig. 7 Computational model: a car passing through a hairpin curve.



Computational stock car model and surface grid

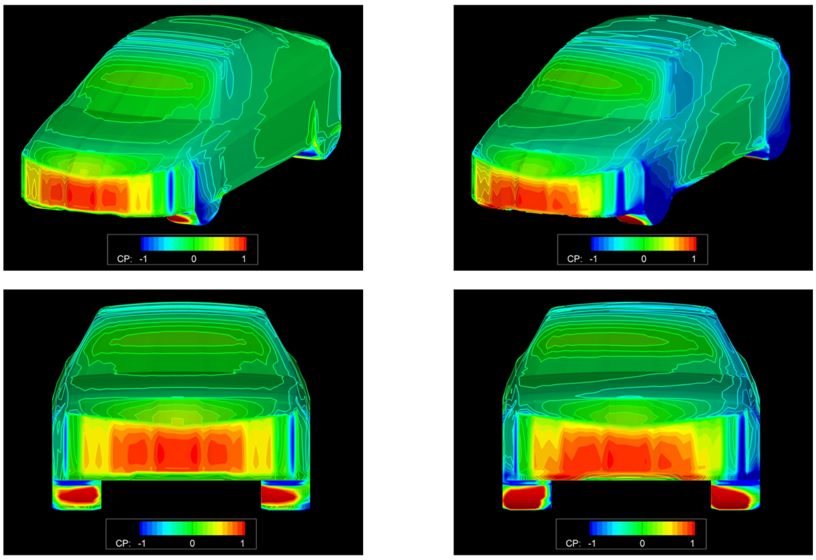
Multiblock grid around the car

Fig. 8 Stock car and computational grid of multi-block structure

4. Application

4.1. Flow Around a Car Passing Through a Hairpin Curve

The present MCD method has been applied to unsteady flow driven by a car passing through a hairpin curve. A model is a stock car of length L, width $0.377L$ and height $0.310L$ of sedan type. The size of the computational domain is $5L$ for forward and $10L$ for backward directions and 3.5L for left, right and upper sides. Figure 7(a) illustrates the computational domain with the car inside. Multi-block/block decomposition approach[13] was utilized considering efficiency and flexibility for complex geometry. The computational grid of the single global block/multiple sub-block structure[13] was generated by the elliptic grid generation method[14]. The computational domain was decomposed into 65 sub-blocks in the present computation. Figure 8 shows the computational model and the grid system of $235 \times 163 \times 86$ points. This computational domain moves rigidly according to the motion of the car, as illustrated in Fig.7(b). The motion of the car is the first 20L straight-line run and then shifts to the turning run with constant radius of 6.43L, which is patterned on the hairpin curve of Fuji-Speed Way. The constant speed U of the car was set to 128km/h (Mach number $M = 0.105$) and the related Reynolds number was 1.66×10^6. Although the flow Mach number is low, no preconditioning technique was used in the computations. To assure time accuracy, five to ten sub-iterations were performed at each time step. The simulation was performed from $t = 0$ to $t = 35$ with fixed time step of $\Delta t = 0.01$. The inflow boundary condition was given to the front side of the domain, while the outflow condition was applied to the back side of the domain. Mixed inflow/outflow conditions were used for the left, right and upper side boundaries. The bottom of the computational domain is the ground and non-slip condition was given. The non-slip condition was used for the surface of the car. Each tire, although the half of it was covered, was assumed to be rotating and non-slip condition was also applied. A change in the steering angle of the front tire was ignored for simplicity in the computation.



