AIAA JOURNAL
Vol. 49, No. 11, November 2011

Turbulent Wall Model for Immersed Boundary Methods

Francesco Capizzano*
Centro Italiano Ricerche Aerospaziali, 81043 Capua, Italy

DOI: 10.2514/1.J050466

This paper describes the development of a wall model to extend the applicability of immersed boundary methods to high-Reynolds-number flows. A two-layer approach, based on a decomposition of the near-wall region, is adopted. An outer region is governed by the compressible Reynolds-averaged Navier–Stokes equations, which are solved numerically by using a classical finite volume method. In the proximity of the wall, an inner zone is established and modeled by a simplified version of the thin-boundary-layer equations. The simulation platform is based on Cartesian meshes and an immersed boundary technique. It is able to solve the steady Euler/Reynolds-averaged Navier–Stokes equations in two- and three-dimensional coordinates. The robustness and the accuracy of the methodology are discussed. At present, this work represents the last advance of a research activity for which the final goal is a fast predesign tool for aeronautical/industrial applications.

Nomenclature

C_f	=	skin-friction coefficient
C_P	=	pressure coefficient
E	=	total energy per unit mass
g	=	$1/\sqrt{C_\mu}\omega$
H	=	total enthalpy per unit mass
h	=	enthalpy per unit mass
k	=	turbulent kinetic energy per unit mass
Mach	=	Mach number
p	=	pressure
Pr	=	Prandtl number
Pr_t	=	turbulent Prandtl number
Re	=	Reynolds number
S_{ij}	=	rate of strain tensor
T	=	temperature
u, v, w	=	cartesian velocity components
u_i	=	velocity components in local curvilinear coordinates
u_τ	=	friction velocity
x_1, x_3	=	tangential to wall directions
x_2	=	normal to wall direction
α	=	angle of incidence
ε	=	turbulent dissipation per unit mass
μ	=	molecular dynamic viscosity
μ_t	=	eddy viscosity
ρ	=	density
τ_{ij}	=	total stress tensor
τ_{w1}, τ_{w3}	=	wall shear stress components
τ_{ij}^R	=	Reynolds-stress tensor
τ_{ij}^v	=	viscous stress tensor
ω	=	turbulent dissipation rate per unit mass

Subscript

∞	=	freestream quantity

Superscript

$+$	=	quantity in wall units

Presented as Paper 2010-0712 at the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, FL, 4–7 January 2010; received 8 February 2010; revision received 7 April 2011; accepted for publication 19 April 2011.

*Research Engineer, Aerodynamic and Aeroacoustic Methods.

I. Introduction

MESH generation is considered one of the major bottlenecks in computational fluid dynamics (CFD). Body-conforming methods are largely diffused thanks to the accuracy of the near-wall treatment. However, a high level of expertise is necessary during the mesh design phase, which is time-consuming, especially for three-dimensional (3-D) complex geometries. Automation is possible only for a particular class of problems. Conversely, Cartesian grid methods are characterized by a high level of automation and a certain flexibility in treating arbitrary geometries. The overlapping zone between Cartesian cells and the body surface can be represented by polyhedral cut cells having proper intersections [1,2]. Alternatively, a body forcefield can be added in the momentum equation to simulate the presence of the immersed boundaries (IBs) [3,4]. The main advantage of the IB methods with respect to cut-cell ones consists of avoiding complex geometrical algorithms to intersect the hexahedra with the solid surfaces.

A number of IB applications are available in literature [5], most of them simulating Euler [6–8] and laminar [9–11] flows by using basically the same discrete forcing approach based on linear interpolation stencils. Turbulent simulations are still possible by using the above linear operators, provided that a very fine mesh is generated near the wall [12]. Moreover, even if a local grid refinement (LGR) strategy is adopted, it is not possible to obtain high-aspect-ratio cells in the wall proximity. This de facto limits the application of Cartesian methods to low Reynolds numbers. The use of a wall model is a way to avoid large Cartesian meshes, particularly in case of 3-D turbulent flows at high Reynolds numbers.

