博士論文

Turbulent Flow Simulations around Aircraft using Hierarchical Cartesian Grids and the Immersed Boundary Method

(階層型直交格子および埋め込み境界法を用いた航空機周り乱流解析手法に関する研究)

A Dissertation
Submitted to the Department of Aeronautics and Astronautics
of the University of Tokyo Graduate School of Engineering
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Adviser: Dr. Taro Imamura

By
Yoshiharu Tamaki (玉置義治)

March 2018

Dissertation Committee

Chairperson:	Dr. Taro Imamura
Member:	Prof. Kenichi Rinoie
Member:	Prof. Kojiro Suzuki
Member:	Prof. Susumu Teramoto
Member:	Dr. Yuichi Matsuo

Abstract

Cartesian grid methods are becoming popular as a way to reduce the computational cost of grid generation for flow simulations. Cartesian grids can be generated automatically and rapidly based on tree-structures. This is a favorable property for aerodynamic design including shape-optimization problems. The wall boundary in the Cartesian grid is, however, represented as step-wise faces. Thus, an additional treatment for wall boundaries is necessary.

In this study, a simulation methodology for high-Reynolds-number flow simulations on a Cartesian grid is developed. The proposed framework aims to enable a more efficient aerodynamic design of aircraft. Thus, we use the immersed boundary method (IBM) for treating the wall boundary in a Cartesian grid because this method retains the simplicity and robustness of the Cartesian grid. The IBM, however, has difficulty in resolving the thin boundary layer in high-Reynolds-number flows. To solve this problem, we develop a wall boundary condition using a turbulent wall function to alleviate the requirement for the grid size on the wall in high-Reynolds number flows. Furthermore, uncertainty exists in the evaluation of the aerodynamic force acting on the immersed body, and thus, a new method is proposed to clarify this point.

Chapter 3: Wall Boundary Conditions for Turbulent Flows To develop the wall boundary condition for turbulent flows, the boundary-layer approximation of the Reynolds-averaged Navier–Stokes equations and the turbulence model are firstly analyzed. Then, the wall boundary condition is developed based on this analysis. The near-wall velocity profile is modified so that it is linear with respect to the wall distance to avoid numerical problems. This is because the original velocity profile defined by the wall-function is nonlinear and cannot be resolved on the grid. Simultaneously, the eddy-viscosity profile is modified by changing the wall-damping function of the SA turbulence model to retain the balance of the shear stress that is derived from the near-wall approximation of the RANS equations. This modification requires a cut-off distance, and thus, the calculation methods for the cut-off distance are also described.

Chapter 4: Force Integration on Cartesian Grids For the force integration, a polygon-based method and a flux-based method are compared. In the former method, the force is integrated on the actual surface of the immersed body. The physical quantities on the surface are extrapolated from the Cartesian grid. However, arbitrariness exists in the extrapolation formula of this method, and additional numerical errors may be introduced in the calculated force. Thus, a flux-based method is developed based on the balance of the numerical fluxes. The force integration in this method is conducted on the step-wise faces near the wall boundary. The calculated force directly reflects the accuracy of the flow calculation because the same numerical flux as that used in the flow calculation is used to calculate the aerodynamic force. Then, test problems are solved, and it is shown that the method is more accurate in predicting the drag coefficient in inviscid flow than the polygon-based method. The time history of the aerodynamic coefficient can also be reproduced by the flux-based method in an unsteady problem without introducing a volume-integration term.

Chapter 5: Test Cases for the Immersed Boundary Method The developed simulation framework is validated through simulations of simple 2D flows. The first validation problem is the turbulent flat-plate boundary layer. When the near-wall cell boundary does not coincide with the surface of the plate (i.e., the surface is immersed in the grid), the baseline IBM cannot predict the skin friction accurately. In contrast, the modified IBM can reproduce accurate distributions of the skin friction. The variable profile in the boundary layer is then investigated. When the modified IBM is used, the shear stress near the wall becomes constant with respect to the wall distance as implicated in the analysis of the governing equation. Furthermore, the criterion for the grid setting is examined. To predict the skin friction with less than 5% error, the cell size should be about one-fifties of the local boundary layer thickness.

The second validation problem is simulations of turbulent flows over a 2D bump. In this scenario, flows with a mild pressure gradient are simulated. The modified IBM again shows an improvement over the original IBM in the prediction of skin friction although the pressure gradient is neglected in the derivation of the modified IBM.

The last problem is the turbulent flow around a NACA 0012 airfoil. Here, the prediction accuracy of the aerodynamic coefficients is investigated. When the grid size on the wall is 1/2000 of the chord length or smaller, the drag at an angle of attack of 0° is predicted within 5% error. Simulation of a flow at a high angle of attack requires a smaller cell size on the wall. Although the precise prediction of the separation point near the trailing edge is difficult, distribution of the skin friction in the other region is predicted accurately when the grid size on the wall is smaller than 1/4000 of the chord length. It is also confirmed that the modified IBM improves the prediction accuracy of the skin friction coefficient and the drag coefficient at low angles of attack.

Chapter 6: Simulations of Turbulent Flows around Aircraft To demonstrate the capability and efficiency of the proposed framework for more practical problems, 3D flows around an aircraft are simulated. The first problem for this case is the prediction of the transonic flows around the NASA Common Research Model. In this case, a medium grid (approx. 62 million cells) is generated within 16 min using one core of Xeon E5–2643 v3. On the medium grid, the y^+ on the upper surface of the wing is less than 600, and more than a thousand cells are located on the mean aerodynamic chord. Aerodynamic coefficients of each component show a trend of grid convergence toward the values computed on a conventional body-fitted grid, although the drag coefficient is slightly overestimated. Compared with the grid converged value, the drag coefficient excluding the estimated induced drag differs by 19 drag counts (0.0019, 8.9% of the reference value) and 11 drag counts (5.1%) for the medium and fine grids (approximately 117 million cells), respectively. At a high angle of attack, the qualitative flow features including the flow separation behind the shock show fair agreement with the computational results on the body-fitted grids. In addition, the nonlinearity of the lift coefficient at high angles of attack observed in the wind-tunnel experiment is reproduced.

In the simulation of the JAXA standard high-lift model, the capability of the proposed framework for the prediction of flows around complex geometries is demonstrated. The computational grid (approximately 110 million cells) is generated within 38 min using one core of Xeon E5–2643 v3. The surface features, including the small gaps between the wing and flap, are well resolved in the generated grid. The result shows that lift coefficients at $\alpha = 4.36°–14.54°$ are predicted within a 2.7% range compared with the experimental data. In addition, the surface pressure coefficient at low angles of attack shows good agreement with the reference experimental data. In contrast, accurate prediction of the flow separation and the maximum lift coefficient is difficult at the moment, and the use of finer grids and/or a different turbulence model may improve the prediction accuracy.

Through the studies described above, the accuracy and applicability of the proposed framework are confirmed. The proposed framework can be used to investigate the basic flow features around a complex geometry within a short time. Thus, the time spent and manual work for the simulation of one case can be reduced compared to the conventional simulation on the body-fitted grids. Although the accuracy of CFD simulations on body-fitted grids may be better once a body-fitted well-tailored grid is prepared, the proposed framework can also predict the flow with a certain degree of accuracy. Therefore, the proposed methodology will be beneficial in terms of improvement in the efficiency of aerodynamic designs.

Acknowledgement

初めに、指導教員の今村太郎准教授に対しまして、修士課程入学より5年間にわたる御指導の御礼を申し上げます。また本論文の審査に当たって、副査の李家賢一教授、鈴木宏二郎教授、寺本進教授、松尾裕一博士に感謝申し上げます。

研究を進めていく上で、様々な人々と議論できたことが大きな支えになったと感じます。特にJAXA数値解析ユニットの皆様、JAXA FQUROHプロジェクトの皆様、東北大学澤田・河合研究室の皆様とは学会等多くの場所でお会いし、議論を通じて様々なことを発見いたしました。

博士1年次の終わりには、外務省の研究者派遣事業でUtah State Universityに3か月滞在しました。全く違う環境で研究を進める中で、自身の見識を深めることができました。ご指導くださったDr. Aaron Katzと、および派遣の斡旋等お世話になりました外務省、JTBコーポレートセールスおよびJIPTの皆様に感謝申し上げます。

また本研究では,日本学術振興会(JSPS)科研費、特別研究員奨励費16J07740の支援を受けました。また元となる直交格子ソルバについてはJSPS科研費、若手研究(A)(基金)15H05559の成果を用いました。これらの支援によりまして、大型計算機やワークステーション、ソフトウェアの利用や、海外での国際学会を含む多くの研究発表を行うことができ、研究活動の大きな助けとなりました。またJSPSより特別研究員に対する研究奨励金も受給しました。以上につきまして厚く御礼申し上げます。

最後に、これまで支えてくれた家族および友人に感謝を申し上げ、謝辞といたします。

2018年3月
玉置 義治

Contents

Nomenclature

Abbreviations

AUSM	Advection Upstream Splitting Method
CFD	Computational Fluid Dynamics
CRM	Common Research Model
DNS	Direct Numerical Simulation
DPW	Drag Prediction Workshop
FVM	Finite-Volume Method
IBM	Immersed Boundary Method
IP	Image Point
JAXA	Japan Aerospace Exploration Agency
JSM	JAXA Standard high-lift Model
LES	Large-Eddy Simulation
MFGS	Matrix-Free Gauss–Seidel
MPI	Message Passing Interface
MUSCL	Monotonic Upwind Scheme for Conservation Laws
NASA	National Aeronautics and Space Administration
QCR	Quadratic Constitutive Relationship
RANS	Reynolds-Averaged Navier–Stokes
SA	Spalart–Allmaras
SLAU	Simple Low-dissipation AUSM
TMR	Turbulence Modeling Resource
UTCart	the University of Tokyo Cartesian-grid-based automatic flow solver
WLSQ	Weighted Least-SQuares method

Superscripts

n	index for the time step

Subscripts

∞	free-stream quantity
F	quantity at the fluid cell next to the wall cell

f	quantity at the far-field boundary
FC	quantity at the face center where the wall boundary condition is imposed
i, j, k, α, β	index for spatial dimensions (from 1 to 3, summation is taken only for these subscripts)
IP	quantity at the image point
L	face-left quantity
l	quantity at the own cell
lr	quantity on the face between cells l and r
n	normal component
P	quantity at the cell including the image point
R	face-right quantity
r	quantity at the adjacent cell
t	tangential component

Variables

$(R.H.S.)$	right-hand side of the discretized governing equation
α	angle of attack
χ	non-dimensionalized working variable of the SA turbulence model $\tilde{\nu}/\nu$
Δt	time increment
Δx	cell size
$\dot{M}$	mass source
γ	ratio for the specific heat
Γ_F	far-field boundary
Γ_G	(step-wise) cell boundary near the wall
Γ_S	object surface
$\hat{n}_i$	face-normal vector
$\mathcal{S}$	source terms in the SA turbulence model
μ	molecular viscosity
ν	kinematic viscosity
ν^+	non-dimensional eddy viscosity ν_t/ν
ν_t	kinematic eddy-viscosity
$\tilde{\nu}$	working variable for the Spalart–Allmaras turbulence model
ϕ	wall distance function
ψ	cutoff distance function
ρ	density
σ_A	spectral radius

τ	pseudo time
τ_{ij}	stress tensor
$\tilde{\mathbf{F}}$	inviscid numerical flux
$\tilde{\mathbf{G}}$	viscous numerical flux
$\mathbf{A}$	flux Jacobian
$\mathbf{q}$	vector of the conservative variables $[\rho, \rho u_i, \rho E]^T$
$\mathbf{q}$	vector of the primitive variables $[\rho, u_i, p]^T$
Pr	laminar Prandtl number (=0.72)
Pr_t	turbulent Prandtl number (=0.90)
Re	Reynolds number
a	sound speed
AR	aspect ratio
c	chord length
c_{ref}	mean aerodynamic chord
C_D	drag coefficient
C_d	sectional drag coefficient
c_f	skin friction coefficient
C_L	lift coefficient
C_l	sectional lift coefficient
C_M	pitching moment coefficient
C_m	sectional pitching moment coefficient
C_n	Courant number
C_p	surface pressure coefficient
c_p	specific heat at constant pressure
d	distance to the nearest wall
d_{cutoff}	cutoff distance
E	specific total energy
f_{SA}	function of the SA wall model
f_{v1}	damping function for the SA turbulence model
F_i	aerodynamic force acting on objects
f_i	body force
L	reference length
n_i	wall normal vector
p	pressure
q_i	heat flux

R	gas constant
r_{IP}	ratio of the IP distance to the cell size $d_{IP}/\Delta x$
r_d	ratio of the cutoff distance to the wall distance d_{cutoff}/d
s	face area
T	temperature
t	physical time
t_i	wall tangential vector in the flow direction
u^+	non-dimensional velocity u/u_τ
u_τ	friction velocity
u_i	velocity
V	cell volume
w	Riemann invariants
x_i	coordinate
y^+	non-dimensional distance from the wall du_τ/ν

Chapter 1

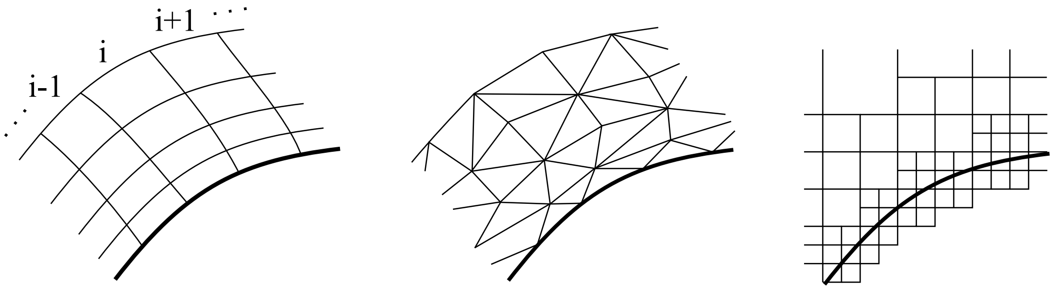
Introduction

Over the past decades, simulation methodology based on the computational fluid dynamics (CFD) has been studied and improved as a tool that can be used to understand fluid phenomena and to design aerodynamic shapes. Moreover, computational speeds are dramatically increasing due to massive parallelization, and very large-scale calculations are becoming possible. For example, a direct numerical simulation (DNS) of decaying isotropic turbulence using 12288^3 grid points was conducted in [1]. According to the CFD Vision 2030 [2] summarized by the National Aeronautics and Space Administration (NASA), simulations with 10 to 100 billion grid points will become commonplace in the 2030s. As the speed of computation increases, the geometry of the target objects for simulations generally becomes more complex. However, the time spent for the grid generation becomes a significant problem when the geometry is very complicated. This is because the procedure for grid generation includes manual tasks, and thus, parallelization of grid generation is difficult. In addition, a conventional body-fitted structured grid (Fig. 1.1 (a)) around a complex geometry often skews, which can degrade the accuracy and stability of flow computation. This grid generation process can become a bottleneck in aerodynamic design, where many cases with different shapes need to be tested.

To alleviate this bottleneck, unstructured grid methods [3–5] (Fig. 1.1 (b)) have been developed and widely used recently. Automatic generation of unstructured grids has also been attempted, for example in [4]. However, generating high-quality unstructured grids still depends on personal skill, and fully automated grid generation is difficult. In addition, a low-quality unstructured grid can have problems not only for its numerical accuracy but also when it comes to the robustness of the computation.

To reduce the time spent for grid generation, Cartesian grid methods (Fig. 1.1 (c)) are an attractive proposition. Specifically, the hierarchical Cartesian grid methods [6–12] has become popular because they are more efficient than uniform Cartesian grids in capturing the flow and surface features. Hierarchical Cartesian grids are generated based on block-based refinement [6, 7, 13], tree-structure [8, 9, 11], or a combination of the two [10, 12]. When these strategies are adopted, the grid generation only involves the detection of cells intersecting the objects, and accordingly, the automation of the grid generation is easy. In addition, the quality of the generated grid is high because the grid is orthogonal. Thus, these methods have advantages in terms of efficiency and robustness of the flow computation. These method are also used for solution-adaptive simulations [6–9, 13].

However, Cartesian grids have one critical problem when it comes to the representation of object surfaces. When the shape of an object is not aligned to the Cartesian coordinate, the Cartesian grid expresses the object surface as sets of step-wise facets. Thus, some special treatment for the wall boundary is required to reconstruct a smooth object surface.



(a) Structured grid (b) Unstructured grid (c) Cartesian grid

Figure 1.1: Classification of computational grids for CFD simulations.

1.1 Wall Boundary Conditions for Cartesian Grids

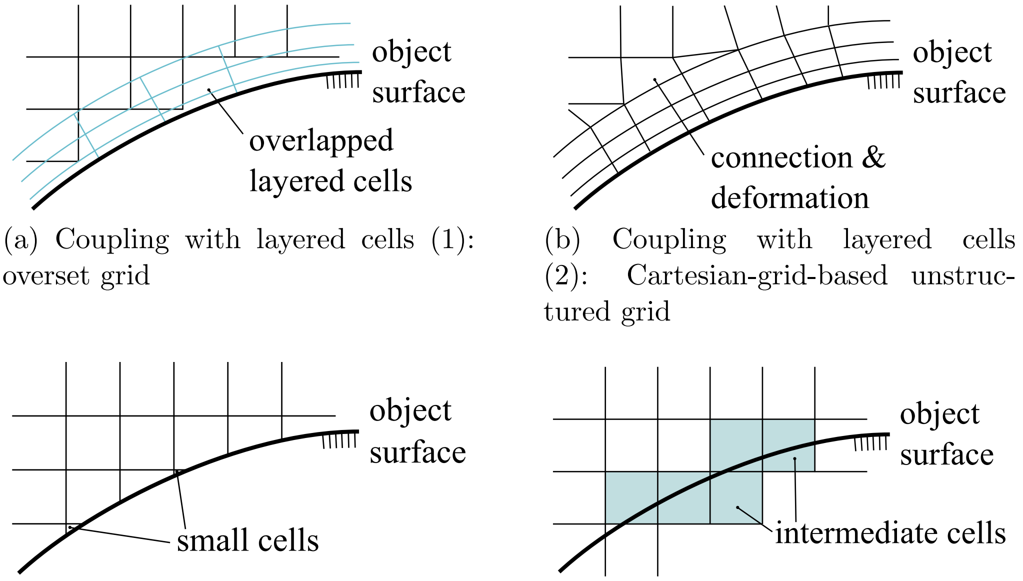
The treatments for the wall boundary can be classified into three main categories (Fig. 1.2):

- coupling with body-fitted layered grids (Fig. 1.2 (a) and (b));
- cut-cell methods (Fig. 1.2 (c));
- immersed boundary methods (IBMs) (Fig. 1.2 (d)) .

Here, the advantage and challenges of these three methods are briefly reviewed.

1.1.1 Coupling with Body-Fitted Layered Grids

Some flow simulations use Cartesian grids in conjunction with body-fitted layered grids, e.g., those using overset grids [10,14–17] or Cartesian-grid-based unstructured grids [18–21]. One advantage of these methods is that the grid near the wall is fitted to the object surfaces, and thus, the treatment for the wall boundary is easy. To resolve the boundary layer on the object, thin-layered cells can also be used.



(a) Coupling with layered cells (1): overset grid

(b) Coupling with layered cells (2): Cartesian-grid-based unstructured grid

(c) Cut-cell method

(d) Immersed boundary method

Figure 1.2: Classification of the wall treatment for Cartesian grids.

However, the overset method can introduce additional interpolation errors, and the conservation law near the overlapping region is generally not strict. In addition, the multidomain parallelization of the flow simulation becomes complicated because the communication between the background Cartesian grid and the layered grids as well as that between each domain is needed. [14–16]. The Cartesian-grid-based unstructured grid methods may generate skewed cells where the layered grids are connected to the background Cartesian grid. Although such methods have advantages in terms of ease of boundary treatment, a question arises when it comes to its robustness of automatic grid generation and flow simulation.

1.1.2 Cut-Cell Methods

Another type of wall treatment is called the cut-cell method [22–27]. Using this method, the cells near the body are "cut" by the object to fit the cell boundary to the object surface. The cell boundary is attached to the actual surface of the object, and thus, the conservation law is strict. The cells off the wall are not deformed and the robustness of the computation is retained.

The most significant problem of the cut-cell method is, however, the complexity of cell-cutting. The cell-cutting procedure includes classification of the cells depending on the condition of the cutting, and many cutting conditions must be considered. This adds complexity to the grid generation, especially in 3D cases. Furthermore, the classification becomes more complicated when multiple planes cut the cell into several segments. Another problem is the very small cells generated by the cell cutting (Fig. 1.2 (c)). To avoid numerical stiffness of the flow simulation, the small cells must be merged with the ambient cells [24], and this procedure can introduce additional complexity and arbitrariness. For these reasons, a large effort is needed to use the cut-cell method in simulations of flows around complicated 3D geometries.

1.1.3 Immersed Boundary Methods

IBMs [28–41] reproduce virtual smooth surfaces on Cartesian grids. When these methods are used, there is no need to modify the background Cartesian grid. Thus, the object surface is immersed in the grid, and some additional treatments for the intermediate cells (Fig. 1.2 (d)) must be implemented. IBMs can be classified into two categories: direct (continuous) forcing methods [28, 30, 31, 34, 42] and discrete forcing methods [33, 36–39, 41].

Direct Forcing Methods The direct forcing or continuous forcing methods adds a body force term to the governing equation to simulate the flow displacement induced by the objects. The force acting on the immersed objects is calculated using Hooke's law, which imitates an elastic body, and the reaction is imposed on the surrounding fluid cells. These methods are suitable for simulations of flows with elastic bodies, such as biological flows [28,29]. Furthermore, these methods are also often employed for moving bodies [28,29,31] or particle-laden flows [34] because no special treatment on the stencil near the wall boundary is necessary.

However, the spring constant in Hooke's law becomes infinity when a rigid body is simulated, and the numerical stability and accuracy then deteriorate. Some methods to mitigate these problems have been developed [30, 42], but additional parameters related to the rigidness of the object and the numerical stability must be introduced. In addition, the area where the body force is imposed generally has a finite thickness, which can also cause inaccuracies in the result.

Discrete Forcing Methods The discrete forcing methods determine the wall boundary conditions for the discretized governing equation. A typical example of this type of methods is the ghost-cell method [32, 37, 43, 44], which imposes the boundary conditions on the embedded cell by extrapolating the physical quantities from the ambient fluid cells. Some variants of this method that impose the boundary conditions on the cell interface [36,38,41] have also been proposed. Another variant of the method [39] defines the wall boundary condition at the point on the surface between the regular grid points, and the flow is then solved by the finite-difference method. This class of methods can represent wall boundary sharply and rigidly.

One problem concerning the discrete-forcing IBMs is the determination of the physical quantities on the boundary. The nodes or cell centers generally do not exist exactly on the object surface, and thus, some interpolation/extrapolation procedure is needed to determine the physical quantities on the boundary. This procedure can introduce numerical errors into the flow calculation. Another problem with IBMs is the evaluation of the aerodynamic force working on the immersed body. To calculate the force, a method to extrapolate or interpolate the surface pressure and friction has been proposed [45]. However, this method can introduce additional errors into the calculated aerodynamic force, and accordingly, uncertainty exists about the actual force acting on the immersed body.

Although some challenges exist in the IBMs as described above, they are widely used in inviscid and low-Reynolds-number viscous flows, including moving boundary problems. One positive aspect of the IBMs is that these methods can retain the original benefit of the Cartesian grids. This is because the cells around the wall boundary are not deformed, and thus, the representation of the surface is simple. Furthermore, the flow simulation is generally robust owing to the orthogonality of the grids.

1.2 Turbulent Flow Simulation around an Aircraft

In general, the Reynolds number, Re, of the flow around an aircraft is high enough for the boundary layer to become turbulent. At the actual scale of a commercial aircraft, Re based on the chord length is on the order of 10^7, and Re is approximately 10^6 to 10^7 for a scale model in a large wind tunnel [46, 47]. In addition, modern aircraft have high-aspect-ratio wings and long fuselages, which increase the surface area. Therefore, a DNS or a large eddy simulation (LES) around an entire aircraft is still very expensive in terms of the computational cost at the moment. For example, [48] reported that more than 10^8 cells are required in the simulation of a wing whose aspect ratio is 4, even when a wall model is used with the LES. Furthermore, unsteady turbulent vortices must be resolved in these simulations, and thus, a time accurate simulation is necessary. The simulation requires many time steps because the time scale of the vortex is several orders of magnitude smaller than that of the mean flow.

For the reason described above, the Reynolds-averaged Navier–Stokes (RANS) simulation is widely used for simulations of external flows around an aircraft. In the derivation of the RANS equation, the temporal fluctuation component and mean component are decomposed. The computation is conducted only on the mean component, and accordingly, a steady state solution can be obtained unless a strong instability (e.g., separated flows behind a bluff body, or artificial oscillating motion) exists in the flow field.

Under the cruise condition for aircraft, the flow is mostly attached to the surface, and thus, RANS simulations are known to be fairly accurate. For example, in the AIAA CFD Drag Prediction Workshops (DPWs) [49], the capability of the RANS simulation for the aerodynamic prediction of an aircraft was widely tested, and the results were extensively validated on structured and unstructured grids [50–53]. The computational grids for RANS simulations are generally designed to resolve the viscous sublayer of the turbulent boundary layer, so high-aspect-ratio cells conforming to the surface are used.

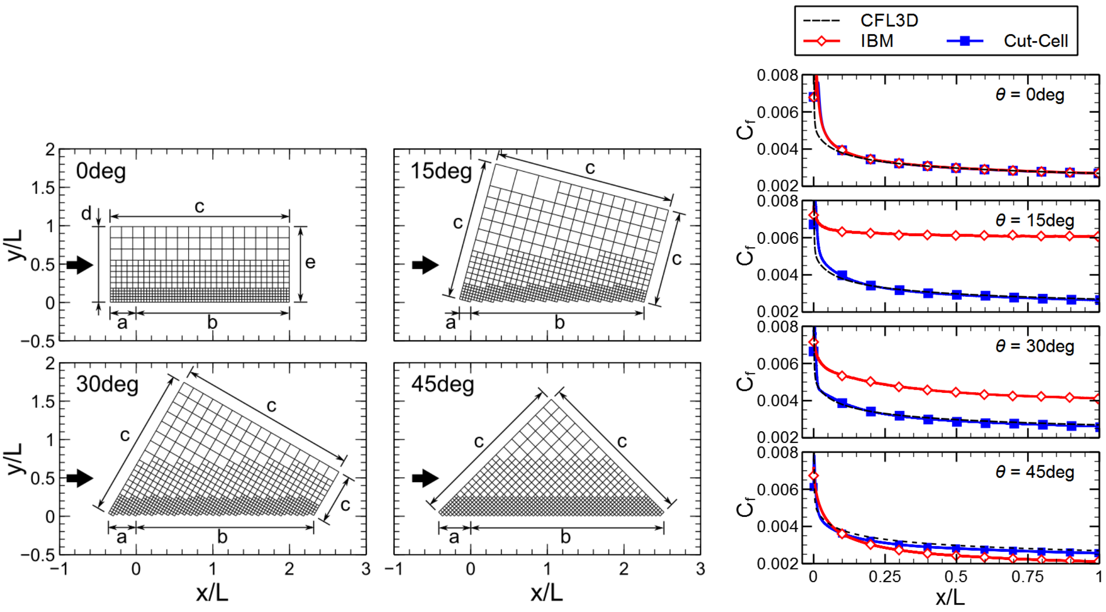
However, Cartesian grids, especially ones that use IBMs for the wall treatment, are not easily applicable to such high-Reynolds number flow simulations because the aspect ratio of the cells near the wall is fixed to one when Cartesian grids and IBM are used. To resolve the viscous sublayer, therefore, a large number of cells are required. In simple 2D problems, it has already been proved [38, 54] that turbulent boundary layers can be reproduced when the viscous sublayer is correctly resolved. However, it is not realistic to simulate 3D turbulent flows using such a fine grid. In particular, simulations of flows around high-aspect-ratio wings are quite difficult to achieve because a uniform cell size is also applied in the span-wise direction.

1.2.1 Wall Modeling

Modeling of the near-wall region of the boundary layer is one of the methods used to mitigate the requirement for the cell size near the wall boundary. This idea was originally developed to avoid the integration of the equation onto the wall when a high-Reynolds-number turbulence model is used (e.g. the standard k–ϵ model [55]). For example, the wall-function methods [56, 57] and the methods that use a diffusion equation [58–60] have been presented. This class of methods is attractive because the variable profiles near the wall are almost universal when they are appropriately non-dimensionalized. For example, a log-law is universally satisfied near the wall where the effects of the growth of the boundary layer and the pressure gradient are small [61].

On body-fitted grids, these methods have achieved results with acceptable accuracy, for example, in [59]. The wall models have been also used for LESs to reduce the computational cost to resolve small vortices near the wall [62–64]. These days, wall models are also used in conjunction with Cartesian grids to alleviate the requirement for the cell size. IBMs has been used with wall models in [26,36,38,40,41,65,66] for RANS simulations. Some attempts to perform wall modeled LESs using IBMs have also been presented [67,68]. Furthermore, other researchers have used the cut-cell methods with wall models [25–27].

However, previous research [26] has reported that the IBM with a wall function cannot reproduce the distribution of the skin friction on a turbulent flat plate accurately when the cell boundary does not coincide with the plate surface (Fig. 1.3). They concluded that the conservation law near the wall may be the limiting factor; the IBM does not satisfy the strict conservation requirements near the wall boundary. However, the mechanism of the inaccuracy has not been investigated in detail. Other researchers [36, 38, 40, 41, 65, 66] have obtained reasonable results in some cases, but neither an extensive grid convergence study nor a detailed analysis of how the wall function works on the Cartesian grid has been presented to date.



(a) Computational grids over a flat plate (coarsened for visualization)

(b) Skin friction coefficient

Figure 1.3: Skin friction coefficient on a turbulent flat plate calculated by an IBM and a cut-cell method using a turbulent wall function [26]. "CFL3D" in the figure shows the reference data computed on a body-fitted structured grid using the same turbulence model. When the IBM is used, the cell boundary does not coincide with the plate surface the except for the 0° case.

1.3 Purpose of this Research

This research aims to establish a methodology to simulate high-Reynolds-number flows using hierarchical Cartesian grids. This framework is intended to be used in the aerodynamic design of aircraft including shape-optimization problems. To achieve this goal, we adopt an IBM as the wall boundary condition because this method can retain the simplicity and robustness of the Cartesian grid. In addition, a RANS-based turbulence model is used to enable rapid analyses of the turbulent flows.

To perform accurate flow prediction using this framework, the following two topics are investigated as the first step:

- wall boundary conditions with a wall function to reproduce a turbulent boundary layer;
- a method to calculate the aerodynamic force acting on the immersed bodies.

For both subjects, the background issues are clarified from the viewpoints of the governing equations and the numerical schemes. In addition, novel methods based on these analyses are proposed to enhance the accuracy of the flow simulations.

Furthermore, the accuracy and applicability of the proposed framework are studied in various situations. First, they are investigated using simple 2D validation problems such as a turbulent flat plate and the flow around a NACA 0012 airfoil. Then, 3D turbulent flows around aircraft are simulated to demonstrate the capability and efficiency of the proposed framework. Finally, the benefits and current status of the proposed framework are discussed.

1.4 Outline of this Article

First, the numerical methods for the framework that simulate the turbulent flow around aircraft are developed. Chapter 2 describes the baseline flow solver using Cartesian grids and the IBM. Then, Chapter 3 describes the development of numerical methods for the turbulent flow simulations. Next, Chapter 4 introduces a method to calculate the aerodynamic forces acting on the immersed body.

To validate the framework described above, numerical studies are conducted. Chapter 5 provides the result of numerical tests on simple 2D problems, e.g., a turbulent flat-plate boundary layer. Then, numerical simulations of turbulent flows around an aircraft are conducted in Chapter 6. Finally, Chapter 7 provides the concluding remarks for this study.

Chapter 2

Baseline Cartesian-Grid-Based Flow Solver

In this chapter, the specification of the baseline flow solver "UTCart" (the University of Tokyo Cartesian-grid-based automatic flow solver) is described. UTCart consists of an automatic Cartesian grid generator and a compressible flow solver. In the following sections, the basic computational methods used in UTCart are explained.

2.1 Generation of Hierarchical Cartesian Grids

UTCart automatically generates computational grids using based on tree-structures; i.e., the quad-tree (2D) or oct-tree (3D). In the grid generation, the cells intersecting with or including the input object are treated as "wall cells." In addition, the cells inside and those outside the object are classified as "body cells" and "fluid cells," respectively (Fig. 2.1).

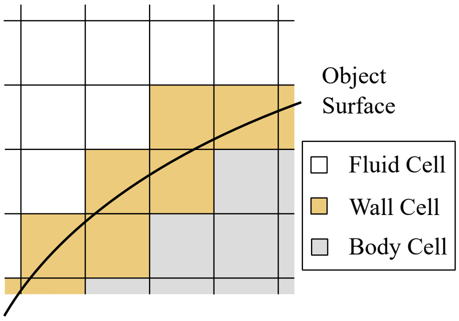


Figure 2.1: Classification of cells in UTCart.

The shapes of input objects are defined by sets of line segments in 2D, or by Standard Triangulated Language (STL) files (i.e., sets of triangular facet segments) in 3D. For each input geometry, a grid size and a boundary condition are specified. Then, binary-tree structures and bounding boxes (Fig. 2.2) are constructed for each object to search the nearest segments quickly. Furthermore, line and box sources for local grid refinement are also specified.

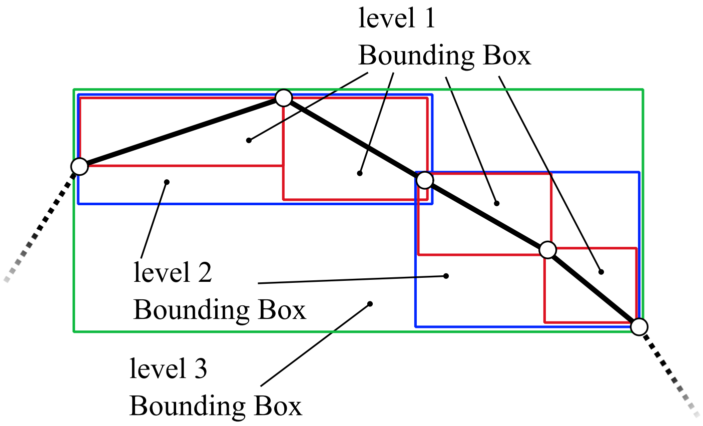
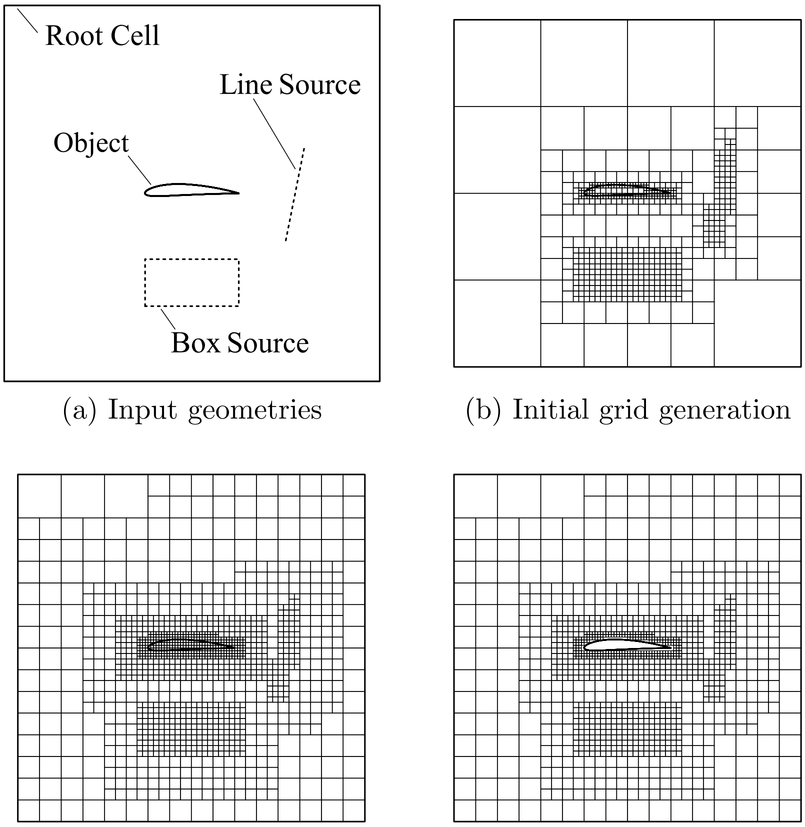


Figure 2.2: Binary tree structure for input object.

The overview of the grid generation is as follows. First, the entire domain is named "root cell," and the intersection with the input objects is examined (Fig. 2.3 (a)). When the objects exist inside the computational domain, the root cell is classified as a wall cell. Then, the wall cells are successively divided into four (2D) or eight (3D) child cells until the cell size becomes smaller than the value specified in the definition file (Fig. 2.3 (b)). Simultaneously, the cells in the refinement sources are also refined. Next, the grid is smoothed to limit the size ratio of the adjacent cell to two (Fig. 2.3 (c)). In this procedure, the number of cells in each layer can be specified, as described in Fig. 2.4. Finally, the cells completely outside the object are detected, and they are specified as fluid cells (Fig. 2.3 (d)). Through this grid generation, a quad-tree or an oct-tree structure is constructed based on the successive parent-children relationship (Fig. 2.5).

After the grid generation, the grid is modified by the following procedures to enhance the robustness of the flow calculation. At a fluid cell surrounded by wall cells, stencils for flow calculation become insufficient. If the two adjacent cells facing each other are wall cells (e.g., cell A in Fig. 2.6), therefore, the fluid cell in between is converted to a wall cell. Furthermore, hanging-nodes should not exist near the wall boundary. Here, the situation illustrated in Fig. 2.7 is considered. One of the adjacent cells of the left large cell has child cells, and (at least) one of the children is a wall. In this situation, numerical problems may arise because the image point (introduced in Section 2.4) can exist in the large cell. To avoid this, the large cell is split as shown in the figure.



(a) Input geometries

(b) Initial grid generation

(c) Smoothing

(d) Specifying fluid cells

Figure 2.3: Generation of Cartesian grid.

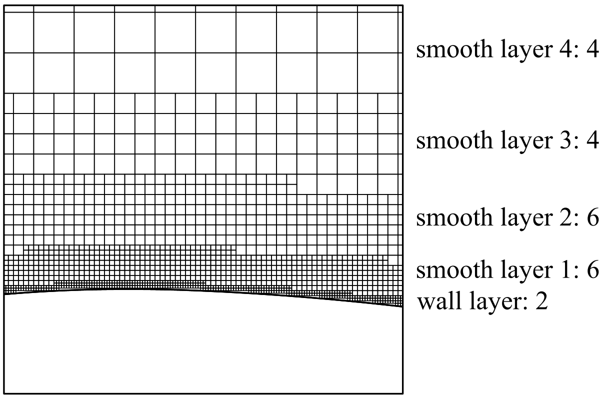


Figure 2.4: Specification of smooth layers.

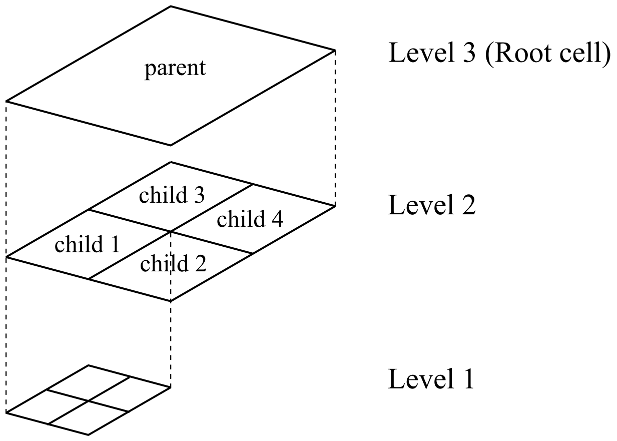


Figure 2.5: Tree-structure of the generated Cartesian grid.

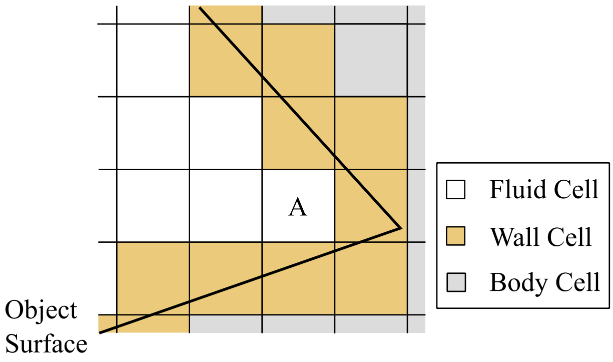


Figure 2.6: A fluid cell surrounded by wall cells.

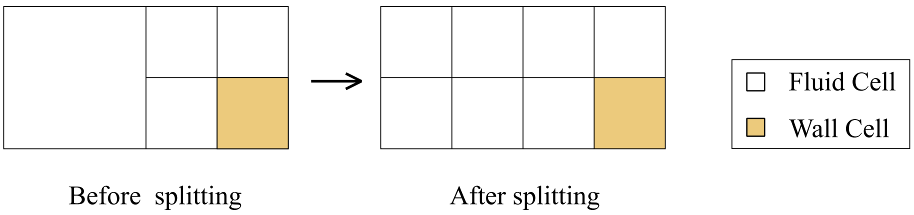


Figure 2.7: A hanging-node near the wall.

2.2 Governing Equations

The flow simulation by UTCart is based on the compressible Navier–Stokes equations in a conservation form. For high-Reynolds-number flows, RANS simulations are conducted using a turbulence model to reduce the computational cost. The governing equations are as follows.

2.2.1 Mean Flow

In turbulent flow simulations, the Favre-averaged Navier–Stokes equations are adopted as the governing equations. The conservation form of the equations is

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0, \tag{2.1}$$

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j) + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j}, \tag{2.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial}{\partial x_j}\left[(\rho E + p)u_j\right] = \frac{\partial}{\partial x_j}(u_i \tau_{ij} - q_j), \tag{2.3}$$

where ρ is the density, u_i are the components of the velocity vector, p is the pressure, and E is the specific total energy. Here, the subscripts i, j and k are the indices for spatial dimensions (from 1 to 3, summation is taken for these variables). Note that the variables in (2.1–2.3) are ensemble-averaged quantities. In addition, the specific total energy is calculated based on the assumption of the ideal gas:

$$E = \frac{RT}{\gamma - 1} + \frac{1}{2} u_i u_i, \tag{2.4}$$

where R is the gas constant, $T = p/(\rho R)$ is the temperature, and γ is the ratio of the specific heat (= 1.4 for air).

The stress tensor is derived using the Boussinesq eddy-viscosity approximation as

$$\tau_{ij} = 2\rho(\nu + \nu_t)\left[S_{ij} - \frac{1}{3} S_{kk}\delta_{ij}\right], \quad S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{2.5}$$

where ν and ν_t are the kinematic molecular viscosity and kinematic eddy-viscosity, respectively. ν is a function of the temperature, which is calculated by Sutherland's law [69] as

$$\mu = \mu_0 \frac{T_0 + C}{T + C}\left(\frac{T}{T_0}\right)^{3/2}, \tag{2.6}$$

where the molecular viscosity $\mu = \rho\nu$. For air, the constants are

$$\mu_0 = 1.716 \times 10^{-5} \ \ [\mathrm{Pa\cdot s}], \quad T_0 = 273.15 \ \ [\mathrm{K}], \quad C = 110.5 \ \ [\mathrm{K}]. \tag{2.7}$$

In addition, the heat flux is approximated as

$$q_j = -\rho \left(\frac{\nu}{\mathrm{Pr}} + \frac{\nu_t}{\mathrm{Pr}_t} \right) \frac{\gamma}{\gamma - 1} \frac{\partial (p/\rho)}{\partial x_j}, \tag{2.8}$$

with the laminar Prandtl number $\mathrm{Pr} = 0.72$ and the turbulent Prandtl number $\mathrm{Pr}_t = 0.90$. When no turbulence model is used, ν_t is set to zero in the equations above. Furthermore, $\nu = \nu_t = 0$ in an inviscid flow simulation.