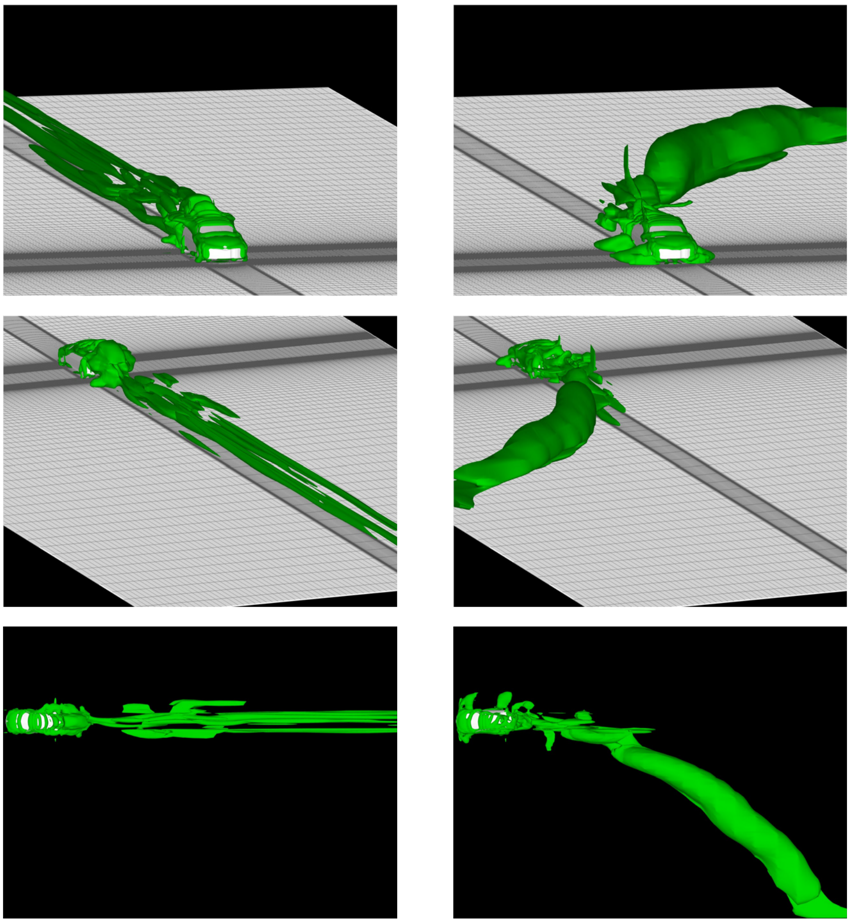
t = 20 (Straight run)

t = 30 (Turn run)

Fig. 9 Pressure distribution on the surface of the car at $t = 20$(left: straight run) and $t = 30$(right: turn run)

4.2. Numerical Results

The car was impulsively started at time $t = 0$ at the speed U. The time is normalized by L/U and thus the car arrived at the end of the straight-line path at the normalized time $t = 20$ and then entered the hairpin curve. Figure 9 shows a comparison of the pressure contours on the body surface between at $t = 20$, the last moment of straight-line run, and at $t = 30$, the turning run. Figure 10 shows a comparison of iso-surface plots of the vorticity at $t = 20$ and $t = 30$. Figure 11 shows: (a) iso-surface of velocity magnitude of 1.0(dark red) and 0.1(light blue), observing from forward-left position and (b) plane view of the surface pressure distribution at $t = 30$. When the car is running along the straight-line path, the pressure distribution on the surface is almost symmetric. However, at the time $t = 20$, the



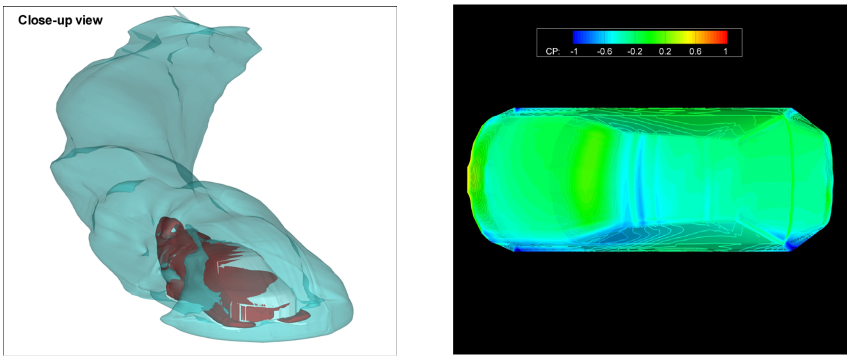
t = 20(Straight motion) t = 30(Turn motion)

Fig. 10 Vorticity contour at t = 20 and t = 30

straight-line run, the flow shows essentially unsteady behavior and some swirling flow can be seen. As expected, when the car is running on the curve, there exists clear difference of the pressure distribution between the left and right sides of the car, as can be seen in Fig. 11(b). The surface pressure on the right side of the car is higher than that on the left side. This pressure difference causes the flow around the car such that the shape of the wake follows just like the car trajectory, as seen in Fig.10(right). As shown in Fig. 10, swirling flow is detected behind the car. At the curve run, the higher velocity magnitude exists at the right side of the car. This is because of the difference of the speed between the right and left sides of the car. It is reflected in the contours of iso-surface of the vorticity as seen in Fig. 10.