The first idea is to take advantage of the wall-function technique developed for body-conforming Reynolds-averaged Navier–Stokes (RANS) methods. Different analytical formulations based on the existence of a logarithmic law of the wall are available, but only a few of them have been already tested in the framework of the IB methods [13–15]. A procedure based on a power-law function coupled with a discrete continuity equation is described in [16] to reconstruct locally the near-wall velocity vector. Adaptive wall functions based on numerical lookup tables have been successfully tested on attached and slightly separated flows by using body-fitted meshes [17], but only a few attempts refer to the IB methods [18].

A second choice uses the two-layer wall models proposed in the field of large eddy simulations (LESs). A discrete bibliography is available on the application of these models, and an extended review is available in [19]. The basic idea is to obtain the wall shear stress by solving a set of boundary-layer equations on an embedded fictitious mesh that originates from the first offwall mesh point down to the solid surface. A two-layer wall model for LES-IB applications is proposed in [20] along with an operative procedure for the reconstruction of the wall shear stress starting from forcing points inside the fluid region. Such a model is expected to be more accurate

with respect to simple analytical wall functions, especially in the case of complex separated flows.

In this paper, a novel RANS-IB methodology for high-Reynolds-number flows is described. An automatic Cartesian mesh generator is coupled with a flow solver based on a finite volume (FV) method. The system adopts a fully unstructured data management that allows for anisotropic LGRs. A discrete forcing approach mimics the effect of the body on the flow by means of a direct boundary condition (BC). Turbulence is introduced to deal with medium/high-Reynolds-number flows, and a two-layer wall modeling is adopted to correctly control the turbulent quantities at the IB surfaces. Particular attention has be paid to the design of vectorizable algorithms, which provide better performance on the Centro Italiano Ricerche Aerospaziali (CIRA) vectorial machines. Furthermore, the code has been parallelized by using the open multiprocessing (OpenMP) standard directives.

II. Geometry Representation and Mesh Generation

A key feature of the IB method is the ability to manage the immersed objects using CAD output files directly. The need to translate surfaces from CAD tools to a mesh generator implies the choice of a surface representation. The IB approach, different from the cut-cell alternative, does not make use of wall surface patches for the reconstruction of the fluxes. Only intersection points between the wall and proper rays from the wall-proximity Cartesian cells are needed. This fact suggests the use of simplified formats based on surface triangulations rather than more complicated representations based on semianalytical forms (e.g., bisplines, NURBS, etc.).

The present method imports CAD objects described by watertight surface triangulations and generates an initial uniform Cartesian mesh in a user-defined box domain. The immersed geometries are detected by using a ray-tracing technique [21] that enables cells inside and outside the fluid region to be identified. A recursive LGR process is carried out to cluster cells near the solid surfaces until the desired spacings are obtained. The final objective is to assure a good resolution of the wall surface along with a smooth jump between zones of different cell sizes in order to limit instabilities that may arise during the flow solution procedure.

There are two typical methods of storing a Cartesian mesh with local adaptation: fully unstructured [10,22] or a hierarchical parent–child tree structure [1,2]. The fully unstructured approach requires more memory storage than the tree structure, but the latter requires more computational effort when calculations involve cell neighbors. In addition, a tree structure obliges to isotropic refinements while fully unstructured data permit anisotropic adaptation with clear advantages in reducing the amount of cells.

The fully unstructured technique adopted here consists of dynamic linked lists that are used as containers of objects. The basic entities are cells, faces, and vertices, each of them storing explicitly an array of pointers to the neighboring entities. The initial uniform mesh is put in a list structure, and new cells resulting from refinements are added to its end. Generally, unstructured data management is based on a complete cell-face-edge-vertex mapping. In the present method, a reduced cell-face-vertex mapping is sufficient to allow for the functionalities of the flow solver and the postprocessor. The face-to-cell connection is fundamental for the finite volume flow solver, while both vertex-to-face and vertex-to-cell connections are used for postprocessing purposes. A local index system is adopted to uniquely define the topological space in a fashion similar to the one used in [10].