2.2.2 Turbulence Model

For the closure of the RANS equations (2.1–2.3), turbulence models are necessary. In this research, the Spalart–Allmaras (SA) turbulence model [70] is employed to calculate the eddy-viscosity. The SA turbulence model solves one transport equation about a working variable $\tilde{\nu}$, which is related to the eddy-viscosity using a damping function f_{v1} as

$$\nu_t = \tilde{\nu} f_{v1}, \tag{2.9}$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}, \quad c_{v1} = 7.1. \tag{2.10}$$

The transport equation for $\tilde{\nu}$ is

$$\begin{aligned} &\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} \\ &\quad = (\mathit{Production}) - (\mathit{Destruction}) + \frac{1}{\sigma} \left[\frac{\partial}{\partial x_j} \left((\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right) + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} \right]. \end{aligned} \tag{2.11}$$

Here, the source terms are

$$(\mathit{Production}) = c_{b1} \tilde{S} \tilde{\nu}, \quad (\mathit{Destruction}) = c_{w1} f_w \left(\frac{\tilde{\nu}}{d} \right)^2, \tag{2.12}$$

where d is the distance to the nearest wall. Note that the trip term in the original equation is omitted following [71]. In addition, the model functions in (2.12) are

$$\begin{aligned} \tilde{S} &= \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \\ f_w &= g \left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{1/6}, \quad g = r + c_{w2}(r^6 - r), \\ r &= \min \left[\frac{\tilde{\nu}}{\tilde{S} \kappa^2 d^2}, 10 \right], \quad \Omega = \sqrt{2 W_{ij} W_{ij}}, \quad W_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right), \end{aligned} \tag{2.13}$$

where the model constants are

$$\begin{aligned} &c_{b1} = 0.1355, \quad \sigma = 2/3, \quad c_{b2} = 0.622, \quad \kappa = 0.41, \\ &c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}. \end{aligned} \tag{2.14}$$

In Chapter 6, the SA turbulence model is used with minor modifications to improve the prediction accuracy for some flows. The rotation correction [72, 73] reduces excessive production of the eddy-viscosity in a vortex core by replacing Ω in the production term with

$$\Omega + C_{rot}\min(0, S - \Omega), \tag{2.15}$$

where $S = \sqrt{2S_{ij}S_{ij}}$, and C_{rot} is set to unity to avoid negative production following the reference [73]. The formulation (2.15) eliminates production of the eddy-viscosity when only a pure rotational motion exists. Another modification, the quadratic constitutive relationship (QCR) [74], is a simple nonlinear eddy-viscosity model. This modification improves the prediction accuracy of the secondary flow in the square duct [74], or flow separation around the wing-body junction of aircraft [75]. In this model, the stress tensor τ_{ij} in (2.2) is replaced by

$$\tau_{ij,QCR} = \tau_{ij} - C_{cr}[O_{ik}\tau_{jk} + O_{jk}\tau_{ik}], \tag{2.16}$$

where $C_{cr} = 0.3$, and

$$O_{ik} = 2W_{ik} / \sqrt{\frac{\partial u_\alpha}{\partial x_\beta}\frac{\partial u_\alpha}{\partial x_\beta}}, \tag{2.17}$$

with α and β being indices for spatial dimensions (from 1 to 3, summation is taken for these variables).

2.3 Spatial Discretization

2.3.1 Finite-Volume Method

The UTCart flow solver treats the generated grid as an unstructured grid, and a face-based algorithm is adopted in the flow solver. This algorithm is simple even when multiple cells are attached to a large cell in one direction. Here, the Navier–Stokes equation (2.1–2.3) is spatially discretized by a cell-centered finite-volume method (FVM) as

$$\left.\frac{\partial \mathbf{Q}}{\partial t}\right|_l + \frac{1}{V_l}\sum_{r \in l}\left(\tilde{\mathbf{F}}_{lr} - \tilde{\mathbf{G}}_{lr}\right) s_{lr} = 0, \tag{2.18}$$

where s is the area of the interface between two cells,

$$\mathbf{Q} = [\rho, \ \ \rho u_i, \ \ \rho E]^T \tag{2.19}$$

is a vector of the conservative variables,

$$\tilde{\mathbf{F}} = [\rho u_j \hat{n}_j, \ \ (\rho u_i u_j + p\delta_{ij})\hat{n}_j, \ \ (\rho E + p) u_j \hat{n}_j]^T \tag{2.20}$$

is the inviscid numerical flux,

$$\tilde{\mathbf{G}} = [0, \ \ \tau_{ij}\hat{n}_j, \ \ (u_i \tau_{ij} - q_j)\hat{n}_j]^T \tag{2.21}$$

is the viscous numerical flux with $\hat{n}_j$ being the components of $\hat{\mathbf{n}}$, the normal vector of the cell interface pointing outward. Furthermore, the subscripts l and r are the indices of the target cell and the adjacent cells of cell l, respectively. The subscript lr denotes the quantity on the face between cells l and r. Note that summation is not taken for these subscripts.

The transport equation of the SA turbulence model is similarly discretized by the FVM as

$$\left.\frac{\partial \tilde{\nu}}{\partial t}\right|_l + \frac{u_{j,l}}{V_l}\sum_{r\in l}(\tilde{\nu}\hat{n}_j)_{lr}s_{lr} - \frac{1}{V_l}\sum_{r\in l}\left(\frac{\nu+\tilde{\nu}}{\sigma}\frac{\partial \tilde{\nu}}{\partial x_j}\hat{n}_j\right)_{lr} s_{lr} = \mathcal{S}_l, \tag{2.22}$$

where the source term is

$$\mathcal{S}_l = \left[(Production)_l - (Destruction)_l + \frac{c_{b2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j}\right)_l\right]. \tag{2.23}$$

The third term in the right-hand side of (2.23) is discretized further as follows:

$$\begin{aligned}\frac{c_{b2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j}\right)_l &= \frac{c_{b2}}{\sigma}\left[\frac{\partial}{\partial x_j}\left\{(\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right\} - (\nu+\tilde{\nu})\frac{\partial^2 \tilde{\nu}}{\partial x_j^2}\right]_l \\ &= \frac{c_{b2}}{\sigma V_l}\sum_{r\in l}\left[\{(\nu+\tilde{\nu})_{lr} - (\nu+\tilde{\nu})_l\}\left(\frac{\partial \tilde{\nu}}{\partial x_j}\hat{n}_j\right)_{lr}\right] s_{lr}.\end{aligned} \tag{2.24}$$

2.3.2 Gradient Estimation

At the beginning of each time step, the conservative variables $\mathbf{Q}$ are converted to the primitive variables $\mathbf{q} = [\rho, u_i, p]^T$. Then, the gradients of the primitive variables are estimated using the weighted least-squares method (WLSQ) [76,77]. The following gradient estimation procedure is conducted also for $\tilde{\nu}$.

Using the WLSQ, the derivatives of the primitive variables are calculated as

$$\left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_l = M_{jk}^{-1} \sum_{r \in l} w_{lr} (\mathbf{q}_r - \mathbf{q}_l)(x_{k,r} - x_{k,l}), \tag{2.25}$$

where w_{lr} is the weight function, and the matrix M_{jk} is

$$M_{jk} = \sum_{r \in l} w_{lr} (x_{j,r} - x_{j,l})(x_{k,r} - x_{k,l}). \tag{2.26}$$

In UTCart, the weight function is defined using the face area and the distance between two cells as

$$w_{lr} = s_{lr} \{(x_{i,r} - x_{i,l})(x_{i,r} - x_{i,l})\}^{-\frac{3}{2}}. \tag{2.27}$$

The inverse matrix M_{jk}^{-1} is independent of the flow solution. Thus, this matrix is calculated at the beginning of the flow calculation and stored. Note that the boundary conditions are not included in the stencils for the gradient estimation. Thus, the stencils for the gradient estimation becomes one-sided, as shown in Fig. 2.8.

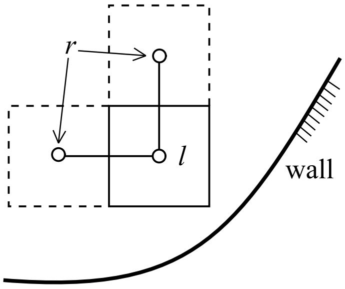


Figure 2.8: Stencils for the gradient estimation near the wall boundary.

2.3.3 Calculation of Inviscid Flux

The inviscid flux is computed by an upwind scheme. The upwind flux is written using the face-left and the face-right quantities as

$$\tilde{\mathbf{F}}_{lr} = \tilde{\mathbf{F}}_{lr}(\mathbf{q}_L, \mathbf{q}_R, \hat{\mathbf{n}}_{lr}), \tag{2.28}$$

where the subscripts L and R denotes the face-left and right quantities, respectively. In UTCart, a variant of the advection upstream splitting method (AUSM), the simple low-dissipation AUSM (SLAU) [78] is used to calculate the inviscid flux. The numerical flux of the SLAU is defined as follows:

$$\tilde{\mathbf{F}}_{lr} = \frac{\tilde{m} + |\tilde{m}|}{2} \mathbf{\Phi}_L + \frac{\tilde{m} - |\tilde{m}|}{2} \mathbf{\Phi}_R + \tilde{p} \mathbf{N}_{lr}, \tag{2.29}$$

where

$$\begin{aligned}
&\mathbf{\Phi} = [1, u_i, (E + p/\rho)]^T, \quad \mathbf{N} = [0, \hat{n}_i, 0]^T,\\
&\tilde{m} = \frac{1}{2}\left[(\rho V_n)_L + (\rho V_n)_R - |\overline{V_n}|(\rho_R - \rho_L)\right](1 - \bar{g}) - \frac{\xi}{2\bar{c}}(p_R - p_L),\\
&V_n = u_i \hat{n}_i, \quad \xi = (1 - \hat{M}^2),\\
&\hat{M} = \min\left[1.0, \frac{1}{\bar{c}}\sqrt{\frac{(u_i u_i)_L + (u_i u_i)_R}{2}}\right],\\
&|\overline{V_n}| = \frac{\rho_L |V_{n,L}| + \rho_R |V_{n,R}|}{\rho_L + \rho_R}\\
&\bar{g} = -\max[\min(M_L, 0), -1]\min[\max(M_R, 0), 1],\\
&M = \frac{V_n}{\bar{c}}, \quad \bar{c} = \frac{a_L + a_R}{2},\\
&\tilde{p} = \frac{p_L + p_R}{2} + \frac{\beta^+ + \beta^-}{2}(p_L + p_R) + (1 - \xi)(\beta^+ + \beta^- - 1)\frac{p_L + p_R}{2},\\
&\beta^+ = \begin{cases} \frac{1}{4}(2 - M_L)(M_L + 1)^2 & \text{for } |M_L| < 1 \\ \frac{1}{2}[1 + \mathrm{sign}(M_L)] & \text{otherwise} \end{cases},\\
&\beta^- = \begin{cases} \frac{1}{4}(2 + M_R)(M_R - 1)^2 & \text{for } |M_R| < 1 \\ \frac{1}{2}[1 + \mathrm{sign}(-M_R)] & \text{otherwise} \end{cases},
\end{aligned}$$

with $a = (\gamma p/\rho)^{1/2}$ being the sound speed. For the SA equation, the advection term is also calculated by an upwind scheme:

$$(\tilde{\nu}\hat{n}_j)_{lr} = \begin{cases} \tilde{\nu}_L(\hat{n}_j)_{lr} & (\text{for } \hat{V}_n \geq 0) \\ \tilde{\nu}_R(\hat{n}_j)_{lr} & (\text{for } \hat{V}_n < 0) \end{cases}, \tag{2.30}$$

where

$$\hat{V}_n = \frac{u_{i,l} + u_{i,r}}{2}(\hat{n}_i)_{lr}. \tag{2.31}$$

Variable Extrapolation For a first order scheme, $\mathbf{q}_L = \mathbf{q}_l$ and $\mathbf{q}_R = \mathbf{q}_r$. To construct a scheme with second or higher-order spatial accuracy, the face-left and right quantities are extrapolated using the estimated gradients (2.25). Here, the monotonic upwind scheme for conservation laws (MUSCL) [79, 80] is used for the extrapolation. The face-right quantities are

$$\begin{aligned}
\mathbf{q}_L &= \mathbf{q}_l + \frac{1}{4}\left[(1 - \kappa_a)\mathbf{\Delta}_l^- + (1 + \kappa_a)\mathbf{\Delta}_l^+\right],\\
\mathbf{\Delta}_l^- &= 2\left.\frac{\partial \mathbf{q}}{\partial x_i}\right|_l (x_{i,r} - x_{i,l}) - (\mathbf{q}_r - \mathbf{q}_l), \qquad (2.32)\\
\mathbf{\Delta}_l^+ &= \mathbf{q}_r - \mathbf{q}_l,
\end{aligned}$$

and the face-right quantities are

$$\begin{aligned}\mathbf{q}_R &= \mathbf{q}_r - \frac{1}{4}\left[(1-\kappa_a)\boldsymbol{\Delta}_r^+ + (1+\kappa_a)\boldsymbol{\Delta}_r^-\right],\\ \boldsymbol{\Delta}_r^- &= \mathbf{q}_r - \mathbf{q}_l,\\ \boldsymbol{\Delta}_r^+ &= 2\left.\frac{\partial\mathbf{q}}{\partial x_i}\right|_r (x_{i,r}-x_{i,l}) - (\mathbf{q}_r-\mathbf{q}_l).\end{aligned} \tag{2.33}$$

This extrapolation achieves third-order spatial accuracy with $\kappa_a = 1/3$. Note that the formal spatial accuracy of this scheme is second-order even with the third-order extrapolation in multi-dimensional nonlinear problems. This is because the spatial accuracy of the integration of the numerical flux over the cell interface is second-order [81].

Where the grid size changes, the extrapolation formulae are slightly changed. At first, two points l' and r' are set as shown in Fig. 2.9. The pairs of points $l - r'$ and $r - l'$ are point-symmetry with respect to the center of the face, respectively. The primitive variables at l' and r' are computed as

$$\begin{aligned}\mathbf{q}_{l'} &= \mathbf{q}_l + \left.\frac{\partial\mathbf{q}}{\partial x_i}\right|_l \left(x_{i,l'}-x_{i,l}\right),\\ \mathbf{q}_{r'} &= \mathbf{q}_r + \left.\frac{\partial\mathbf{q}}{\partial x_i}\right|_r \left(x_{i,r'}-x_{i,r}\right).\end{aligned} \tag{2.34}$$

Then, the quantities (primitive variable and the coordinate) at r in (2.32) and l in (2.33) are replaced by those at r' and l', respectively.

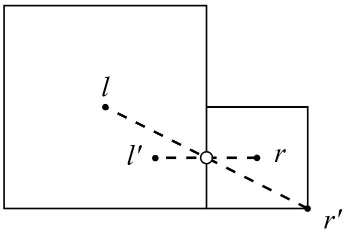


Figure 2.9: Description of the points used in the MUSCL extrapolation.

A higher-order scheme is also implemented in UTCart. This scheme can achieve a fourth-order spatial accuracy in the uniform region and retains a second-order accuracy near the interface where the cell size changes. For the description of this scheme, see [81].

Shock-Capturing When the flow includes discontinuities, the extrapolation formulae (2.32) and (2.33) result in oscillatory solutions. Thus, a total-variation diminishing scheme is used instead. Here, the minmod limiter [82] is employed to limit the extrapolation; $\boldsymbol{\Delta}^+$ and $\boldsymbol{\Delta}^-$ in (2.32) and (2.33) are replaced by

$$\bar{\boldsymbol{\Delta}}^{\pm} = \mathrm{minmod}(\boldsymbol{\Delta}^{\pm}, b\boldsymbol{\Delta}^{\mp}), \tag{2.35}$$

where $b = (3 - \kappa_a)/(1 - \kappa_a)$, and the minmod function is

$$\mathrm{minmod}(x, y) = \begin{cases} \min(x, y) & \text{for } x > 0 \text{ and } y > 0 \\ \max(x, y) & \text{for } x < 0 \text{ and } y < 0 \\ 0 & \text{otherwise} \end{cases} . \tag{2.36}$$

2.3.4 Calculation of Viscous Flux

To calculate the viscous flux, a central difference scheme is used. In this scheme, the primitive variables on the face are calculated as

$$\mathbf{q}_{lr} = \frac{\Delta x_r \mathbf{q}_l + \Delta x_l \mathbf{q}_r}{\Delta x_l + \Delta x_r}. \tag{2.37}$$

For the calculation of the gradient on the face, a modification is included to avoid the even-odd decoupling following [83, 84]. Here, the simple average of the gradient is firstly taken:

$$\left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_{temp} = \frac{1}{\Delta x_l + \Delta x_r}\left[\left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_l \Delta x_r + \left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_r \Delta x_l\right]. \tag{2.38}$$

Then, the modified gradient is calculated as

$$\left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_{lr} = \left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_{temp} - \left[\left.\frac{\partial \mathbf{q}}{\partial x_i}\right|_{temp} \overline{r_{lr}}_i - \frac{\mathbf{q}_r - \mathbf{q}_l}{d_{lr}}\right] \overline{r_{lr}}_j, \tag{2.39}$$

where $\overline{r_{lr}}_j$ are the components of the unit direction vector from cell l to r, and d_{lr} is the distance between the two cells. In addition, the diffusive part of the transport equation of the SA turbulence model is also calculated by the same procedure.

2.4 Boundary Conditions

2.4.1 Wall Boundary Conditions

UTCart imposes the wall boundary condition at the center of the face between fluid and wall cells (point FC in Fig. 2.10). To determine the boundary conditions, a discrete-forcing IBM is used. Here, the baseline IBM for inviscid and low-Reynolds-number viscous flows are explained. Note that the method for high-Reynolds-number flow is presented in Chapter 3.

To determine the physical quantities at FC, an image point (IP) is set on the wall-normal line through FC, and one-dimensional variable profiles between the IP and wall is assumed. d_{IP}, the distance between the IP and wall, is related to the size of the ambient cells Δx as

$$d_{IP} = r_{IP}\Delta x, \tag{2.40}$$

where r_{IP} is the ratio of the IP distance to the cell size on the wall and is set to a constant value. The minimum value for r_{IP} is $\sqrt{2}$ in 2D and $\sqrt{3}$ in 3D to assure the IPs to be located in the fluid cells. Note that the dependency on the r_{IP} value will be discussed in Chapter 5. An exception may occur where two walls locate close to each other. At this condition, the IP sometimes locates inside the wall. To avoid a numerical problem, the wall boundary is considered as a step-wise face, and the value at FC is determined using the fluid cell including FC.

The formulae to determine the boundary condition are as follows. In the explanation below, the quantities at the IP and FC are represented with subscripts IP and FC, respectively. First, the primitive variables at the IP is linearly interpolated as

$$\mathbf{q}_{IP} = \mathbf{q}_P + \left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_P (x_{j,IP} - x_{j,P}), \tag{2.41}$$

where the subscript P denotes the value at the center of the cell including the IP. Then, the primitive variables at FC are calculated using the quantities at the IP. For example, the density and pressure are assumed to satisfy the zero gradient condition on the wall. Thus, the density and pressure at FC are

$$\rho_{FC} = \rho_{IP}, \quad p_{FC} = p_{IP}. \tag{2.42}$$

The wall-normal velocity must satisfy the non-penetration condition, i.e., the normal velocity is zero on the wall. Thus, a linear profile between the IP and wall is assumed, and the normal velocity at FC is

$$u_{n,FC} = u_{n,IP}\frac{d_{FC}}{d_{IP}}, \tag{2.43}$$

where u_n is the normal velocity and d_{FC} is the distance between FC and wall.

The boundary condition for the tangential velocity u_t depends on whether the wall is slip or non-slip. For the slip wall,

$$u_{t,FC} = u_{t,IP}, \tag{2.44}$$

to satisfy the zero-gradient condition of the tangential velocity on the wall. For the non-slip wall, the tangential velocity is zero on the wall; thus,

$$u_{t,FC} = u_{t,IP} \frac{d_{FC}}{d_{IP}}, \tag{2.45}$$

and the velocity gradient on the wall is

$$\frac{\partial u_t}{\partial n} = \frac{u_{t,IP}}{d_{IP}}. \tag{2.46}$$

Furthermore, the boundary condition for $\tilde{\nu}$ is

$$\tilde{\nu}_{FC} = \tilde{\nu}_{IP} \tag{2.47}$$

for the slip wall, and

$$\tilde{\nu}_{FC} = \tilde{\nu}_{IP} \frac{d_{FC}}{d_{IP}} \tag{2.48}$$

for the non-slip wall.

The primitive variables at FC calculated by (2.42–2.45) are used to calculate the numerical flux at FC. For the inviscid flux, an upwind scheme is used to stabilize the computation:

$$\tilde{\mathbf{F}} = \tilde{\mathbf{F}}(\mathbf{q}_L, \mathbf{q}_{FC}, \hat{\mathbf{n}}). \tag{2.49}$$

Here, $\mathbf{q}_L$ are the primitive variables extrapolated from cell F, the cell including FC, as

$$\mathbf{q}_L = \mathbf{q}_F + \left.\frac{\partial \mathbf{q}}{\partial x_j}\right|_F (x_{j,FC} - x_{j,F}), \tag{2.50}$$

where the subscript F denotes the value at the center of cell F. The viscous flux is calculated using only the quantities at FC. Using the velocity gradient (2.46), the velocity gradient tensor is

$$\tau_{ij} = \rho_{FC}(\nu + \nu_t)_{FC} \left.\frac{\partial u_t}{\partial n}\right|_{FC} (n_i t_j + n_j t_i), \tag{2.51}$$

where t_j are the components of a unit vector in the flow direction. Furthermore, the heat flux q_j is assumed to be zero for an adiabatic wall. Consequently, the viscous flux (2.21) is calculated using the stress tensor (2.51) and the primitive variables at FC.

2.4.2 Far-Field Boundary Conditions

At the far-field face, the flux is calculated by an upwind scheme as

$$\tilde{\mathbf{F}}_{lf} = \tilde{\mathbf{F}}(\mathbf{q}_l, \mathbf{q}_f, \hat{\mathbf{n}}_{lf}), \tag{2.52}$$

where the subscript f denotes the far-field quantities. In UTCart, the following boundary conditions are implemented following [85]:

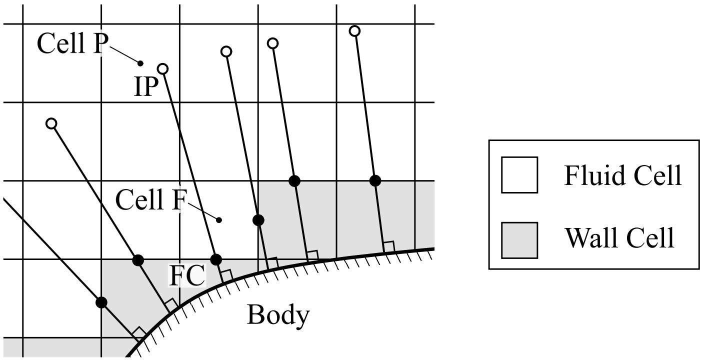


Figure 2.10: Schematic of the wall boundary condition for the immersed surface.

1. free-stream (Riemann) condition;
2. symmetry condition;
3. subsonic inflow condition with constant total pressure and total temperature ;
4. subsonic outflow condition with constant static pressure condition.

Free-Stream (Riemann) Condition This condition is used to simulate external flows. If the flow in the direction normal to the face is supersonic, the boundary conditions are

$$\mathbf{q}_f = \mathbf{q}_\infty, \tag{2.53}$$

for inflow, and

$$\mathbf{q}_f = \mathbf{q}_l, \tag{2.54}$$

for outflow, where the subscript ∞ denotes the free-stream quantities. For subsonic flows, the characteristic waves through the far-field face are considered. The three Riemann invariants are

$$\begin{aligned} w_1 &= \frac{p}{\rho^\gamma}, \\ w_2 &= u_n + \frac{2a}{\gamma - 1}, \\ w_3 &= u_n - \frac{2a}{\gamma - 1}, \end{aligned} \tag{2.55}$$

where $u_n = u_i \hat{n}_i$ is the face-normal velocity, and the face normal $\hat{n}_i$ is pointing outside the domain. The characteristic speeds for these invariants are u_n, $u_n + c$ and $u_n - c$, respectively. From (2.55), the normal velocity and the sound speed are

$$\begin{aligned} u_n &= \frac{w_2 + w_3}{2}, \\ a &= \frac{\gamma - 1}{4}(w_2 - w_3). \end{aligned} \tag{2.56}$$

When the flow is subsonic, w_2 is determined by the value in the domain, while w_3 is coming from the outside. Thus, the face-normal velocity and sound speed at the boundary are computed as

$$\begin{aligned} u_{n,f} &= \frac{w_{2,l} + w_{3,\infty}}{2}, \\ a_f &= \frac{\gamma - 1}{4}(w_{2,l} - w_{3,\infty}). \end{aligned} \tag{2.57}$$

Here, the face-tangential component of the velocity is depending on the direction of the flow. Thus, the velocity at the far-field boundary is

$$u_{i,f} = \begin{cases} u_{n,f}\hat{n}_i + [u_{i,l} - (u_{j,l}\hat{n}_j)\hat{n}_i] & (u_{n,f} \geq 0) \\ u_{n,f}\hat{n}_i + [u_{i,\infty} - (u_{j,\infty}\hat{n}_j)\hat{n}_i] & (u_{n,f} < 0) \end{cases}. \tag{2.58}$$

In addition, the other invariant w_1 also depends on the sign of the face-normal speed:

$$w_{1,f} = \frac{p_f}{\rho_f^\gamma} = \begin{cases} p_l/\rho_l^\gamma & (u_{n,f} \geq 0) \\ p_\infty/\rho_\infty^\gamma & (u_{n,f} < 0) \end{cases}, \tag{2.59}$$

and the boundary values for the density and pressure are

$$\rho_f = \left(\frac{a_f^2}{\gamma w_{1,f}}\right)^{\frac{1}{\gamma-1}}, \quad p_f = \frac{\rho_f a_f^2}{\gamma}. \tag{2.60}$$

Symmetry Condition When a half-span model is simulated, a symmetry boundary condition is required. For this condition, only the sign of the face-normal velocity is reversed:

$$\rho_f = \rho_l, \quad u_{i,f} = u_{i,l} - 2u_{j,l}\hat{n}_j\hat{n}_i, \quad p_f = p_l. \tag{2.61}$$

Subsonic Inflow Condition with Constant Total Pressure and Total Temperature This condition is used as the inflow condition for internal flows. Here, the total pressure $p_{total,set}$ and the total temperature $T_{total,set}$ are subscribed. The total enthalpy H_t is preserved at the boundary, and thus,

$$\begin{aligned} H_t &= \frac{\gamma}{\gamma - 1}\frac{p_l}{\rho_l} + \frac{1}{2}u_{i,l}u_{i,l} \\ &= \frac{a_f^2}{\gamma - 1} + \frac{1}{2}u_{i,f}u_{i,f}. \end{aligned} \tag{2.62}$$

Furthermore, one of the Riemann invariants, w_2, is constant between the boundary face and the cell center:

$$\begin{aligned} w_2 &= u_{i,l}\hat{n}_i + \frac{2a_l}{\gamma - 1} \\ &= u_{n,f} + \frac{2a_f}{\gamma - 1}. \end{aligned} \tag{2.63}$$

From (2.62) and (2.63), the following second-order quadratic equation about a_f is achieved:

$$\begin{gathered} Aa_f^2 - Ba_f + C = 0, \\ A = 1 + \frac{2}{\gamma - 1}, \quad B = -2w_2, \quad C = \frac{\gamma - 1}{2}(w_2^2 - 2H_t). \end{gathered} \tag{2.64}$$

The solution of (2.64) is

$$a_f = \frac{-B + \sqrt{B^2 - 4AC}}{2A}, \tag{2.65}$$

and the Mach number on the face is calculated as

$$M_f = \frac{2a_f + (\gamma - 1)w_2}{(\gamma - 1)a_f}, \tag{2.66}$$

where w_2 is calculated by the quantities at l. Finally, the boundary conditions are computed as follows:

$$\begin{aligned} & p_f = p_t \left[1 + \frac{\gamma - 1}{2} M_f\right]^{-\frac{\gamma}{\gamma-1}}, \\ & T_f = T_t \left[1 + \frac{\gamma - 1}{2} M_f\right]^{-1}, \quad \rho_f = \frac{p_f}{RT_f}, \\ & u_{i,f} = u_{n,f}\hat{n}_i. \end{aligned} \tag{2.67}$$

Subsonic Outflow Condition with Constant Static Pressure This condition is used as the outflow condition for internal flows. Here, the boundary value of the static pressure p_{set} is subscribed. The boundary values are computed as

$$p_f = p_{set}, \quad u_{i,f} = u_{i,l}, \quad \rho_f = \frac{\rho_l p_f}{p_l}. \tag{2.68}$$

Boundary Condition for the SA Equation The far-field boundary condition $\tilde{\nu}$ is as follows. For the symmetry condition, always $\tilde{\nu}_f = \tilde{\nu}_l$. For the other boundary conditions, $\tilde{\nu}$ is depending on the face-normal velocity:

$$\tilde{\nu} = \begin{cases} \tilde{\nu}_l & (u_{n,f} \geq 0) \\ \tilde{\nu}_\infty & (u_{n,f} < 0) \end{cases}, \tag{2.69}$$

where the far-field value $\tilde{\nu}_\infty$ is set to $3\nu_\infty$.

2.5 Temporal Discretization

2.5.1 Implicit Time Integration

The time is advanced by the Matrix-Free Gauss–Seidel (MFGS) [86]. This method is derived from the spatially discretized governing equations (2.18) as follows. Here, the time is discretized by the Euler implicit method as

$$\frac{\Delta \mathbf{Q}_l^n}{\Delta t} = -\frac{1}{V_l}\sum_{r\in l}\left(\tilde{\mathbf{F}}_{lr}^{n+1} - \tilde{\mathbf{G}}_{lr}^{n+1}\right) s_{lr}, \tag{2.70}$$

where the superscripts n and $n+1$ denotes the values at those time steps, and

$$\Delta \mathbf{Q}_l^n = (\mathbf{Q}_l^{n+1} - \mathbf{Q}_l^n). \tag{2.71}$$

The fluxes can be linearized as

$$\begin{aligned} \tilde{\mathbf{F}}^{n+1} &= \tilde{\mathbf{F}}^n + \left.\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{Q}}\right|^n \Delta \mathbf{Q}^n, \\ \tilde{\mathbf{G}}^{n+1} &= \tilde{\mathbf{G}}^n + \left.\frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{Q}}\right|^n \Delta \mathbf{Q}^n, \end{aligned} \tag{2.72}$$

Then, the Jacobian matrix

$$\mathbf{A} = \left(\frac{\partial \tilde{\mathbf{F}}}{\partial \mathbf{Q}} - \frac{\partial \tilde{\mathbf{G}}}{\partial \mathbf{Q}}\right) \tag{2.73}$$

is split into the upwind and downwind components:

$$\mathbf{A} = \mathbf{A}^+ + \mathbf{A}^-, \tag{2.74}$$

where the superscripts + and − denote the upwind and downwind components, respectively. These components are identical with the Jacobian matrix which consists of only the positive or negative eigenvalues. The Jacobian should be evaluated by a first-order upwind method to assure numerical stability. Thus, the upwind component of the Jacobian matrix is calculated by the physical quantities of cell l, and the downwind component by those of cell r. Using this Jacobian splitting, the fluxes at the $n+1$ step in (2.70) become

$$\tilde{\mathbf{F}}_{lr}^{n+1} + \tilde{\mathbf{G}}_{lr}^{n+1} = \tilde{\mathbf{F}}_{lr}^{n} + \tilde{\mathbf{G}}_{lr}^{n} + \mathbf{A}_l^{+,n}\Delta \mathbf{Q}_l^n + \mathbf{A}_r^{-,n}\Delta \mathbf{Q}_r^n. \tag{2.75}$$

Substituting (2.75) into (2.70) yields

$$\begin{aligned} \left[\frac{V_l}{\Delta t} + \sum_{r\in l}\mathbf{A}_l^{+,n} s_{lr}\right]\Delta \mathbf{Q}_l^n + \sum_{r\in l}\mathbf{A}_r^{-,n} s_{lr}\Delta \mathbf{Q}_r^n \\ = \sum_{r\in l}\left(\tilde{\mathbf{F}}_{lr}^n - \tilde{\mathbf{G}}_{lr}^n\right) s_{lr}. \end{aligned} \tag{2.76}$$

In the explanation below, the superscript n is omitted, and the right-hand side of (2.76) is just written as $(R.H.S.)$ for brevity. Here, the Jacobian matrix is approximated as follows:

$$\mathbf{A}^{\pm} = \frac{\mathbf{A} \pm \sigma_A \mathbf{I}}{2}, \tag{2.77}$$

where σ_A is called "spectral radius" which is the maximum eigenvalue of $\mathbf{A}$, and $\mathbf{I}$ is the identity matrix. The spectral radius for the RANS equation is

$$\sigma_A = |u_n| + a + \frac{2(\nu + \nu_t)}{\Delta x}, \tag{2.78}$$

where u_n is the normal velocity to the face between cells l and r. Substituting (2.77) into (2.76) yields

$$\left[\frac{V_l}{\Delta t} + \frac{1}{2}\sum_{r\in l}\sigma_{A,l}s_{lr}\right]\Delta\mathbf{Q}_l + \frac{1}{2}\sum_{r\in l}\left(\mathbf{A}_{rl} - \sigma_{A,r}\mathbf{I}\right)s_{lr}\Delta\mathbf{Q}_r = (R.H.S.), \tag{2.79}$$

because

$$\sum_{r\in l}\mathbf{A}_{lr}s_{lr} = \mathbf{0}. \tag{2.80}$$

In addition, $\mathbf{A}_{rl}\Delta\mathbf{Q}_r$ in the second term of the left-hand side of (2.79) is approximated as

$$\begin{aligned}\sum_{r\in l}\mathbf{A}_{rl}\Delta\mathbf{Q}_r s_{lr} &\simeq \sum_{r\in l}\left[\tilde{\mathbf{F}}(\mathbf{Q}_r + \Delta\mathbf{Q}_r) - \tilde{\mathbf{F}}(\mathbf{Q}_r)\right]s_{lr}\\ &\simeq \sum_{r\in l}\left[\tilde{\mathbf{F}}(\mathbf{Q}_l + \Delta\mathbf{Q}_r) - \tilde{\mathbf{F}}(\mathbf{Q}_l)\right]s_{lr}\\ &\simeq \sum_{r\in l}\tilde{\mathbf{F}}(\mathbf{Q}_l + \Delta\mathbf{Q}_r)s_{lr}.\end{aligned} \tag{2.81}$$

Here, $\tilde{\mathbf{F}}(\mathbf{Q}_r) \simeq \tilde{\mathbf{F}}(\mathbf{Q}_l)$ is assumed to reduce the computational cost [86]. Finally, the time evolution of $\mathbf{Q}$ is achieved as follows:

$$\begin{aligned}&\Delta\mathbf{Q}_l =\\ &\left[\frac{V_l}{\Delta t} + \frac{1}{2}\sum_{r\in l}\sigma_{A,l}s_{lr}\right]^{-1}\left[-\frac{1}{2}\sum_{r\in l}\left(\tilde{\mathbf{F}}(\mathbf{Q}_l + \Delta\mathbf{Q}_r) - \sigma_{A,r}\Delta\mathbf{Q}_r\right)s_{lr} + (R.H.S.)\right].\end{aligned} \tag{2.82}$$

(2.82) is solved by a Gauss–Seidel method because the right hand side of the equation includes $\Delta\mathbf{Q}_r$. In UTCart, four Gauss–Seidel iterations are conducted for each time step. Although the derivation described above includes many approximations, a steady-state solution is obtained as $\Delta\mathbf{Q} \to 0$. Thus, the steady-state solution is

$$(R.H.S.) = 0, \tag{2.83}$$

which is same as the solution of the original discretized equation (2.18).

For the SA turbulence model, the governing equation includes source term $\mathcal{S}$. The source terms at the next time step are evaluated by a similar linearization as (2.72):

$$\mathcal{S}^{n+1} = \mathcal{S}^n + \left.\frac{\partial\mathcal{S}}{\partial\tilde{\nu}}\right|^n\Delta\tilde{\nu}^n, \tag{2.84}$$

where the Jacobian for the source term is

$$\begin{aligned}
&\frac{\partial \mathcal{S}}{\partial \tilde{\nu}} = \frac{\partial (Production)}{\partial \tilde{\nu}} - \frac{\partial (Destruction)}{\partial \tilde{\nu}}, \\
&\frac{\partial (Production)}{\partial \tilde{\nu}} = c_{b1}\left(\tilde{\nu}\frac{\partial \tilde{S}}{\partial \tilde{\nu}} + \tilde{S}\right), \\
&\frac{\partial (Destruction)}{\partial \tilde{\nu}} = c_{w1}\left(f_w \frac{2\tilde{\nu}}{d^2} + \frac{\partial f_w}{\partial \tilde{\nu}}\frac{\tilde{\nu}^2}{d^2}\right), \\
&\frac{\partial \tilde{S}}{\partial \tilde{\nu}} = \frac{1}{\kappa^2 d^2}\left(f_{v2} + \tilde{\nu}\frac{\partial f_{v2}}{\partial \tilde{\nu}}\right), \quad \frac{\partial f_w}{\partial \tilde{\nu}} \simeq \frac{\partial g}{\partial \tilde{\nu}} = \frac{\partial r}{\partial \tilde{\nu}}\left\{c_{w2}(6r^5 - 1) + 1\right\}, \\
&\frac{\partial f_{v2}}{\partial \tilde{\nu}} = \frac{1}{\nu}\frac{\chi^2\dfrac{\partial f_{v1}}{\partial \chi} - 1}{(1 + \chi f_{v1})^2}, \quad \frac{\partial f_{v1}}{\partial \chi} = \frac{3c_{v1}^3\chi^2}{(\chi^3 + c_{v1}^3)^2}, \quad \frac{\partial r}{\partial \tilde{\nu}} = \frac{1}{\tilde{S}\kappa^2 d^2}\left(1 - \frac{\tilde{\nu}}{\tilde{S}}\frac{\partial \tilde{S}}{\partial \tilde{\nu}}\right).
\end{aligned} \tag{2.85}$$

To accelerate the convergence, the source term is treated explicitly when the production term exceeds the destruction term [87]. Thus, the Jacobian for the source term is modified as

$$\mathcal{J} = \frac{1}{2}\left(\frac{\partial \mathcal{S}}{\partial \tilde{\nu}} - \left|\frac{\partial \mathcal{S}}{\partial \tilde{\nu}}\right|\right). \tag{2.86}$$

Consequently, the equation for the Gauss–Seidel iterations (corresponding to (2.82)) is obtained as

$$\begin{aligned}
\Delta\tilde{\nu}_l = &\left[\frac{V_l}{\Delta t} + \frac{1}{2}\sum_{r\in l}\sigma_{N,l}s_{lr} - \mathcal{J}_l V_l\right]^{-1} \\
&\left[-\frac{1}{2}\sum_{r\in l}\left(\Delta\tilde{\nu}_r u_n - \sigma_{N,r}\Delta\tilde{\nu}_r\right)s_{lr} + (R.H.S.)\right],
\end{aligned} \tag{2.87}$$

where spectral radius for the transport equation of the SA turbulence model is

$$\sigma_N = u_n + \frac{\nu + \tilde{\nu}}{\sigma\Delta x}. \tag{2.88}$$

2.5.2 Local Time-Stepping

All the computations in this research are steady, and thus, a local time-stepping method is introduced to accelerate convergence. In this method, the time increment Δt is related to the local cell size by the Courant number C_n as

$$\Delta t_l = C_n \frac{\Delta x_l}{a_l + \sqrt{u_{i,l}u_{i,l}}}. \tag{2.89}$$

Convergence to the steady state is accelerated because the time increment is large in the cells away from the wall boundary.

2.6 Parallelization of the Flow Computation

For large scale simulations, the flow computation part in UTCart is parallelized by the Message Passing Interface (MPI). The generated Cartesian grid is partitioned using the METIS library [88] for the parallel computation. In each divided domain of the grid, sleeve cells for communication with other domains are specified. In addition, the physical quantities at the IP must be available to determine the boundary condition. Therefore, the cell including the IP is added to the list of sleeve cells if the cell belongs to a different domain from where the boundary condition is imposed. Note that the parallelization of UTCart is also described in [89].

Chapter 3

Wall Boundary Conditions for Turbulent Flows

In this chapter, wall boundary conditions for UTCart to simulate turbulent flows are presented. First, the approximation of the RANS equations in near-wall region of a turbulent boundary layer is analyzed in Section 3.1. Then, in Section 3.2, the wall boundary conditions are developed based on this approximation. Finally, Section 3.3 describes the calculation method for the cutoff-distance required in the proposed boundary condition.

3.1 Boundary Layer Approximation of the Governing Equations

In the near-wall region (inner-layer) of a turbulent boundary layer, the flow is quasi-one-dimensional, i.e., the wall normal component of the velocity is much smaller than the wall tangential component. This inner layer is also assumed to be quasi-steady when a RANS simulation is considered. In addition, the influence of the pressure gradient is generally smaller than that of the viscosity in the inner layer, as discussed in [61]. Here, a flat-plate turbulent boundary layer in the $x-y$ plane is considered. The x- and y-axes are set to the wall tangential and normal directions, respectively. In the following subsections, the approximations of the governing equations in this situation are examined.

3.1.1 Momentum Equation

Under the assumption in the inner layer of the boundary layer, the momentum equation (2.2) is reduced to an ordinary differential equation about y:

$$\frac{1}{\rho}\frac{\mathrm{d}\tau_{xy}}{\mathrm{d}y} = \frac{\mathrm{d}}{\mathrm{d}y}\left[(\nu+\nu_t)\frac{\mathrm{d}u}{\mathrm{d}y}\right] = 0. \tag{3.1}$$

Integrating (3.1) about y from the wall yields

$$(\nu+\nu_t)\frac{\mathrm{d}u}{\mathrm{d}y} = \frac{\tau_w}{\rho}, \tag{3.2}$$

where τ_w is the magnitude of the shear stress on the wall. (3.2) indicates that the shear stress is constant with respect to y. Note that the shear stress increases or decreases in an actual boundary layer as y increases, due to the advection or pressure gradient neglected in (3.1). Here, (3.2) is non-dimensionalized as follows:

$$(1+\nu^+)\frac{\mathrm{d}u^+}{\mathrm{d}y^+} = 1, \tag{3.3}$$

where the non-dimensional velocity, length and eddy-viscosity are

$$u^+ = \frac{u}{u_\tau}, \quad y^+ = \frac{yu_\tau}{\nu}, \quad \nu^+ = \frac{\nu_t}{\nu}, \tag{3.4}$$

with the friction velocity $u_\tau = \sqrt{\tau_w/\rho}$.

3.1.2 SA Turbulence Model

The SA turbulence model is also reduced to an ordinary differential equation under the assumption in the inner layer of the boundary layer as

$$\begin{aligned}
0 &= (Production) - (Destruction) + (Diffusion),\\
(Production) &= c_{b1}\left(\frac{\mathrm{d}u}{\mathrm{d}y} + \frac{\tilde{\nu}}{(\kappa y)^2}f_{v2}\right)\tilde{\nu},\\
(Destruction) &= c_{w1}f_w\left(\frac{\tilde{\nu}}{y}\right)^2,\\
(Diffusion) &= \frac{1}{\sigma}\left[\frac{\mathrm{d}}{\mathrm{d}y}\left((\nu+\tilde{\nu})\frac{\mathrm{d}\tilde{\nu}}{\mathrm{d}y} + c_{b2}\left(\frac{\mathrm{d}\tilde{\nu}}{\mathrm{d}y}\right)^2\right)\right].
\end{aligned} \tag{3.5}$$

Note that the wall boundary exist at $y = 0$, and thus, $d = y$. Here, (3.5) is non-dimensionalized with dividing each term by u_τ^2:

$$\begin{aligned}
0 &= (Production') - (Destruction') + (Diffusion')\\
(Production') &= c_{b1}\left(\frac{\mathrm{d}u^+}{\mathrm{d}y^+} + \frac{\chi}{(\kappa y^+)^2}f_{v2}\right)\chi\\
(Destruction') &= c_{w1}f_w\left(\frac{\chi}{y^+}\right)\\
(Diffusion') &= \frac{1}{\sigma}\left[\frac{\mathrm{d}}{\mathrm{d}y^+}\left((1+\chi)\frac{\mathrm{d}\chi}{\mathrm{d}y^+} + c_{b2}\left(\frac{\mathrm{d}\chi}{\mathrm{d}y^+}\right)^2\right)\right].
\end{aligned} \tag{3.6}$$

The solution of (3.6) is linear with respect to y^+:

$$\chi = \kappa y^+. \tag{3.7}$$

With a dimensional form, the solution is

$$\tilde{\nu} = \kappa u_\tau y. \tag{3.8}$$

When this solution is satisfied, the production term becomes

$$(Production') = c_{b1} \left\{ \chi \frac{\mathrm{d}u^+}{\mathrm{d}y^+} + \left(1 - \frac{\chi}{1 + \chi f_{v1}} \right) \right\}. \tag{3.9}$$

In addition, the following equation is derived from (3.3):

$$\frac{1}{1 + \chi f_{v1}} = \frac{\mathrm{d}u^+}{\mathrm{d}y^+}. \tag{3.10}$$

Substituting (3.10) into (3.9) yields

$$(Production') = c_{b1}. \tag{3.11}$$

Furthermore, r in the definition of f_w is 1 when the solution (3.7) is satisfied. Thus, the other terms in (3.6) become

$$\begin{aligned} (Destruction') &= c_{w1}\kappa^2, \\ (Diffusion') &= \frac{1 + c_{b2}}{\sigma}\kappa^2. \end{aligned} \tag{3.12}$$

As shown above, all the three terms in (3.6) are constant when the near-wall solutions of the velocity and $\tilde{\nu}$ are satisfied. Note that the actual solution deviates from this near-wall solution when y increases because the advection term or the derivative of $\tilde{\nu}$ in the x-direction becomes not negligible.

3.1.3 Wall Function

The solution for the velocity under the inner-layer approximation is called "wall function". In the inner layer except for the region very close to the wall, $\tilde{\nu}$ is much larger than ν, and thus, $f_{v1} \simeq 1$. Therefore, substituting $\nu^+ \simeq \chi = \kappa y^+$ into (3.3) and integrating it about y^+ from the wall yields

$$\frac{\mathrm{d}u^+}{\mathrm{d}y^+} \simeq \frac{1}{\kappa y^+}. \tag{3.13}$$

When this equation is integrated about y^+, the following equation is obtained:

$$u^+ \simeq \frac{1}{\kappa} \log y^+ + C, \tag{3.14}$$

where C is an integration constant. The relationship (3.14) is generally called "log-law." This is a universal law for the velocity profile in the inner layer of a turbulent boundary layer.