5. Concluding Remarks

In this paper, the Moving-Computational-Domain Method has been proposed and applied to the flow around the high speed car passing through the hairpin curve. The method is based on the Moving-Grid Finite-Volume method and has been shown to satisfy both the physical and geometric conservation laws simultaneously. Application to the flow around the car passing thorough the hairpin curve has shown the promising feature of the method. We have applied the method only to rigid turn motion with constant radius. As previously described, it is possible to simulate the flow along the full course of track. The MCD method is also allowed to deform the grid system inside the computational domain simultaneously. By use of the MCD method it is theoretically possible for cars to run with relative motion each other in the moving computational domain. Thus the MCD method will be able to simulate the case that a car overtakes other cars in the full track, just like a track race.



(a) Iso-suraface of velocity magnitude

(b) Surface Cp distribution

Fig. 11 (a)Pespective view of iso-surface of velocity magnitsude, 1.0(dark red) and 0.1(light blue), and (b) Plane view of surafce pressure distribution, (at t=30, in the curve)

Acknowledgement

This work was supported by Grant-in-Aid for Scientific Research(21560175).

References

(1) Katz, J., Aerodynamics of Race Cars, *Annual Reviews of Fluid Mechanics*, Vol.38, (2006), pp.27-63.

(2) Steger, J.L., Dougherty, F.C. and Benek, J.A., A Chimera Grid Scheme; Advances in Grid Generation, *American Society of Mechanical Engineers, Fluids Engineering Division*, Vol.5 (1983), pp.55-70.

(3) Mihara, K., Matsuno, K., An Iterative Finite-Volume Scheme on a Moving Grid, *Transactions of the Japan Society of Mechanical Engineers, Series B*, Vol.65, (1999), pp.2945-2953 (in Japanese).

(4) Matsuno, K., Mihara, K. and Satofuka, N., A Moving-Mesh Finite-Volume Scheme for Compressible Flows, *Computational Fluid Dynamics 2000*, Springer, (2001), pp.705-710.

(5) Yamakawa, M. and Matsuno, K., Unstructured Moving-Grid Finite-Volume Method for Unsteady Shocked Flows, *Journal of Computational Fluids Engineering*, Vol.10, No.1,(2005) pp.24-30.

(6) Zhang, H., Reggio, M. Trepanier, J.Y. and Camarero, R., Discrete Form of the GCL for Moving Meshes and Its Implementation in CFD schemes, *Computers and Fluids*, Vol.22, (1993), pp.9-23.

(7) Roe, P.L., Approximate Riemann Solvers, Parameter Vectors, and Difference schemes, *Journal of Computational Physics*, Vol.43, (1981), pp.458-476.

(8) Harten, A. and Osher, S., Uniformly High-Order Accurate Non-Oscillatory Schemes I., *SIAM Journal on Numerical Analysis.* Vol.24 ,(1987), pp.279-309.

(9) Spalart, P.R. and Allmaras, S.R., A One-Equation Turbulence Model for Aerodynamics Flows, *AIAA Paper 92-0439*, (1992), pp.1-22.

(10) Rumsey, C.L., Santetrik, M.D., Biedron, R.T., Melson, N.D. and Parlette, E.B., Efficiency and Accuracy of Time-Accurate Navier-Stokes Computations, *Computers and Fluids* Vol.25, (1996),pp.217-236.

(11) Jameson, A., and Yoon, S., Lower-Upper Implicit Scheme with Multiple Grids for the Euler Equations. *AIAA Journal*, Vol.25, (1987), pp.929-935.

(12) Hirsch,C., *Numerical Computation of Internal and External Flows, Volume 2 Computational Methods for Inviscid and Viscous Flows*, Wiley, (1988), p.346.

(13) Matsuno, K., Flexible and Efficent Parallel Computation for Complex Flows Using Building-Multi-Block and Block-Decomposition Method, *Parallel Computational Fluid*

Dynamics - Multidiciplinary Applications, Edited by Winner, C. Elsevier, (2005), pp.377-381.

(14) Matsuno, K., Solution-Adaptive Methods for Elliptic Grid Generation, *Computational Fluid Dynamics Review 1998*, Edited by Hafez, M.M. and Oshima, K., World Scientific Pab., (1998), pp.127-139.