The mesh refinement takes place based on different criteria. The first is based on geometric considerations. It is clear that all the near-wall cells, named IB cells, have to be refined; thus, a procedure to locate them is required. At each refinement stage, the whole mesh is first divided into fluid and solid cells using a ray-casting check. Simply put, the boundary between these two zones contains the cells to be refined. These interface cells are further divided into IB-solid and IB-fluid groups, depending on whether the cell center is inside or outside the solid region, respectively. Only fully fluid and IB-fluid cells are passed to the flow solver.

Table 1 Mesh generation after seven refinement levels around a sphere surface

N_{tri}	M_{cells}	CPU_{time}, sec	CPU_{time}/cell
288	149,828	17.8	$1.19 \cdot 10^{-4}$
1,152	152,908	18.9	$1.24 \cdot 10^{-4}$
4,608	153,492	19.34	$1.26 \cdot 10^{-4}$
18,432	154,148	21.32	$1.38 \cdot 10^{-4}$

At the input phase, the user can define layers of cells having the same size. These are named as buffer layers and are obtained by applying the following procedure during the refinement process. Starting from the IB-fluid cells, a fixed number of neighbors are marked in all directions and then obliged to have the same refinement level [23]. This feature is particularly useful in the case of viscous computations to limit spurious entropy productions caused by zones of different cell sizes inside the boundary layer.

A third refinement criteria comes from an a posteriori analysis of the flow solution. Proper sensors can be defined based on flowfield quantities such as pressure gradient, velocity divergence, and vorticity. When a cell does not meet the required criteria, it is flagged for refinement. The description of the mesh-tool features and implementation details are given in [24]. An interesting aspect is the evaluation of the mesh generation performance. A large number of tests on different geometries with various surface discretizations shows, as expected, that high resources are spent for the cell tagging process. Particularly, the intersection between the rays from each cell and the surface triangulation represent the major CPU overhead. In general, the number of checks required for M cells with respect to N surface triangles is of order $\mathcal{O}(M \cdot N)$. Filters based on the bounding-box concept can alleviate the problem but do not resolve it.

A way to reduce the computational effort is the use of a binary-tree structure to store the CAD surface facets. If the tree is built properly, it returns only a small number of triangles that potentially intersect the ray casting from a querying cell. An alternating digital tree (ADT) structure is adopted [25] that guarantees, for a single ray, a query search complexity of order $\mathcal{O}(\log N)$. Table 1 shows the results of some numerical experiments on a desktop PC. Starting from the same uniform Cartesian mesh, a fixed number of seven geometric refinements are carried out around a sphere surface discretized by a variable number of facets. The first and second columns refer to the number of CAD surface elements and the final mesh cells, respectively. The third and fourth columns consider the total CPU time and the CPU time per cell generated during the refinement process. The graph shown in Fig. 1 relates the ability of ADT structure in maintaining a low influence of CAD surface resolution compared with the mesh generation CPU resources.

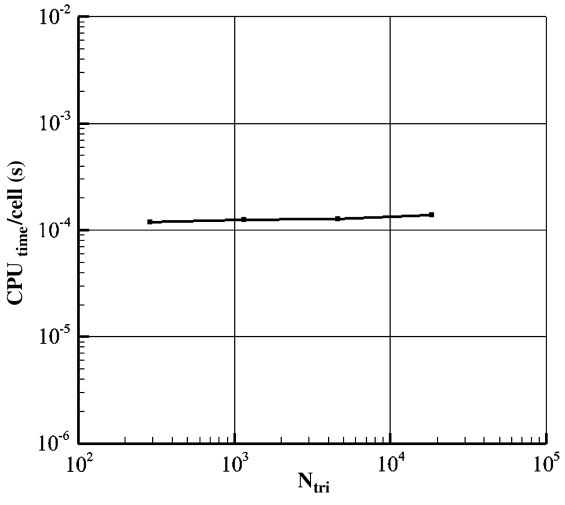


Fig. 1 Mesh generation performances on a single-core PC; 3.0 GHz.