In the region very close to the wall, the eddy-viscosity is relatively small compared to the molecular viscosity. The derivation above does not include the effect of the molecular viscosity, and thus, (3.14) is not applicable to this region. To include the effect of the neglected molecular viscosity and construct a universal law for the wall, the SA wall model is developed in [90]. First, the near-wall solution $\chi = \kappa y^+$ is substituted to (3.10):

$$\frac{\mathrm{d}u^+}{\mathrm{d}y^+} = \frac{c_{v1}^3 + (\kappa y^+)^3}{c_{v1}^3 + (\kappa y^+)^3 + (\kappa y^+)^4}. \tag{3.15}$$

Then, integrating (3.15) about y^+ yields a near-wall asymptotic solution of the SA turbulence model:

$$\begin{aligned} u^+ &= f_{SA}(y^+) \\ &= \bar{B} + c_1 \log\left((y^+ + a_1)^2 + b_1^2\right) + c_2 \log\left((y^+ + a_2)^2 + b_2^2\right) \\ &- c_3 \mathrm{Arctan}[y^+ + a_1, b_1] - c_4 \mathrm{Arctan}[y^+ + a_2, b_2], \end{aligned} \tag{3.16}$$

where

$$\begin{aligned} \bar{B} &= 5.03339088, \quad a_1 = 8.1482158, \quad a_2 = -6.92870938, \\ b_1 &= 7.46008761, \quad b_2 = 7.46814579, \\ c_1 &= 2.54967735, \quad c_2 = 1.33016516, \\ c_3 &= 3.59945911, \quad c_4 = 3.63975319. \end{aligned}$$

The shape of this function is shown in Fig. 3.1. The region with $y^+ < 5$ is called "viscous sublayer", where work of the molecular viscosity dominates. In contrast, the effect of the molecular viscosity is negligible and the log-law of the velocity is satisfied for $y^+ > 30$. This region is called "inertial sublayer". In addition, the region in between these two layers is the buffer layer, where the velocity gradient rapidly changes.

3.2 Boundary Conditions using a Turbulent Wall Function

For turbulent flow simulations, the IBM described in Section 2.4 is combined with the turbulent wall function. First, the skin friction on the wall is determined using the turbulent wall function. The tangential velocity at the IP must satisfy the velocity profile defined by the SA wall model:

$$u_{t,IP} = u_\tau f_{SA}(y_{IP}^+), \tag{3.17}$$

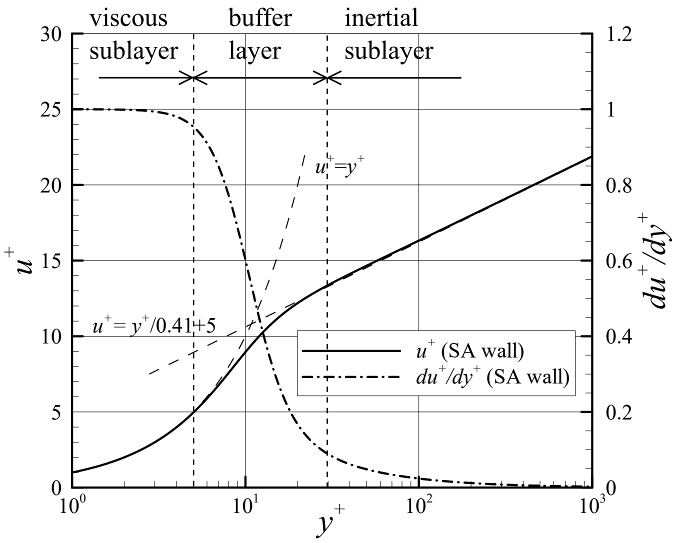


Figure 3.1: The SA wall model developed in [90].

where $f_{SA}(y^+)$ is the SA wall model defined by (3.16). y^+ in (3.17) includes u_τ in its definition, and thus, iterations are needed to determine u_τ. Then, the friction velocity is calculated by Newton iterations as

$$u_\tau^{new} = u_\tau - \frac{u_\tau f_{SA}(y_{IP}^+) - u_{t,IP}}{f_{SA}(y_{IP}^+) + u_\tau \dfrac{\mathrm{d}f_{SA}}{\mathrm{d}u_\tau}}, \tag{3.18}$$

where u_τ^{new} is the friction velocity at the next iteration cycle, and

$$\frac{\mathrm{d}f_{SA}}{\mathrm{d}u_\tau} = \frac{y^+}{u_\tau}\frac{\mathrm{d}f_{SA}}{\mathrm{d}y^+}. \tag{3.19}$$

The first derivative of the SA wall model is calculated by (3.15). The iteration starts from the converged value at the previous time step of flow calculation. Convergence of u_τ is generally obtained with less than ten iteration steps, and the required number of iterations decreases as the flow converges to a steady state.

After u_τ is calculated, the physical quantities at FC are determined. In the original IBM proposed by [38], the tangential velocity at FC is calculated as

$$u_{t,FC} = u_\tau f_{SA}(y_{FC}^+). \tag{3.20}$$

The boundary conditions for the normal velocity and pressure are the same as that for the non-slip walls described in Section 2.4. Furthermore, the temperature at FC is calculated by the Crocco–Busemann relationship [69] as

$$T_{FC} = T_{IP} + \frac{Pr^{1/3}}{2c_p}(u_{t,IP}^2 - u_{t,FC}^2), \tag{3.21}$$

where c_p is the specific heat at constant pressure $(= \gamma R/(\gamma - 1))$. Then, the density at FC is calculated:

$$\rho_{FC} = \frac{p_{FC}}{RT_{FC}}. \tag{3.22}$$

In addition, the boundary condition for $\tilde{\nu}$ is calculated by the near-wall solution of the SA turbulence model:

$$\tilde{\nu}_{FC} = \kappa u_\tau d_{FC}. \tag{3.23}$$

The inviscid flux at FC is calculated by the same method as that used for the slip/non-slip walls. In addition, the viscous flux is calculated using the calculated u_τ. Here, the stress tensor at FC (see (2.51) for the non-slip wall boundary condition) is

$$\tau_{ij} = \rho u_\tau^2 (n_i t_j + n_j t_i). \tag{3.24}$$

Note that the velocity is zero at the separation point, and $u_\tau = 0$ accordingly. Under this condition, both velocity and viscous flux at FC are zero.

However, the method described above cannot reproduce turbulent boundary layers accurately when the cell boundary is not attached to the actual object surface, as reported in the previous study [26]. Modifications for this method are therefore presented in the following subsections.

3.2.1 Modification for the Near-Wall Velocity Profile

The velocity profile defined by the wall function is a nonlinear function. However, a spatial scheme with second-order spatial accuracy can reconstruct only a linear (first-order polynomial) profile within a cell. As a result, the velocity profile assumed by the wall function cannot be reproduced correctly on the grid. This seems to be the origin of the inaccuracy observed in the results obtained by the previous method.

To overcome this problem, the velocity profile should be modified so that it is linear. This idea originates in [36]. The profile of the tangential velocity is modified using the first derivative of the SA wall model (3.15):

$$u^+(y^+) = u^+_{IP} - \left.\frac{\mathrm{d}f_{SA}}{\mathrm{d}y^+}\right|_{IP} (y^+_{IP} - y^+). \tag{3.25}$$

u^+ at FC is calculated with $y^+ = y^+_{FC}$ in (3.25). In this velocity profile (Fig. 3.2), the tangential velocity at $y = 0$ is nonzero, and thus, a virtual slip velocity is imposed on the wall.

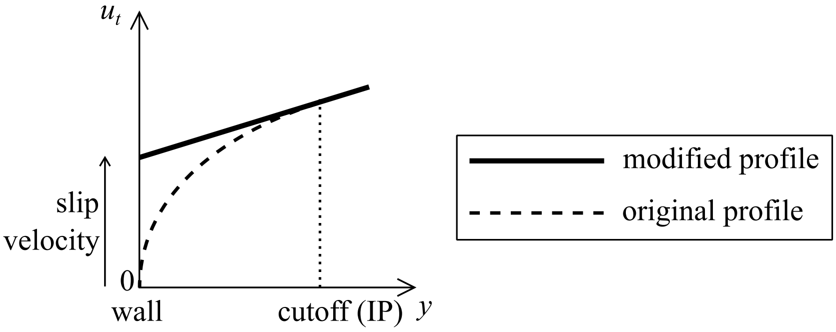


Figure 3.2: Modification of the velocity profile.

3.2.2 Modification of the Eddy-Viscosity Profile

Although [36] proposed the modification for the velocity profile, the near-wall eddy-viscosity profile has not been discussed in detail. The key issue here is that the balance of the shear stress (see (3.2)) must be maintained. Thus, a modification is required on the eddy-viscosity profile corresponding to the modification for the velocity profile.

In the modified velocity profile (3.25), the velocity gradient ($\mathrm{d}u/\mathrm{d}y$) is constant. Accordingly, ν_t must be constant in the region between the IP and wall to maintain the constant shear stress implied by (3.2). Here, the near-wall solution of $\tilde{\nu}$ is retained, and only the wall-damping function f_{v1} in (2.10) is modified to avoid additional complexity. To realize the constant profile of the eddy-viscosity, the wall-damping function must be

$$f_{v1} \sim \frac{1}{d}, \tag{3.26}$$

because the near-wall solution of $\tilde{\nu}$ is proportional to the wall distance d.

For the implementation, the profile of f_{v1} must be continuous. Thus, the wall-damping function is redefined as

$$f_{v1} = \begin{cases} f_{v1,original} & (d \geq d_{cutoff}) \\ f_{v1,near-wall} & (d < d_{cutoff}) \end{cases}, \tag{3.27}$$

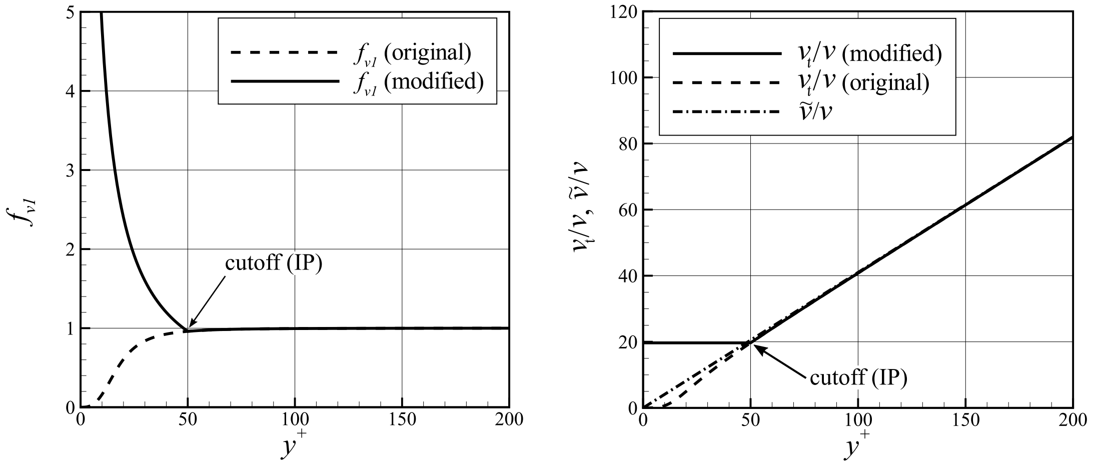
where d_{cutoff} is a cutoff distance that equals to the distance between the IP and wall. $f_{v1,original}$ is the original damping function:

$$f_{v1,original} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \tag{3.28}$$

and $f_{v1,near-wall}$ is a modified damping function near the wall defined as

$$f_{v1,near-wall} = r_d \frac{(\chi r_d)^3}{(\chi r_d)^3 + c_{v1}^3}, \tag{3.29}$$

where $r_d = d_{cutoff}/d$. Note that $f_{v1,near-wall}$ is a product of r_d and the original f_{v1} at $d = d_{cutoff}$. When the IP is located in the log-layer of the turbulent boundary layer, $f_{v1,near-wall}$ is approximately equal to r_d. This function depends on the relative position of the IP in the boundary layer. For example, the shape of the function with $y^+_{IP} = 50$ is shown in Fig. 3.3.



(a) Wall damping function

(b) Eddy viscosity

Figure 3.3: Modification for the eddy-viscosity profile

The modified eddy-viscosity profile has a kink at the cutoff point. Thus, the following technique is adopted to calculate when the viscous fluxes on the faces are calculated. Here, face lr, the face between cell l and r is considered. To calculate the viscous flux on face l, r, $\nu_{t,lr}$, the eddy-viscosity on the face is needed. However, the simple average of $\nu_{t,l}$ and $\nu_{t,r}$ is different from the true value on the profile if the kink exists between cell l and r, and thus, numerical errors may be caused.

Thus, the following procedure is adopted to eliminate the effect of the kink. The averages of $\tilde{\nu}$, ν and d are firstly calculated:

$$\begin{aligned} \tilde{\nu}_{lr} &= r_{lr}\tilde{\nu}_l + (1 - r_{lr})\tilde{\nu}_r, \\ \nu_{lr} &= r_{lr}\nu_l + (1 - r_{lr})\nu_r, \\ d_{lr} &= r_{lr}d_l + (1 - r_{lr})d_r, \end{aligned} \tag{3.30}$$

where r_{lr} is the ratio of cell sizes:

$$r_{lr} = \frac{\Delta x_r}{\Delta x_l + \Delta x_r}. \tag{3.31}$$

Then, the eddy-viscosity is calculated by those quantities:

$$\nu_t|_{lr} = \tilde{\nu}_{lr} f_{v1}(\chi_{lr}, d_{lr}), \tag{3.32}$$

where $\chi_{lr} = \tilde{\nu}_{lr}/\nu_{lr}$. The profile of $\tilde{\nu}$ is nearly linear in near the wall, and the numerical error becomes smaller than taking the simple average of ν_t.

3.2.3 Thermal Boundary Condition

In the original method, the temperature on the wall is calculated based on the Crocco–Busemann relationship. Corresponding to the modification for the velocity and eddy-viscosity profiles, the thermal boundary condition is now reconsidered. Differentiating the Crocco–Busemann relationship (3.21) yields

$$\frac{\mathrm{d}T}{\mathrm{d}y} = -\frac{\mathrm{Pr}^{\frac{1}{3}}}{c_p} u_t \frac{\mathrm{d}u_t}{\mathrm{d}y}. \tag{3.33}$$

In the modified velocity profile (3.25), the normal gradient of the tangent velocity is constant below the IP. Here, u_t is also assumed to be nearly constant because the velocity gradient in the log-layer is small.

Thus, the temperature gradient is nearly constant below the IP, and the temperature profile becomes a linear profile (Fig. 3.4) as

$$T_{FC} = T_{IP} - \frac{\mathrm{d}T}{\mathrm{d}y}_{IP}(y_{IP} - y_{FC}), \tag{3.34}$$

where the temperature gradient at IP is calculated by (3.33) as

$$\left.\frac{\mathrm{d}T}{\mathrm{d}y}\right|_{IP} = -\frac{\mathrm{Pr}^{\frac{1}{3}}}{c_p} u_{t,IP} \left.\frac{\mathrm{d}u_t}{\mathrm{d}y}\right|_{IP}. \tag{3.35}$$

In addition, the velocity gradient is calculated using the first derivative of the SA wall model (3.15) as

$$\left.\frac{\mathrm{d}u_t}{\mathrm{d}y}\right|_{IP} = \frac{u_\tau^2}{\nu} \left.\frac{\mathrm{d}f_{SA}}{\mathrm{d}y^+}\right|_{IP}. \tag{3.36}$$

Finally, the heat flux at FC is calculated using the temperature gradient (3.35) directly:

$$q_{j,FC} = -\rho_{FC} c_p \left(\frac{\nu}{\mathrm{Pr}} + \frac{\nu_t}{\mathrm{Pr_t}}\right)_{FC} \left.\frac{\mathrm{d}T}{\mathrm{d}y}\right|_{IP} n_j. \tag{3.37}$$

This heat flux is added as a part of the boundary viscous flux at FC.

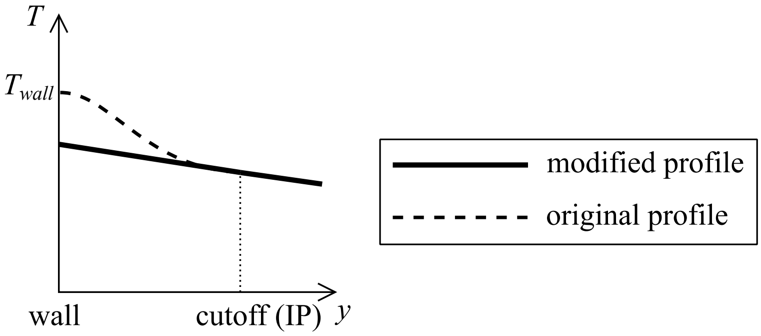


Figure 3.4: Modification for the temperature profile.

3.2.4 Influence on the Near-Wall Solution of the SA Turbulence Model

Here, it is confirmed that the modifications for the velocity and eddy-viscosity profiles do not change the near-wall solution of the SA turbulence model (3.8). As examined in (3.11), the production term is constant when the near-wall solution of the SA turbulence model and the balance of the shear stress (3.2) are maintained. This is independent of the choice of f_{v1}. In addition, $\tilde{S} = u_\tau/(\kappa y)$ is independent of f_{v1} or the velocity gradient. Thus, the destruction and diffusion terms are not influenced by the changes in the near-wall profiles. Consequently, the near wall solution $\tilde{\nu} = \kappa u_\tau y$ is still a solution of the approximated SA equation (3.5) even when the near-wall profiles are modified.

3.3 Calculation of Wall and Cutoff Distances

The SA turbulence model requires information of the distance to the nearest wall. If the nearest wall is directly searched, the computational cost rapidly increases as the grid become finer. This is because the cost is proportional to the product of the number of cells and the number of the line (or facet) segments composing the object surface. Thus, the level-set method [91] is used to reduce the computational cost. The transport equation for the distance function $\phi = \phi(x_i, \tau)$ is

$$\frac{\partial \phi}{\partial \tau} = 1 - \sqrt{\frac{\partial \phi}{\partial x_i}\frac{\partial \phi}{\partial x_i}}, \tag{3.38}$$

where τ is the pseudo time. The distance d is obtained as the solution of (3.38) as $\tau \to \infty$.

In addition, a cutoff distance is required to calculate the modified damping function (3.27). This cutoff distance is equal to the value of the IP distance d_{IP} assigned to the nearest wall from the cell. When multiple objects (or parts) with different assigned cell sizes exist in the computational domain, all the cells in the domain need to know which object is the nearest. To avoid the direct searching, a method to detect the nearest wall using an advection equation is developed. The cutoff distance is convected from the wall in the wall normal direction because the value must be constant in that direction. Thus, the equation for the cutoff distance function ψ is defined as

$$\frac{\partial \psi}{\partial t} + n_i \frac{\partial \psi}{\partial x_i} = 0, \tag{3.39}$$

where n_i are the components of the wall normal vector $\mathbf{n}$. The cutoff distance is obtained as the solution as $\tau \to \infty$ Furthermore, the components of the wall normal vector are calculated by differentiating the distance function ϕ as

$$n_i = \frac{\partial \phi}{\partial x_i}. \tag{3.40}$$

In the solver, the WLSQ [77] is used to approximate this differentiation.

Here, (3.39) is discretized by the FVM:

$$\frac{\partial \psi_l}{\partial \tau} = -\frac{1}{V_l} \sum_{r \in l} \tilde{H}_{lr} s_{lr}, \tag{3.41}$$

where $\tilde{H}$ is an numerical flux, and the subscripts are identical with those used in (2.18). The wall boundary condition is imposed at FC as $\psi_{FC} = r_{IP} \Delta x_F$ with Δx_F being the size of the fluid cell that includes FC. In addition, a zero-gradient condition is imposed at the far-field.

To stabilize the computation, an upwind scheme is used to determine the numerical flux. The upwind numerical flux is calculated by the face-left and right values as

$$\tilde{H}_{lr} = \begin{cases} \psi_l \alpha_l & (\alpha_l > 0) \\ \frac{\psi_l + \psi_r}{2} \alpha_l & (\alpha_l \geq 0 \text{ and } \alpha_r < 0) \\ \psi_r \alpha_l & (\alpha_l \leq 0 \text{ and } \alpha_r \leq 0) \end{cases}, \tag{3.42}$$

where the advection speed α is the inner product of $\mathbf{n}$ and the face normal $\tilde{\mathbf{n}}$. Note that the upwind flux (3.42) is that seeing from cell l, and the flux seeing from cell r is different (i.e., $\tilde{H}_{lr} \neq \tilde{H}_{rl}$). This upwinding suppresses the numerical oscillation when informations from two different walls collide.

The wall and cutoff distances are time-invariant if the surface of the object is fixed to the space. Thus, they are calculated at the beginning of the flow calculation and used throughout the whole simulation. The calculation is similarly parallelized using the MPI library, as described in Section 2.6.

Figure 3.5 shows an example of the result of the cut-off distance calculation. The cell sizes assigned on the lower and upper airfoils are 0.02 and 0.04, respectively. In addition, a local refinement with $\Delta x = 0.01$ is specified near the leading edge of the lower airfoil. Here, the computational domain is divided into three domains, depending on the cell size assigned to the nearest wall. Furthermore, the calculated cutoff distance is almost equal to the intended value.

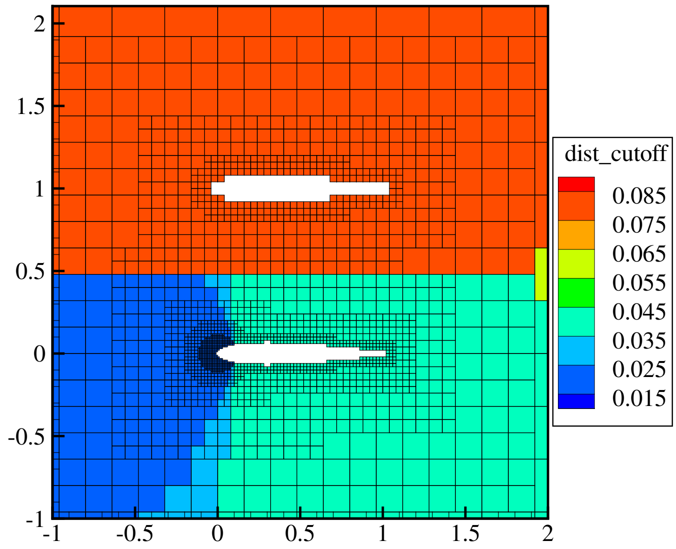


Figure 3.5: Calculation example of the cut-off distance.

3.4 Summary of Chapter 3

In this chapter, the wall boundary conditions for UTCart to simulate turbulent flows were presented. First, the near-wall behavior of the RANS equation and the SA turbulence model was examined in Section 3.1. From on the near-wall approximation of the RANS equation, the shear stress is implied to be nearly constant in the wall normal direction. Under this condition, the SA turbulence model has a linear solution with respect to the wall distance.

Then, the IBM for turbulent flow simulations was developed in Section 3.2. The turbulent wall function is used to determine the friction velocity on the wall, and the viscous flux at the boundary is directly calculated by this friction velocity. In addition, physical quantities at the boundary are determined to calculate the convective flux at the boundary. To avoid numerical problems, the profile of the wall tangential velocity was modified so that it is linear. This is because the original velocity profile defined by the wall function is nonlinear, and it cannot be resolved on the grid when a second-order spatial scheme is used. The eddy viscosity profile was also modified to maintain the balance of the shear stress that is discussed in Section 3.1. Furthermore, the thermal boundary condition was reconsidered corresponding to the modification for these profiles.

The developed boundary condition requires the cutoff distance, which is equal to the IP distance on the nearest wall. In Section 3.3, the calculation method for this cutoff distance was developed. The method is based on an advection equation, which can eliminate the direct searching of the nearest wall.

In the verification problems shown later, the baseline method described in Section 3.2 and the method with the modifications described in Sections 3.2.1–3.2.3 are referred to as the “original IBM” and “modified IBM,” respectively. The specification of these methods is tabulated in Table 3.1.

Table 3.1: Specification of the original and modified IBMs.

	Original IBM	Modified IBM
Velocity profile	following the wall function (3.20)	linear (3.25)
f_{v1}	original definition in the SA (2.10)	modified (3.27)
Temperature profile	Crocco–Busemann relationship (3.21)	linear (3.34)

Chapter 4

Force Integration on Cartesian Grids

In this chapter, methods to calculate the aerodynamic force acting on immersed bodies are investigated. To compute the aerodynamic force, a method based on the integration over the input CAD surface, which is called "polygon-based method" in [45], is often used. In this method, the physical quantities (e.g., pressure) on the Cartesian grid is extrapolated onto the CAD surface and then integrated. This method is the same as the force integration method on conventional body-fitted grids except for the extrapolation. However, arbitrariness exists in the extrapolation formula, and thus, the computed force contains additional numerical errors caused by the extrapolation. In addition, the accuracy of the integration depends on the resolution of the CAD surface. For these reasons, the relation between the flow solution and the actual force acting on the immersed body is unclear.

To clarify this point, a new force integration method is developed based on the balance of the numerical flux. This idea is similar to the far-field methods [92, 93], but the integration surface is the step-wise cell boundary near the wall boundary. Unlike the far-field method, the pressure and viscous component of the force can be calculated using this method because the integration surface is near the object surface. The force can also be decomposed into that acting on each object when multiple objects exist in the computational domain. Furthermore, an unsteady force can be evaluated without a volume integration.

The structure of this chapter is as follows. In Section 4.1, the formulation of the force integration over the CAD surface is described. The method based on the balance of the numerical flux is then developed in Section 4.2. The force integration methods are then implemented in UTCart, and test problems are conducted in Section 4.3 to check the validity and accuracy of the proposed methods.

In the explanation below, an immersed body Γ_S in Cartesian grids, as shown in Fig. 4.1, is considered. The step-wise cell boundary near the wall and the far-field boundary are named Γ_G and Γ_F, respectively. Note that the normal vectors of Γ_G and Γ_F are pointing outside the computational domain. Furthermore, the domains between Γ_G and Γ_F, and that between Γ_G and Γ_S are named V_1 and V_2, respectively.

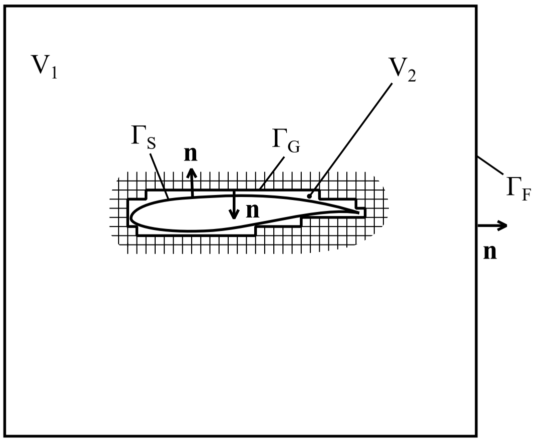


Figure 4.1: Description of the computational domain and the boundaries.

4.1 Force Integration on a CAD Surface

In this method, the aerodynamic force is integrated over the object surface Γ_S. First, a body force term is added to the momentum equation (2.2):

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j) + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} + f_i, \tag{4.1}$$

where f_i represents the body force. The object is considered as the source of the body force in the fluid. Then, the momentum equation (4.1) is integrated within the object as

$$\int_{V_B} \left[\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j) + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} - f_i \right] dV = 0, \tag{4.2}$$

where V_B denotes the volume inside the object. Here, the aerodynamic force acting on the object F_i is the reaction of the body force in (4.2):

$$F_i = -\int_{V_B} f_i dV. \tag{4.3}$$

The flow velocity inside the object is zero, and thus,

$$F_i = \int_{V_B} \left[-\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \right] dV. \tag{4.4}$$

Finally, applying the divergence theorem to (4.4) yields

$$F_i = \int_{\Gamma_S} \left(-(p - p_\infty)\delta_{ij} + \tau_{ij} \right) n_j dS, \tag{4.5}$$

where p_∞ is the free-stream pressure, and n_j are the components of the wall normal vector $\mathbf{n}$. The integral of $p_\infty \delta_{ij} n_j$ works as a compensation for a non-closed surface (e.g., a half-cut model with a symmetry plane) and is zero if the integration surface is closed. For the implementation, (4.5) is discretized for each surface element (i.e., line segment or triangular facet) composing Γ_S.

This method is similar to the integration method on the conventional body-fitted grids, and the pressure and viscous components of the force can be divided by integrating the equation term by term. However, the pressure and stress tensor on Γ_S is not available on the Cartesian grid, and extrapolation of the physical quantities is necessary. To determine the pressure and stress tensor on each surface element, two reference points $I1$ and $I2$ are set on the wall normal line through the gravity center of each element (see Fig. 4.2). Here, d_1 and d_2, the distances from the wall to $I1$ and $I2$ are 1.5 and 3 times the size of the nearest cell, respectively. Note that d_2 is modified so that it is r_{IP} times the cell size when r_{IP} in (2.40) is greater than 3. This is because at least one of the probes should be located outside the region where the velocity profile is modified (see Section 3.2).

In the explanation below, the physical quantities at $I1$ and $I2$ are represented with subscripts 1 and 2, respectively. The physical quantities at $I1$ and $I2$ are linearly interpolated using the gradient at the cell-center including these probes. Here, the stress tensor on the surface is

$$\tau_{ij} = \left(\mu \frac{\partial u_t}{\partial n} \right) n_i t_j, \tag{4.6}$$

where t_j are the components of the unit vector in the flow direction. For a slip wall, the stress the velocity gradient on the wall is zero:

$$\frac{\partial u_t}{\partial n} = 0. \tag{4.7}$$

When the non-slip condition is imposed on the surface, the velocity gradient is calculated by a quadratic polynomial with the velocity being zero on the wall:

$$\frac{\partial u_t}{\partial n} = \frac{u_{t,1} d_2^2 - u_{t,2} d_1^2}{d_1 d_2 (d_2 - d_1)}. \tag{4.8}$$

For the turbulent boundary condition described in Chapter 3, the wall shear stress is calculated by the wall function. Here, the matching of the wall function is calculated at $I2$ similarly as (3.17–3.18). Then, using the calculated u_τ, the wall shear stress is estimated as

$$\tau_{ij} = \rho u_\tau^2 n_i t_j. \tag{4.9}$$

For the surface pressure p_s, arbitrariness exists in the extrapolation formula; for example,

$$p_s = p_1. \tag{4.10}$$

This formula employs a P0 polynomial (constant value) for the extrapolation of the surface pressure. Another method is possible when a P1 (linear) polynomial is used:

$$p_s = p_1 - \frac{d_1}{d_2 - d_1}(p_2 - p_1). \tag{4.11}$$

If the pressure gradient in the wall normal direction is assumed to be zero, a P2 (quadratic) polynomial can also be applied:

$$p_s = p_1 - \frac{d_1^2}{d_2^2 - d_1^2}(p_2 - p_1). \tag{4.12}$$

The extrapolation formulae (4.10–4.12), the methods are named polygon-based (P0), polygon-based (P1) and polygon-based (P2) methods, respectively. As shown above, this method has arbitrariness in the determination of the surface pressure. This process can introduce additional numerical error independent of the actual accuracy of the flow computation.

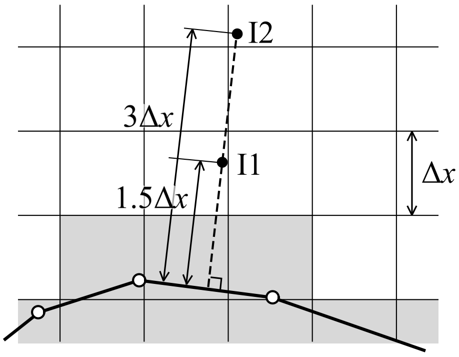


Figure 4.2: Probes for the polygon-based method.

4.2 Force Integration based on the Balance of Numerical Fluxes

Here, a new method to calculate the aerodynamic force based on the balance of the numerical flux is developed. This method is derived from the governing equation as follows.

4.2.1 Far-Field Flux-Based Method

The force integration over the far-field boundary is firstly described for comparison. The aerodynamic force is obtained by integrating the momentum equation with the body force term (4.1) at the far-field Γ_F. Integrating the momentum equation over the volumes V_1 and V_2 yields,

$$\int_{V_1+V_2} \frac{\partial(\rho u_i)}{\partial t} dt + \int_{\Gamma_F} (\rho u_i u_j + p\delta_{ij} - \tau_{ij})\, n_j dS = -F_i + \dot{M} U_{\infty,i}, \tag{4.13}$$

where $U_{\infty,i}$ are the components of the free-stream velocity, and $\dot{M}$ is the mass source in the computational domain. Although there is no mass source in the original governing equation, this term is added in (4.13) because the IBM does not assure strict mass conservation near the wall boundary. Here, the sourced mass is assumed to recover the free-stream velocity at the far-field. The mass source exists only in V_2, and thus, it can be calculated by the integration of the continuity equation over V_2 as

$$\dot{M} = \int_{\Gamma_G} \rho u_j n_j dS. \tag{4.14}$$

Therefore, the aerodynamic force F_i is

$$F_i = -\int_{V_1+V_2} \frac{\partial(\rho u_i)}{\partial t} dt - \int_{\Gamma_F} (\rho u_i u_j + p\delta_{ij} - \tau_{ij})\, n_j dS + \int_{\Gamma_G} \rho u_j U_{\infty,i} n_j dS. \tag{4.15}$$

If the flow is unsteady, the first term in the left-hand side is nonzero. Thus, the volume integration of time derivative of density is essential. When multiple objects exist in the computational domain, the force acting on one of the objects cannot be calculated. In addition, this method cannot divide the viscous and pressure components of the aerodynamic force.

4.2.2 Near-Field Flux-Based Method

To enable component-wise integration of the aerodynamic force, the integral over Γ_F is replaced by that over Γ_G. First, the momentum equation is integrated over domain V_1 assuming neither mass source nor body force exists in the domain:

$$\int_{V_1} \frac{\partial(\rho u_i)}{\partial t} dV + \int_{\Gamma_F} (\rho u_i u_j + p\delta_{ij} - \tau_{ij}) n_j dS + \int_{\Gamma_G} (\rho u_i u_j + p\delta_{ij} - \tau_{ij}) n_j dS = 0. \tag{4.16}$$

Then, subtracting (4.16) from (4.13) yields

$$\int_{V_2} \frac{\partial \rho u_i}{\partial t} dV - \int_{\Gamma_G} (\rho u_i u_j + p\delta_{ij} - \tau_{ij}) n_j dS = -F_i + \dot{M} U_{\infty,i}. \tag{4.17}$$

Here, the volume integral over V_2 is neglected because the volume V_2 is small (becomes zero on extremely fine grid), and the flow near the wall is assumed to be nearly steady. Thus, substituting (4.14) into (4.17) yields the near-field integration formula for the aerodynamic force:

$$F_i = \int_{\Gamma_G} \{(\rho(u_i - U_{\infty,i})u_j + (p - p_\infty)\delta_{ij} - \tau_{ij}\} n_j dS. \tag{4.18}$$

For the implementation, eq. (4.18) is discretized on the faces that compose Γ_G:

$$F_i = \sum_{face \in \Gamma_G} [\{\langle(\rho u_i u_j + p\delta_{ij})\hat{n}_j\rangle - \langle\rho u_j \hat{n}_j\rangle U_{\infty,i} - p_\infty \delta_{ij} - \langle\tau_{ij}\hat{n}_j\rangle\}]_{face}, \tag{4.19}$$

where $\langle(\rho u_i u_j + p\delta_{ij})\hat{n}_j\rangle$ are the momentum components of the inviscid flux, $\langle\rho u_j \hat{n}_j\rangle$ is the mass component of inviscid flux, and $\langle\tau_{ij}\hat{n}_j\rangle$ are the momentum components of the viscous flux. Here, the integral of only the viscous flux is considered as the viscous component of the aerodynamic force, and the remainder is considered as pressure component. The aerodynamic forces acting on each part of the immersed body (or each object) can also be decomposed when the faces are classified with respect to the nearest part or object.

The important point for this method is to use the same inviscid and viscous numerical fluxes as those used in the flow calculation for the flux components in (4.19). The evaluated force therefore directly reflects the accuracy of the flux used in the flow calculation, and no additional numerical error is produced.

4.3 Test Problems

To compare the accuracy of the polygon-based and flux-based methods, the following flows are simulated:

- inviscid flow around a NACA 0012 airfoil;
- steady and unsteady viscous flow around a circular cylinder;
- inviscid flow around a 3D rectangular wing.

In the following explanation, the near-field flux-based method is referred to as "flux-based method".

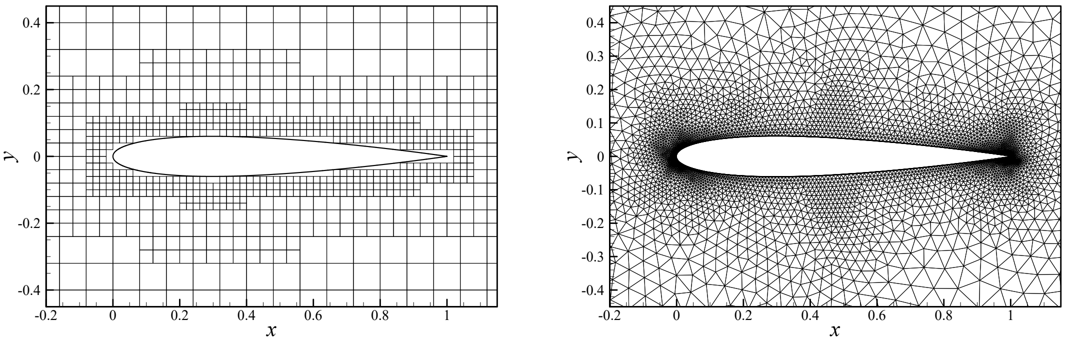
4.3.1 Inviscid Flow around a NACA 0012 Airfoil

Settings The first problem is an inviscid flow around a NACA 0012 airfoil. The free-stream Mach number is 0.2, and the angle of attack is fixed to 3°. The original NACA 0012 airfoil has a blunt trailing edge. However, the trailing edge should be sharp in inviscid flow simulations where no flow separation will occur. Thus, the definition of the airfoil is slightly modified to

$$\begin{aligned} y = \pm 0.594689181[&0.298222773\sqrt{x} - 0.127125232x \\ &-0.357907906x^2 + 0.291984971x^3 - 0.105174606x^4]. \end{aligned} \tag{4.20}$$

This modified definition is presented in [94] for the verification study of the turbulence models.

Five grids with different grid resolution (see table 4.1) are used to check the trend of grid convergence. The size of the computational domain is 655×655, and the far-field boundary condition is imposed on the far-field boundaries. For the wall boundary, the slip wall condition is used. Figure 4.3 (a) shows the near field of grid 1. The input definition of the NACA 0012 airfoil consists of 10,093 line segments, and the size of each line segment is small enough compared to the minimum cell size of grid 5. In addition, computation by Japan Aerospace Exploration Agency (JAXA)'s unstructured flow solver "FaSTAR" [95] on a body-fitted grid (16,697 cells) is conducted to obtain the reference solution.



(a) Grid for UTCart (grid 1)

(b) Grid for FaSTAR

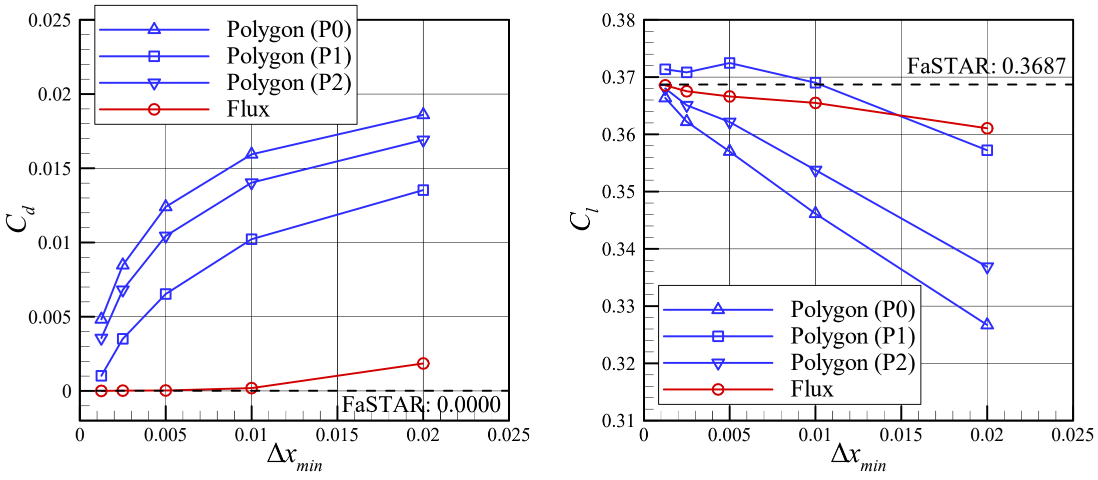
Figure 4.3: Computational grid around the NACA 0012 airfoil. Only the near-field is shown.

Table 4.1: Setting of computational grids around the NACA 0012 airfoil.

	Minimum grid size	Total cell number
Grid 1	0.02	2,510
Grid 2	0.01	4,710
Grid 3	0.005	8,598
Grid 4	0.0025	15,372
Grid 5	0.00125	33,096

Results Figure 4.4 shows grid convergences of the drag and lift coefficients. The ideal value for the drag coefficient is zero because this case is a 2D inviscid flow simulation without shock waves. The flux-based method on grid 3 calculates the drag coefficient with 1 drag count ($= 10^{-4}$) error, whereas the polygon-based methods calculate it with more than 50 drag counts error. Even though the result of the flow computation is the same, the drag coefficient varies by more than 40 drag counts depending on the extrapolation methods for the surface pressure when the polygon-based methods are used. This implies that there is an additional numerical error unrelated to the accuracy of the flow computation.

Figure 4.5 shows the surface pressure coefficient $C_p = p/(\rho u_\infty^2/2)$ on the airfoil for the grid 3. This surface pressure is used in the polygon-based methods to calculate the pressure component of the force. The areas confined by the line in $x/c - C_p$ plot and in $C_p - y/c$ plot represent the magnitudes of the vertical and horizontal forces, respectively. Depending on the extrapolation formula of the surface pressure, the distribution near the leading edge varies. For example, the P1 extrapolation results in a strong suction peak, and thus, the vertical force is overestimated. In addition, the surface pressure near the leading edge in all the results is higher than the FaSTAR result, and this seems to be the cause of the overestimation of the drag coefficient computed by the polygon-based methods.



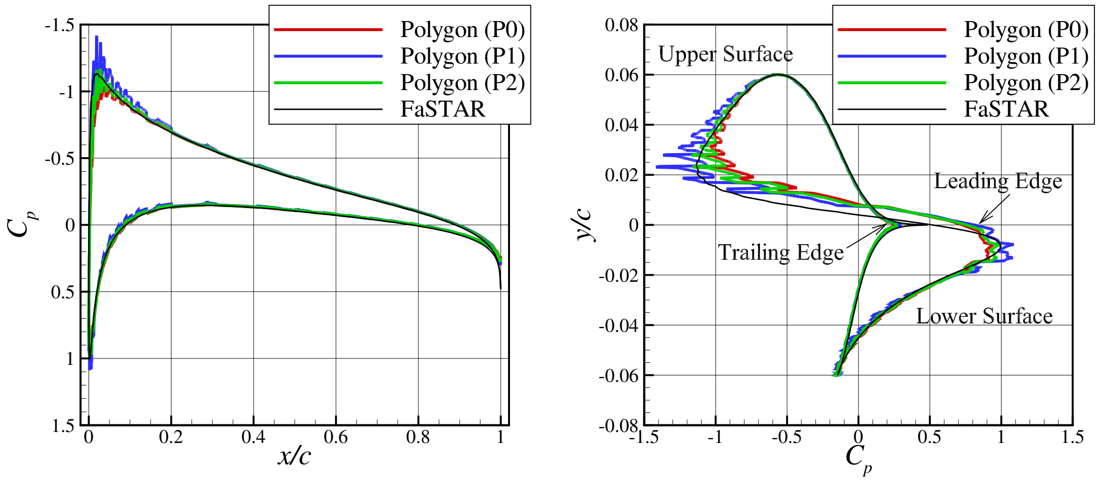
(a) Drag coefficient

(b) Lift coefficient

Figure 4.4: Grid convergence of aerodynamic coefficients of the NACA 0012 airfoil.

4.3.2 Inviscid Flow around a 3D Rectangular Wing

Settings To investigate the accuracy of the force integration methods in 3D flows, an inviscid flow around a rectangular wing is simulated. The section of the wing is defined by the modified NACA 0012 airfoil (4.20), and the aspect ratio $AR = 4$. The symmetry boundary condition is imposed on the $y = 0$ plane, and a half-cut model is employed. The



(a) $x/c - C_p$ (b) $C_p - y/c$

Figure 4.5: Surface pressure on the NACA 0012 airfoil (Grid 3).

surface mesh of the wing is shown in Fig. 4.7. The free-stream Mach number is 0.2, and the angles of attack are 0, 3, 6, 9, and 12°. Figure 4.6 shows the overview of the computational grid. For the wall boundary, the slip wall condition is used. The size of the computational domain is 640^3, and the Riemann boundary condition is imposed on the far-field boundaries except for the $y = 0$ plane. Two grids with different resolutions are prepared to check the trend of grid convergence.