III. Mathematical Model

The flow solver is able to simulate compressible, steady, turbulent two-dimensional (2-D)/3-D flows by means of Favre-averaged RANS equations for which the Cartesian differential form in tensor notation read

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ji}}{\partial x_j} \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j H) = \frac{\partial}{\partial x_j}\left[u_i \tau_{ij} + (\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_j} - q_j\right] \tag{3}$$

which are formally identical to the Navier–Stokes equations, except that all the dependent variables are averaged quantities describing the mean flow. The basic conservative variables are the density ρ, the mass flux vector $(\rho u, \rho v, \rho w)$, and the total energy ρE per unit volume. The total energy E is defined as the sum of the internal energy, the kinetic energy, and the turbulent kinetic energy per unit mass, respectively, $E = e + u_i u_i/2 + k$. A perfect gas model assumption is made:

$$p = \rho RT, \qquad h = c_p T = \frac{\gamma}{\gamma - 1}\frac{p}{\rho}$$

$$H = E + p/\rho, \qquad e = c_v T = \frac{1}{\gamma - 1}\frac{p}{\rho}$$

where R is the gas constant, h is the enthalpy per unit mass, H is the total enthalpy per unit mass, c_p is the specific heat coefficient at constant pressure, c_v is the specific heat coefficient at constant volume, and $\gamma = c_p/c_v$. The total stress tensor τ_{ij} is the sum of the viscous stress tensor τ_{ij}^v and the Reynolds-stress tensor τ_{ij}^R:

$$\tau_{ij} = \tau_{ij}^v + \tau_{ij}^R$$

For Newtonian fluids and Stokes' hypothesis, the viscous stress tensor is related to the rate of strain tensor S_{ij}:

$$S_{ij} = \frac{1}{2}\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right], \qquad \tau_{ij}^v = 2\mu\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right]$$

with μ as the molecular dynamic viscosity, which is assumed to vary with the temperature as described by the Sutherland law. The Reynolds stresses are modeled by using the Boussinesq hypothesis based on the eddy viscosity concept μ_t:

$$\tau_{ij}^R = 2\mu_t\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] - \frac{2}{3}\rho k \delta_{ij}$$

Analogously, the total heat flux vector is the sum of the molecular and turbulent heat conduction:

$$q_j = -c_p\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial x_j}$$

where Pr and Pr_t are the laminar and turbulent Prandtl numbers, respectively. The closure of the RANS equations is obtained by using two different k-ω turbulence models that do not contain an explicit dependence on the wall distance. In general, the k-ω models have better performance for wall-bounded flows than k-ε models and are numerically more robust, especially when strongly coupled with the RANS equations. However, they suffer from the well-known problem of numerical results, which are dependent on the freestream ω values. The following two models try to resolve the problem by using different strategies.

A. *k-g* Model

The k-g turbulence model is a modified version of the standard Wilcox k-ω model. It is obtained by the change of variable $g = 1/\sqrt{C_\mu \omega}$, as suggested in [18]. Thus, the eddy viscosity is defined as

$$\mu_t = \rho k (C_\mu g^2)$$

The corresponding transport equations are

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial(\rho u_j k)}{\partial x_j} = \tau_{ij}\frac{\partial u_i}{\partial x_j} - C_\mu \rho \omega k + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_k \mu_t)\frac{\partial k}{\partial x_j}\right] \tag{4}$$