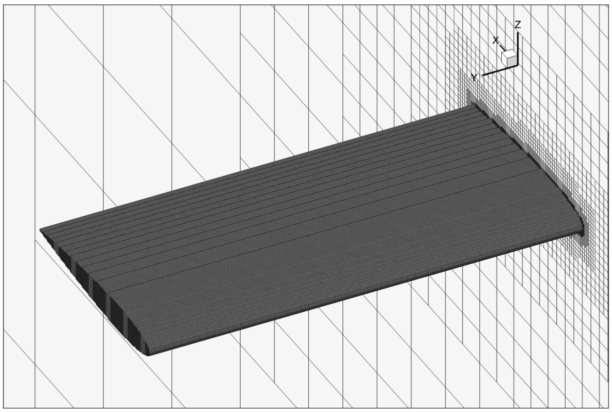


Figure 4.6: Computational grid around the rectangular wing.

Results Figure 4.8 shows the calculated drag coefficient versus the lift coefficient. Note that the “Theory” curve in the figure is calculated by the theoretical relationship used in

Table 4.2: Setting of computational grid around rectangular wing.

	Minimum grid size	Total cell number
Grid 1	0.0048	1,004,218
Grid 2	0.0024	3,770,743

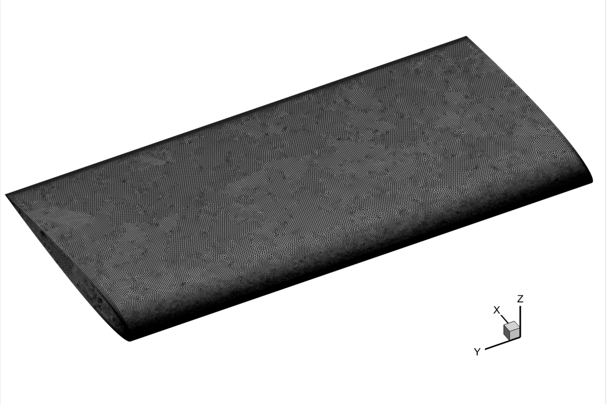


Figure 4.7: Surface polygon mesh of the rectangular wing.

the conceptual design of aircraft:

$$C_D = \frac{C_L^2}{e\pi AR}, \tag{4.21}$$

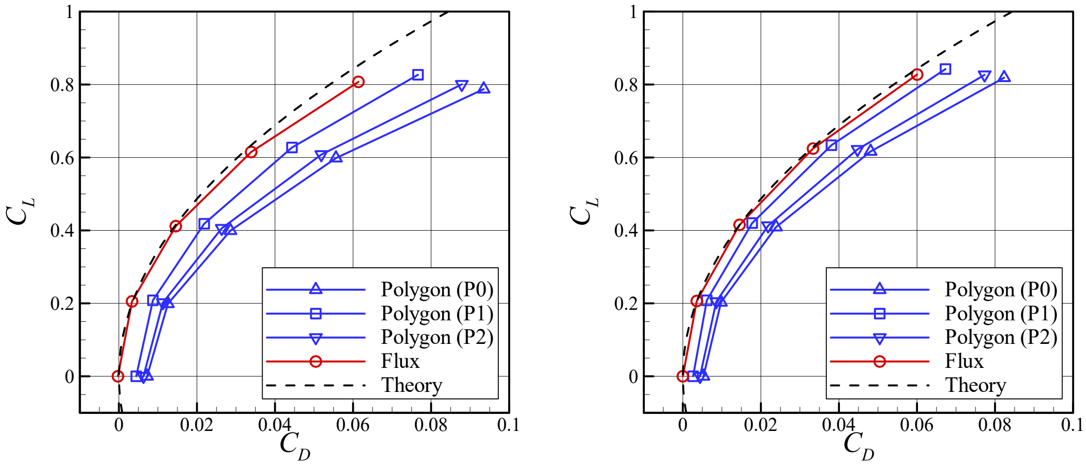
where the Oswald span efficiency factor is calculated using a statistical relationship [96] as

$$\begin{aligned} e &= 1.78(1 - 0.045AR^{0.68}) - 0.64 \\ &= 0.934. \end{aligned} \tag{4.22}$$

On both computational grids, the curve calculated by the flux-based method has less than 1 drag count error at $C_L = 0$. The curves calculated by the polygon-based methods, on the other hand, have more than 40 drag counts error compared with the ideal curve, and the magnitude of the error increases at the higher angles of attack. Furthermore, the curve by the flux-based method shows closer agreement with the ideal curve on grid 2. This indicates that the trend of the grid convergence is correct.

4.3.3 Steady Viscous Flow past a Circular Cylinder

Settings To investigate the accuracy of the force calculation methods in a viscous flow, the flow past a circular cylinder with the diameter $D = 1$ is simulated. Here, the surface of the circular cylinder is a non-slip wall. The Reynolds number based on the diameter and the free-stream velocity is 20. In addition, the free-stream Mach number and the free-stream



(a) Grid 1

(b) Grid 2

Figure 4.8: Drag-lift polar curves of the rectangular wing.

temperature is 0.2 and 293.15 K, respectively. Four resolutions of grids (Table 4.3) are prepared to investigate the trend of grid convergence. The size of the computational domain is 262×262 and the Riemann boundary condition is imposed on the far-field boundaries. Figure 4.9 (a) shows the near field of grid 1. As shown in the figure, a box source for grid refinement is placed in the wake. The grid size in the box source is four times the grid size on the wall. The input definition of the cylinder consists of 3,600 line segments, and the length of each line segment is small enough compared to the minimum cell size of grid 4.

Table 4.3: Setting of computational grid around a circular cylinder.

	Minimum grid size	Total cell number
Grid 1	0.016	4,830
Grid 2	0.008	11,418
Grid 3	0.004	34,004
Grid 4	0.002	113,744

Results Figure 4.10 shows the calculated drag coefficients on each grid. Note that the three polygon-based methods have no difference in the calculation of the viscous force, and thus, only polygon-based method (P1) is shown in Fig. 4.10 (b). The converged value of the total drag coefficient is expected to be around 2.05, and it is identical in all the force calculation methods presented here. The reported values for the total drag coefficient in

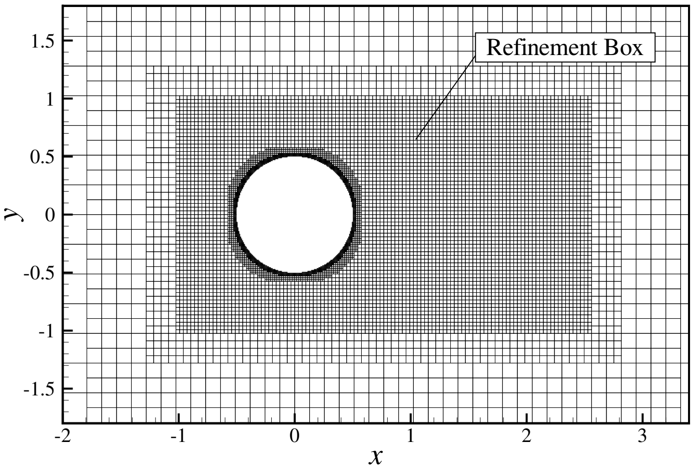
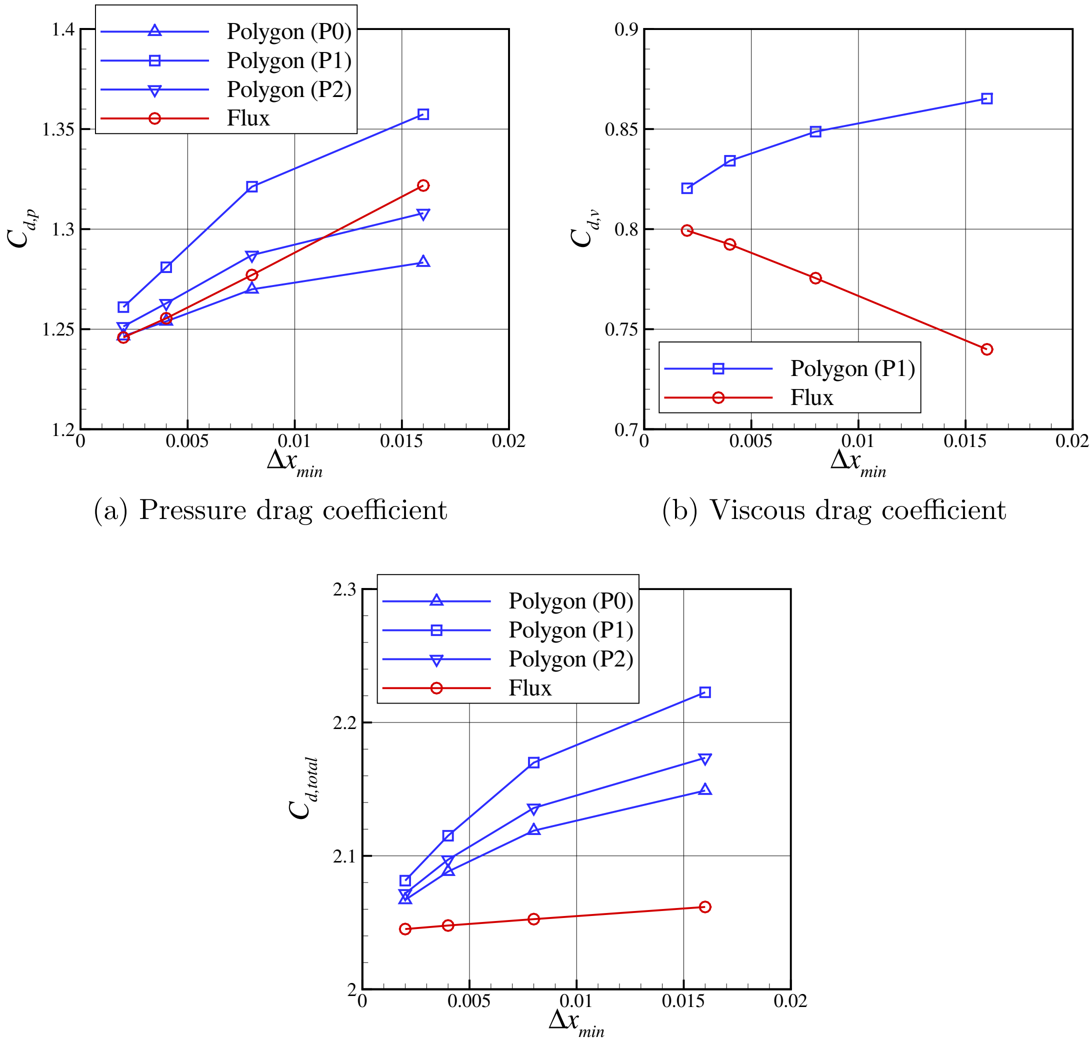


Figure 4.9: Computational grid around a circular cylinder.

the previous studies [11, 97, 98] are within the range from 2.043 to 2.053, and those are consistent with the present values. The pressure drag coefficient on the coarse grid tends to be overestimated regardless of the force calculation method. Furthermore, being similar to the flow around the NACA 0012 airfoil, the pressure drag varies by the choice of the extrapolation formula. The viscous drag is overestimated by the polygon-based methods and underestimated by the flux-based method. A typical laminar boundary layer profile is represented by the Blasius solution [69], in which the velocity gradient becomes small where the distance from the wall increases. When the IP is located farther from the wall, the velocity gradient u_{IP}/d_{IP} is expected to become smaller. Thus, it is natural that the viscous drag is underestimated on the coarse grid. However, the polygon-based method overestimates the viscous drag, and only the flux-based method follows the trend stated above.

In addition, the drag related to the mass source is shown in Fig. 4.11. The magnitude is smaller than that in the inviscid calculations. This is because the flow near the wall is slow due to the presence of the boundary layer. Consequently, the mass flux through the cell boundary becomes small, and the drag related to the mass source is small in this problem.



(a) Pressure drag coefficient

(b) Viscous drag coefficient

(c) Total drag coefficient

Figure 4.10: Grid convergence of aerodynamic coefficients of a circular cylinder.

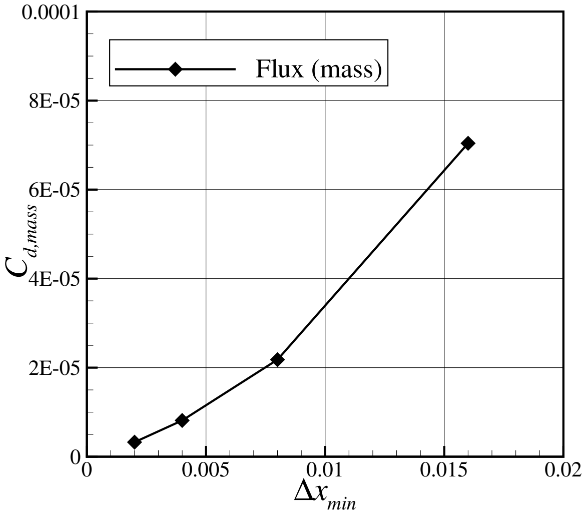


Figure 4.11: Mass conservation in the flow around a circular cylinder.

4.3.4 Unsteady Viscous Flow past a Circular Cylinder

Settings The Reynolds number for the diameter of the cylinder is here enhanced to 200. This problem is conducted to confirm the capability of the near-field flux-based method for tracking time history of the aerodynamic force in unsteady problems. The computational grid is identical with grid 4 in the previous problem. Here, the polygon-based method (P1), the far-field flux-based method and the near-field flux-based method are compared. Note that the volume integration of the unsteady term in the far-field flux-based method is omitted.

Results Figure 4.12 shows the convergence history of the drag coefficient. The value calculated by the far-field flux-based method does not follow the other two convergence histories. This is because the unsteady term in (4.15) is omitted. In addition, the aerodynamic force caused by the cylinder delays and attenuates before reaching the far-field. Therefore, the far-field flux-based method cannot be applied to unsteady problems without including the volume integration over the entire domain. In contrast, the convergence histories by the polygon-based method and the near-field flux-based method show close agreement. Although the unsteady term in (4.17) is omitted in the integration, there is almost no delay between the polygon-based method and the near-field flux-based method. Thus, it is confirmed that the near-field flux-based method can be applied also to unsteady problems.

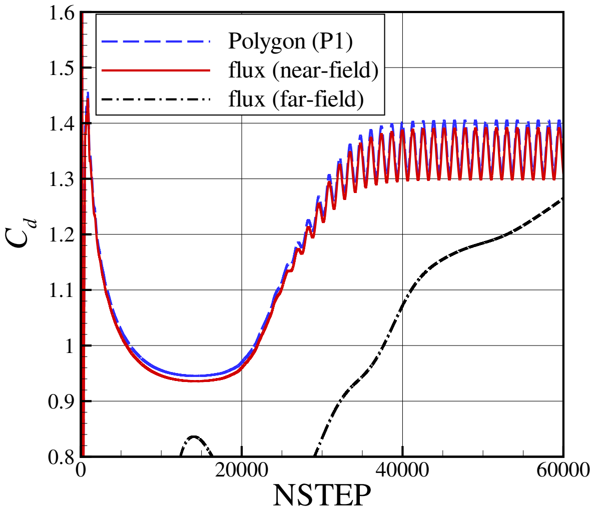


Figure 4.12: History of the total drag coefficient in an unsteady flow simulation around a cylinder at Re = 200.

4.4 Summary of Chapter 4

In this chapter, the methods to calculate aerodynamic forces acting on the immersed bodies were investigated. The calculation is not as straightforward as that on the body-fitted grid because no physical quantities exist on the immersed body. For the force calculation methods, the polygon-based method and the flux-based method were compared.

In the polygon-based method, the quantities on the immersed body are extrapolated from the computational grids. For the interpolation, probes are set in the fluid cells. The integration of the force is the same as the near-field method on the body-fitted grid, but additional numerical error and arbitrariness are introduced by the interpolation.

On the other hand, the flux-based method calculates the aerodynamic force by integrating the numerical flux through the step-wise cell boundary. This method is similar to the far-field integration, but it can be applied also to unsteady problems without introducing a volume-integration term. The calculated aerodynamic force can be decomposed into the pressure and viscous components. In addition, the force acting on one of the objects can be calculated when multiple objects exist in the computational domain. The accuracy of this method is consistent with the accuracy of the flow computation because the same numerical flux as that employed in the flow computation is used.

Then, inviscid and viscous test problems were solved to investigate the accuracy of the force calculation methods. The pressure component of the force varies by the extrapolation formula of the surface pressure when the polygon-based method is used. In contrast, the flux-based method could calculate the pressure drag more accurately than the polygon-based methods. The oscillating flow past a circular cylinder was also simulated. The time history of the aerodynamic coefficient could be reproduced by the flux-based method.

Chapter 5

Test Cases for the Immersed Boundary Method

To investigate the accuracy and validity of the proposed simulation methodology for turbulent flows, studies on simple 2D flows defined at the NASA turbulence modeling resource (TMR) [94] are conducted. The test cases include the simulations of the following flows:

- turbulent flat-plate boundary layers;
- subsonic flows over a bump;
- subsonic flows around the NACA 0012 airfoil.

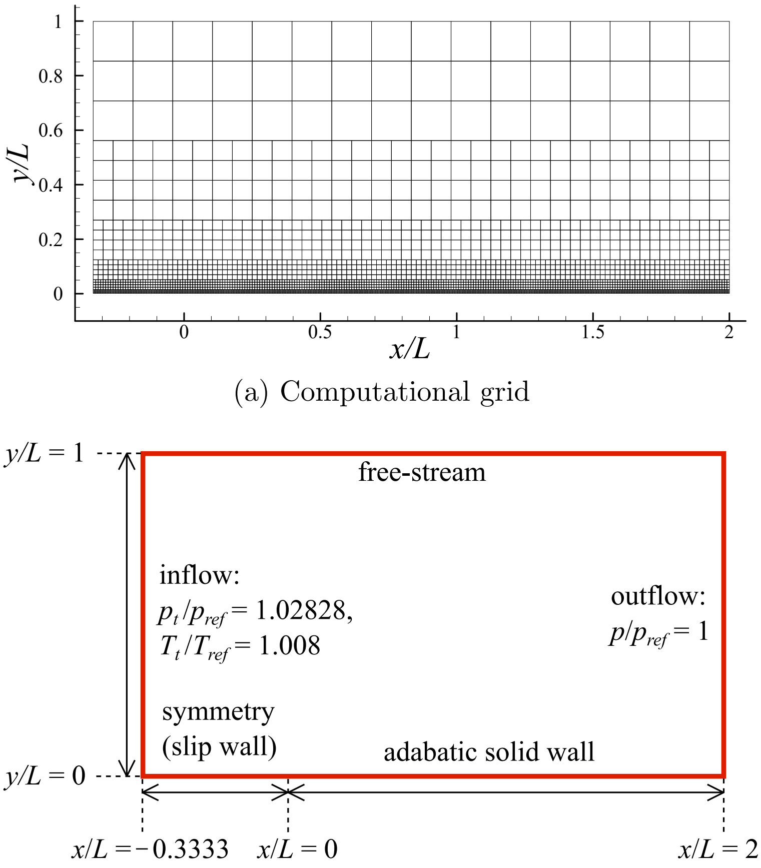
In this chapter, the original and modified IBMs (see Section 3.4) are implemented in UTCart, and these two methods are compared. For every problem, multiple grids with different grid resolution are prepared to check the trends of grid convergence. The test cases in this section are conducted to confirm that UTCart can produce similar results to those obtained on the conventional body-fitted grids. Thus, the computational results are compared with the reference results provided in the TMR. These reference results are computed by a structured-grid-based flow solver CFL3D [99] on body-fitted structured grids using the same turbulence model. In the TMR, the grid convergence of the results has been strictly verified, and therefore the results are here considered as the "correct" solutions of the governing equations.

5.1 Turbulent Flat-Plate Boundary Layer on Aligned Grids

As the simplest validation study for the proposed IBM, turbulent flows over a flat plate are simulated at first. The definition of the problem is following the benchmark problem presented in the TMR. The Reynolds number based on the unit length $L = 1$, the free-stream Mach number and the free-stream temperature are 5.0×10^6, 0.2 and 300K, respectively. The reference solution for this problem is provided in the TMR, which is computed by CFL3D on the 545×385 structured grid.

The computational grid and the boundary conditions for this case are shown in Fig. 5.1.

Note that the computational grid is coarsened for visualization. This setting is a special case for the IBM because the boundary of the cell coincides with the plate surface. Thus, this grid can be classified as a body-fitted grid. Here, the original and modified IBMs (see Section 3.4) are compared on five different grids, as tabulated in Table 5.1. The y^+_{IP} values in the table are estimated using u_τ at $x/L = 0.97$ of the reference result. For all the cases, the ratio of the IP distance to the cell size on the wall, r_{IP} (see (2.40)), is fixed to 3.



(a) Computational grid

(b) Boundary conditions

Figure 5.1: The computational grid over the flat plate (aligned grid).

5.1.1 Computational Results

Figure 5.2 compares the distributions of the skin friction coefficient on the flat plate calculated by the original and modified IBMs. The skin friction coefficient c_f is defined as

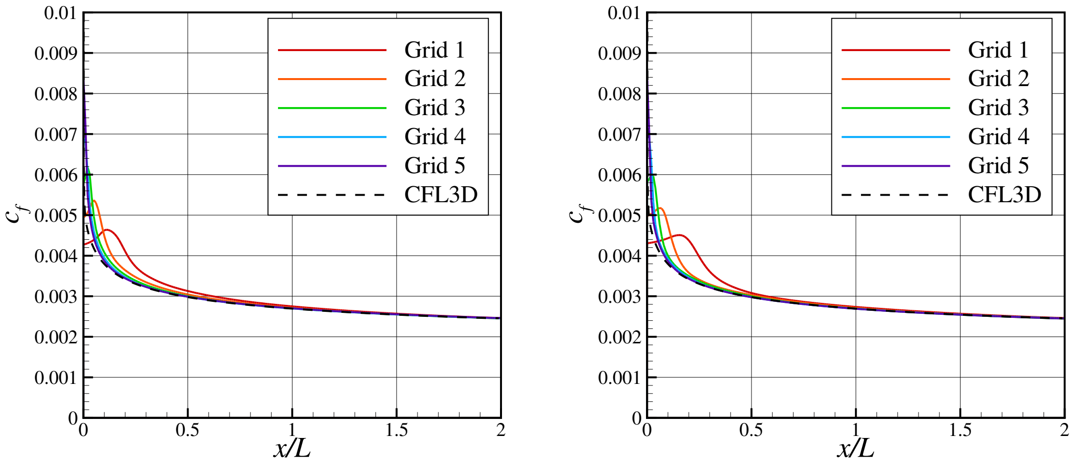
$$c_f = \tau_w / q_\infty, \tag{5.1}$$

Table 5.1: Setting of computational grid over the flat plate (aligned grid).

Grid	Minimum grid size	Cell number	y^+_{IP} at $x/L = 0.97$ (estimated)
1	1.14×10^{-3}	18,224	628
2	5.70×10^{-4}	41,264	314
3	2.85×10^{-4}	89,648	157
4	1.42×10^{-4}	191,624	78.6
5	7.12×10^{-5}	398,384	39.3

where $q_\infty = (\rho_\infty u_\infty^2)/2$ is the dynamic pressure of the free-stream. The results by both IBMs show good agreement with the reference result by CFL3D, except for the region very close the leading edge. This discrepancy occurs because the boundary layer in this region is very thin, and thus, the IP is not located in the inner layer of the boundary layer correctly.

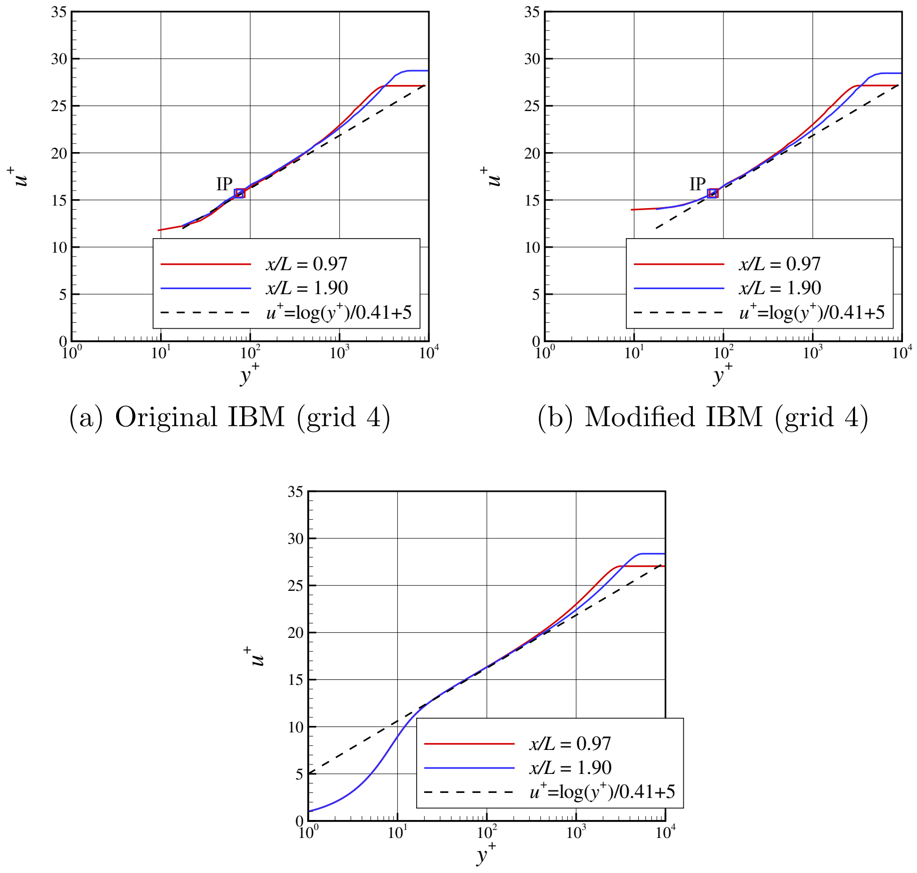
Figure 5.3 visualizes the profiles of the wall tangential velocity at $x/L = 0.97$ and 1.90. In addition, Fig. 5.4 shows the profiles of the non-dimensional eddy-viscosity $\nu^+ = \nu_t/\nu$ at the same location. The profiles above the IP shows good agreement with the CFL3D result, and the difference between the two IBMs are small. In the modified IBM result, the tangential velocity below the IP becomes larger than the actual profile in the viscous sublayer. Furthermore, the eddy-viscosity profile between the IP and wall is nearly constant. These results confirm that the profile modifications work as intended.



(a) Original IBM

(b) Modified IBM

Figure 5.2: Skin friction coefficient on the flat plate (aligned grid).



(a) Original IBM (grid 4)

(b) Modified IBM (grid 4)

(c) CFL3D

Figure 5.3: Profiles of the tangential velocity in the boundary layer (aligned grid).

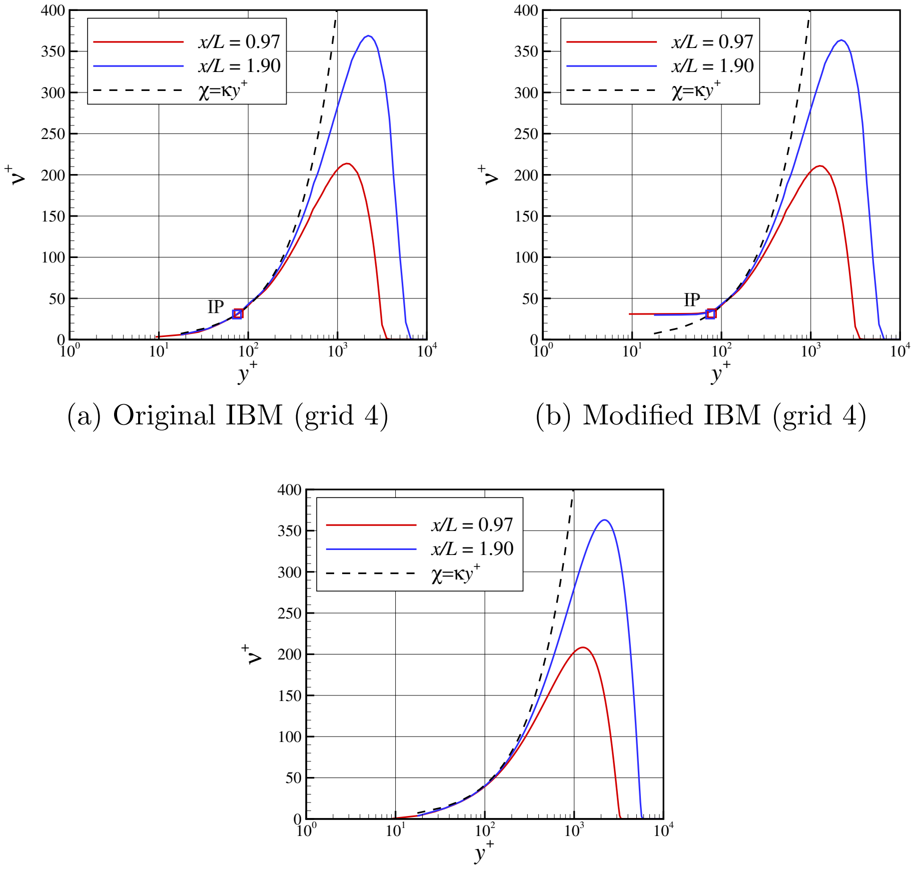
5.1.2 Discussion

Both original and modified IBMs can reproduce the growth of the turbulent boundary layer accurately on this grid. This is assumed to be because the cell boundary is attached to the plate. Here, the mechanisms for the wall function to reproduce the turbulent boundary layer on this grid is analyzed in detail.

A control volume from the wall to the upper edge of the boundary layer is considered (Fig. 5.5). The x- and y-axes are set in the wall tangential and normal directions, respectively. The pressure gradient in this control volume and the shear stress at the upper edge are zero. Thus, the balance of the x-direction momentum flux is

$$\tau_w l_x = \int_0^\delta \left\{ (\rho u^2)_{in} - (\rho u^2)_{out} \right\} \mathrm{d}y, \tag{5.2}$$

where τ_w is the shear stress determined by the wall function, l_x is the width of the control



(a) Original IBM (grid 4)

(b) Modified IBM (grid 4)

(c) CFL3D

Figure 5.4: Profiles of the eddy-viscosity in the boundary layer (aligned grid).

volume, and δ is the boundary layer thickness. The right-hand side of (5.2) represents the development of the boundary layer, and this term is equated with the wall shear stress. In addition, the advection flux through the lower edge of the control volume is zero when a body-fitted grid is used. Therefore, the balance of the momentum flux (5.2) is automatically satisfied, and thus, the development of the boundary layer is correctly reproduced when the wall shear stress is correct.

The cell-wise balance of the momentum flux is also examined. In the first cell from the wall, the advection term is small compared to the viscous term because the cell is assumed to be in the inner layer of the boundary layer. Thus, only the viscous fluxes through the upper and lower edges remain. The time evolution of the wall tangential velocity u becomes

$$\frac{u^{new} - u^{old}}{\Delta t} = \frac{1}{\rho} \frac{\tau_N - \tau_w}{\Delta y}, \tag{5.3}$$

where τ_N denotes the shear stress on the upper face. If τ_w is greater than τ_N, u decreases and the velocity gradient at the upper edge increases. Consequently, τ_N increases, and it balances with τ_w at the converged state. Under this converged state, the near-wall velocity profile is not resolved correctly. However, the balance of the x-direction momentum is automatically maintained, and the shear stress imposed on the wall is correctly conveyed to the outer layer of the boundary layer. As a result, even the original IBM can reproduce the boundary layer accurately on the body-fitted grid.

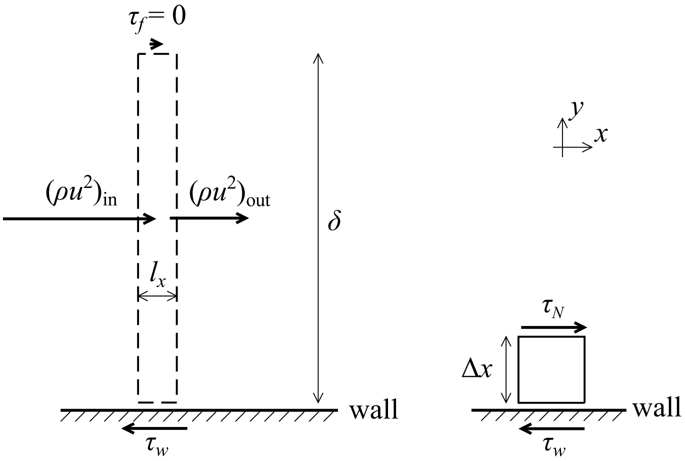


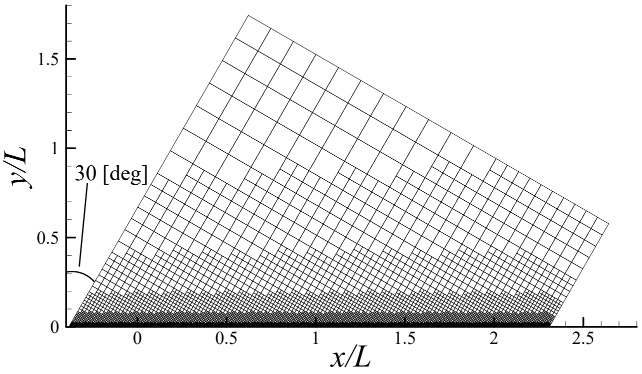
Figure 5.5: Balance of the x-direction momentum in the boundary layer (left: the entire boundary layer, right: the first cell to the wall).

5.2 Turbulent Flat-Plate Boundary Layer on Inclined Grids

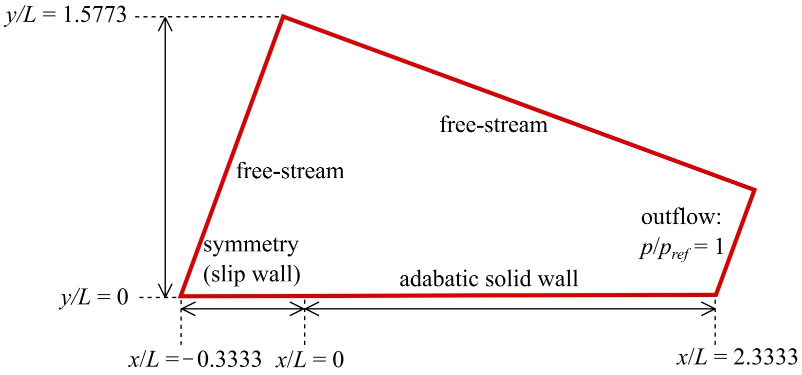
The same problem as that in the previous section is solved on inclined grids visualized in Fig. 5.6. This case is more general for the IBM than the previous problem because the plate surface does not coincide with the cell boundary. For this problem, nine different grids described in Table 5.2 are prepared. Here, the following two topics are examined:

- dependency on the r_{IP} value;
- effect of the profile modifications presented in Section 3.2.

The former topic is examined on grid 1–9 to check the grid convergence toward a very fine grid where the IP locates in the viscous sublayer. For the latter topic, the computations are conducted on grid 1–5, where the IP is expected to be within the log-layer.



(a) Computational grid



(b) Boundary conditions

Figure 5.6: Computational grid over the flat plate (inclined grid).

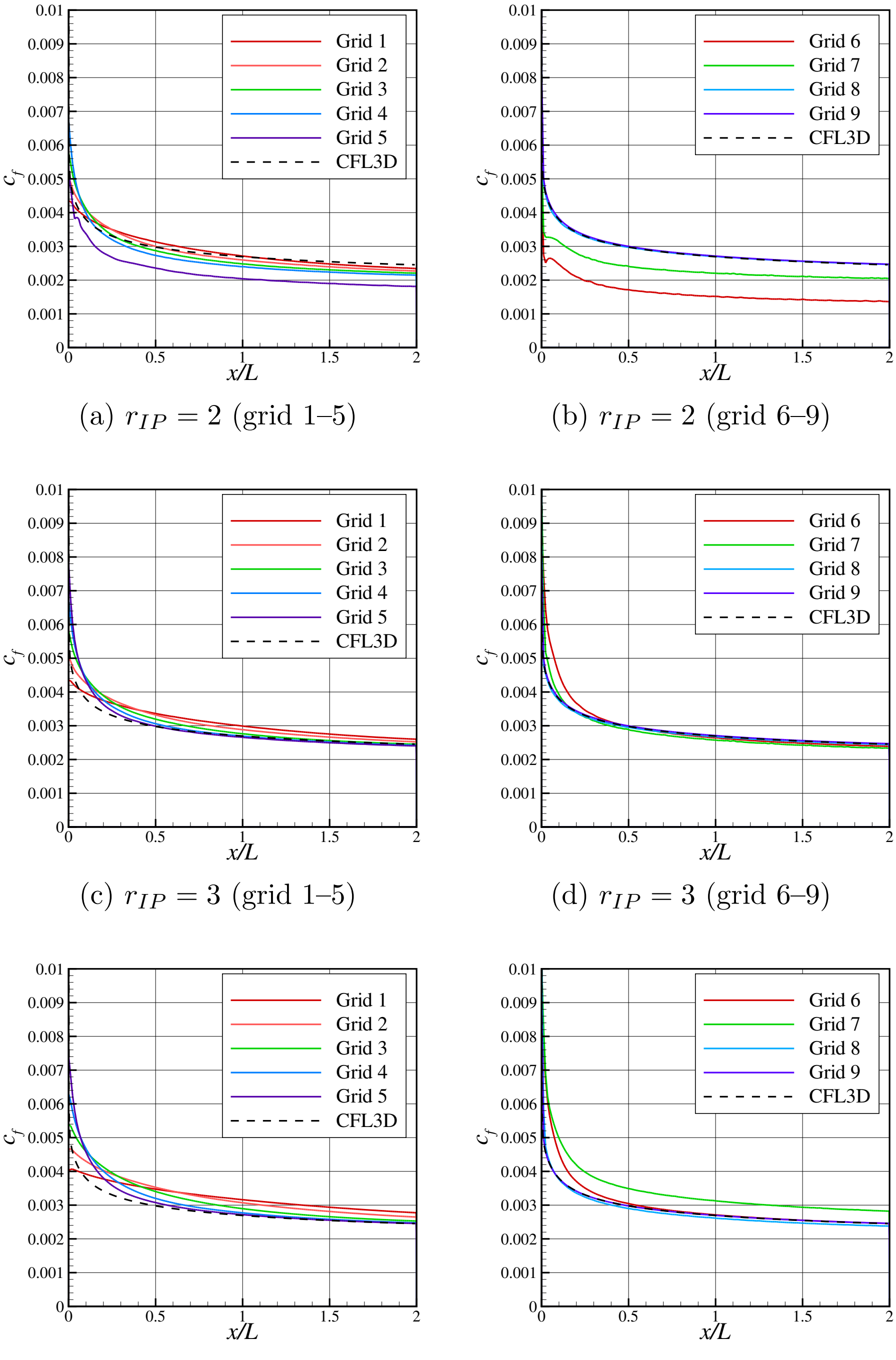
Table 5.2: Setting of computational grid over the flat plate (inclined grid).

Grid	Minimum cell size	Cell number	y^+_{IP} at $x/L = 0.97$ (estimated)		
			$r_{IP} = 2$	$r_{IP} = 3$	$r_{IP} = 4.5$
1	1.14×10^{-3}	28,878	419	628	943
2	5.70×10^{-4}	74,047	209	314	471
3	2.85×10^{-4}	164,449	105	157	236
4	1.42×10^{-4}	345,279	52.4	78.6	118
5	7.12×10^{-5}	707,083	26.2	39.3	58.9
6	3.56×10^{-5}	1,430,643	13.1	19.6	29.5
7	1.78×10^{-5}	2,877,773	6.55	9.82	14.7
8	8.90×10^{-6}	5,772,169	3.27	4.91	7.37
9	4.45×10^{-6}	11,561,019	1.64	2.46	3.68

5.2.1 Dependency on the IP Setting

Figure 5.7 shows the distributions of the skin friction coefficient c_f on each grid. The reference result by CFL3D is also represented in the figure. The skin friction tends to be overestimated on the coarse grid. When r_{IP} is 3 or 4.5, the distribution is generally well predicted, although there is a few exception. With $r_{IP} = 2$, the scatter of c_f for different grids is larger than the other cases.

Figure 5.8 shows the grid convergence of the skin friction coefficients at $x/L = 0.97$ and 1.90. For c_f at $x/L = 0.97$, the grid 3–9 cases with $r_{IP} = 3$ predict the value within the 5% range from the CFL3D result. In addition, with $r_{IP} = 4.5$, the grid 4–9 cases except for grid 7 case also predict the value within this range. The accuracy is better for c_f at $x/L = 1.90$; the calculated values on grid 2 with $r_{IP} = 3$ and grid 3 with $r_{IP} = 4.5$ exist within the 5% range from the CFL3D result. On grid 7 with $r_{IP} = 3$, the skin friction is slightly underestimated. Furthermore, c_f on grid 7 with $r_{IP} = 4.5$ is approximately 15% larger than the CFL3D result. These phenomena seem to occur when the IP is located in the buffer layer of the boundary layer ($5 < y^+_{IP} < 30$) where the nonlinearity of the velocity profile is strong. In contrast, c_f is accurately calculated being independent of the r_{IP} value when the IP is located in the viscous sublayer of the boundary layer.



(a) $r_{IP} = 2$ (grid 1–5)

(b) $r_{IP} = 2$ (grid 6–9)

(c) $r_{IP} = 3$ (grid 1–5)

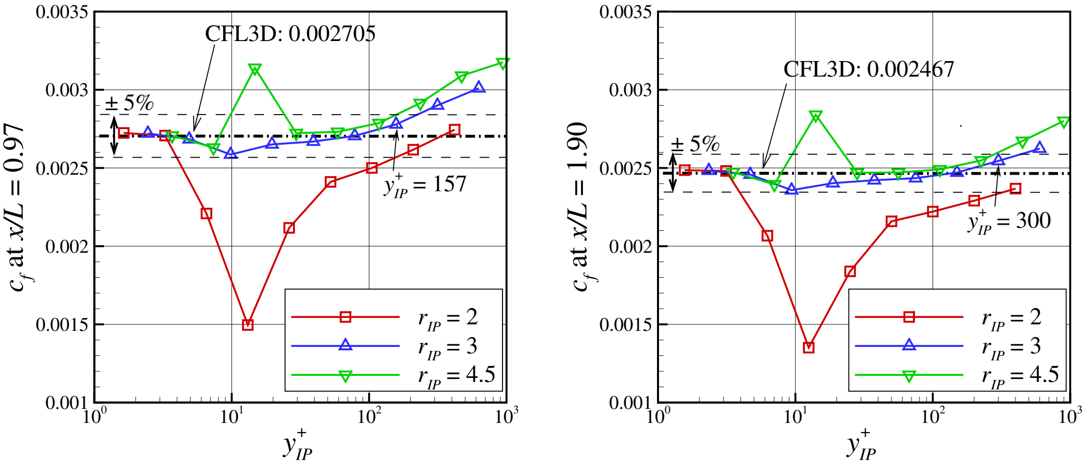
(d) $r_{IP} = 3$ (grid 6–9)

(e) $r_{IP} = 4.5$ (grid 1–5)

(f) $r_{IP} = 4.5$ (grid 6–9)

Figure 5.7: Distribution of the skin friction coefficient on the plate.

Furthermore, Figs. 5.9–5.11 show the profiles of the tangential velocity, eddy-viscosity and shear stress τ_{xy} in each grid 4 case. Note that u_τ used in the non-dimensionalization



(a) $x/L = 0.97$ (b) $x/L = 1.90$

Figure 5.8: Grid convergence of the skin friction coefficient.

is calculated by the computed value of c_f in each result. In addition, the shear stress is calculated at each cell using the cell-centered values of the kinematic viscosity, eddy-viscosity, and velocity gradient. With $r_{IP} = 2$, the velocity deviates from the log-law, and the eddy-viscosity becomes larger than that in the other cases. The ideal solution of the shear stress $\tau_{xy}/(\rho u_\tau^2)$ near the wall is unity, as indicated in (3.3). However, the shear stress above the IP becomes larger than unity in the $r_{IP} = 2$ cases. This phenomenon is also observed in the $r_{IP} = 3$ cases, although the magnitude of the deviation is smaller.

One possible cause for this difference is the numerical inaccuracy in the reconstruction in the cell around the IP. Here, assume that a cell center (point P) is located in the log-layer, and the cells have a uniform size Δx. The y-axis is set to the wall normal direction, and $y = 0$ on the wall. The velocity distribution within the cell is represented by the Taylor expansion around point P as

$$u(y) = \left.\frac{\mathrm{d}u}{\mathrm{d}y}\right|_P (y - y_P) + \left.\frac{\mathrm{d}^2u}{\mathrm{d}y^2}\right|_P (y - y_P)^2 + O(\Delta x^3). \tag{5.4}$$

The first derivative of the velocity (see Section 3.1) is derived by differentiating the log-law profile:

$$\left.\frac{\mathrm{d}u}{\mathrm{d}y}\right|_P = \frac{u_\tau}{\kappa y_P}. \tag{5.5}$$

The second derivative is

$$\left.\frac{\mathrm{d}^2 u}{\mathrm{d}y^2}\right|_P = \frac{-u_\tau}{\kappa y_P^2}. \tag{5.6}$$

Substituting (5.5) and (5.6) into (5.4) yields

$$u(y) = \frac{u_\tau}{\kappa y_P}(y - y_P) - \frac{u_\tau}{\kappa y_P^2}(y - y_P)^2 + O(\Delta x^3), \tag{5.7}$$

The second and higher-order terms in the right-hand side of (5.7) become the truncation error when a linear reconstruction scheme (i.e., a second-order scheme) is used. Here, the approximate magnitude of the first order term is

$$\frac{u_\tau}{\kappa y_P}(y - y_P) \sim \left(\frac{\Delta x}{y_P}\right), \tag{5.8}$$

because the reconstruction within a cell is considered (i.e., $y - y_P \sim \Delta x$). Similarly, the magnitude of the second order term is approximated as

$$-\frac{u_\tau}{\kappa y_P^2}(y - y_P)^2 \sim \left(\frac{\Delta x}{y_P}\right)^2. \tag{5.9}$$

Thus, the effect of the truncation error is depending on $(\Delta x / y_P)$.