$$\frac{\partial}{\partial t}(\rho g) + \frac{\partial(\rho u_j g)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[(\mu + \sigma_g \mu_t)\frac{\partial g}{\partial x_j}\right] - \alpha_g \frac{g}{2\kappa}P_\kappa + \frac{\beta \rho}{2C_\mu g} - \frac{3}{g}(\mu + \sigma_g \mu_t)\frac{\partial g}{\partial x_j}\frac{\partial g}{\partial x_j} \tag{5}$$

along with the model constants:

$$C_\mu = 0.09, \qquad \sigma_k = 0.5, \qquad \alpha_g = \tfrac{5}{9}$$
$$\beta = \tfrac{5}{6}C_\mu, \qquad \sigma_g = 0.5$$

The peculiarity of the model is its ability to enforce the correct large value of ω at the boundary-layer edge. Note that the g quantity goes to zero at the wall.

B. *k-ω* Turbulent Nonturbulent Model

The k-ω turbulent nonturbulent (TNT) turbulence model [26] resolves the freestream dependency by adding a cross-diffusion source term into the ω transport equation:

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \alpha_\omega \frac{\omega}{k}\tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta \rho \omega^2 + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_\omega \mu_t)\frac{\partial \omega}{\partial x_j}\right] + C_d \tag{6}$$

where

$$C_d = \sigma_d \frac{\rho}{\omega}\max\left[\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 0\right]$$

The added source term appears to work only away from the wall and does not influence the near-wall region. The k transport equation is unchanged [see Eq. (4)], and the eddy viscosity is

$$\mu_t = \frac{\rho k}{\omega}$$

The closure constants are

$$C_\mu = 0.09, \qquad \sigma_k = \tfrac{2}{3}, \qquad \alpha_\omega = \tfrac{5}{9}$$
$$\beta = \tfrac{5}{6}C_\mu, \qquad \sigma_\omega = 0.5, \qquad \sigma_d = 0.5$$

IV. Numerical Method

The integral form of the governing equations (1–6) are discretized by a cell-centered finite volume method. The basic scheme described in [27] is extended to treat physical diffusion and source terms. The convective and diffusive fluxes are approximated by using a second-order accurate in space central difference scheme (CDS). In general, a simple average of the flow variables on either side of the cell face is sufficient. Moreover, the Cartesian LGR process generates faces that share cells of different sizes. Here, the generic face-center quantity ϕ_{ik} is reconstructed by a linearly exact interpolation involving the ith cell and its kth neighbor. In case of an x-normal directed face, the cell-

center quantities ϕ_i and ϕ_k have to be interpolated, as a first step, along the y and z directions on an auxiliary node aligned with the face-center unit normal vector. Then, the classic CDS scheme is applied along the x direction. The general formula is

$$\phi_{ik} = \left(\frac{x - x_i}{x_k - x_i}\right)\left[\phi_k + (y - y_k)\left(\frac{\partial\phi}{\partial y}\right)_k + (z - z_k)\left(\frac{\partial\phi}{\partial z}\right)_k\right] + \left(\frac{x_k - x}{x_k - x_i}\right)\left[\phi_i + (y - y_i)\left(\frac{\partial\phi}{\partial y}\right)_i + (z - z_i)\left(\frac{\partial\phi}{\partial z}\right)_i\right] \tag{7}$$

Consider the case shown in Fig. 2, where an x-normal directed face is shared by two cells of different sizes. The face completely spans the refined kth cell, and the first expression inside the square brackets of Eq. (7) reduces to ϕ_k. The centroid of the coarser ith cell does not lie on the face-center normal direction; thus, the ϕ_i quantity has to be y and z interpolated:

$$\phi_i' = \phi_i + (y - y_i)\left(\frac{\partial\phi}{\partial y}\right)_i + (z - z_i)\left(\frac{\partial\phi}{\partial z}\right)_i$$