When the modified IBM is used, the profiles below the IP is modified so that it is linear. Thus, the cell just above the IP is considered, where $(\Delta x / y_P)$ is approximately equal to $(1/r_{IP})$. Therefore, the truncation error becomes more critical when the r_{IP} value is small. This is assumed to be the reason the result with $r_{IP} = 2$ is less accurate than the other results. This error does not converge even when Δx is set to a smaller value because y_P is proportional to Δx. The high-order terms in (5.4) have a similar relationship, and thus, this problem cannot be solved by a higher-order scheme. Furthermore, the truncation error becomes large in the buffer layer of the boundary layer because the second derivative is large.

The examination above can be reconsidered following the discussion on the cause of the log-layer mismatch in the wall-modeled LES [64]. In the log-layer, velocity gradient is proportional to y^{-1}. Thus, the averaged length scale is proportional to y:

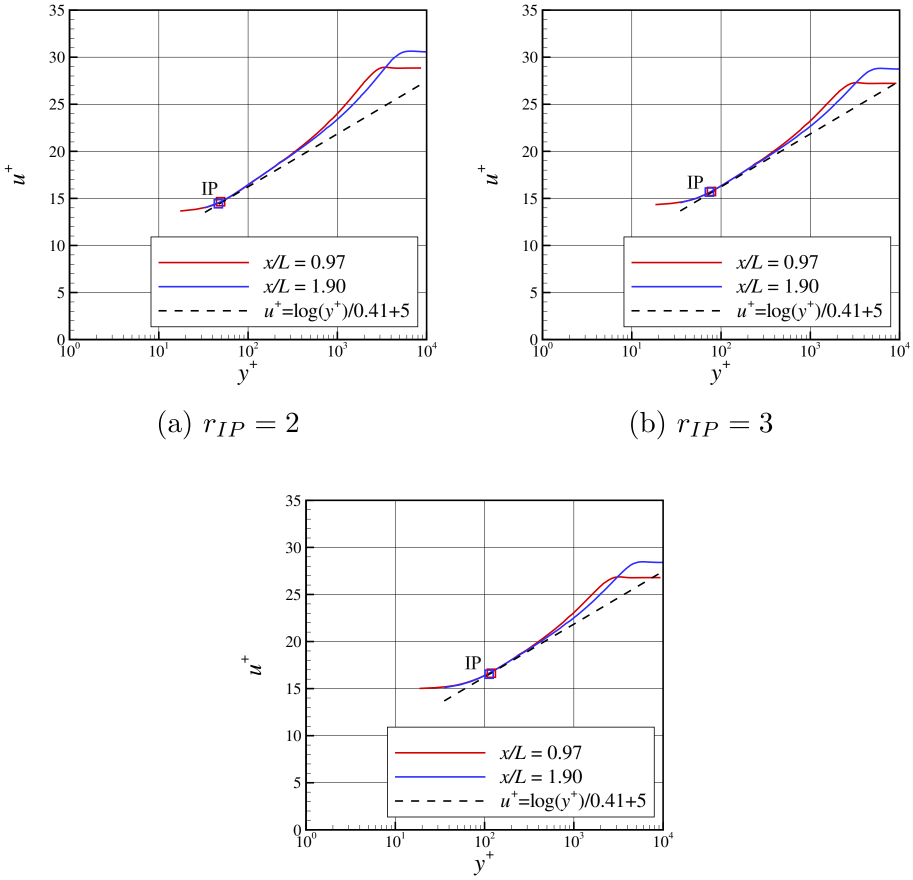
$$l = \frac{|\mathrm{d}u/\mathrm{d}y|}{|\mathrm{d}^2 u/\mathrm{d}y^2|} \sim y, \tag{5.10}$$

because the scale of the vortex is limited by the existence of the wall. This length scale must be resolved by the grid, and thus, $l/\Delta x$ is required to be larger than a certain value. Namely, the requirement for the grid is

$$\frac{y}{\Delta x} > C, \tag{5.11}$$

where C is the minimum value of the wavelength that can be resolved by the grid.

The length scale takes a minimum value at the cell just above the cutoff region. When $r_{IP} = 2$, C must be smaller than two. It is difficult to satisfy this requirement because of the limitation of the Nyquist frequency. With $r_{IP} = 3$, the boundary layer is reproduced with a certain degree of accuracy although $C \sim 3$ seems to be a severe requirement. In a RANS simulation, only the averaged profile is computed. This is different from an LES where each vortex must be resolved, and thus, the requirement for the grid size may be milder. Note that the relationship (5.10) is not applicable in the viscous sublayer where the length scale is determined by the kinematic viscosity. This is the reason grid 9 can reproduce the boundary layer accurately being independent of the r_{IP} value.

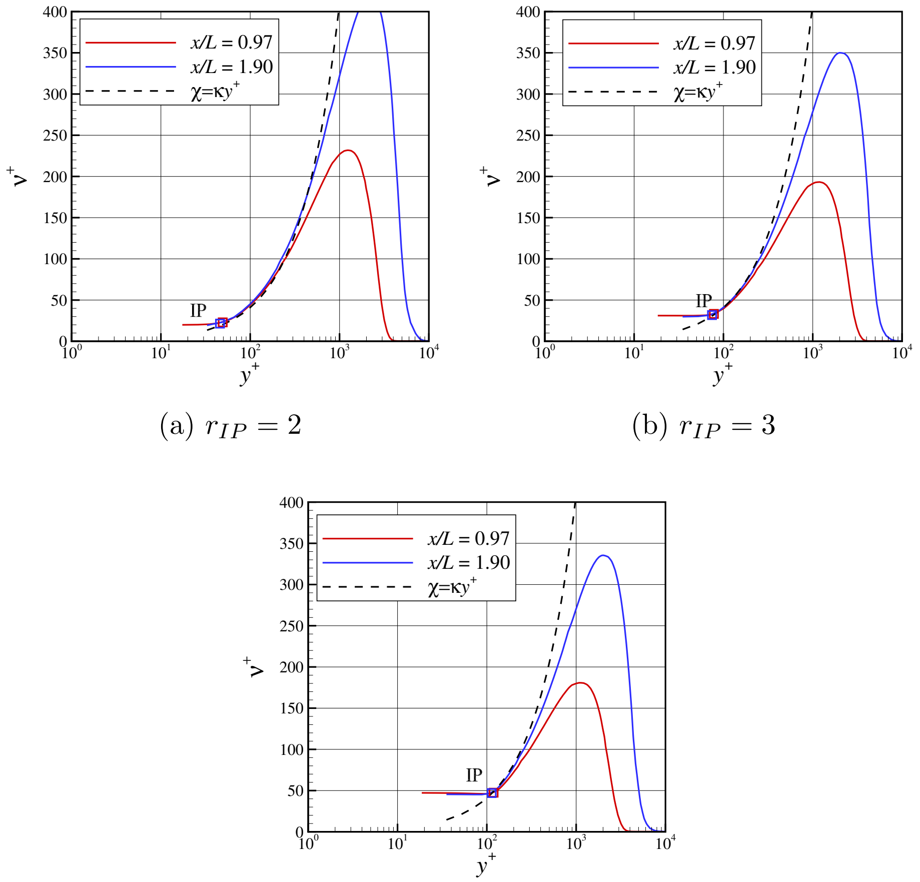


(a) $r_{IP} = 2$

(b) $r_{IP} = 3$

(c) $r_{IP} = 4.5$

Figure 5.9: Profiles of the tangential velocity in the boundary layer (grid 4).

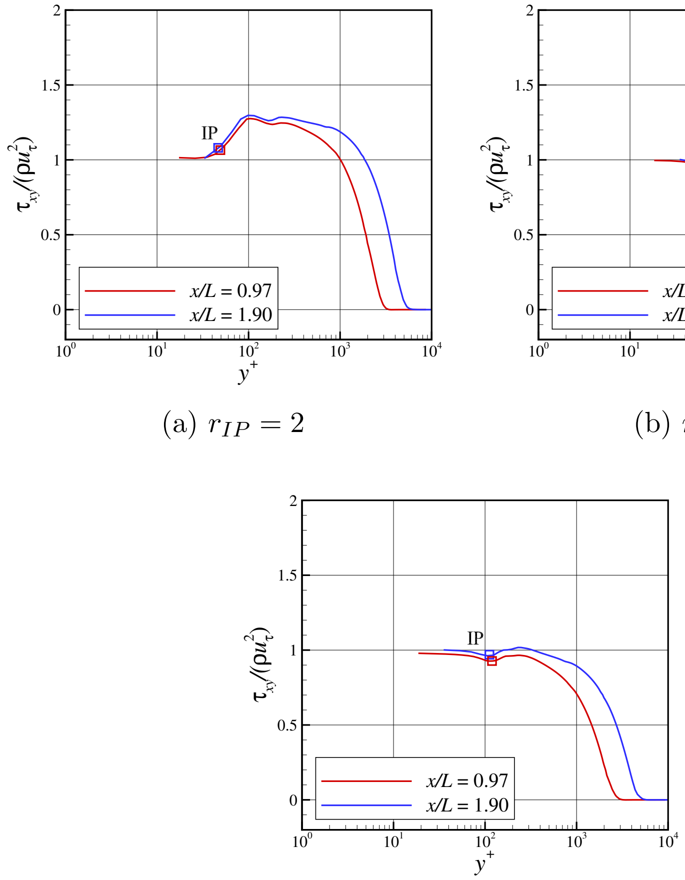


(a) $r_{IP} = 2$

(b) $r_{IP} = 3$

(c) $r_{IP} = 4.5$

Figure 5.10: Profiles of $\tilde{\nu}$ in the boundary layer (grid 4).



(a) $r_{IP} = 2$

(b) $r_{IP} = 3$

(c) $r_{IP} = 4.5$

Figure 5.11: Profiles of the shear stress in the boundary layer (grid 4).

5.2.2 Effect of Profile Modifications

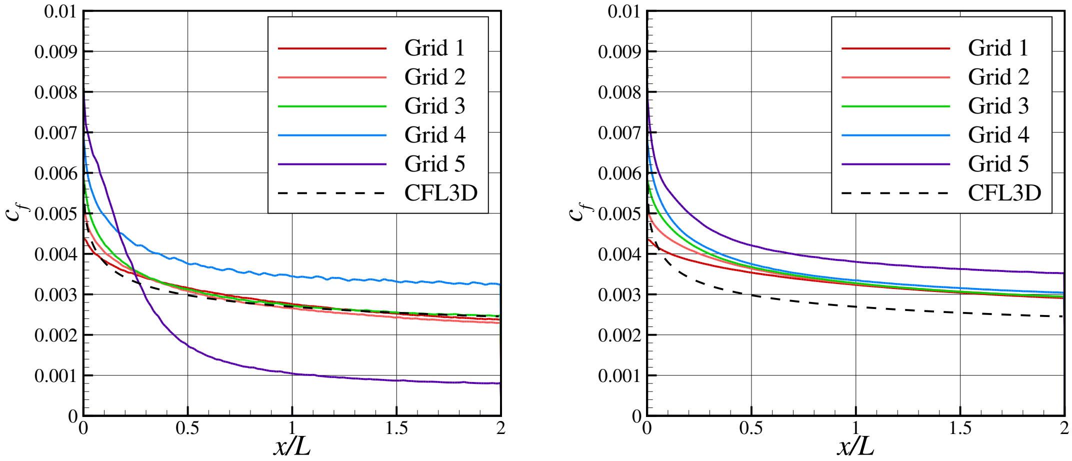
The modified IBM employs the modifications for both velocity and eddy-viscosity profiles. Here, the effects of the two profile modifications are examined. In addition to the original and modified IBMs, the IBM only with the modified velocity profile (3.25) is tested. In this IBM, the velocity profile is modified so that it is linear, but the balance of the shear stress (see Section 3.2.2) is not considered. Note that r_{IP} is fixed to 3 in this simulation.

The distributions of the skin friction calculated by the two IBMs are visualized in Fig. 5.12. In contrast to the results on the aligned grid, the original IBM cannot predict the skin friction accurately, especially on the fine grids. The IBM only with the modified velocity profile also cannot reproduce accurate distribution. Furthermore, the profiles of the tangential velocity, eddy-viscosity, and shear stress are shown in Figs. 5.13, and 5.14.

In the original IBM result, the shear stress has an unphysical peak near the IP (Fig. 5.13 (c)). In the results of the IBM only with the modified velocity profile, the shear stress stays at a low level compared with the ideal solution $\tau_{xy}/(\rho u_\tau^2) = 1$. The distributions of the tangential velocity and the eddy-viscosity are also different from the ideal solution.

The difference between the results on the aligned grid and the inclined grid can be discussed as follows. In the inclined grid, the cell boundary does not coincide with the plate surface, and the conservation law at the wall is not strict. Therefore, the balance of the fluxes fails when the velocity profile is not resolved. This results in the fluctuation of the shear stress in the inner layer. The modified IBM avoids this problem by changing the velocity profile below the IP, and thus, the method can reproduce accurate results.

The results here also confirm that the modifications for both the velocity and eddy-viscosity profiles are essential. The modified eddy-viscosity profile retains the balance of the shear stress as discussed in Section 3.2. Once the balance is lost, the shear stress given by the viscous flux at cell boundary near the wall is not correctly conveyed into the computational domain, as indicated in Fig. 5.14.



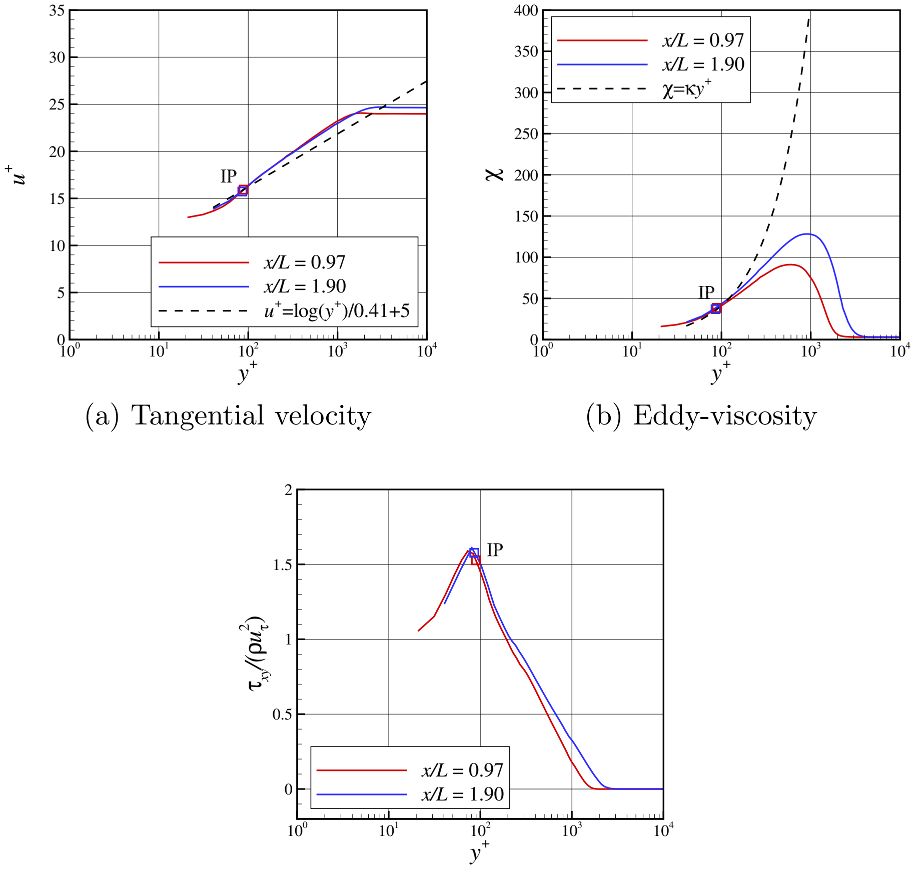
(a) Original IBM

(b) IBM only with the modified velocity profile

Figure 5.12: Distribution of the skin friction coefficient on the plate (original IBM).

5.2.3 Criterion for the Grid Setting

The criterion for grid setting for flows with different Reynolds numbers is considered. If the wall function is not used, y^+ at the first point off the wall (or the IP) is required to be



(a) Tangential velocity

(b) Eddy-viscosity

(c) Shear stress

Figure 5.13: Variable profiles in the boundary layer (original IBM, grid 4, $r_{IP} = 3$).

smaller than a certain value C_1. This is the case with a RANS simulation or a wall-resolved LES on body-fitted grids. Under this condition, the requirement for the minimum cell size on the wall Δy is

$$\frac{u_\tau \Delta y}{\nu} < C_1. \tag{5.12}$$

For the turbulent flat plate, the skin friction coefficient is estimated [69] as

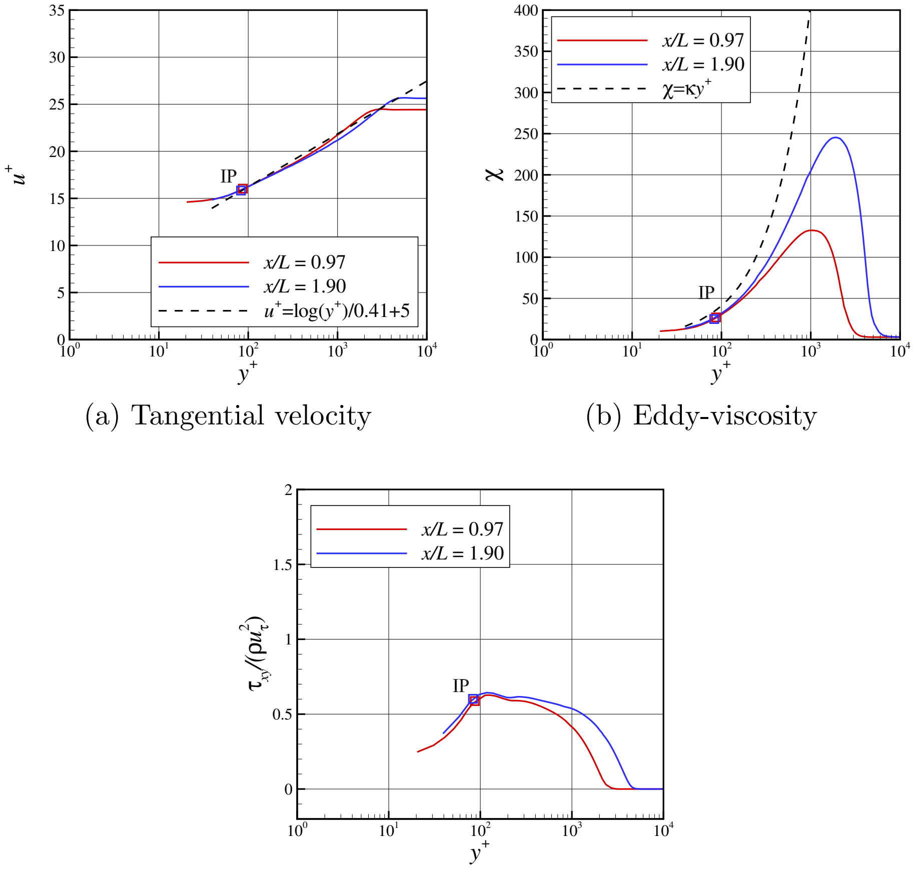
$$c_f \sim 0.026\mathrm{Re}_x^{-1/7}, \tag{5.13}$$

where $\mathrm{Re}_x = u_\infty x/\nu$ and x is the distance from the leading edge of the plate. Thus, the friction velocity is

$$\frac{u_\tau}{u_\infty} = \frac{1}{u_\infty}\sqrt{\frac{c_f u_\infty^2}{2}} \sim 0.11\mathrm{Re}_x^{-1/14}. \tag{5.14}$$

Substituting (5.14) into (5.12) yields the requirement for the grid size as

$$\frac{\Delta x}{x} < 8.77 C_1 \mathrm{Re}_x^{-13/14}. \tag{5.15}$$



(a) Tangential velocity

(b) Eddy-viscosity

(c) Shear stress

Figure 5.14: Variable profiles in the boundary layer (IBM only with the modified velocity profile, grid 4, $r_{IP} = 3$).

In contrast, the wall function method just requires the IP to be located in the region where the log-law is satisfied. As shown in Fig. 5.9, the log-layer extends to higher y^+ when the Reynolds number is higher. In addition, the velocity profiles in the outer layer of the turbulent boundary layer is generally a function of only y/δ as noted in [61], where δ is the boundary layer thickness. Thus, the upper limit of the log-layer is also defined by a function of y/δ.

To determine a universal criterion for the grid setting, the requirement for the grid size on the wall should be related to y/δ. Namely, the requirement is defined using a constant C_2 as

$$\frac{\Delta x}{\delta} < C_2. \tag{5.16}$$

The boundary layer thickness is assumed by the empirical relationship [69] as

$$\frac{\delta}{x} \simeq 0.37 \mathrm{Re}_x^{-1/5}. \tag{5.17}$$

Using (5.16) and (5.17), the requirement for the grid size is estimated:

$$\frac{\Delta x}{x} < 0.37 C_2 \mathrm{Re}_x^{-1/5}. \tag{5.18}$$

This requirement is milder than (5.15), and accordingly, the cell number can be less.

With $r_{IP} = 3$, the skin friction coefficient at $x/L = 0.97$ can be predicted within the 5% range on grid 3 ($\Delta x = 2.85 \times 10^{-4}$) and the finer grids. For the skin friction coefficient at $x/L = 1.90$, grid 2 ($\Delta x = 5.70 \times 10^{-4}$) and the finer grids can predict with this accuracy. In addition, the boundary layer thicknesses δ/L at $x/L = 0.97$ and 1.90 are 0.0164 and 0.0282, respectively. For each case, this thickness is approximately 50 times the grid size required for the prediction within the 5% range. Thus, the following criterion is derived: the grid size on the wall should be smaller than one-fiftieth of the boundary layer thickness. Furthermore, the IP should be located at $y^+ > 30$ to avoid the numerical problem in the buffer layer. Note that the grid of the wall is coarsened in the region off the wall, and the actual number of cells in the boundary layer is approximately 20, as shown in Fig. 5.15.

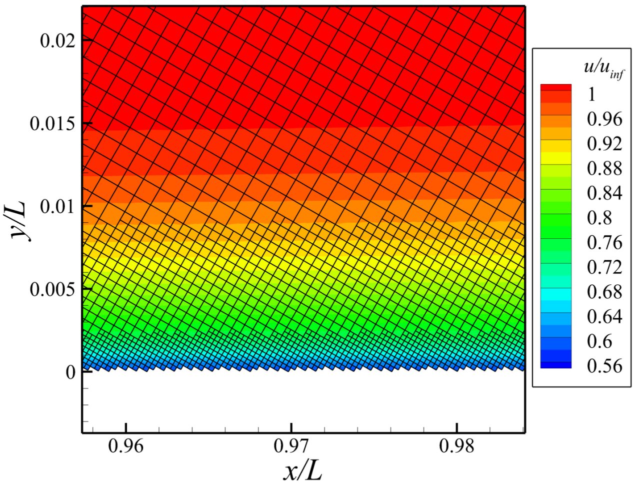


Figure 5.15: Computational grid and the distribution of the tangential velocity in the boundary layer (grid 3, $r_{IP} = 3$).

5.3 Subsonic Flow past a 2D Bump

The second test case is the subsonic flow past a 2D bump. In (3.1), the baseline approximated governing equation for the proposed IBM, the effect of the stream-wise pressure gradient is assumed to be small compared with that of viscous force and neglected. This assumption is valid except for the location close to a separation point, as discussed in [61]. In this problem, the validity of the proposed IBM in a flow with mild pressure gradient is therefore investigated.

This problem is also defined in the TMR [94]. The shape of the bump is defined as

$$\frac{y}{L} = \begin{cases} 0.05 * \left\{\sin\left(\frac{\pi x}{0.9L} - \frac{\pi}{3}\right)\right\}^4 & \text{for } 0.3 < x/L < 1.2 \\ 0 & \text{for } x/L \leq 0.3 \text{ and } x/L \geq 1.2 \end{cases}, \tag{5.19}$$

where $L = 1$ is the reference length. The Reynolds number for L, the free-stream Mach number, and the free-stream temperature are 3×10^6, 0.2 and 300 K, respectively.

The overview of the grid and the boundary conditions are illustrated in Fig. 5.16. Five grids with different grid resolutions are prepared to check the trend of grid convergence as tabulated in Table 5.3. In addition, r_{IP} is fixed to 3 in this problem.

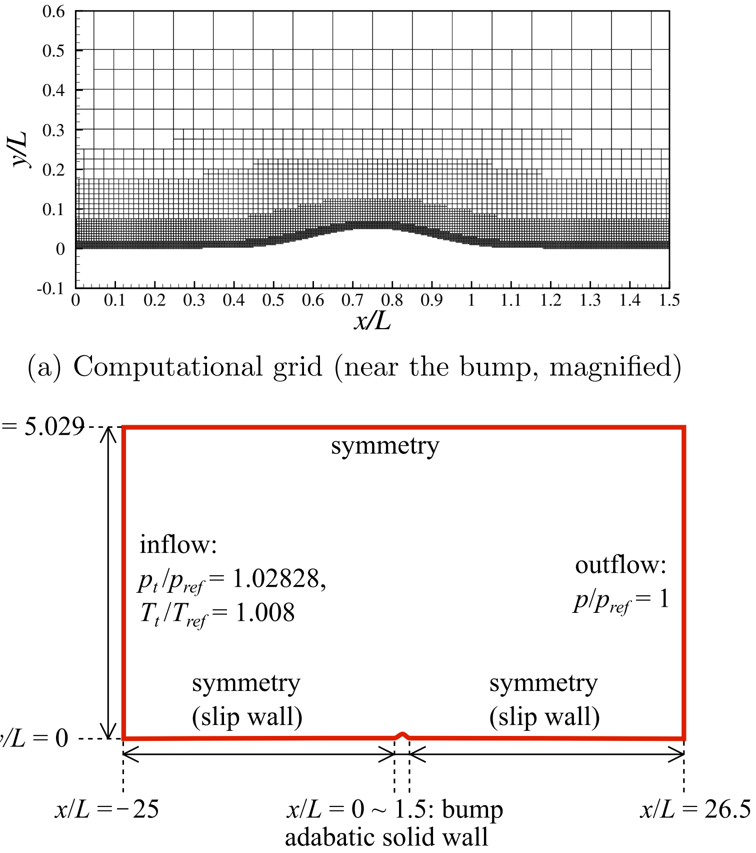
For this problem, the reference computational result is computed by CFL3D on the 1409×641 grid. This reference computational results are also provided in the TMR. The y^+_{IP} values in Table 5.3 is estimated by c_f of this reference result.

Table 5.3: Setting of computational grids over the 2D bump.

Grid	Minimum grid size	Cell number	y^+_{IP} at $x/L = 0.75$ (estimated)
1	1.57×10^{-3}	21,762	784
2	7.86×10^{-4}	43,246	392
3	3.93×10^{-4}	82,978	196
4	1.96×10^{-4}	164,638	98.0
5	9.82×10^{-5}	325,698	49.0

5.3.1 Computational Results

The distributions of the pressure and skin friction coefficient on the bump are shown in Figs. 5.17 and 5.18, respectively. The reference result by CFL3D is also shown in the figures. On one hand, large oscillation is observed on the pressure coefficient in the original IBM results, and the skin friction deviates from the reference result. This trend is especially obvious in the fine grids; the result on grid 5 predict the peak of c_f at a different location,



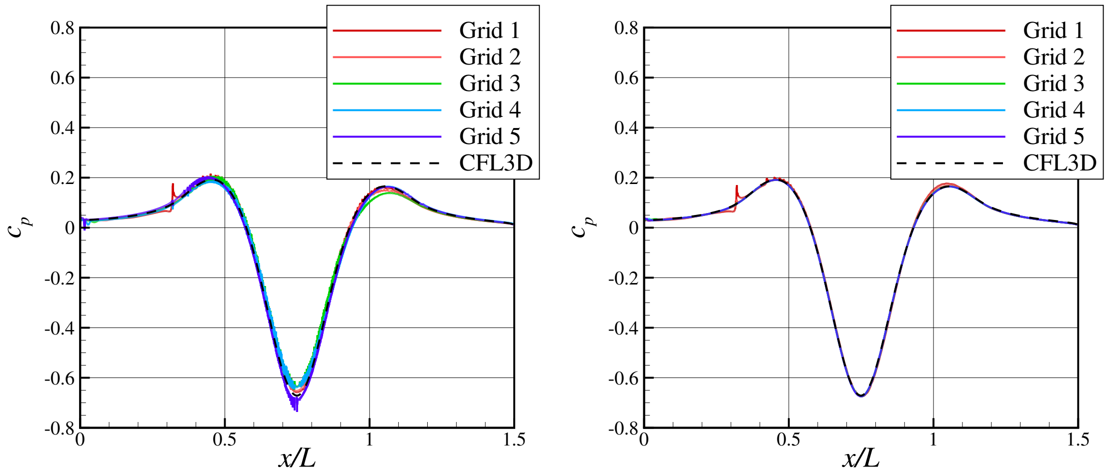
(a) Computational grid (near the bump, magnified)

(b) Boundary conditions

Figure 5.16: Computational grid over the bump.

and the magnitude of c_f is approximately 30% smaller than the reference result. As a result, no trend of grid convergence is observed in the original IBM results. On the other hand, the modified IBM reproduces the distribution of c_f more accurately. The oscillation on the pressure coefficient is smaller than the original IBM result, and the magnitude of C_p is also more accurate. In addition, the skin friction on the finer grids shows better agreement with the reference result, and thus, a correct trend of grid convergence toward the reference result is confirmed.

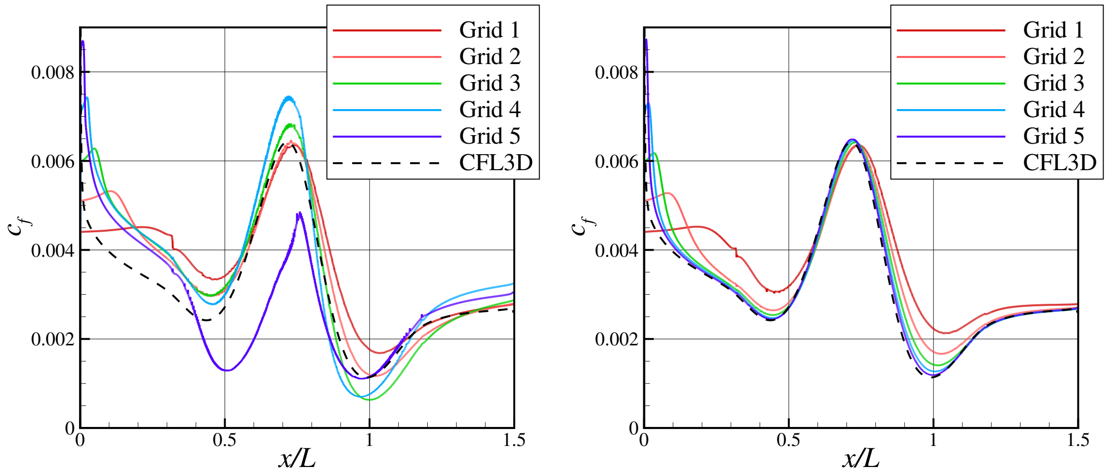
Figures 5.19 and 5.20 compares the profiles of the tangential velocity and the eddy-viscosity at $x/L = 0.75$ and 1.20. Note that the original and modified IBM results are those on grid 4. For the peak value of the eddy-viscosity, the modified IBM results show better agreement with the reference CFL3D result than the original IBM result. Furthermore, the velocity profile in the outer layer of the boundary layer is more accurately predicted in the modified IBM result.



(a) Original IBM

(b) Modified IBM

Figure 5.17: Distribution of the pressure coefficient on the bump.

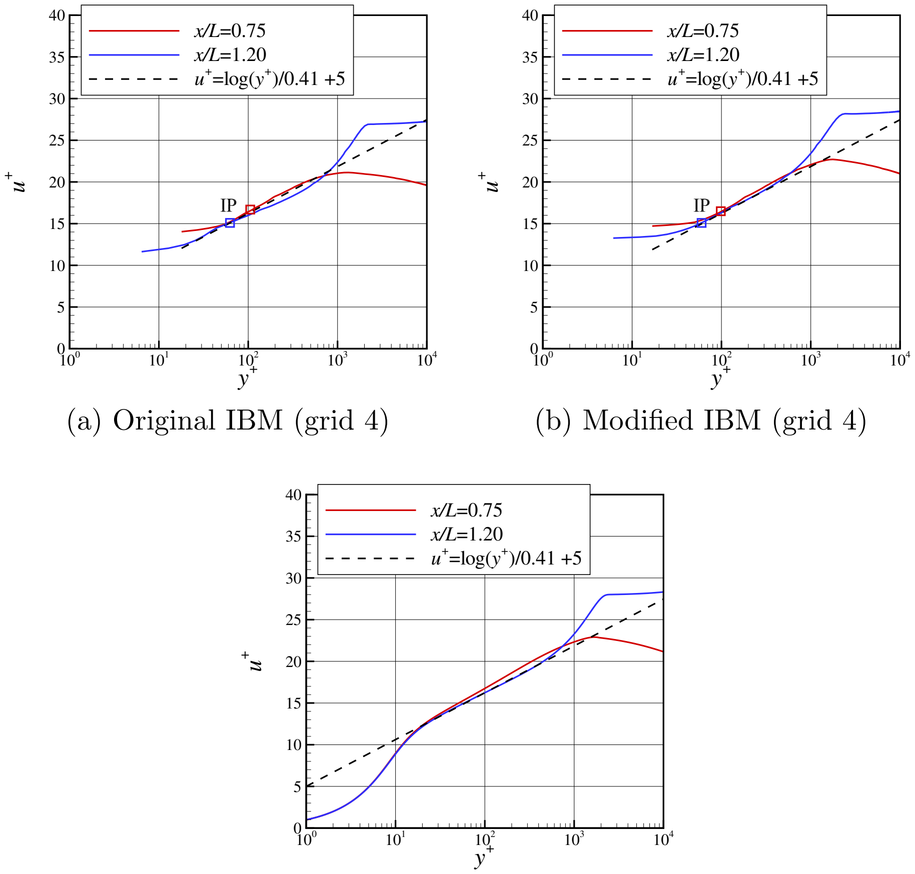


(a) Original IBM

(b) Modified IBM

Figure 5.18: Distribution of the skin friction coefficient on the bump.

The modified IBM is developed on the assumption that the effect of the pressure gradient is negligible, and that the log-layer appears. Here, the results show that the velocity profile in the inner layer still satisfies the log-law mostly, although a mild pressure gradient exists. Therefore, the modified IBM can reproduce this flow with a certain degree of accuracy.

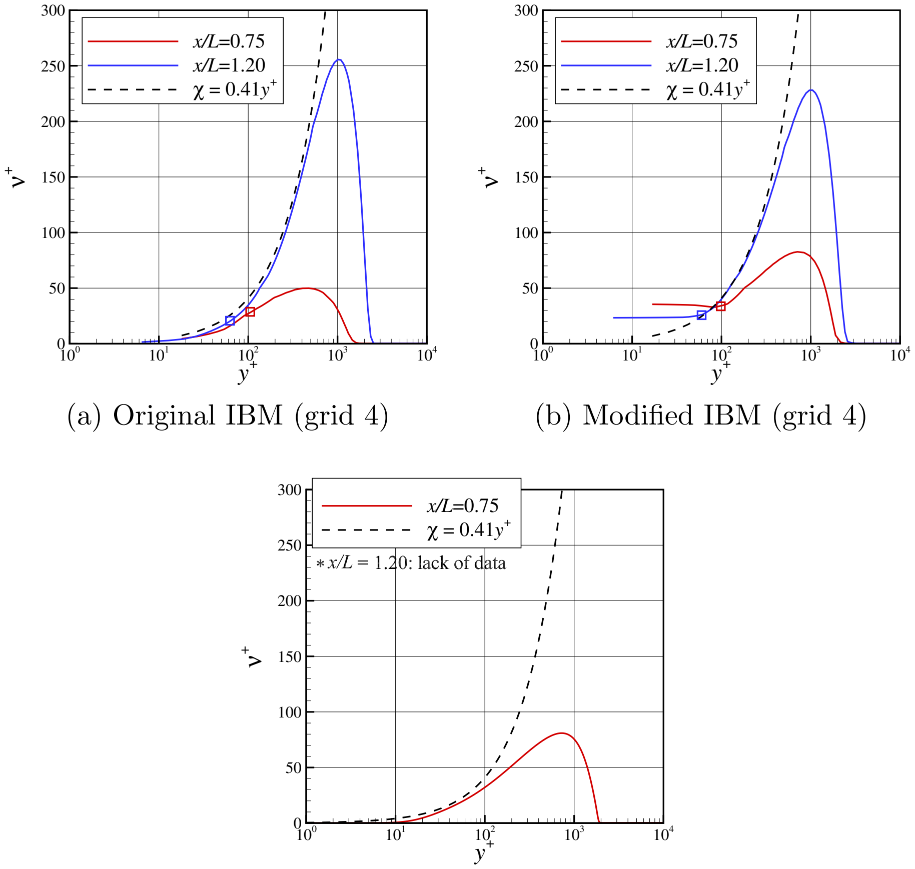


(a) Original IBM (grid 4)

(b) Modified IBM (grid 4)

(c) CFL3D

Figure 5.19: Profiles of the tangential velocity.



(a) Original IBM (grid 4)

(b) Modified IBM (grid 4)

(c) CFL3D

Figure 5.20: Profiles of the eddy-viscosity.

5.4 Subsonic Flow around a NACA 0012 Airfoil

To check the present IBM in terms of prediction accuracy of aerodynamic forces acting on an airfoil, subsonic flows around a NACA 0012 airfoil are simulated. Following the definition in the TMR [94], the Reynolds number for the airfoil chord c, the free-stream Mach number, and the free-stream temperature are set to 6×10^6, 0.15, and 300 K, respectively. The angles of attack simulated here are 0, 5, 10, and 15°. For the 0, 10 and 15 degrees, the reference results by CFL3D on 897×257 grid provided in the TMR are available.

The airfoil used in this problem is defined by (4.20), which has a sharp trailing edge. The size of the far-field boundary is 524×524. For the far-field boundary, the free-stream condition is imposed. Note that r_{IP} is fixed to 3 in all the calculation in this section. An example of the computational grid is shown in Fig. 5.21, and the settings of the computational grid are tabulated in Table 5.4.

The calculation cases are set as follows. At first, flows at all the angle of attack written above are computed using the modified IBM on all the grids. Then, the original and modified IBMs are compared at $\alpha = 0°$ on all the grids. In addition, the flows at all the four angles are simulated on grid 4 using the two IBMs.

Table 5.4: Setting of the computational grids around the NACA 0012 airfoil.

Grid	Minimum grid size	Cell number
1	1.00×10^{-3}	22,516
2	5.00×10^{-4}	55,140
3	2.50×10^{-4}	118,136
4	1.25×10^{-4}	244,906
5	6.25×10^{-5}	496,550

5.4.1 Alpha-Sweep

Figure 5.22 shows the calculated aerodynamic coefficients. The experimental data [100] are also shown in the figure for reference. Here, the flux-based method is used to calculate the aerodynamic force acting on the airfoil. On grid 1, the lift coefficient at $\alpha = 15°$ is approximately 30% lower than the reference result, and the pitching moment shows a different trend. On the finer grids, the lift coefficient at $\alpha = 15°$ becomes larger, while the drag coefficient becomes smaller. This trend seems to be depending on the magnitude of flow separation near the trailing-edge (Fig. 5.23).

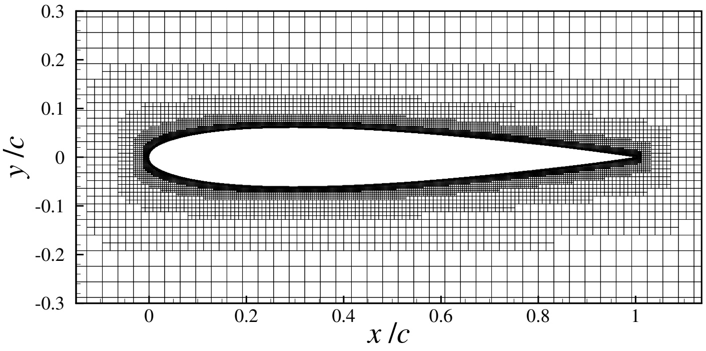


Figure 5.21: Computational grid around the NACA 0012 airfoil.

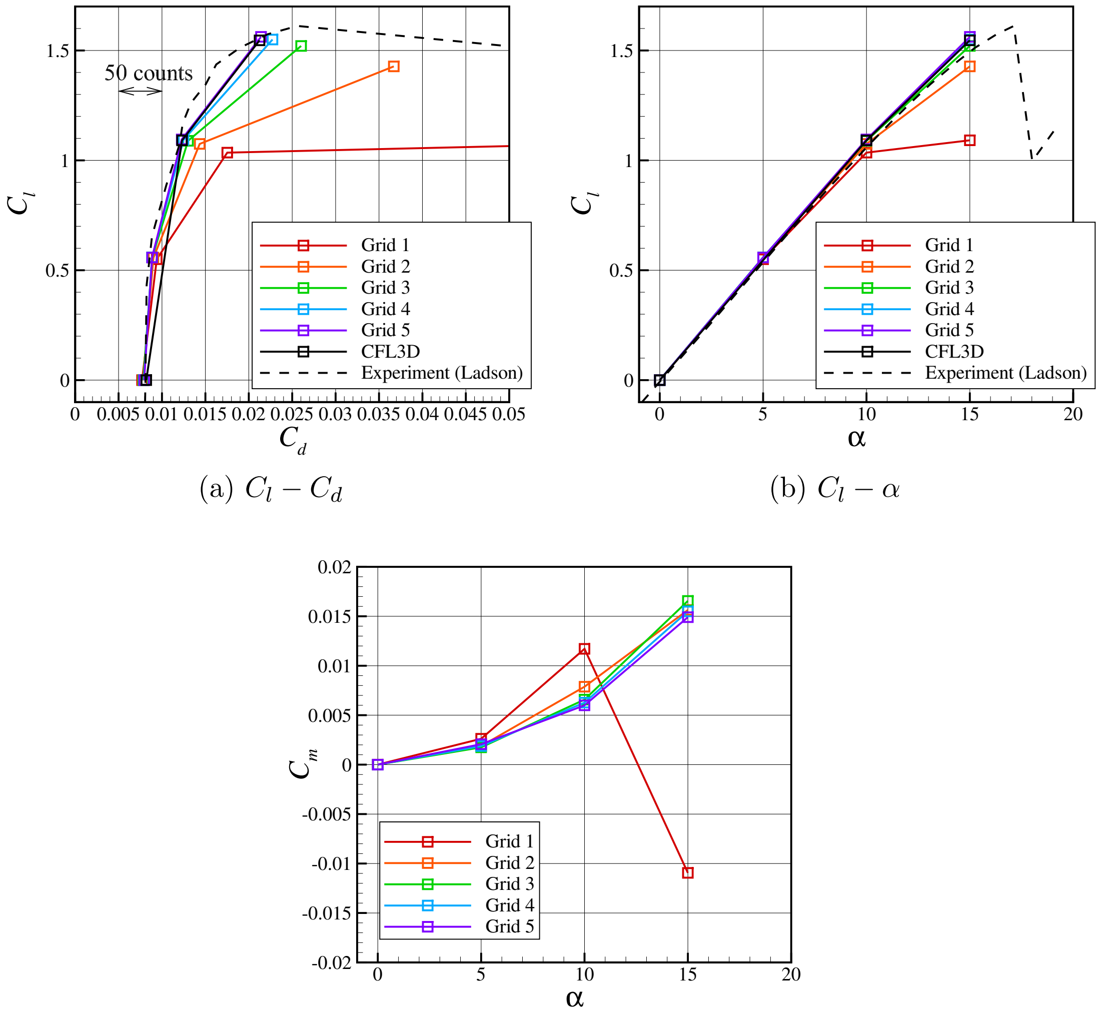
To analyze the flow at $\alpha = 15°$ in more detail, the surface pressure and friction coefficients are shown in Fig. 5.24. Note that the friction coefficient shown in the figure is the magnitude of the friction coefficient multiplied by the sign of its x-component. The separation point moves downstream when the grid is refined. The separated region is too large in the coarse grid, and it causes the underprediction of the lift coefficient the overprediction of the drag coefficient. Furthermore, the distribution of the skin friction except for the region close to the separation point is accurately predicted in the results on grid 4 and 5.

5.4.2 Comparison between the Original and Modified IBMs

Figure 5.25 shows the distribution of the skin friction coefficient at $\alpha = 0°$. On one hand, the original IBM results overestimate the skin friction near the leading edge and underestimate near the trailing edge. This means the growth of the boundary layer is not correctly reproduced. These trends are more obvious in the fine grid, and thus, the trend of the grid convergence is incorrect.

On the other hand, the modified IBM can predict the skin friction more accurately. In particular, the skin friction near the trailing edge shows good agreement with the reference result. One discrepancy is observed near the leading edge; the skin friction in this region is slightly overestimated. This is because the boundary layer is very thin in this region due to the favorable pressure gradient. Furthermore, the results on the finer grid show better agreement with the reference result.

Figure 5.26 compares the aerodynamic coefficients by the original and modified IBMs on grid 4. The original IBM overestimates the drag coefficient at the low angles of attack. Furthermore, the lift coefficient at 15°, and the drag is lower. This is because the trailing edge separation does not appear in the original IBM result. Compared with this original IBM result, the modified IBM predicts the aerodynamic force more accurately.



(a) $C_l - C_d$

(b) $C_l - \alpha$

(c) $C_m - \alpha$

Figure 5.22: Calculated aerodynamic coefficients of the NACA 0012 airfoil.

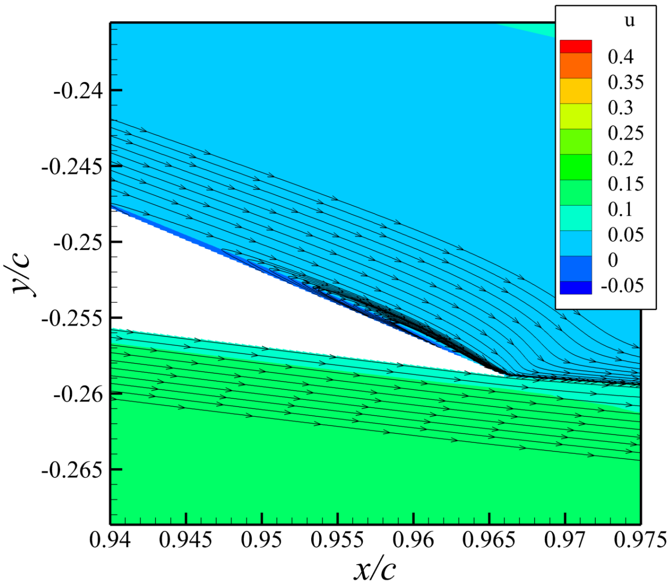
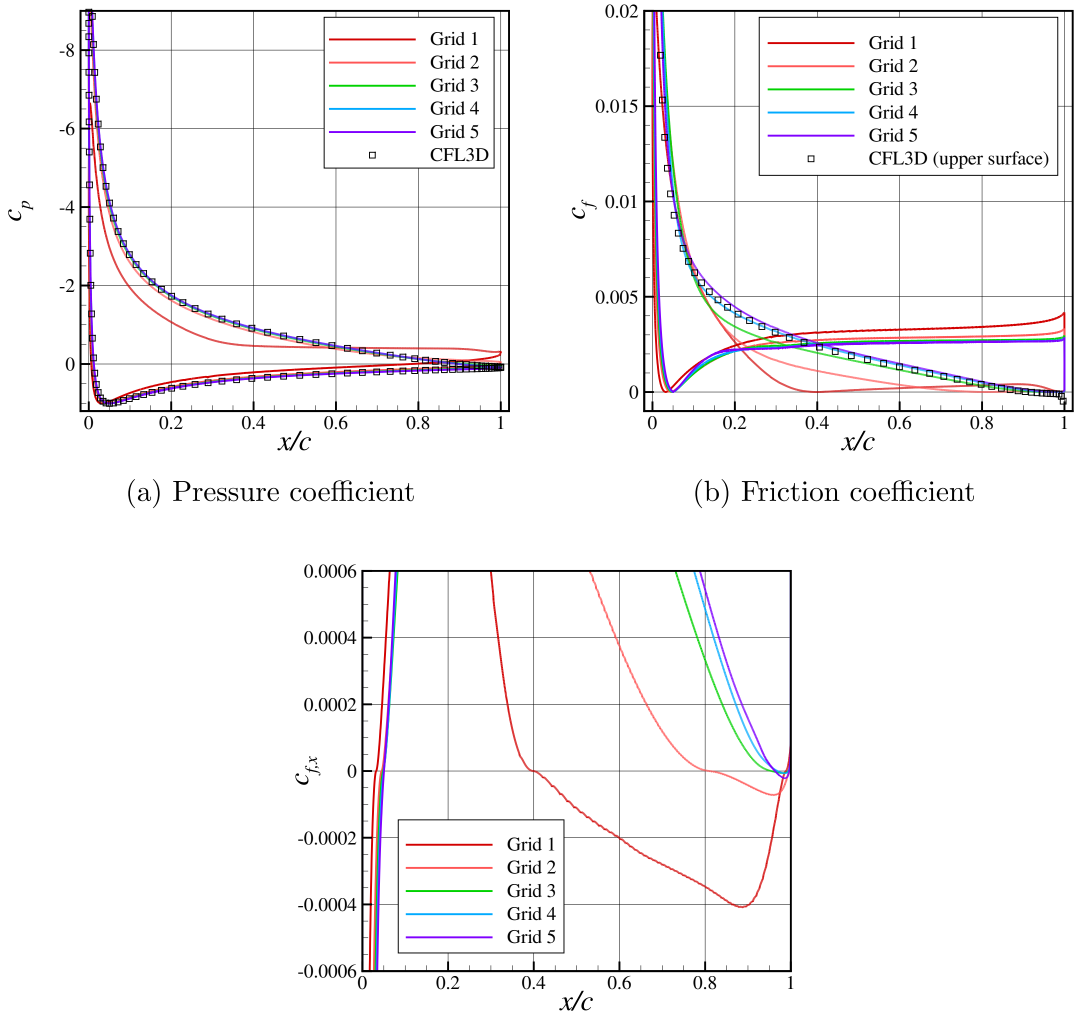


Figure 5.23: Streamlines around the trailing edge (grid 5, $\alpha = 15°$).

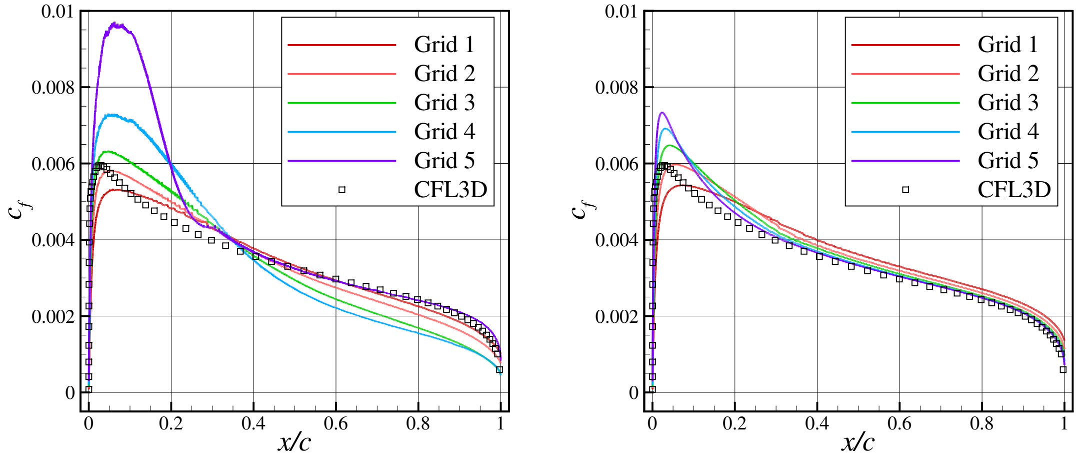


(a) Pressure coefficient

(b) Friction coefficient

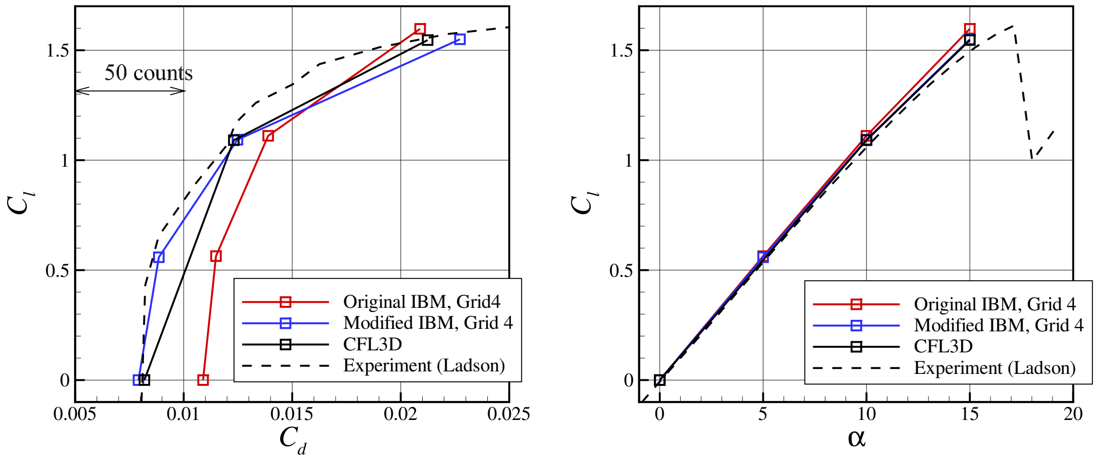
(c) Friction coefficient (near trailing edge, magnified)

Figure 5.24: Distributions of the surface pressure and skin friction coefficients on the NACA 0012 airfoil at $\alpha = 15°$.



(a) Original IBM (b) Modified IBM

Figure 5.25: Skin friction coefficient on the NACA 0012 airfoil ($\alpha = 0°$).



(a) $C_l - C_d$ (b) $C_l - \alpha$

Figure 5.26: Comparison of the aerodynamic coefficient of the NACA 0012 airfoil (grid 4).

5.4.3 Discussion

Here, the criterion of grid setting for this case is discussed. The error of the calculated aerodynamic coefficients in the modified IBM result is tabulated in Table 5.5. The CFL3D result is regarded as the reference value of this problem, and the relative error compared to the CFL3D result is indicated in the table.

For C_d at $\alpha = 0°$, grid 2 and the finer grids can predict the value with less than 5% error. The C_l is also relatively easy to be predicted, except for the $\alpha = 15°$ case where the flow separation occurs. Furthermore, the scatter of C_d and C_l at $\alpha = 5°$ is small as shown in Fig. 5.22. In contrast, accurate prediction of C_d at $\alpha = 10°$ and 15° requires finer resolution of grids.

Figure 5.27 shows the distribution of the estimated boundary layer thickness relative to the minimum cell size $\delta/\Delta x$. Note that y^+_{IP} in this figure is calculated using the c_f value in the reference result with $r_{IP} = 3$. Furthermore, δ is calculated by the formula for the turbulent flat plate (5.17), and thus, this value may be therefore different from the actual boundary layer thickness on the airfoil. Following the discussion in Section 5.2.3, the grid size should be smaller than one-fiftieth of the local boundary layer thickness. In this simulation, grid 3 satisfy this requirement at the trailing edge. In contrast, grid 1 and 2 do not satisfy it over the entire airfoil, and this may be the reason the skin friction (Fig. 5.25) is slightly overestimated in these two cases.

The criterion for the grid setting is concluded as follows. If no flow separation occurs, grid density similar to grid 3 is recommended; the cell size is smaller than one-fiftieth of the boundary layer thickness at the trailing edge. Under this condition, the calculated lift and drag coefficients are expected to have less than 5% error. For the simulation of only the flow at the low angle of attack, lower grid density is acceptable. This may be also related to the shape representation of the airfoil; grid 1 (1000 cells per chord) seems to be the acceptable value. Furthermore, for the simulation of flows with a smooth body separation, the quantitative feature can be reproduced if grid size is sufficiently small; in this case, at least grid density of grid 3 is recommended. Note that this criterion will not be the case with the flows with fixed separation point (e.g., flow around a sharp-edge).

Table 5.5: Percentage of the relative error of the modified IBM results. The reference value is taken from the CFL3D results.

	Modified IBM					Original IBM
	Grid 1	Grid 2	Grid 3	Grid 4	Grid 5	Grid 4
C_d $(\alpha = 0°)$ %	5.934	4.786	4.151	3.321	2.943	33.00
C_d $(\alpha = 10°)$ %	42.32	16.52	5.621	1.357	0.3006	12.85
C_l $(\alpha = 10°)$ %	5.110	1.484	0.1799	0.2002	0.4024	1.819
C_d $(\alpha = 15°)$ %	270.0	72.80	22.53	7.010	0.7721	1.629
C_l $(\alpha = 15°)$ %	29.42	7.667	1.656	0.2418	1.028	3.309

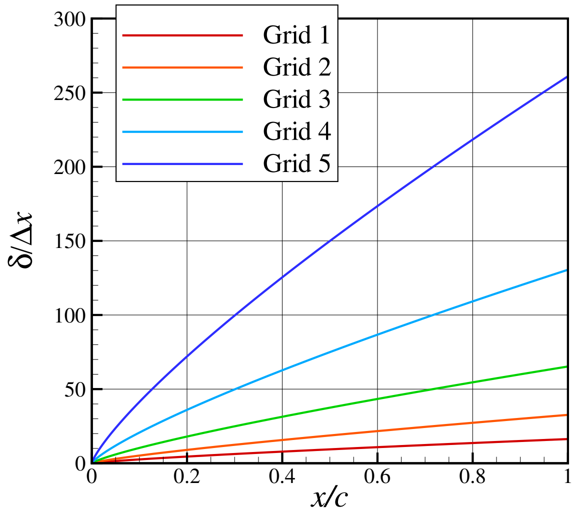


Figure 5.27: Estimated distribution of the relative boundary layer thickness to the minimum cell size on the NACA 0012 airfoil.

5.5 Summary of Chapter 5

In this chapter, the developed IBM was tested in simple 2D turbulent flow simulations defined in the TMR. For all the problems, the simulation was conducted on multiple grid resolutions to check the trend of the grid convergence. The IBM with the profile modifications (modified IBM) and without them (original IBM) were also compared.

The first problem was a turbulent flat plate boundary layer. The simulation was conducted on aligned (body-fitted) and inclined (non-body-fitted) series of grids. On the aligned grid, the original and modified IBMs similarly produce relatively accurate results. In contrast, only the modified IBM can predict accurate skin friction on the inclined grid. The variable profiles in the boundary layer were also investigated; When the modified IBM is used, the shear stress near the wall becomes constant with respect to the wall distance as implicated in the analysis of the governing equation. The setting of the grid size and the distance of the IP was also investigated. The distribution of the skin friction can be accurately reproduced when r_{IP}, the ratio of the IP distance to the cell size at the wall, is larger than 3. When the grid size on the wall is smaller than one-fiftieth of the local boundary layer thickness, the skin friction is predicted with less than 5% error.

The second problem was the turbulent flow past a 2D bump. In this problem, flows with a mild pressure gradient was simulated. The modified IBM again shows improvement from the original IBM in the prediction of the skin friction although the pressure gradient is neglected in the derivation of the modified IBM.

The last problem was the turbulent flows around a NACA 0012 airfoil. In the α-sweep, a correct trend of the grid convergence is observed. In addition, the modified IBM predicts the drag coefficient and the skin friction on the airfoil more accurately than the original IBM. Finally, the criterion for the grid settings of aerodynamic prediction of the airfoil was discussed. When the cell size is smaller than one-fiftieth of the estimated boundary layer thickness at the trailing edge, the lift and drag coefficients are predicted with less than 5% error. Furthermore, the cell size can be larger if only the flows at low angles of attack are simulated.

Chapter 6

Simulations of Turbulent Flows around Aircraft

In this chapter, the capability and efficiency of the proposed framework for turbulent simulations around an aircraft are demonstrated. The benchmark problems are as follows:

- transonic flows around the NASA Common Research Model (CRM) [101];
- subsonic flows around the JAXA standard high-lift model (JSM) [102].

For those problems, the modified IBM is used for the wall boundary condition in UTCart. In addition, the aerodynamic force is evaluated by the flux-based method.

6.1 Transonic Flows around the NASA Common Research Model

To investigate the capability of the proposed framework for the aerodynamic prediction on a civil transport aircraft, transonic flows around the NASA Common Research Model (CRM) [101] are simulated. The NASA CRM was developed as a benchmark problem in the Drag Prediction Workshops (DPWs) [49]. This geometry is widely tested in wind tunnel experiments [103, 104] and numerical simulations [50–53, 75, 105, 106].

Recently, a domestic workshop in Japan, the Aerodynamic Prediction Challenge [107] workshop, was held to investigate the accuracy of the aerodynamic prediction of the NASA CRM using CFD simulations. The geometry tested in this workshop consists of a fuselage, main wings, and horizontal tails with the incident angle of attack of $i_H = 0°$. The calculation setting in this section is adjusted to the condition of the experiment [104] in JAXA's transonic wind tunnel, in which a 2.16% scale model (the mean aerodynamic chord c_{ref} =151.31 mm) was used. The free-stream Mach number is 0.847, the free-stream temperature is 284K, and the Reynolds number based on the mean aerodynamic chord is 2.26×10^6. The angles of attack are from -1.79 to 5.72°.

In the wind tunnel experiment, the wing is deformed by the aerodynamic force acting on the wing. In the sixth DPW, it is reported that prediction accuracy of the surface pressure and the aerodynamic coefficient improves by considering the aerodynamic deformation of the wing [106]. The deformation (twist and bend) of the wing was measured [104], and the data is provided in [107]. The geometry used in this simulation is also deformed based on the experimental data, as shown in Fig. 6.1.

6.1.1 Simulation Settings

UTCart uses oct-tree-based Cartesian grids (see Fig. 6.2). Here, a symmetry boundary condition is assigned on the $y = 0$ plane, and a half-span model is simulated. To reduce the computational cost, two different cell sizes are specified on the wall. The wing upper surface and the tail are covered by the finest level of the cell because the flow features in those regions are expected to be critical to simulate the aerodynamic force accurately. The other parts, the fuselage and wing lower surface, are covered with the second finest level of the cell to save computational cost. The ratio of the IP distance to the cell size, r_{IP}, is set to three.

To check grid sensitivity, coarse, medium and fine grids are prepared. In addition, a "medium-b" grid is created by changing the number of cells in the smooth layer, as shown in Fig. 6.3. Table 6.1 describes the specification of these grids. The lengths represented in the table is based on the actual scale of the NASA CRM (c_{ref} = 275.8 inches). The cell number slightly changes when the wing is deformed, and the numbers shown in the table are those at $\alpha = 2.94°$. The grids are generated on a workstation (Xeon E5–2643 v3 at 3.4GHz) with one core. The time spent for generation of the medium grid is 16 minutes excluding the time for file I/O and setting the boundary conditions.

The computational cases by UTCart are as follows. First, grid sensitivity is examined at $\alpha = 2.94°$ on the coarse, medium, medium-b, and fine grids. Then, the flows at α = -1.79, 0.62, 2.47, 2.94, 3.55, 4.65, and 5.72° are simulated on the medium grid. Furthermore, reference calculations are conducted by FaSTAR [95] on body-fitted grids. The computational grids are provided in the APC workshop, whose overview is shown in Fig. 6.4. Note that the grid is originally created for a structured-grid-based flow solver, but they are treated as an unstructured grid in FaSTAR. Using FaSTAR, the flows at α = -0.62, 2.94, 4.65 and 5.72° are simulated on the provided medium grid (9,006,808 hexahedral cells), and the flow at α =2.94° is simulated also on the fine grid (30,397,977 hexahedral cells).

The computational methods for each solver are summarized in Table 6.2. The "Hishida" limiter is a slope limiter for unstructured grids, which has a similar functional shape as van Albada's limiter [108]. For the turbulence model, the rotational correction and the QCR (see Section 2.2) are used in conjunction with the SA turbulence model to improve the prediction accuracy of the side-body separation, following [75]. The flow computation is conducted on the Reedbush-U supercomputer of the University of Tokyo (Xeon E5–2697 v4 at 2.1 GHz) [109] using 144 cores.

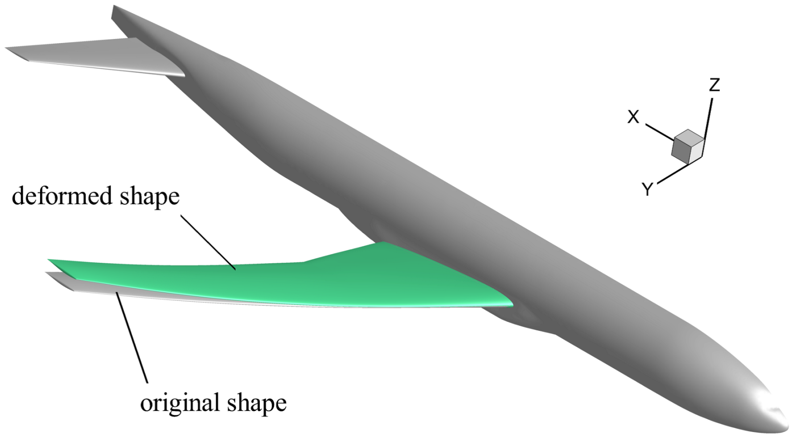


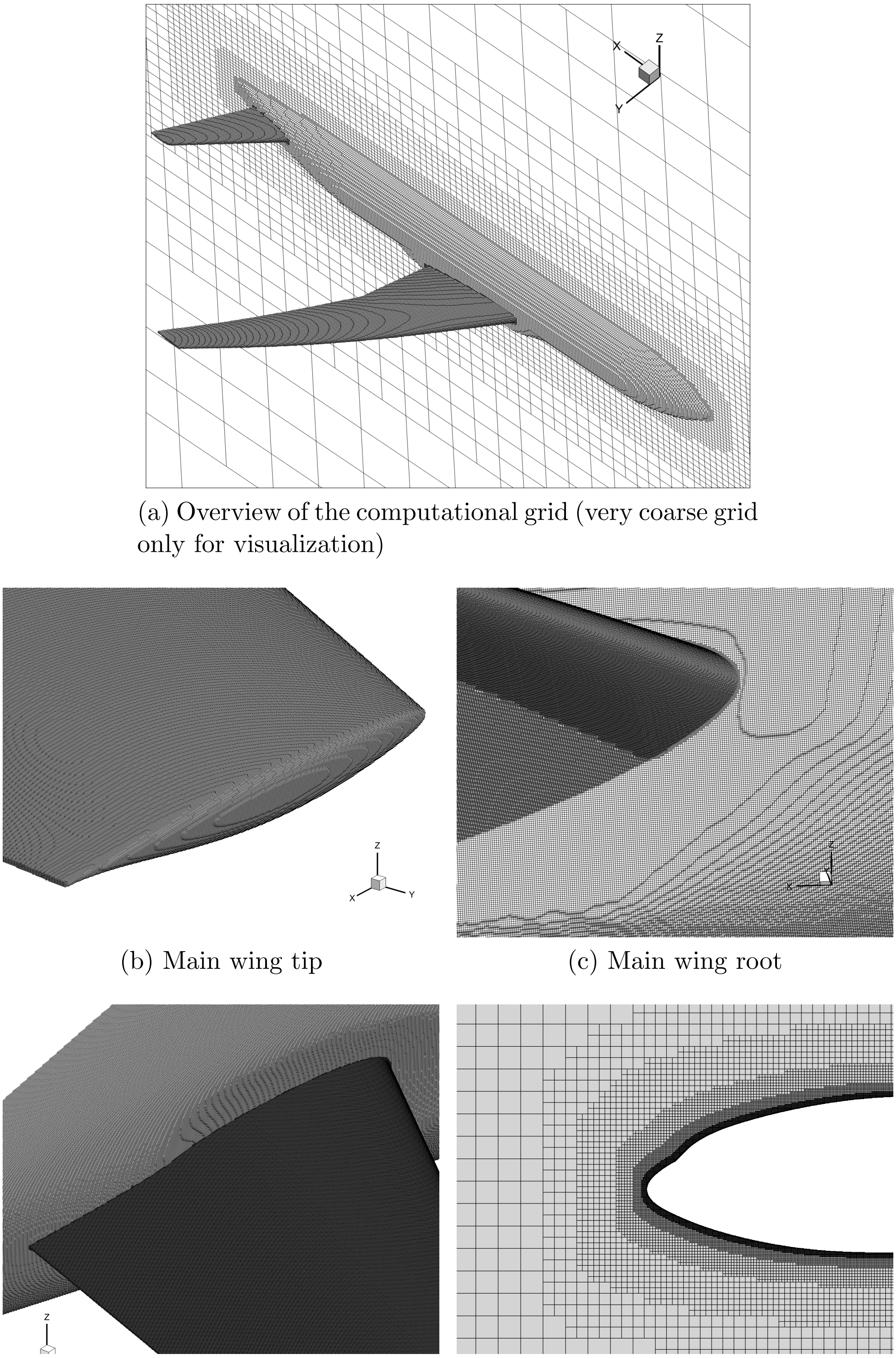
Figure 6.1: Aerodynamic deformation of the wing.

Table 6.1: Settings of computational grid around the NASA CRM for UTCart.

	Coarse	Medium	Medium-b	Fine
Total cell number	31,055,490	61,988,288	54,335,363	117,882,932
Domain size inch	4.80×10^4	3.60×10^4		5.40×10^4
Grid size inch (wing upper surface/tail)	0.732	0.549		0.412
Grid size inch (wing lower surface/fuselage)	0.366	0.274		0.206
Wall layer	3			
Smooth layer (near field)	3	6	3	8
Smooth layer (far field)	3			
c_{ref}/Grid size (wing upper surface)	753	1,004		1,339

Table 6.2: Numerical methods for UTCart and FaSTAR.

Solver	UTCart	FaSTAR
Inviscid flux	SLAU	
Numerical scheme (advection term)	MUSCL	
Limiter	minmod	Hishida [110]
Numerical scheme (viscous term)	Second-order central difference	
Gradient reconstruction	WLSQ	GLSQ [77]
Time integration	MFGS	Lower-Upper Symmetric Gauss–Seidel [111]
Time Stepping	Local	
Courant number	200	50



(a) Overview of the computational grid (very coarse grid only for visualization)

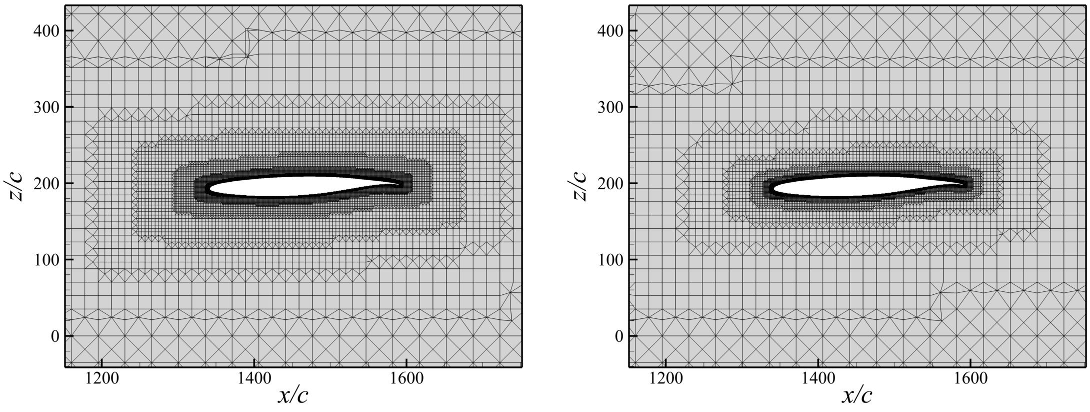
(b) Main wing tip

(c) Main wing root

(d) Tail wing root

(e) Symmetry plane around the nose

Figure 6.2: Computational grid for UTCart (medium grid except for the overview).



(a) Medium grid

(b) Medium-b grid

Figure 6.3: Cross sectional view of the computational grid.

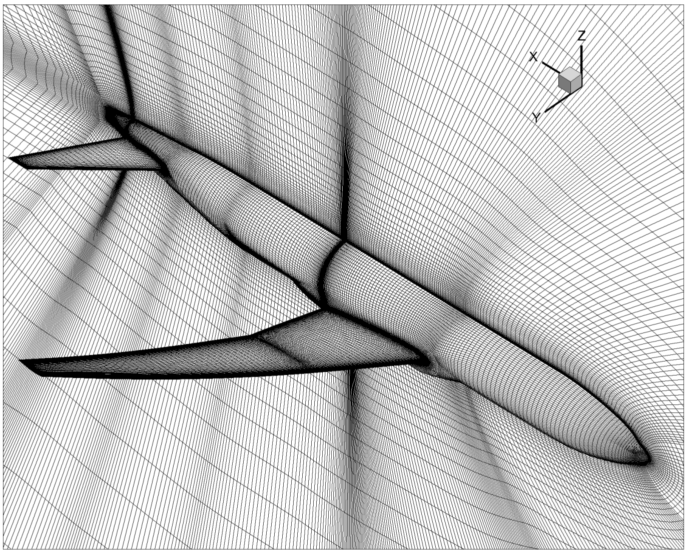


Figure 6.4: Computational grid for FaSTAR (medium grid, provided in [107]).

6.1.2 Grid Sensitivity Study

Figure 6.5 shows the time-step convergence of the drag coefficient on the medium grid at $\alpha = 2.94°$. The magnitude of the oscillation during the final 1000 time-step iterations is smaller than 0.1 drag count (10^{-5}). The time spent for the 8000 time-step iterations was 5.9 hours.

Figure 6.6 compares the distributions of the pressure coefficient on the surface by the two flow solvers. The qualitative features (e.g., the position of the shock on the wing upper surface) show good agreement with each other. The distributions of the surface pressure coefficient on the section of the wing are then shown in Fig. 6.7. The definition of the sections is following the APC workshop, as shown in Fig. 6.8. These sections are identical with the positions of the pressure taps for the experiment. At the inboard sections, the surface pressure coefficient in the UTCart result shows good agreement with the FaSTAR result and the experimental data. The pressure distributions at the outboard sections are slightly different from the FaSTAR result. In the UTCart grid, the grid size on the upper surface of the wing is uniform, and accordingly, the number of cells within the local chord is smaller in the outboard sections. This means the grid resolution relative to the local chord length is low in the outboard sections and is assumed to be one of the causes of the inaccuracy there. Furthermore, the shock at section I in the UTCart result is thinner than that in the FaSTAR result. This seems to be because the computational grid for UTCart has higher grid resolution in the chord-wise direction than the grid for FaSTAR.

The distributions of the skin friction coefficient on the section of the wing are shown in Fig. 6.9. The difference of the skin friction coefficient is more obvious than that of the pressure coefficient. Note that the peak of the friction in the FaSTAR result may be due to the skewness of the cell around the trailing edge, and thus, it is not considered here. On the upper surface, the distribution of the skin friction has a local minimum, where the skin friction decreases due to the adverse pressure gradient caused by the shock. In the UTCart results, this phenomenon is well reproduced at the inboard sections, especially on the fine grid. At the outboard sections, the friction is also less accurate because of the low grid resolution relative to the local chord length. In addition, the skin friction on the lower surface tends to be overestimated, which may be because the coarser level of the grid is assigned to this region.

Figure 6.10 shows the component-wise aerodynamic coefficients. The pressure drag computed by UTCart tends to be overestimated, especially on the coarse grid. The pressure drag in the medium-b grid result is 3 drag count larger than the value of the medium grid result. This indicates the pressure drag is dependent on the grid resolution in the region off the wall, and a proper grid refinement required. Furthermore, the viscous drag is overestimated by 7 drag counts even on the fine grid. The main cause of this prediction is the wing and body, which seems to be related to the overprediction of the skin friction on the wing lower surface. Simultaneously, the lift coefficient in the UTCart result is overestimated compared with FaSTAR result, while the pitching moment coefficient is underestimated. For these two coefficients, the trend of grid convergence is observed toward the FaSTAR result. The main cause of these discrepancies is the main wing. They may be due to the grid resolutions to capture the curvature of the leading edge and the thickness of the trailing edges.

The grid convergence behavior of the drag coefficient is then examined. In this simulation, C_L is not fixed. Therefore, the estimated induced drag $C_L^2/(\pi AR)$ is subtracted from the total drag coefficient to reduce the error caused by comparing different lift conditions. Figure 6.11 shows the grid convergence of $C_D - C_L^2/(\pi AR)$, where the drag coefficients are plotted versus the cell number N to the power of $-2/3$. If the grid is refined uniformly and the spatial order of accuracy is two, the plot becomes straight. Following the Richardson extrapolation process in [105], the converged value is estimated as the y-intercept of the extrapolated line:

$$\mathcal{Y}_0 = \frac{\mathcal{X}_2\mathcal{Y}_1 - \mathcal{X}_1\mathcal{Y}_2}{\mathcal{X}_2 - \mathcal{X}_1}, \tag{6.1}$$

where $\mathcal{X} = N^{-2/3}$ and $\mathcal{Y} = C_D - C_L^2/(\pi AR)$. In addition, the subscripts 0, 1, 2 denote the converged value, the value on grid 1, and that on grid 2, respectively. The converged value is calculated using each grid pair is tabulated in Table 6.3. Compared with the converged value of the FaSTAR results, UTCart overestimates the drag coefficient by 29 (13%), 19 (8.9%), and 11 (5.1%) drag counts on the coarse, medium, and fine grids, respectively. In contrast, the extrapolated values of the UTCart results have less than 2% difference from the converged value of the FaSTAR results.

Finally, Fig. 6.12 visualizes the distribution of y_{IP}^+ on the surface. The value is low on the wing upper surface and tail because smaller Δx is assigned there. On the wing top surface, the maximum value is 533. Note that the values on the coarse and fine grids are approximately 33% larger and 25% smaller than the medium grid value, respectively.

6.1.3 Alpha-Sweep

Figure 6.13 shows the computed and measured aerodynamic coefficients at each angle of attack. The basic trend of each coefficient shows fair agreement between the UTCart and FaSTAR results, and between the UTCart results and the experimental data. In the computational results, however, the lift slope and pitching moment coefficient are underestimated

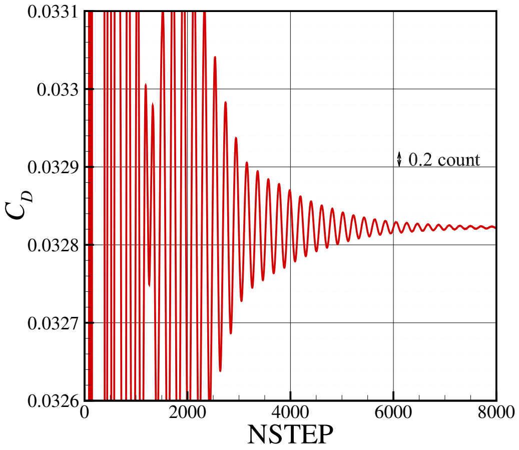
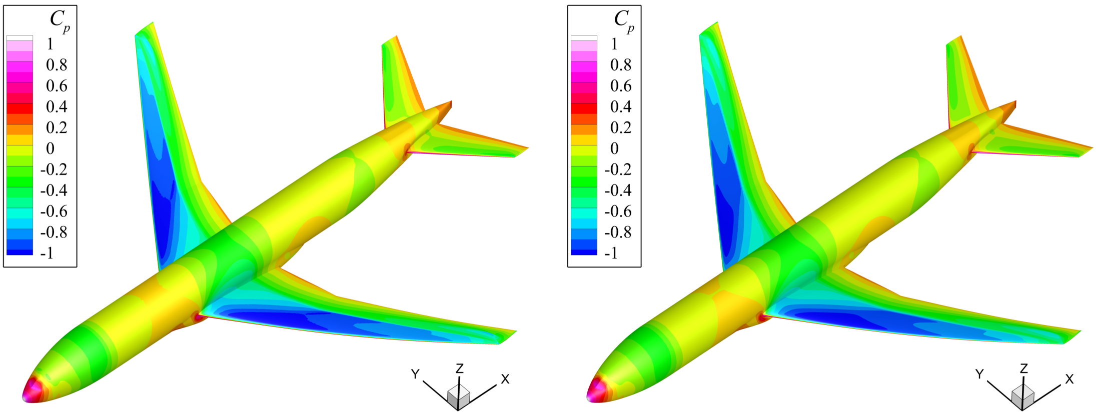


Figure 6.5: Time-step convergence of the drag coefficient ($\alpha = 2.94°$, medium grid).



(a) UTCart (medium grid)

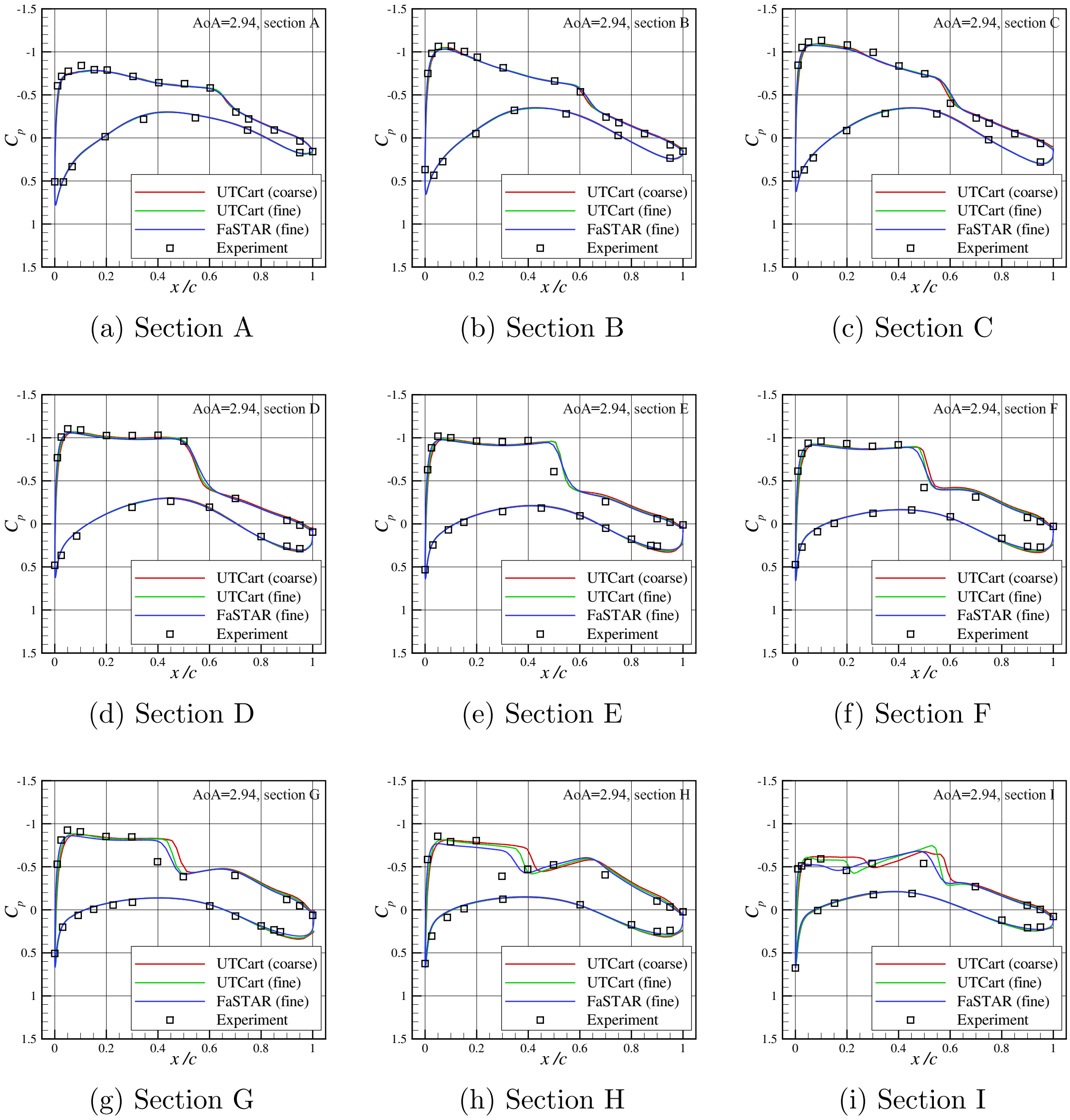
(b) FaSTAR (fine grid)

Figure 6.6: Surface pressure coefficient on the wing ($\alpha = 2.94°$).

Table 6.3: Richardson extrapolation of the drag coefficient excluding the estimated induced drag.

Solver	Grid pair	Extrapolated value
UTCart	Coarse–Medium	0.02154
	Medium–Fine	0.02091
FaSTAR	Medium–Fine	0.02130

compared with the experimental data, whereas the drag coefficient is overestimated. The UTCart and FaSTAR results share these discrepancies, and thus, they may be due to the



(a) Section A

(b) Section B

(c) Section C

(d) Section D

(e) Section E

(f) Section F

(g) Section G

(h) Section H

(i) Section I

Figure 6.7: Surface pressure coefficient on the wing sections ($\alpha = 2.94°$).

difference in the configuration between the experiment and the computation (e.g., the support sting).

Figures 6.14 and 6.15 show the surface pressure coefficient and skin friction coefficient at $\alpha = 4.65°$, respectively. The difference of the pressure on the upper surface between the UTCart result and the experimental data is more visible than that in the $\alpha = 2.94°$ case. This is because flow separation occurs near the wing-body junction and behind the shock.

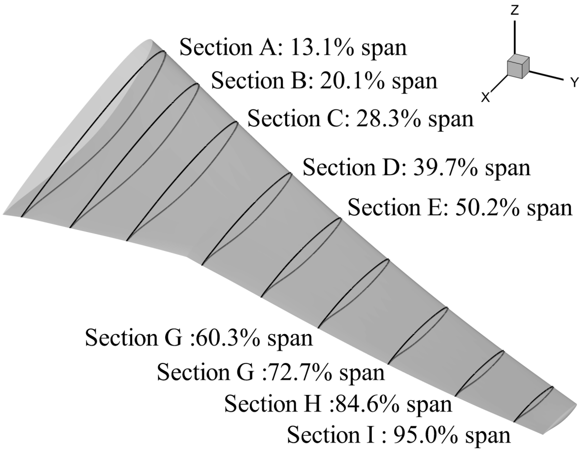


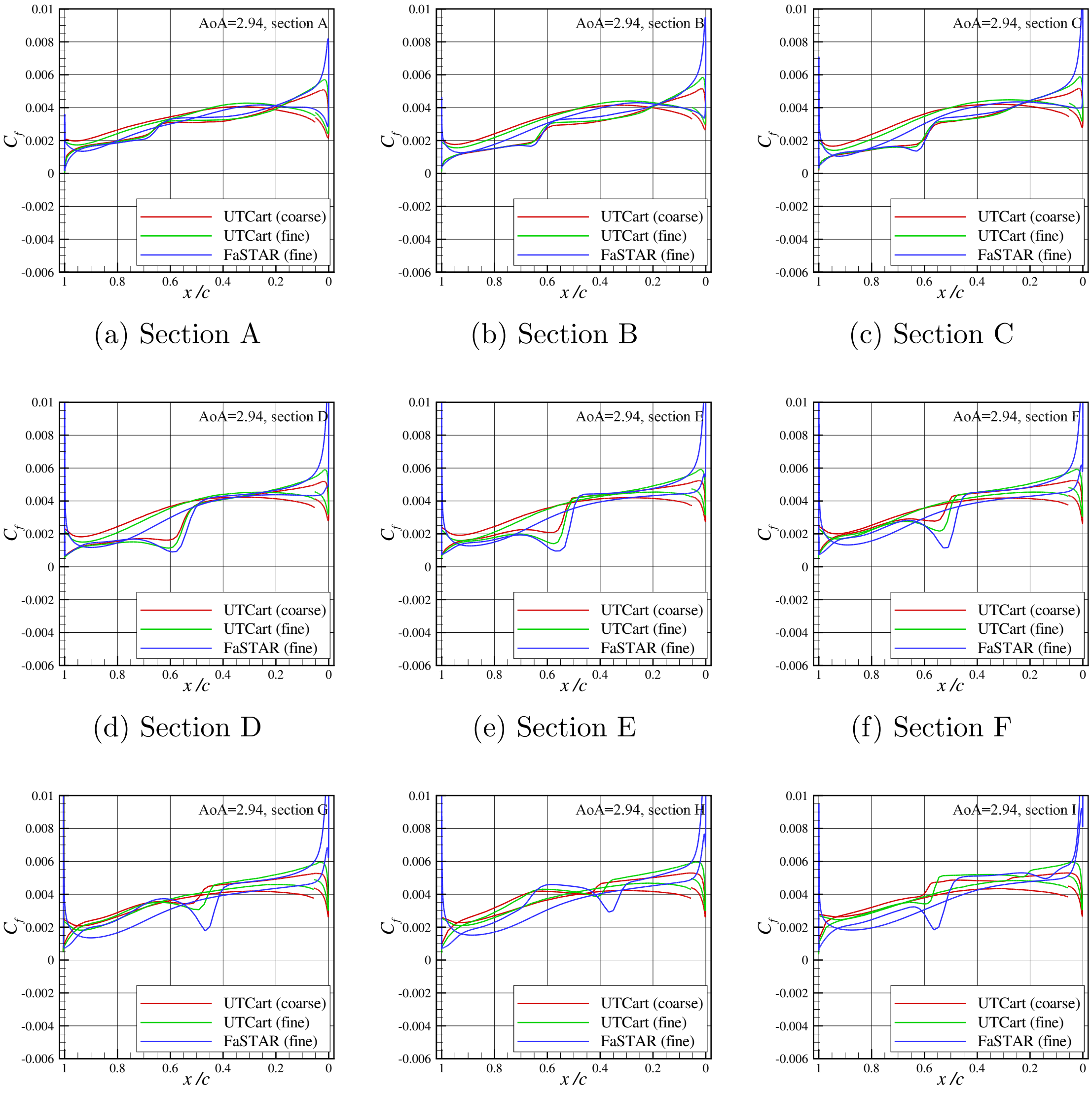
Figure 6.8: Definition of the wing sections of the NASA CRM.

The surface streamlines and distribution of the x-component of the skin friction coefficient, $c_{f,x}$, are shown in Fig. 6.16. The area with $c_{f,x} < 0$ (shown in blue in the figure) represents the region where flow separation occurs. The value in the region near the wing tip is higher in the FaSTAR result. This is because the grid resolution relative to the local chord length is small at the wing tip as stated in the grid sensitivity study. Although both flow solvers reproduce a similar pattern of the flow separation, the shape and area of the separated regions are different. The separated region at the mid-span in the UTCart result is slightly smaller than that in the FaSTAR result. In addition, the absolute value of the friction in the separated region is small in the UTCart result. This is because the wall function is used even in the separated region, and the boundary condition assumes the turbulent boundary layer attaches in the negative direction of the x-axis. Note that prediction of this flow separation is a continuing concern in the aerodynamic prediction of the NASA CRM [50–53,75,106] even on body-fitted grids. Thus, dependency on the turbulence models should also be examined in the future.

6.1.4 Discussion on the Accuracy and Applicability

Here, the results on the medium grid are discussed. The boundary layer thickness at the trailing edge is estimated to be 5.47 inches using the formula (5.17) with the mean aerodynamic chord (275.8 inches) and the Reynolds number for this length. Thus, the grid size on the upper surface of the wing is approximately one-twentieth of the boundary layer thickness at the trailing edge. This resolution is similar to the that of grid 1 for the study on the NACA 0012 in Section 5.4, and thus, there seems to be still room for grid convergence.

As shown in Fig. 6.9, the distributions of the skin friction at the inboard sections shows



(a) Section A

(b) Section B

(c) Section C

(d) Section D

(e) Section E

(f) Section F

(g) Section G

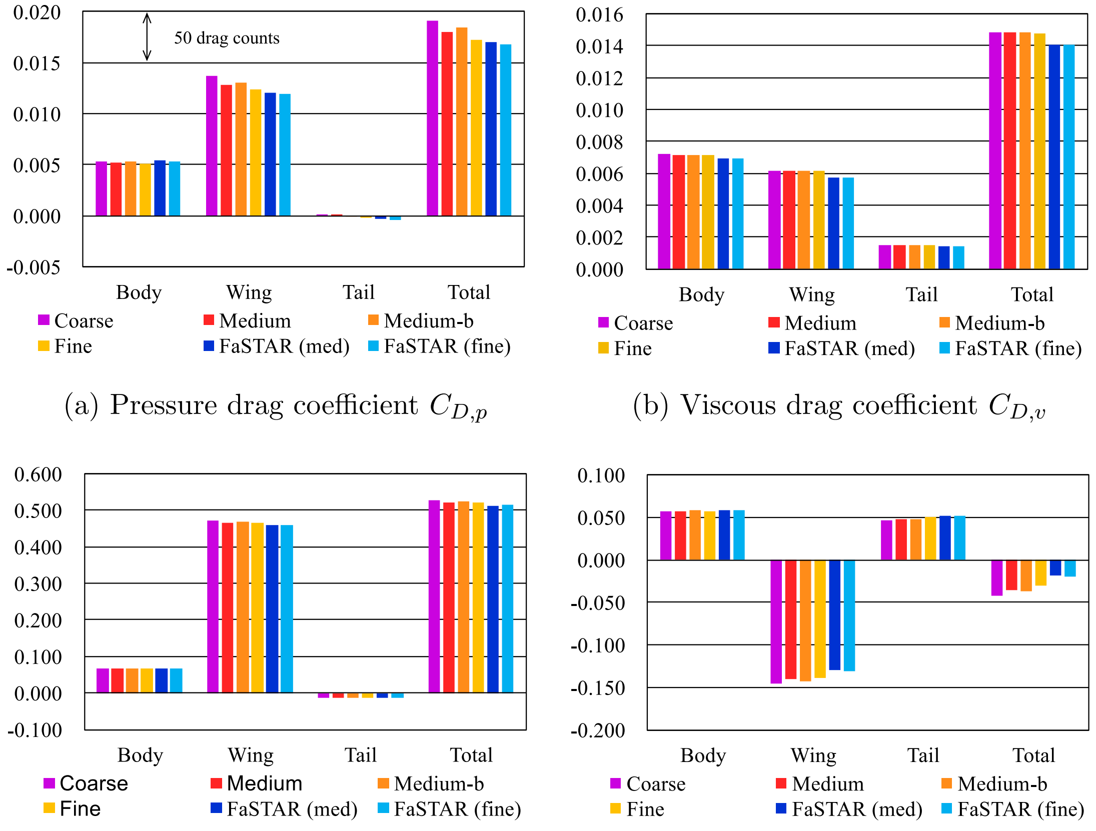
(h) Section H

(i) Section I

Figure 6.9: Skin friction coefficient on the wing sections ($\alpha = 2.94°$). The distribution on the upper surface can be recognized by its local minimum at the mid-chord.

better agreement with the reference result than those at the sections near the wing tip. At the wing root and tip, the local chord lengths are 465 and 108 inches, respectively. Thus, the estimated boundary layer thickness is 8.31 inches at the root trailing edge and is 2.58 inches at the tip trailing edge. As indicated in these values, the grid resolution is relatively low on the outer part of the wing, and the error seems to become large accordingly. For more accurate simulations, the grid size on the outer part of the wing should be smaller.

Even though the grid resolution should be higher to predict the skin friction more



(a) Pressure drag coefficient $C_{D,p}$

(b) Viscous drag coefficient $C_{D,v}$

(c) Lift coefficient C_L

(d) Pitching moment coefficient C_M

Figure 6.10: Comparison of the aerodynamic coefficients of the NASA CRM ($\alpha = 2.94°$).

accurately, the drag and lift coefficient could be predicted with 19 drag counts (8.9%) error from the FaSTAR result. The extrapolated value using the Coarse–Medium grid pair has 2.3 drag counts (1.1%) difference from the FaSTAR result, which is smaller than the standard deviation of the drag coefficient reported in the DPW-4 workshop (8.1 drag counts) [105]. Furthermore, the nonlinearity of the lift and pitching-moment coefficient could be predicted on the medium grid. For the simulation of the NASA CRM, the conventional CFD simulation on body-fitted grids has achieved good accuracy [106]. However, there is a premise that once a body-fitted well-tailored grid has been prepared. One case of the simulation by the proposed framework including the grid generation can be performed within a half days using 144 cores at the moment. It is shown that the proposed framework can also predict the flow with a certain accuracy, and thus, the time spent and manual work for a simulation of one case is reduced compared with the conventional simulation on the body-fitted grids.

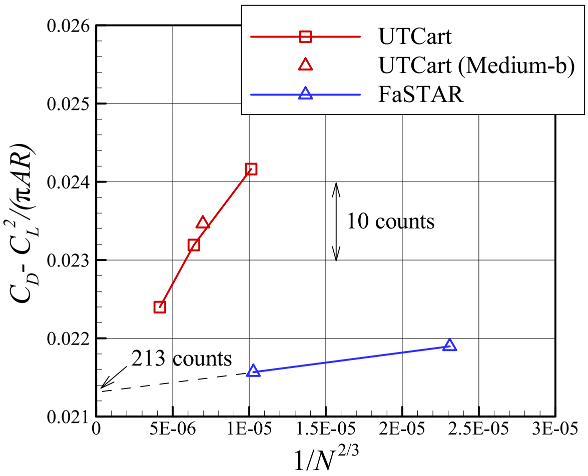


Figure 6.11: Grid convergence of the total drag coefficient excluding the estimated induced drag.

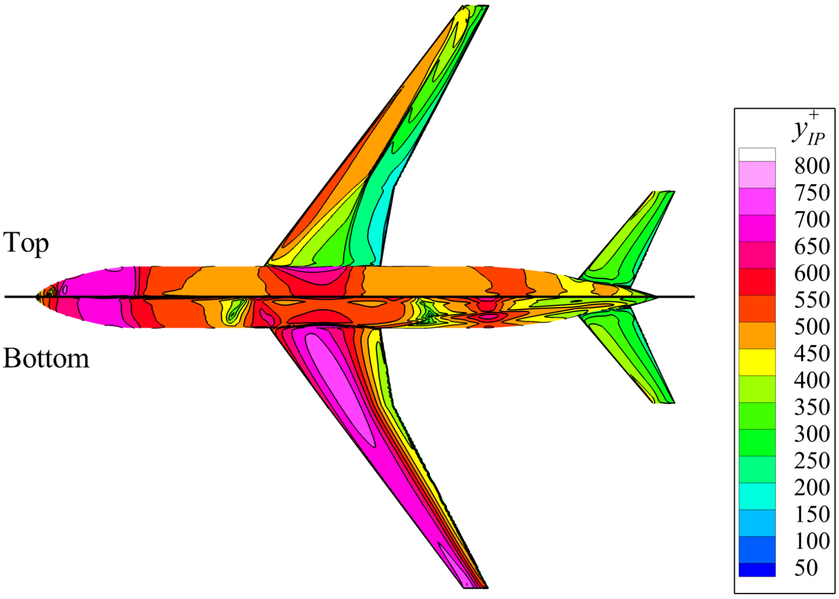
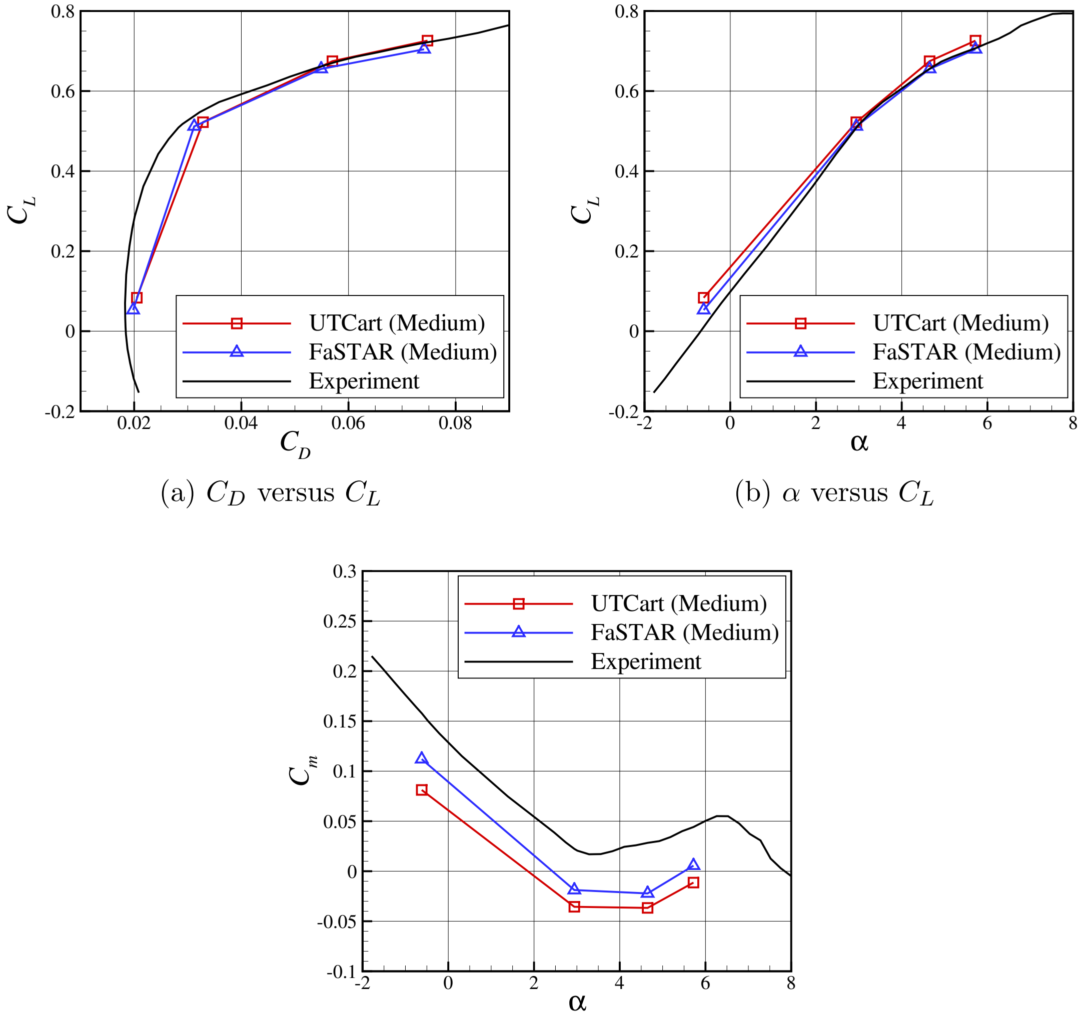


Figure 6.12: Distribution of y_{IP}^+ on the surface ($\alpha = 2.94°$, medium grid).

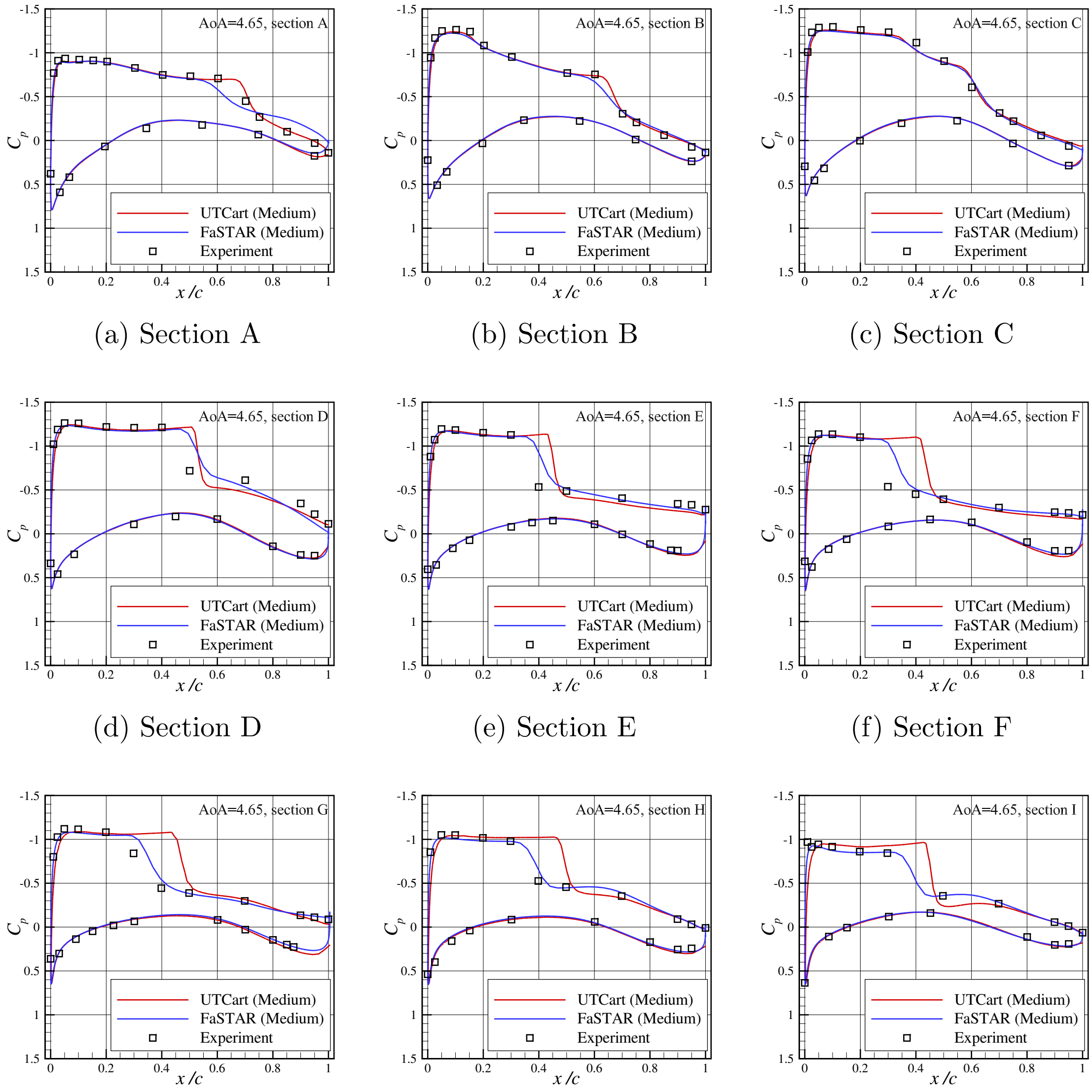


(a) C_D versus C_L

(b) α versus C_L

(c) α versus C_M

Figure 6.13: Aerodynamic coefficients of the NASA CRM at each angle of attack.



(a) Section A

(b) Section B

(c) Section C

(d) Section D

(e) Section E

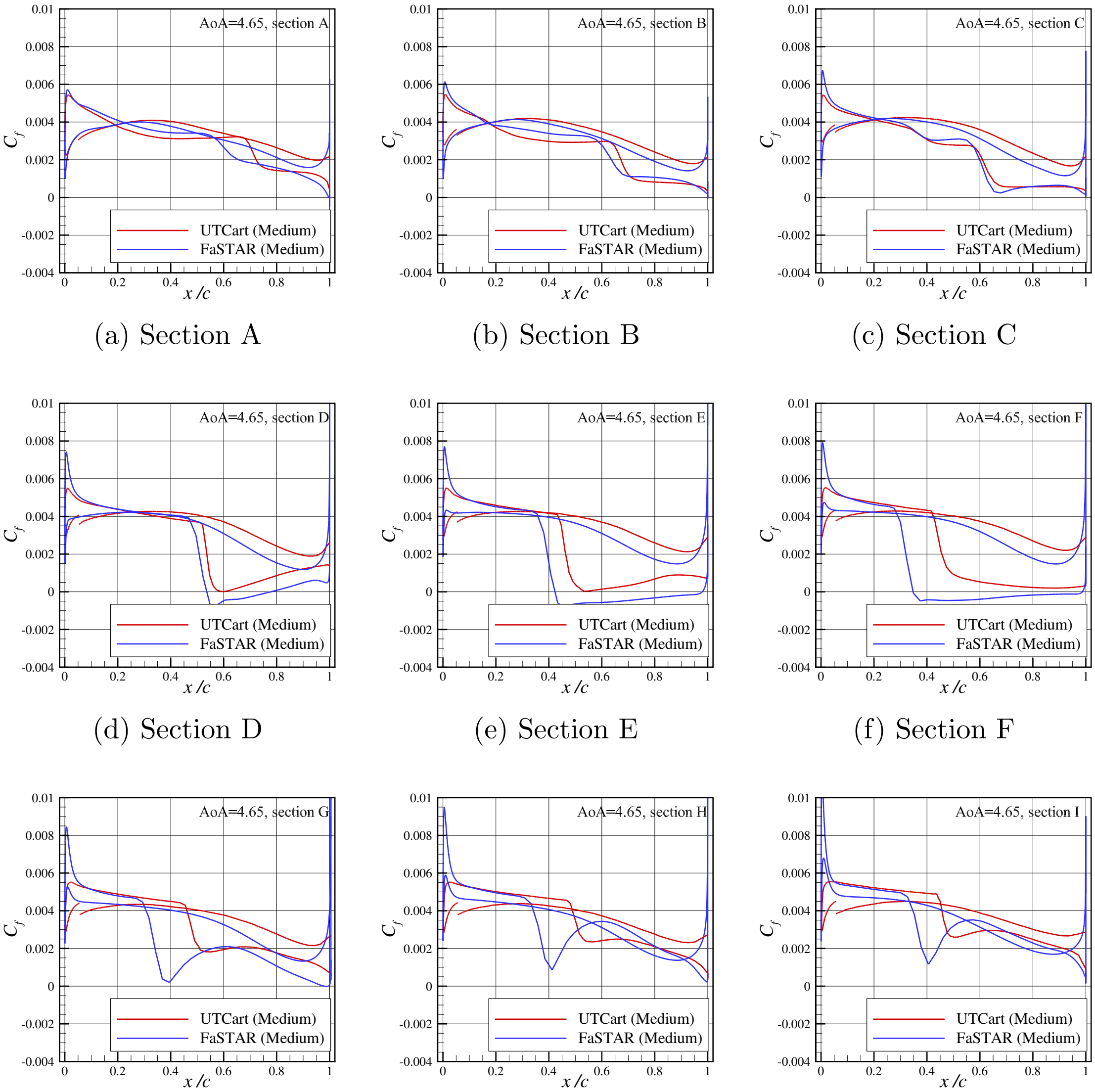
(f) Section F

(g) Section G

(h) Section H

(i) Section I

Figure 6.14: Surface pressure coefficient on the wing sections ($\alpha = 4.65°$).



(a) Section A

(b) Section B

(c) Section C

(d) Section D

(e) Section E

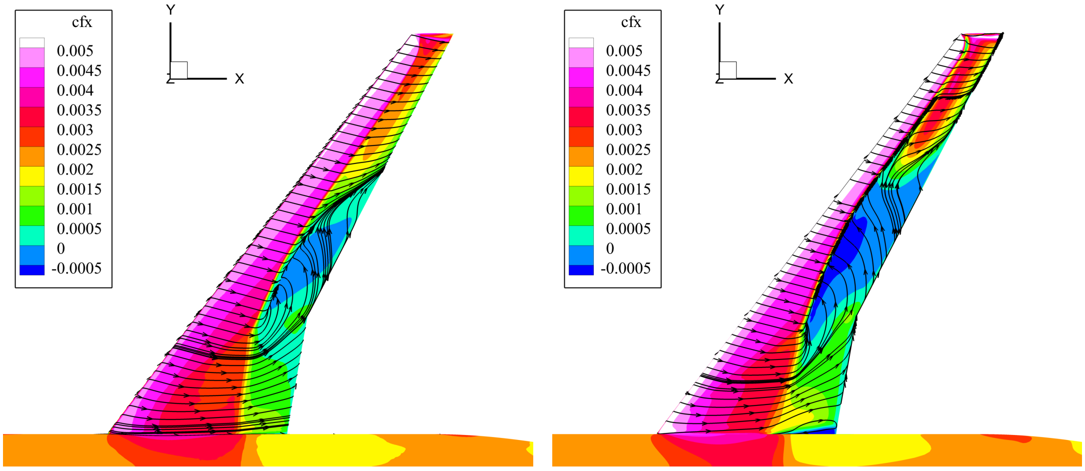
(f) Section F

(g) Section G

(h) Section H

(i) Section I

Figure 6.15: Skin friction coefficient on the wing sections ($\alpha = 4.65°$).



(a) UTCart (medium grid)

(b) FaSTAR (medium grid)

Figure 6.16: Surface streamlines and x-direction friction coefficienton the wing ($\alpha = 4.65°$).

6.2 Subsonic Flows around the JAXA Standard High-Lift Model

To demonstrate the capability of the proposed framework for flows simulations around a more complex geometry, flows around the JAXA standard high-lift model (JSM) are simulated. This geometry has been studied by wind-tunnel experiments [112] and CFD simulations [102, 113, 114]. CFD simulation of this flow is also adopted as a benchmark problem in 3rd AIAA CFD High Lift Prediction Workshop [115]. The geometry used in this computation consists of a fuselage, wings, slats and single-slotted flaps with a deflection angle of $I_F = 30°$. In addition, support brackets of the slats and flaps are also included. The mean aerodynamic chord c_{ref} and the half span are 529.9 and 2300 mm, respectively.

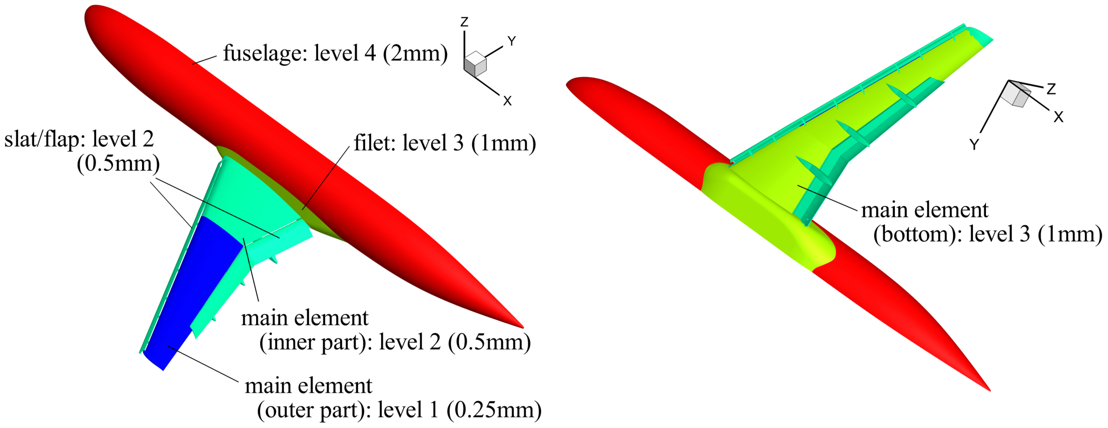
The setting of the computation is adjusted to the experiment in JAXA's 6.5m×5.5m low-speed wind tunnel facility. The Reynolds number for c_{ref} is 1.93×10^6; the free-stream Mach number is 0.172; the free-stream temperature is 306.55K. The simulated angles of attack are 4.36, 10.47, 14.54, 18.58, 20.59 and 21.57°. In the simulation, a symmetry boundary condition is specified on the $y = 0$ plane, and a half-span model is used. The calculated results are compared with the experimental data provided in the workshop.

6.2.1 Computational Grid and Methods

To reduce the computational cost while retaining acceptable computational accuracy, variable cell sizes are specified on the wall as shown in Fig. 6.17. The outboard part of the wing upper surface is covered with the finest level of the cells ($\Delta x = 0.25$ mm) because flow separation is expected to occur over the wing upper surface. The inboard part of the wing is covered by the second finest level of the cells ($\Delta x = 0.5$ mm). Consequently, the cell sizes on the wing upper surface are smaller than one-thousandth of the local chord. The slat and flap are also covered with the second finest level of the cells. The other part is covered by coarser levels of the grid to reduce the total number of the cells. Furthermore, local grid refinement with the finest level of the cells is specified at the gap between the flap and the main wing to resolve it. The number of the wall layer of the grid is set to two to reduce the total cell number. To assure the IPs are located within the wall layer, r_{IP} is set to two. The number of near-field and the far-field smooth layers are set to eight and three, respectively.

The grids are generated using a workstation (Xeon E5–2643 v3 at 3.4GHz) using one core. The time spent for the generation of the grid (114,466,104 cells) is approximately 38 minutes excluding the time for file I/O and setting the boundary conditions. The overview of the generated computational grid is shown in Fig. 6.18. The definition of the A–A section is as shown in Fig. 6.19. As shown in the figure, the small gap between the flap and wing is resolved by multiple cells. Figure 6.20 shows the calculated cut-off distance on slices around the E–E and N–N sections. The cut-off distance is calculated as intended, depending on the cell size assigned to the nearest wall.

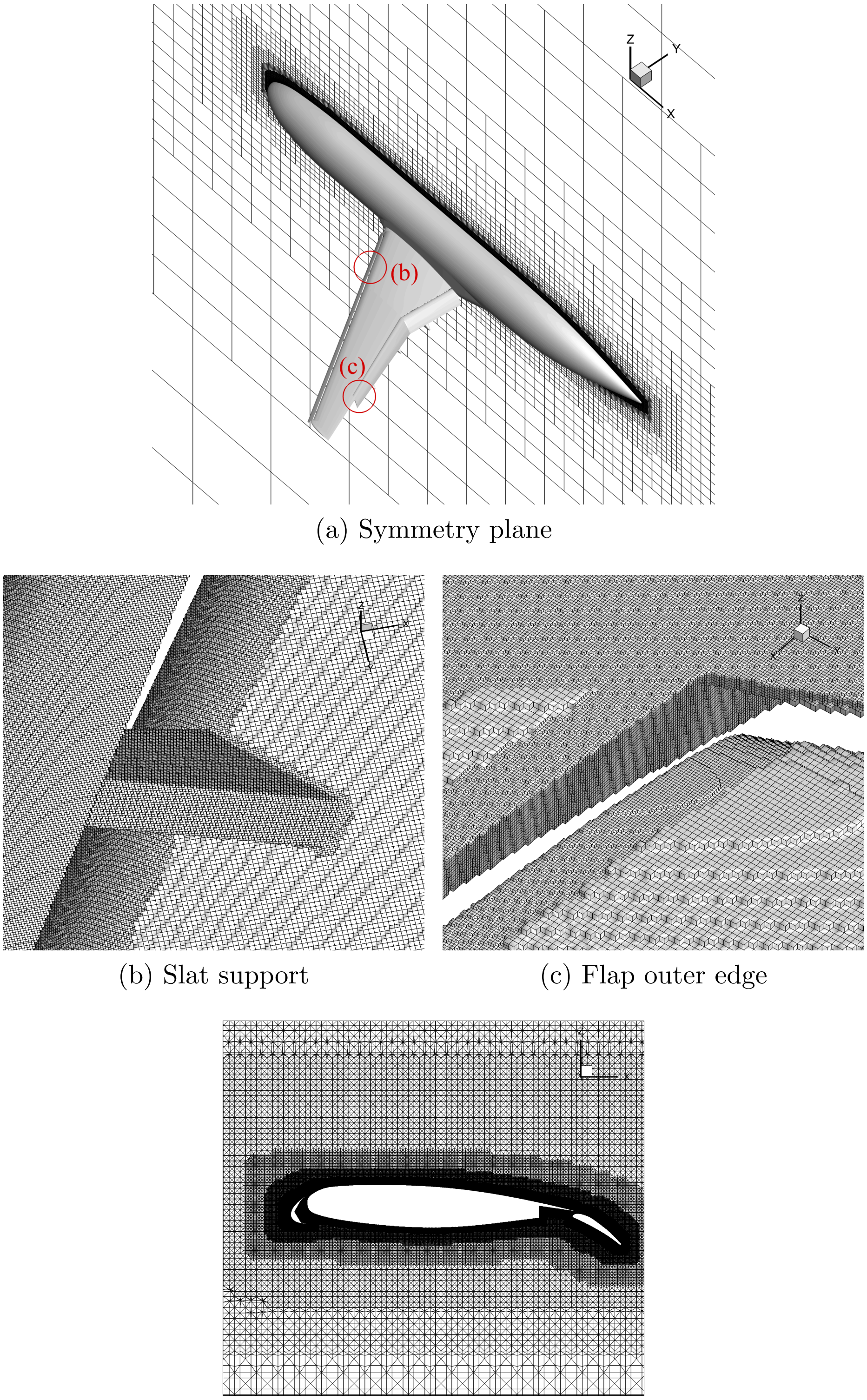
For the turbulence model, the Rotational correction is used with the SA turbulence model. Furthermore, a high-order scheme [81] is used for the inviscid term to reduce numerical viscosity. Note that no limiter or shock-capturing method is employed in this simulation. The flow computation is conducted on the Reedbush-U supercomputer using 144 cores.


fuselage: level 4 (2mm)
Z
Y
X
slat/flap: level 2 (0.5mm)
filet: level 3 (1mm)
main element (inner part): level 2 (0.5mm)
main element (outer part): level 1 (0.25mm)
Z
X
Y
main element (bottom): level 3 (1mm)

(a) Upper surface

(b) Lower surface

Figure 6.17: Setting of the minimum grid sizes on the JSM.



(a) Symmetry plane

(b) Slat support

(c) Flap outer edge

(d) A–A section

Figure 6.18: Computational grids around the JSM.

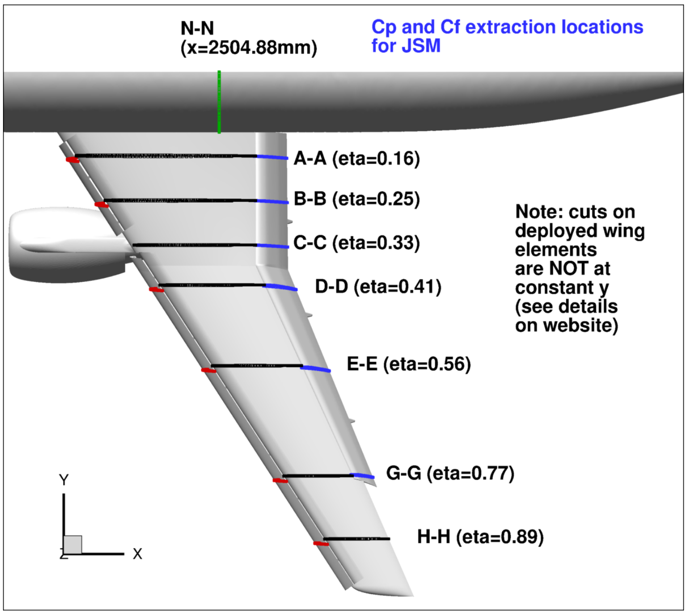
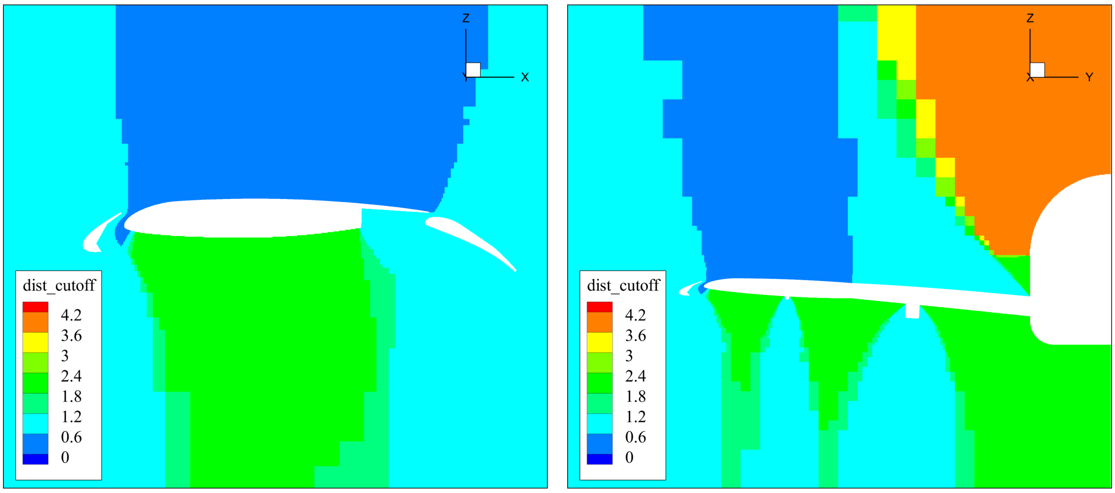


Figure 6.19: Definition of the sections on the wing of the JSM [115].



(a) Around E–E section

(b) Around N–N section

Figure 6.20: Calculated cut-off distance around the JSM.

6.2.2 Alpha-Sweep

Figure 6.21 shows the time-history of the lift coefficient at $\alpha = 14.54°$. Although oscillation remains, the lift coefficient has reached a quasi-steady solution. Thus, the aerodynamic forces are calculated by averaging the result during the last 2000 iterations. In addition, the root-mean-square values of the aerodynamic coefficients during the 2000 iterations are summarized in Table 6.4. The unsteadiness of the aerodynamic coefficient mostly increases as the angle of attack increases. The standard deviation of C_L is 0.34% of the averaged value of C_L at maximum. The following discussion is based on the assumption that this fluctuation remains.

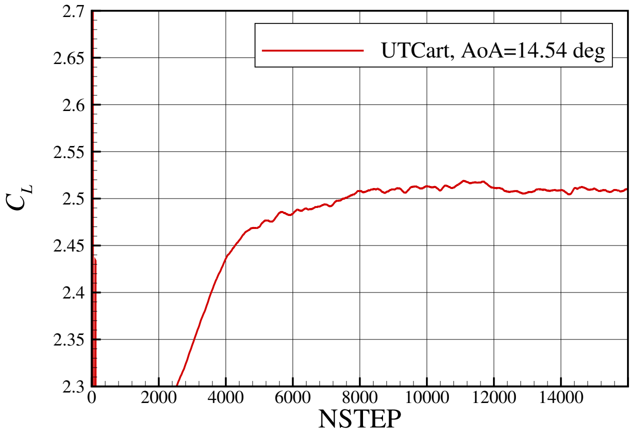


Figure 6.21: Convergence history of the lift coefficient at $\alpha = 14.54°$.

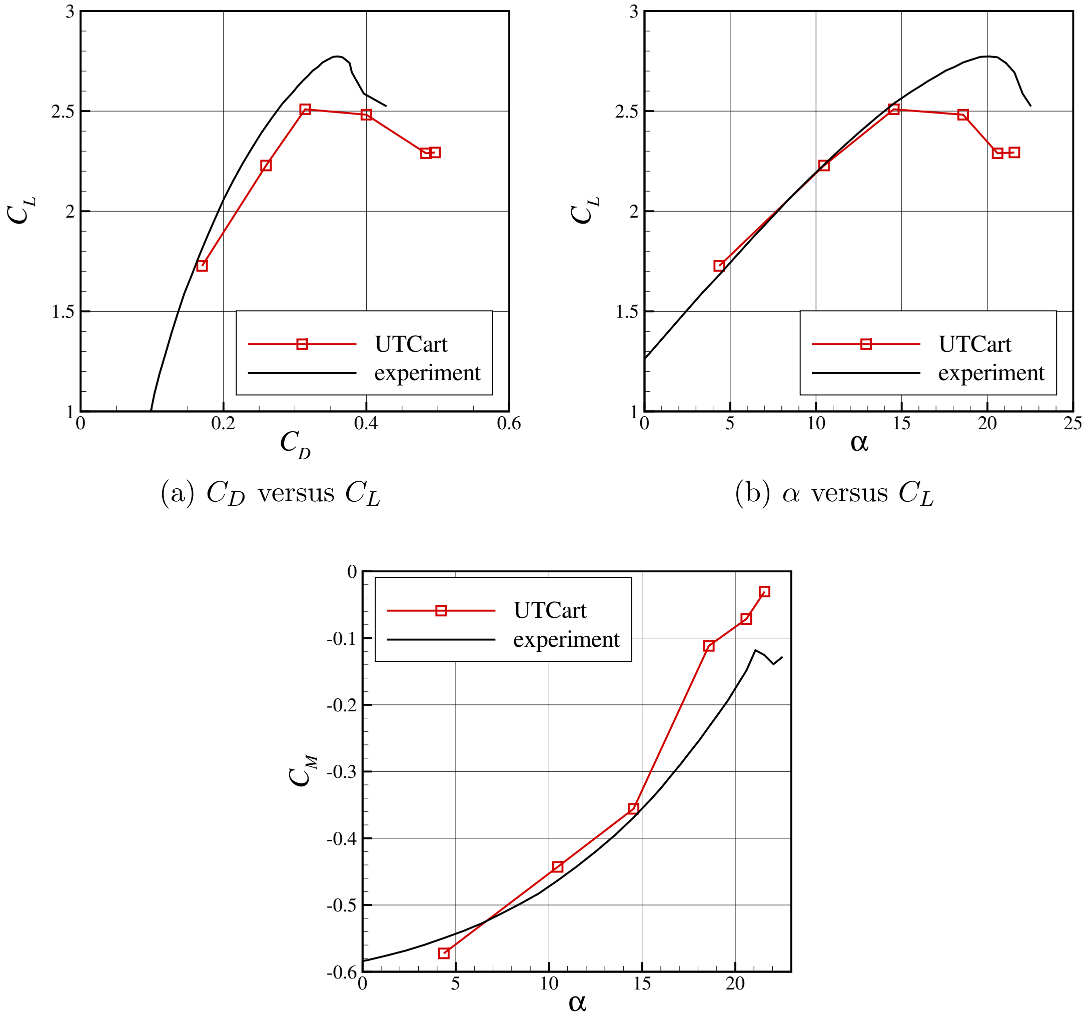
Table 6.4: Variation of the aerodynamic coefficients.

α	$C_{D,rms}$	$C_{L,rms}$	$C_{M,rms}$
4.36	0.00014	0.00072	0.00043
10.47	0.00018	0.00076	0.00035
14.54	0.00043	0.00090	0.00075
18.58	0.00117	0.00522	0.00333
20.59	0.00116	0.00399	0.00167
21.57	0.00201	0.00778	0.00306

Figure 6.22 shows the calculated aerodynamic coefficients. At the low angles of attack, the lift and pitching moment coefficients show good agreement with the experimental data.

In contrast, C_D is overestimated even at the low angles of attack, for which other computational results [102, 113, 114] show the similar trend. This discrepancy seems to be related to the wall interference of the wind tunnel as discussed in [102] or due to insufficient grid resolution.

The lift coefficient saturated at $\alpha = 18.58°$ and decreases at the higher angles of attack. Accordingly, the maximum C_L is approximately 10% lower than the experimental data. To see the flow more closely, the distributions of the surface pressure coefficient over the sections at $\alpha = 14.54°$ and $\alpha = 18.58°$ are compared in Figs. 6.23 and 6.24. At $\alpha = 14.54°$, the pressure distributions show a basically good agreement with the experimental data, although the suction peaks at the D–D, E–E, and G–G sections are underpredicted. This discrepancy may be depending on the lack of the grid resolution relative to the size of the flap. At $\alpha = 18.58°$, the distributions in the inboard sections are still predicted accurately. In contrast, the suction peak of the main wing is lost at the outboard sections. From the surface restricted streamline shown in Fig. 6.25, flow separation is observed in the downstream of the slat support. Figure 6.26 shows the surface flow pattern visualized by the oil-flow method in the experiment (the images are provided in [115]). A similar pattern of flow separation as that in the computation is observed near the outer-most slat support bracket at $\alpha = 18.58°$, and it is more obvious at $\alpha = 21.57°$. In the computation, the flow separation occurs also in the downstream of the inboard two support brackets, and consequently, the lift is underestimated at high angles of attack.

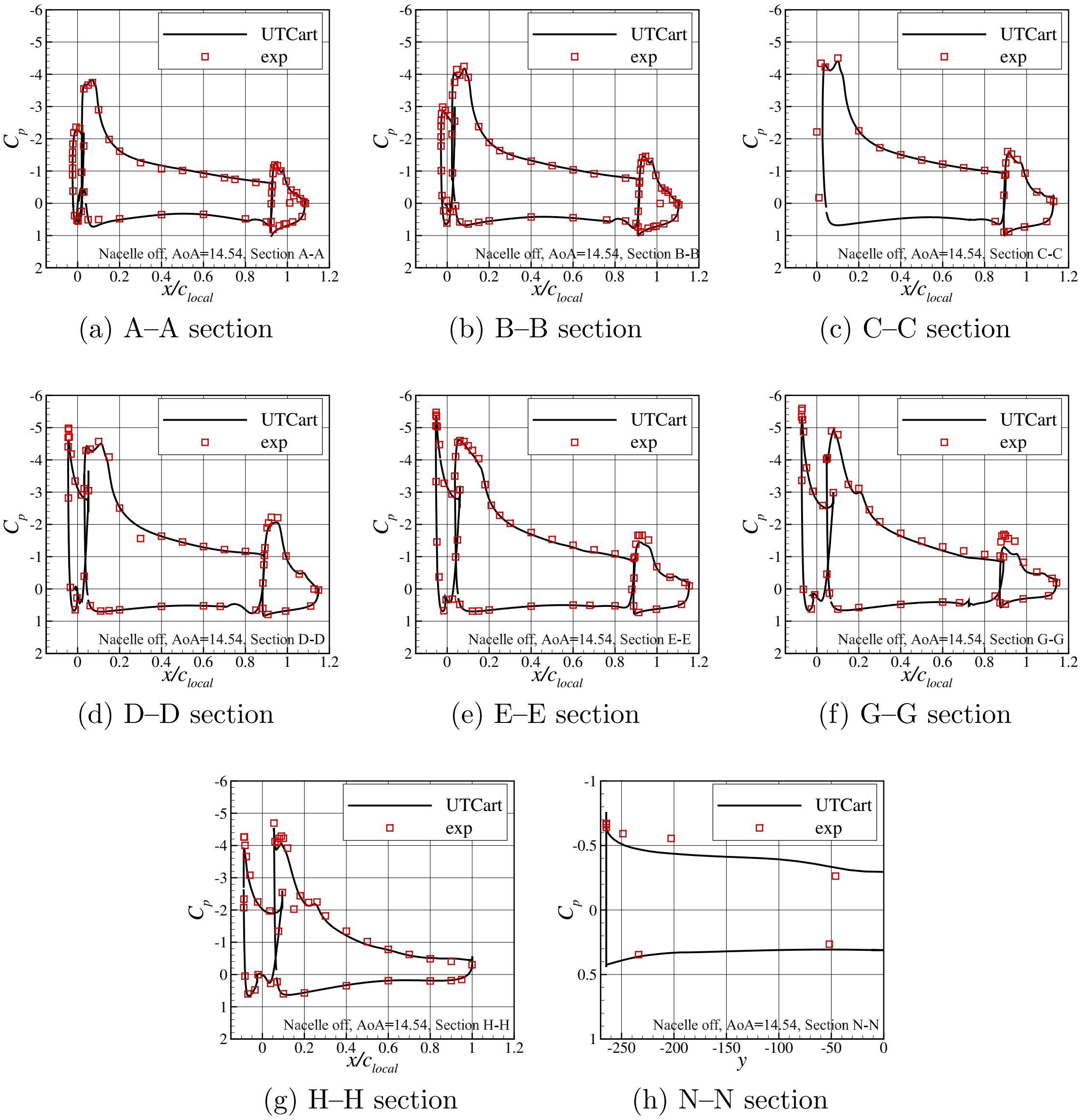


(a) C_D versus C_L

(b) α versus C_L

(c) α versus C_M

Figure 6.22: Aerodynamic coefficients of the JSM at each angle of attack.



(a) A–A section

(b) B–B section

(c) C–C section

(d) D–D section

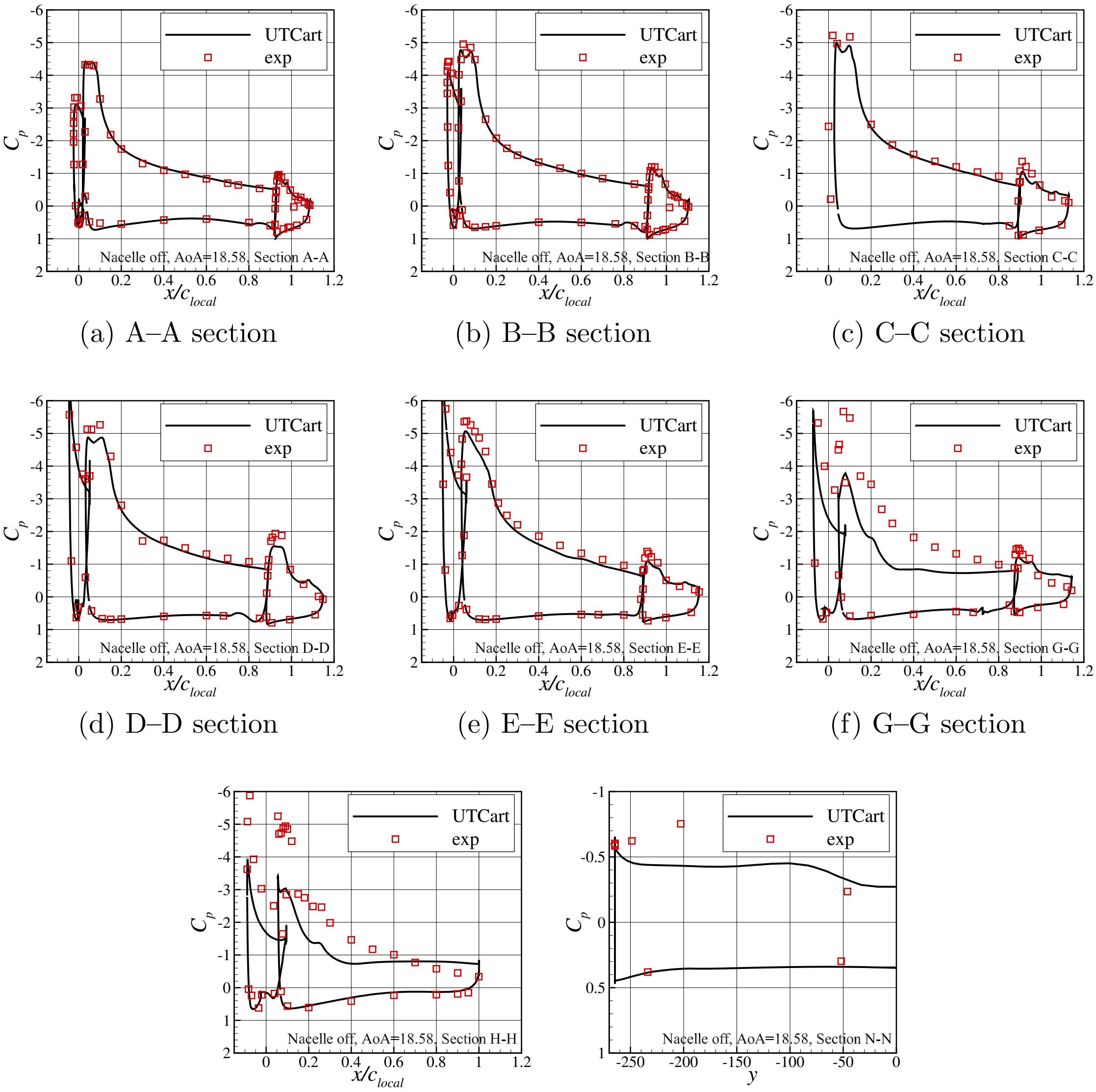
(e) E–E section

(f) G–G section

(g) H–H section

(h) N–N section

Figure 6.23: Surface pressure coefficient on the JSM at $\alpha = 14.54°$.



(a) A–A section (b) B–B section (c) C–C section

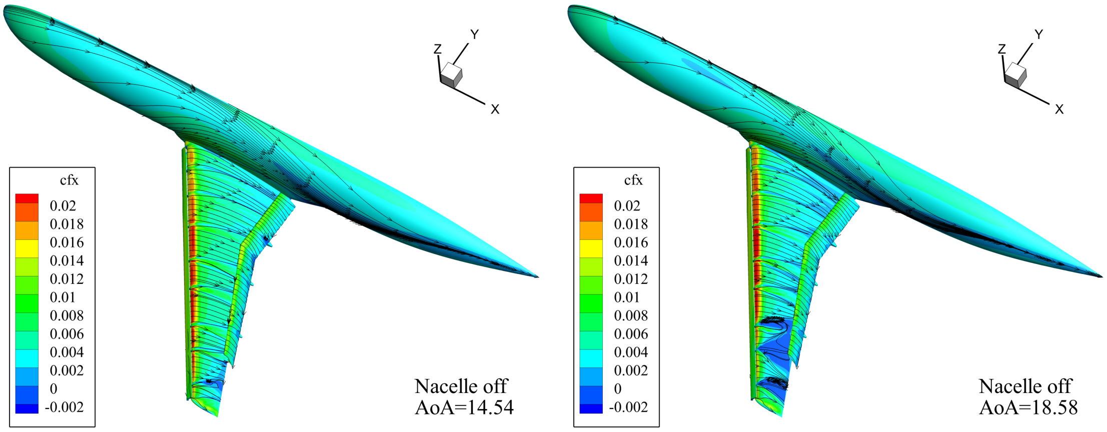
(d) D–D section (e) E–E section (f) G–G section

(g) H–H section (h) N–N section

Figure 6.24: Surface pressure coefficient on the JSM at $\alpha = 18.58°$.

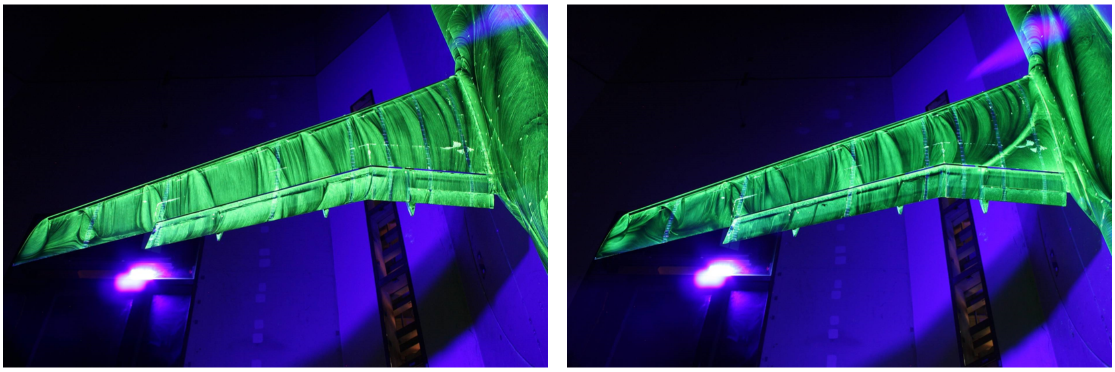
6.2.3 Sensitivity to the Grid and Spatial Accuracy

To detect the underlying problem in the simulation at the high angles of attack, sensitivities of the result to the grid and spatial accuracy of the inviscid term are investigated. Here, the flow at $\alpha = 18.58°$ is simulated on the same grid using the second-order accurate scheme for the inviscid term. In addition, the same flow is also simulated on a grid that has a 10% smaller minimum grid size (140,540,721 cells). Here, cases 1–3 denote the baseline



(a) $\alpha = 14.54°$ (b) $\alpha = 18.58°$

Figure 6.25: Comparison of surface restricted streamlines and x-direction skin friction coefficient on the JSM.



(a) $\alpha = 18.58°$ (b) $\alpha = 21.57°$

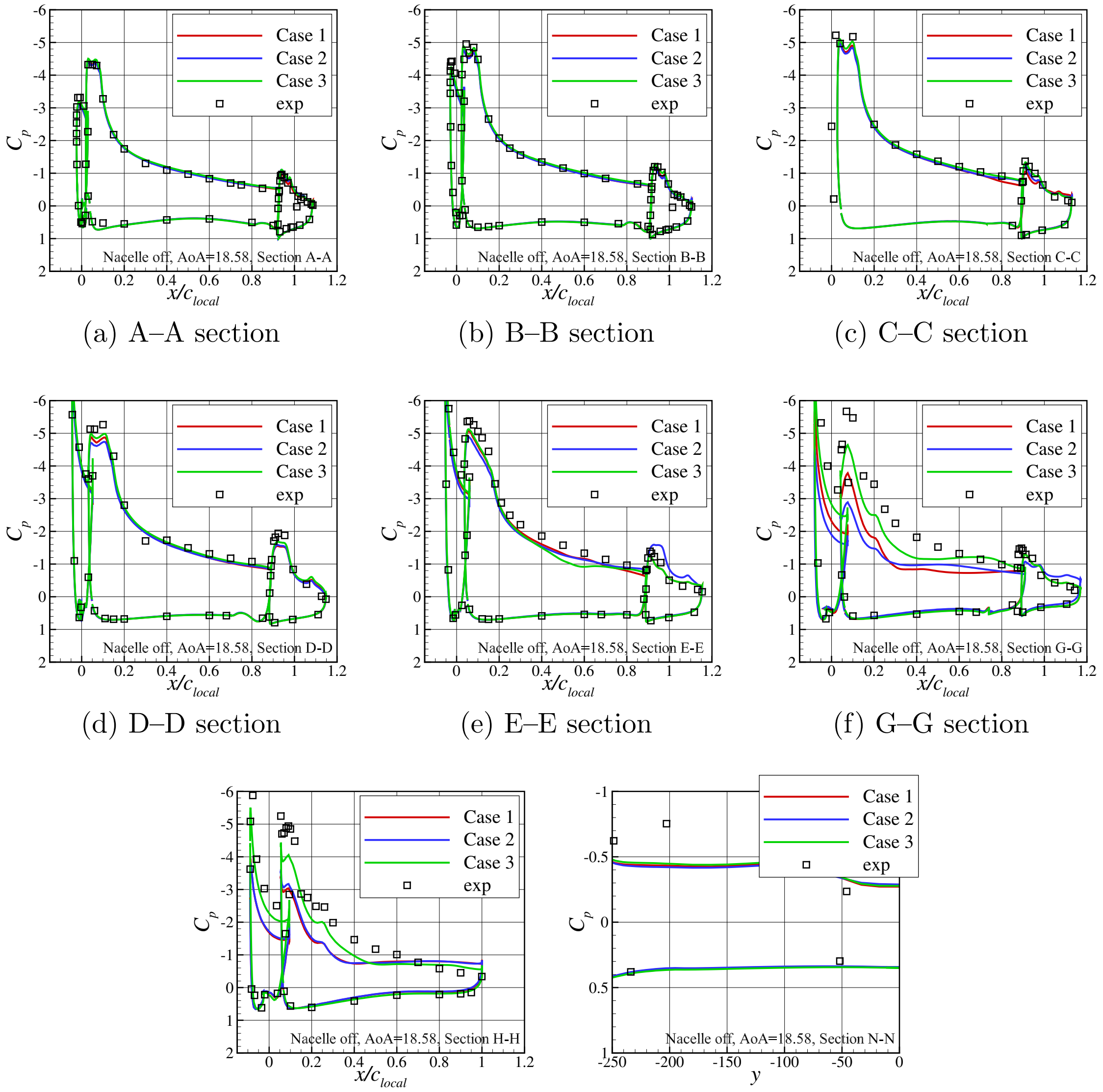
Figure 6.26: Surface flow pattern on the JSM in the wind-tunnel experiment [115].

calculation, the calculation using the second-order accurate scheme, and the calculation on the finer grid, respectively (Table 6.5).

The calculated values of C_L are 2.482, 2.411 and 2.518 in cases 1–3, respectively. The differences between the results are larger than the root-mean-square values represented in Table 6.4, and thus, they are not negligible. Furthermore, the surface pressure coefficient is compared in Fig. 6.27. As indicated at G–G and H–H section, the suction peak at the leading edge of the wing is weaker in case 2 and stronger in case 3. Figure 6.28 shows the surface streamlines. The area of the separated region is larger in case 2 and smaller in case 3. This flow separation leads to the loss of the suction peak and seems to be one of the causes of the difference in C_L.

Table 6.5: Specification of the cases for the grid sensitivity study.

	Case 1 (baseline)	Case 2	Case 3
Spatial accuracy (inviscid term)	fourth	second	fourth
Cell number	114,466,104		140,540,721



(a) A–A section

(b) B–B section

(c) C–C section

(d) D–D section

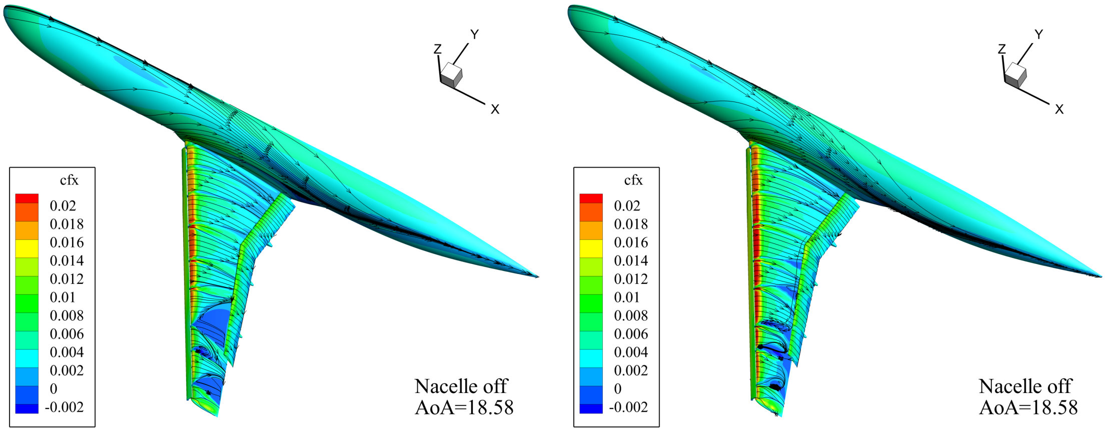
(e) E–E section

(f) G–G section

(g) H–H section

(h) N–N section

Figure 6.27: Comparison of the surface pressure coefficient on the JSM at $\alpha = 18.58°$ in case 1–3.



(a) Case 2

(b) Case 3

Figure 6.28: Surface streamlines and the x-direction skin friction coefficient on the JSM at $\alpha = 18.58°$.

6.2.4 Discussion

In this simulation, the flow-fields at $\alpha = 4.36°$–$14.54°$ were predicted within 2.7% error, and capability of the proposed framework for a complex flow was demonstrated. The time spent for the grid generation for this case is shorter than that for the flow calculation, although some pre-processing work is required to seek a proper grid setting (e.g., detecting a small gap and designating local refinement there). In addition, once the grid specification is determined, the simulation of similar geometries can be automatic. At these angles, therefore some further application may be possible; e.g., investigation on the drag increment by the nacelle, or optimization of the flap relative location to the wing.

In contrast, the prediction of the maximum C_L and the stall angle is difficult at the moment. As discussed in Section 5.4, prediction of the separation is sensitive to the grid resolution. The grid size on the upper surface is approximately one-thousandth of the local chord. This grid resolution is similar to that of grid 1 in the NACA 0012 cases. In the NACA 0012 cases, the flow separation is overestimated in a coarse grid case, and the prediction accuracy improves using the finer grids. Thus, prediction of the stall may improve also in this simulation if the cell size is set to a smaller value. This trend is also indicated in the study on the sensitivity to the grid the grid and spatial accuracy. The use of the fourth-order accurate scheme virtually enhances the grid resolution by reducing the numerical dissipation compared with the second-order accurate scheme. The results of case 2 and 3 indicate the grid resolution is not sufficient at the high angles of attack, and the prediction accuracy may improve by using a finer grid. Note that prediction of the flow separation is also depending on the turbulence model itself, and the improvement of the turbulence model should be attempted in a different context.

6.3 Summary of Chapter 6

In this chapter, the capability and efficiency of the proposed framework for 3D flow simulations around aircraft were demonstrated. In the simulation of transonic flows around the NASA CRM, the medium grid (approximately 62 million cells) was generated within 16 minutes using one core of Xeon E5–2643 v3. On the medium grid, y^+ on the wing upper surface is less than 600, and more than a thousand cells are located within the mean aerodynamic chord. First, grid sensitivity at $\alpha = 2.94°$ was investigated. The trend of each component-wise aerodynamic coefficient shows a correct trend of grid convergence toward the values computed on a conventional body-fitted grid, although the drag coefficient was slightly overestimated. Compared with the grid converged value, the drag coefficient excluding the estimated induced drag differs by 19 drag counts (8.9%) and 11 drag counts (5.1%) for the medium and the fine grids (approximately 117 million cells), respectively. The Richardson extrapolation of the drag coefficient using the coarse-medium grid pair predicts the drag coefficient with 2.3 counts (1.1%) error. Then, the flows at 7 different angles of attack from -1.79 to 5.72° were simulated. At the high angle of attack, the qualitative flow features including flow separation behind the shock show fair agreement with the computational results on the body-fitted grids. In addition, the nonlinearity of the lift coefficient at high angles of attack observed in the wind-tunnel experiment can be reproduced.

In the simulation of the JSM, the capability of the proposed framework for predicting flows around complex geometries was demonstrated. The grid (approximately 110 million cells) was generated within 38 minutes using one core of Xeon E5–2643 v3. The surface features, including the small gaps between the wing and the flap, are well resolved in the generated grid. In the result, the lift coefficients at $\alpha = 4.36°–14.54°$ are predicted within 2.7% range compared with the experimental data. In addition, the surface pressure coefficient at low angles of attack shows a good agreement with the reference experimental data. In contrast, accurate prediction of the flow separation and maximum C_L is difficult at the moment, which may require more cells and/or a different turbulence model.

Chapter 7

Concluding Remarks

In this study, we explored simulation methodology for high-Reynolds-number flow simulations using hierarchical Cartesian grids and an IBM. In the development of the method, we obtained the following conclusions:

Wall Boundary Conditions for Turbulent Flows (Chapter 3) To reduce the computational cost, the wall function, i.e., the model of the near-wall part of the turbulent boundary layer, was combined with the IBM. In the implementation of the wall function, the velocity profile must be modified so that it is linear to avoid numerical problems. We also showed that the modification for the eddy-viscosity is essential because the balance of the shear stress near the wall must be retained.

Evaluation of Aerodynamic Force Acting on Immersed Bodies (Chapter 4) The object surface is immersed in the Cartesian grid, and thus, uncertainty existed in the evaluation of the aerodynamic force acting on the object. We clarified the relation between the aerodynamic force and the numerical flux used in the flow calculation. The test problems showed that this method is more accurate than the conventional calculation method based on extrapolation onto the input CAD surface.

Subsequently, we gained the following knowledge from the validation study of the proposed methods:

Accuracy and Applicability of the Proposed IBM (Chapter 5) In the 2D validation problems, the IBM using the proposed profile modifications always achieved better accuracy than the baseline IBM in predicting the skin friction and aerodynamic coefficients. This is because the balance of the shear stress near the wall is correctly retained in the modified IBM. In these studies, the criterion for the grid setting was also considered. Based on the collected data, when the cell size is smaller than one-fiftieth of the local boundary layer thickness, the error of the skin friction on the airfoil is expected to be less than 5% of the correct value.

Capability of the Methods in Aerodynamic Prediction of Aircraft (Chapter 6) In the flow simulations around an aircraft presented in this study, the total number of cells used varied between 30 and 110 million. The computational grid with this grid resolution was generated automatically and rapidly, and complex features such as a small gap could be reproduced on the grid. For the cruise condition of a transonic aircraft such as the NASA CRM, the prediction accuracy of the drag coefficient could be within 10% using computational grids with 60 million cells. Using the Richardson extrapolation with a grid pair, the accuracy can be further improved. For a more complex geometry (e.g., a high-lift device), basic flow features, including the surface pressure and aerodynamic coefficients at low angles of attack, can be predicted within a short time using the proposed framework. The proposed framework can therefore be used to estimate the basic flow feature around a complex geometry within a short time.

Although the accuracy of the conventional CFD simulation may be better once a well-tailored body-fitted grid is prepared, the proposed framework can also predict the flow with a certain degree of accuracy. The grid generation is fully automatic, and thus, the total workload for the flow simulations is reduced compared to the conventional simulation on body-fitted grids. In addition, shape optimization problems can be conducted without any manual procedure during the sequence of calculations. Therefore, the proposed framework will be beneficial as a tool for aerodynamic design.

To conclude, the possible future research options to extend this study are stated.

Simulation using Different Turbulence Models At the moment, the proposed wall boundary conditions are formulated only for the SA turbulence model. For other turbulence models that use the Boussinesq approximation of eddy viscosity, a similar formulation is possible by introducing an additional model function to modify the eddy viscosity so that it is constant near a wall boundary. However, some turbulence variables (e.g., ω in the $k-\omega$ turbulence model [87]) diverge at the wall, and thus, additional treatments for the variable are necessary. This topic remains as a future research option. In addition, wall modeled LESs can be reconsidered following a similar story.

Improvement in the Wall Model Another possibility for future study is improvement in the wall model. The wall model used in this study neglects the effect of non-equilibrium (i.e., the pressure gradient or the advection in the boundary layer). As reported in [27], considering the effect of non-equilibrium improves the prediction accuracy in some cases such as flows with separation. To include this effect, however, the balance of shear stress must be reconsidered, which is not straightforward. The qualitative features of flows with separation have already been successfully predicted in some cases of this study using the current method. Improvement of the wall model may enhance the prediction accuracy as well as the applicability of this framework to more complex flows.

Bibliography

[1] T. Ishihara, K. Morishita, M. Yokokawa, A. Uno, Y. Kaneda, Energy spectrum in high-resolution direct numerical simulations of turbulence, Physical Review Fluids 1 (8) (2016) 082403. doi:10.1103/PhysRevFluids.1.082403.

[2] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, D. Mavriplis, CFD vision 2030 study: A path to revolutionary computational aerosciences, Tech. rep., National Aeronautics and Space Administration, NASA/CR-2014-218178 (2011).

[3] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, in: Proceedings of the 27th Aerospace Sciences Meeting, 1989, AIAA Paper No. 89-0366. doi:10.2514/6.1989-366.

[4] R. Löhner, Automatic unstructured grid generators, Finite Elements in Analysis and Design 25 (1) (1997) 111–134, adaptive Meshing, Part 1. doi:https://doi.org/10.1016/S0168-874X(96)00038-8.

[5] D. Mavriplis, Unstructured grid techniques, Annual Review of Fluid Mechanics 29 (1) (1997) 473–514.

[6] M. J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of Computational Physics 53 (3) (1984) 484–512. doi:10.1016/0021-9991(84)90073-1.

[7] M. J. Berger, R. J. Leveque, An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries, in: Proceedings of the nineth Computational Fluid Dynamics Conference, 1989, AIAA Paper No. 89-1930. doi:10.2514/6.1989-1930.

[8] D. De Zeeuw, K. G. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, Journal of Computational Physics 104 (1) (1993) 56–68. doi:10.1006/jcph.1993.1007.

[9] W. J. Coirier, K. G. Powell, Solution-adaptive Cartesian cell approach for viscous and inviscid flows, AIAA Journal 34 (5). doi:10.2514/3.13171.

[10] K. Nakahashi, F. Togashi, D. Sharov, Intergrid-boundary definition method for overset unstructured grid approach, AIAA Journal 38 (11) (2000) 2077–2084. doi:10.2514/2.869.

[11] D. Hartmann, M. Meinke, W. Schröder, An adaptive multilevel multigrid formulation for Cartesian hierarchical grid methods, Computers & Fluids 37 (9) (2008) 1103–1125. doi:10.1016/j.compfluid.2007.06.007.

[12] T. Ishida, S. Takahashi, K. Nakahashi, Efficient and robust Cartesian mesh generation for building-cube method, Journal of Computational Science and Technology 2 (4) (2008) 435–446. doi:/10.1299/jcst.2.435.

[13] R. Keppens, M. Nool, G. Tóth, J. Goedbloed, Adaptive mesh refinement for conservative systems: multi-dimensional efficiency evaluation, Computer Physics Communications 153 (3) (2003) 317–339. doi:10.1016/S0010-4655(03)00139-5.

[14] A. M. Wissink, R. L. Meakin, On parallel implementations of dynamic overset grid methods, in: Supercomputing, ACM/IEEE 1997 Conference, IEEE, 1997, pp. 15–15. doi:10.1145/509593.509608.

[15] S. E. Rogers, N. E. Suhs, W. E. Dietz, PEGASUS 5: an automated preprocessor for overset-grid computational fluid dynamics, AIAA journal 41 (6) (2003) 1037–1045. doi:10.2514/2.2070.

[16] R. W. Noack, D. A. Boger, R. F. Kunz, P. M. Carrica, SUGGAR++: An improved general overset grid assembly capability, in: Proceedings of the 19th AIAA Computational Fluid Dynamics Conference, 2009, AIAA Paper No. 2009-3992. doi:10.2514/6.2009-3992.

[17] A. Katz, A. M. Wissink, V. Sankaran, R. L. Meakin, W. M. Chan, Application of strand meshes to complex aerodynamic flow fields, Journal of Computational Physics 230 (17) (2011) 6512–6530. doi:10.1016/j.jcp.2011.04.036.

[18] S. L. Karman Jr, A 3D unstructured Cartesian/prismatic grid CFD code for complex geometries, in: Proceedings of the 33rd Aerospace Science Meeting, 1995, AIAA Paper No. 95-0343. doi:10.2514/6.1995-343.

[19] Z. Wang, A quadtree-based adaptive Cartesian/quad grid flow solver for Navier–Stokes equations, Computers & Fluids 27 (4) (1998) 529–549.

[20] M. Delanaye, M. Aftosimis, M. Berger, Y. Liu, T. Pulliam, Automatic hybrid-Cartesian grid generation for high-Reynolds number flows around complex geometries, in: Proceedings of the 27th Aerospace Sciences Meeting, 1999, AIAA Paper No. 99-0777. doi:10.2514/6.1999-0777.

[21] P. Lahur, A hybrid grid generation using unstructured prismatic and Cartesian grids, in: 41st AIAA Aerospace Sciences Meeting, 2003, AIAA Paper No. 2003-804. doi:10.2514/6.2003-804.

[22] D. Clarke, H. Hassan, M. Salas, Euler calculations for multielement airfoils using Cartesian grids, AIAA Journal 24 (1986) 353–358. doi:10.2514/3.9273.

[23] M. J. Aftosmis, M. J. Berger, J. E. Melton, Robust and efficient Cartesian mesh generation for component-based geometry, AIAA Journal 36 (6) (1998) 952–960. doi:10.2514/2.464.

[24] D. Hartmann, M. Meinke, W. Schröder, A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids, Computer Methods in Applied Mechanics and Engineering 200 (9) (2011) 1038–1052. doi:10.1016/j.cma.2010.05.015.

[25] M. Berger, M. J. Aftosmis, S. R. Allmaras, Progress towards a Cartesian cut-cell method for viscous compressible flow, in: Proceedings of the 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2012, AIAA Paper No. 2012-1301. doi:10.2514/6.2012-1301.

[26] M. Harada, Y. Tamaki, Y. Takahashi, T. Imamura, Simple and robust cut-cell method for high-Reynolds-number-flow simulation on Cartesian grids, AIAA Journal 55 (8) (2017) 2833–2841. doi:10.2514/1.J055343.

[27] M. J. Berger, M. J. Aftosmis, An ODE-based wall model for turbulent flow simulations, AIAA Journal(article in advance). doi:10.2514/1.J056151.

[28] C. S. Peskin, Flow patterns around heart valves: A numerical method, Journal of Computational Physics 10 (2) (1972) 252–271. doi:10.1016/0021-9991(72)90065-4.

[29] R. P. Beyer, A computational model of the cochlea using the immersed boundary method, Journal of Computational Physics 98 (1) (1992) 145–162. doi:10.1016/0021-9991(92)90180-7.

[30] R. P. Beyer, R. J. LeVeque, Analysis of a one-dimensional model for the immersed boundary method, SIAM Journal on Numerical Analysis 29 (2) (1992) 332–364. doi:10.1137/0729022.

[31] L. Zhu, C. S. Peskin, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, Journal of Computational Physics 179 (2) (2002) 452–468. doi:10.1006/jcph.2002.7066.

[32] Y.-H. Tseng, J. H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, Journal of Computational Physics 192 (2) (2003) 593–623. doi:10.1016/j.jcp.2003.07.024.

[33] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, Computers & Fluids 33 (3) (2004) 375–404. doi:10.1016/S0045-7930(03)00058-6.

[34] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, Journal of Computational Physics 209 (2) (2005) 448–476. doi:10.1016/j.jcp.2005.03.017.

[35] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (2005) 239–261. doi:10.1146/annurev.fluid.37.061903.175743.

[36] F. Capizzano, Turbulent wall model for immersed boundary methods, AIAA Journal 49 (11) (2011) 2367–2381. doi:10.2514/1.J050466.

[37] J. H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, Journal of Computational Physics 230 (19) (2011) 7347–7363. doi:10.1016/j.jcp.2011.06.003.

[38] Y. Takahashi, T. Imamura, High Reynolds number steady state flow simulation using immersed boundary method, in: Proceedings of the 52nd aerospace science meeting, 2014, AIAA Paper No. 2014-0228. doi:10.2514/6.2014-0228.

[39] C. Brehm, C. Hader, H. Fasel, A locally stabilized immersed boundary method for the compressible Navier–Stokes equations, Journal of Computational Physics 295 (2015) 475–504. doi:10.1016/j.jcp.2015.04.023.

[40] M. Bernardini, D. Modesti, S. Pirozzoli, On the suitability of the immersed boundary method for the simulation of high-Reynolds-number separated turbulent flows, Computers & Fluids 130 (2016) 84–93. doi:10.1016/j.compfluid.2016.02.018.

[41] F. Capizzano, Coupling a wall diffusion model with an immersed boundary technique, AIAA Journal 54 (2) (2016) 728–734. doi:10.2514/1.J054197.

[42] M.-C. Lai, C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, Journal of Computational Physics 160 (2) (2000) 705–719. doi:10.1006/jcph.2000.6483.

[43] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics 161 (1) (2000) 35–60. doi:10.1006/jcph.2000.6484.

[44] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, Journal of Computational Physics 227 (10) (2008) 4825–4852. doi:10.1016/j.jcp.2008.01.028.

[45] T. Nonomura, J. Onishi, A comparative study on evaluation methods of fluid forces on Cartesian grids, Mathematical Problems in Engineering 2017. doi:10.1155/2017/8314615.

[46] R. Wahls, The national transonic facility: A research retrospective, in: Proceeding of the 39th Aerospace Meeting, 2001, AIAA Paper No. 2001-0754. doi:10.2514/6.2001-0754.

[47] J. Green, J. Quest, A short history of the european transonic wind tunnel etw, Progress in Aerospace Sciences 47 (5) (2011) 319–368. doi:10.1016/j.paerosci.2011.06.002.

[48] H. Choi, P. Moin, Grid-point requirements for large eddy simulation: Chapman ' s estimates revisited, Physics of Fluids 24 (1) (2012) 011702. doi:10.1063/1.3676783.

[49] Drag Prediction Workshop, https://aiaa-dpw.larc.nasa.gov, retrieved on October 12th, 2017.

[50] A. J. Sclafani, J. Vassberg, C. Rumsey, M. DeHaan, T. H. Pulliam, Drag prediction for the NASA CRM wing/body/tail using CFL3D and OVERFLOW on an overset mesh, in: Proceedings of the 28th AIAA Applied Aerodynamics Conference, 2010, AIAA Paper No. 2010-4219. doi:10.2514/6.2010-4219.

[51] A. J. Sclafani, J. C. Vassberg, C. Winkler, A. J. Dorgan, M. Mani, M. E. Olsen, J. Coder, DPW-5 analysis of the CRM in a wing-body configuration using structured and unstructured meshes, in: Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2013, AIAA Paper No. 2013-0048. doi:10.2514/6.2013-0048.

[52] E. M. Lee-Rausch, D. P. Hammond, E. J. Nielsen, S. Pirzadeh, C. L. Rumsey, Application of the FUN3D solver to the 4th AIAA drag prediction workshop, Journal of Aircraft 51 (4) (2014) 1149–1160. doi:10.2514/1.C032558.

[53] A. Hashimoto, K. Murakami, T. Aoyama, K. Yamamoto, M. Murayama, P. R. Lahur, Drag prediction on NASA common research model using automatic hexahedra grid-generation method, Journal of Aircraft 51 (4) (2014) 1172–1182. doi:10.2514/1.C032641.

[54] M. D. de Tullio, P. de Palma, G. Iaccarino, G. Pascazio, M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, Journal of Computational Physics 225 (2) (2007) 2098–2117. doi:10.1016/j.jcp.2007.03.008.

[55] B. E. Launder, D. B. Spalding, The numerical computation of turbulent flows, Computer Methods in Applied Mechanics and Engineering 3 (2) (1974) 269–289. doi:10.1016/0045-7825(74)90029-2.

[56] D. C. Wilcox, Wall matching, a rational alternative to wall functions, in: Proceedings of the 27th AIAA Aerospace Sciences Meeting, 1989, AIAA Paper No. 89-611.

[57] T. Craft, A. Gerasimov, H. Iacovides, B. Launder, Progress in the generalization of wall-function treatments, International Journal of Heat and Fluid Flow 23 (2) (2002) 148–160. doi:10.1016/S0142-727X(01)00143-6.

[58] G. Kalitzin, G. Medic, G. Iaccarino, P. Durbin, Near-wall behavior of RANS turbulence models and implications for wall functions, Journal of Computational Physics 204 (1) (2005) 265–291. doi:10.1016/j.jcp.2004.10.018.

[59] T. Knopp, T. Alrutz, D. Schwamborn, A grid and flow adaptive wall-function method for RANS turbulence modelling, Journal of Computational Physics 220 (1) (2006) 19–40. doi:10.1016/j.jcp.2006.05.003.

[60] R. Bond, F. G. Blottner, Derivation, implementation, and initial testing of a compressible wall-layer model, International Journal for Numerical Methods in Fluids 66 (9) (2011) 1183–1206. doi:10.1002/fld.2309.

[61] H. Tennekes, J. L. Lumley, A first course in turbulence, MIT press, 1972.

[62] U. Piomelli, E. Balaras, Wall-layer models for large-eddy simulations, Annual Review of Fluid Mechanics 34 (1) (2002) 349–374. doi:10.1146/annurev.fluid.34.082901.144919.

[63] M. Wang, P. Moin, Dynamic wall modeling for large-eddy simulation of complex turbulent flows, Physics of Fluids 14 (7) (2002) 2043–2051. doi:10.1063/1.1476668.

[64] S. Kawai, J. Larsson, Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy, Physics of Fluids 24 (1) (2012) 015105. doi:10.1063/1.3678331.

[65] G. Kalitzin, G. Iaccarino, Turbulence modeling in an immersed-boundary RANS method, CTR Annual Briefs (2002) 415–426.

[66] T. Ishida, K. Nakahashi, Immersed boundary method for compressible turbulent flow computations in building-cube method, in: Proceedings of the 21st AIAA Computational Fluid Dynamics Conference, 2013, AIAA Paper No. 2013-2451. doi:10.2514/6.2013-2451.

[67] F. Tessicini, G. Iaccarino, M. Fatica, M. Wang, R. Verzicco, Wall modeling for large-eddy simulation using an immersed boundary method, Annual Research Briefs, Stanford University Center for Turbulence Research, Stanford, CA (2002) 181–187.

[68] F. Cadieux, J. Sadique, X. Yang, R. Mittal, C. Meneveau, Wall-modeled large eddy simulation of laminar and turbulent separated flows, in: Proceedings of the 46th AIAA Fluid Dynamics Conference, 2016, AIAA Paper No. 2016-3189. doi:10.2514/6.2016-3189.

[69] F. M. White, I. Corfield, Viscous fluid flow, McGraw-Hill Higher Education Boston, 1974.

[70] P. R. Spalart, S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, in: Proceedings of the 52nd aerospace science meeting, 1992, AIAA Paper No. 92-0439. doi:10.2514/6.1992-439.

[71] C. L. Rumsey, Apparent transition behavior of widely-used turbulence models, International Journal of Heat and Fluid Flow 28 (6) (2007) 1460–1471. doi:10.1016/j.ijheatfluidflow.2007.04.003.

[72] J. Dacles-Mariani, D. Kwak, G. Zilliac, On numerical errors and turbulence modeling in tip vortex flow prediction, International Journal for Numerical Methods in Fluids 30 (1) (1999) 65–82. doi:10.1002/(SICI)1097-0363(19990515)30:1<65::AID-FLD839>3.0.CO;2-Y.

[73] Z. Lei, Effect of RANS turbulence models on computation of vortical flow over wing-body configuration, Transactions of the Japan Society for Aeronautical and Space Sciences 48 (161) (2005) 152–160. doi:10.2322/tjsass.48.152.

[74] P. R. Spalart, Strategies for turbulence modelling and simulations, International Journal of Heat and Fluid Flow 21 (3) (2000) 252–263. doi:10.1016/S0142-727X(00)00007-2.

[75] K. Yamamoto, K. Tanaka, M. Murayama, Effect of a nonlinear constitutive relation for turbulence modeling on predicting flow separation at wing-body juncture of transonic commercial aircraft, in: Proceedings of the 20th AIAA Applied Aerodynamics Conference, 2012, AIAA Paper No. 2012-2895. doi:10.2514/6.2012-2895.

[76] D. J. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, in: Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, 2003, AIAA Paper No.2003-3986. doi:10.2514/6.2003-3986.

[77] E. Shima, K. Kitamura, T. Haga, Green–Gauss/weighted-least-squares hybrid gradient reconstruction for arbitrary polyhedra unstructured grids, AIAA Journal 51 (11) (2013) 2740–2747. doi:10.2514/1.J052095.

[78] E. Shima, K. Kitamura, Parameter-free simple low-dissipation AUSM-family scheme for all speeds, AIAA Journal 49 (8) (2011) 1693–1709. doi:10.2514/1.J050905.

[79] B. van Leer, Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method, Journal of Computational Physics 32 (1) (1979) 101–136. doi:10.1016/0021-9991(79)90145-1.

[80] H. Luo, J. D. Baum, R. Löhner, A hybrid cartesian grid and gridless method for compressible flows, Journal of Computational Physics 214 (2) (2006) 618–632. doi:doi.org/10.1016/j.jcp.2005.10.002.

[81] Y. Tamaki, T. Imamura, Efficient dimension-by-dimension higher-order finite-volume methods for a Cartesian grid with cell-based refinement, Computers & Fluids 144 (2017) 74–85. doi:10.1016/j.compfluid.2016.12.002.

[82] C. Hirsch, Numerical computation of internal and external flows. Volume 2: Computational Methods for Inviscid and Viscous Flows, John Wiley & Sons, 1984.

[83] J. M. Weiss, J. P. Maruszewski, W. A. Smith, Implicit solution of preconditioned Navier–Stokes equations using algebraic multigrid, AIAA Journal 37 (1) (1999) 29–36. doi:10.2514/2.689.

[84] G. Wang, A. Schwöppe, R. Heinrich, Comparison and evaluation of cell-centered and cell-vertex discretization in the unstructured Tau-Code for turbulent viscous flows, in: ECCOMAS CFD, 2010.

[85] J.-R. Carlson, Inflow/outflow boundary conditions with application to FUN3D, Tech. rep., National Aeronautics and Space Administration, NASA/TM-2011-217181 (2011).

[86] E. Shima, A simple implicit scheme for structured/unstructured CFD, in: Proceedings of the 29th Fluid Dynamics Symposium, 1997, (in Japanese).

[87] D. C. Wilcox, Turbulence modeling for CFD, DCW industries, 2006.

[88] METIS - serial graph partitioning and fill-reducing matrix ordering, http://glaros.dtc.umn.edu/gkhome/metis/metis/overview, retrieved on October 25th, 2017.

[89] T. Imamura, Y. Tamaki, M. Harada, Parallelization of a compressible flow solver (UTCart) on cell-based refinement Cartesian grid with immersed boundary method, in: Proceedings of the 29th International Conference on Parallel Computational Fluid Dynamics, 2017.

[90] S. R. Allmaras, F. T. Johnson, P. R. Spalart, Modifications and clarifications for the implementation of the Spalart–Allmaras turbulence model, in: Proceedings of the 7th International Conference on Computational Fluid Dynamics, 2012, ICCFD7-1902.

[91] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, Journal of Computational physics 114 (1) (1994) 146–159. doi:10.1006/jcph.1994.1155.

[92] C. P. van Dam, Recent experience with different methods of drag prediction, Progress in Aerospace Sciences 35 (8) (1999) 751–798. doi:10.1016/S0376-0421(99)00009-3.

[93] K. Kusunose, J. P. Crowder, Extension of wake-survey analysis method to cover compressible flows, Journal of Aircraft 39 (6) (2002) 954–963. doi:10.2514/2.3048.

[94] NASA Langley Research Center Turbulence Modeling Resource, https://turbmodels.larc.nasa.gov, retrieved on October 12th, 2017.

[95] A. Hashimoto, K. Murakami, T. Aoyama, K. Ishiko, M. Hishida, M. Sakashita, P. Lahur, Toward the fastest unstructured CFD code 'FaSTAR', in: Proceedings of the 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, 2012, AIAA Paper No. 2012-1075. doi:10.2514/6.2012-1075.

[96] D. P. Raymer, Aircraft Design: A Conceptual Approach, AIAA Education Series, 1992.

[97] S. Dennis, G.-Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, Journal of Fluid Mechanics 42 (3) (1970) 471–489.

[98] F. Nieuwstadt, H. Keller, Viscous flow past circular cylinders, Computers & Fluids 1 (1) (1973) 59–71.

[99] CFL3D Version 6 Home Page, https://cfl3d.larc.nasa.gov, retrieved on October 18th, 2017.

[100] C. L. Ladson, Effects of independent variation of Mach and Reynolds numbers on the low-speed aerodynamic characteristics of the NACA 0012 airfoil section, Tech. rep., National Aeronautics and Space Administration, NASA-TM-4074 (1988).

[101] J. C. Vassberg, M. A. DeHaan, S. M. Rivers, R. A. Wahls, Development of a common research model for applied CFD validation studies, in: Proceedings of the 26th Applied Aerodynamics Conference, 2008, AIAA Paper No. 2008-6919. doi:10.2514/6.2008-6919.

[102] M. Murayama, Y. Yokokawa, H. Kato, M. Kanazaki, K. Yamamoto, T. Ito, Computational study for high-lift aerodynamics research in JAXA, in: Proceeding of the 26th International Congress of the Aeronautical Sciences, 2008.

[103] M. B. Rivers, A. Dittberner, Experimental investigations of the NASA common research model, Journal of Aircraftdoi:10.2514/1.C032626.

[104] M. Ueno, M. Kozai, S. Koga, Transonic wind tunnel test of the NASA CRM: Volume 1, Tech. rep., Japan Aerospace Exploration Agency, JAXA-RM-13-017E (2014).

[105] J. C. Vassberg, E. N. Tinoco, M. Mani, B. Rider, T. Zickuhr, D. Levy, O. Brodersen, B. Eisfeld, S. Crippa, R. Wahls, J. H. Morrison, D. J. Mavriplis, M. Mitsuhiro, Summary of the fourth aiaa computational fluid dynamics drag prediction workshop, Journal of Aircraft 51 (4) (2014) 1070–1089. doi:10.2514/1.C032418.

[106] E. N. Tinoco, O. P. Brodersen, S. Keye, K. R. Laflin, E. Feltrop, J. C. Vassberg, M. Mani, B. Rider, R. A. Wahls, J. H. Morrison, Summary of data from the sixth AIAA CFD Drag Prediction Workshop: CRM cases 2 to 5, in: Proceedings of the 55th AIAA Aerospace Sciences Meeting, 2017, AIAA Paper No. 2017-1208. doi:10.2514/6.2017-1208.

[107] Third Aerodynamic Prediction Challenge (APC-III), https://cfdws.chofu.jaxa.jp/apc, retrieved on October 12th, 2017.

[108] G. van Albada, B. Van Leer, W. Roberts Jr, A comparative study of computational methods in cosmic gas dynamics, Astronomy and Astrophysics 108 (1982) 76–84.

[109] Reedbush Supercomputer System, https://www.cc.u-tokyo.ac.jp/system/reedbush/index-e.html, retrieved on October 12th, 2017.

[110] M. Hishida, A. Hashimoto, K. Murakami, T. Aoyama, A new slope limiter for fast unstructured CFD solver FaSTAR, in: Proceedings of the 42th Fluid Dynamics Symposium / 35th Aerospace Numerical Simulation Symposium, 2010, (in Japanese).

[111] S. Yoon, A. Jameson, Lower-upper symmetric-Gauss–Seidel method for the Euler and Navier–Stokes equations, AIAA Journal 26 (9) (1988) 1025–1026. doi:10.2514/3.10007.

[112] Y. Yokokawa, M. Murayama, H. Uchida, K. Tanaka, T. Ito, K. Yamamoto, K. Yamamoto, Aerodynamic influence of a half-span model installation for high-lift configuration experiment, in: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, AIAA Paper No. 2010-0684. doi:10.2514/6.2010-0684.

[113] M. Murayama, Y. Yokokawa, K. Tanaka, K. Yamamoto, T. Ito, Numerical simulation of half-span aircraft model with high-lift devices in wind tunnel, in: Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit, 2008, AIAA Paper No.2008-0333. doi:10.2514/6.2008-0333.

[114] B. Konig, E. Fares, M. Murayama, Y. Ito, Y. Yokokawa, K. Yamamoto, K. Ishikawa, Lattice-Boltzmann simulations of the JAXA JSM high-lift configuration, in: Proccedings of the 34th AIAA Applied Aerodynamics Conference, 2016, AIAA Paper No. 2016-3721. doi:10.2514/6.2016-3721.

[115] High Lift Prediction Workshop, https://hiliftpw.larc.nasa.gov, retrieved on October 12th, 2017.