Expressions similar to Eq. (7) can be obtained in the cases of y- and z-normal directed faces. The cell-center gradient components $\left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right)$ can be computed by using the Green–Gauss or a least-squares reconstruction [10,28]. The face-center gradients, necessary for the viscous fluxes, can be approximated by averaging the gradients at the neighboring cell centers:

$$\overline{(\nabla\phi)}_{ik} = \tfrac{1}{2}[(\nabla\phi)_i + (\nabla\phi)_k]$$

Furthermore, a correction in the direction connecting the cell centers is carried out:

$$(\nabla\phi)_{ik} = \overline{(\nabla\phi)}_{ik} - \left[\overline{(\nabla\phi)}_{ik} \cdot \mathbf{r}_{ik} - \frac{\phi_k - \phi_i}{|\mathbf{r}_{ik}|}\right] \cdot \mathbf{r}_{ik} \tag{8}$$

where $\mathbf{r}_{ik} = (\mathbf{r}_k - \mathbf{r}_i)/|\mathbf{r}_k - \mathbf{r}_i|$ is the direction connecting the neighboring cell centers.

The CDS scheme poses the problem of oscillatory solutions. A way to damp high-frequency modes is to add artificial dissipative terms to the discretized balance equations. Here, a matrix artificial diffusion (MATD) scheme is implemented with proper modifications for unstructured grids [27] and in such a way to allow the solver vectorization [29]. In the MATD scheme, different from the scalar model, the amount of artificial diffusion is controlled by a matrix linked to the Jacobian of the convective fluxes in a way similar to the upwind schemes. The result is an improved solution quality near shock waves and in the boundary layers thanks to a weaker grid dependency when compared with scalar dissipation.

The complete set of Eqs. (1–6) are solved by using a multistage explicit Runge–Kutta pseudotime relaxation in a fully coupled way. That is to say that the RANS and the turbulence model equations are marching in the pseudotime simultaneously. The advantages of this choice are a faster convergence rate and a compact algorithm with respect to an uncoupled strategy. No special treatment is adopted for the turbulent source terms, except for the turbulent kinetic energy production, which is upper-bounded to correct the stagnation problem anomaly [30]. Usual limiters based on the freestream values are applied to the turbulent quantities to ensure their positivity. A local time stepping is performed in each cell to accelerate convergence. The stability limit of the whole scheme can be further increased by replacing the original residuals at the ith cell center $\mathbf{R}_i$ with averaged residuals $\tilde{\mathbf{R}}_i$:

$$\tilde{\mathbf{R}}_i = \mathbf{R}_i + \epsilon\nabla^2\tilde{\mathbf{R}}_i \tag{9}$$

where ϵ is a smoothing factor, and

$$\nabla^2\tilde{\mathbf{R}}_i = \sum_{k(i)}(\tilde{\mathbf{R}}_k - \tilde{\mathbf{R}}_i)$$

is the undivided Laplacian operator constructed using the residuals $\mathbf{R}_k$ of the neighboring cells. Equation (9) can be solved by using a small number of Jacobi iterations at every stage of the Runge–Kutta time stepping. A good level of efficiency is obtained by using three Jacobi iterations and $\epsilon \approx 0.2$.

V. Immersed Boundary Technique

The proposed IB method consists of a discrete forcing approach with a direct BC imposition. The forcing is face-based in order to force conservation at the near-wall cells. Indeed, the IB-fluid cells make use of the same FV scheme as that applied to the fully fluid cells. Effectively, the cell-center flow-state vector $\mathbf{Q}$ is obtained by the balance of the convective and diffusive fluxes through its faces. Three different face types are identified: 1) the domain-boundary face, being the face located at the far boundary; 2) the fully fluid face, being the face sharing two fully fluid cells; and 3) the IB face, being the face sharing an IB-fluid cell and an IB-solid cell.

Note that the ensemble of the IB faces is a closed surface. Consider the simplification of a 2-D near-wall region, shown in Fig. 3. The fully fluid and IB-fluid cells are colored gray and black, respectively. The correct flux balance inside the IB-fluid cells requires the flow-state vector $\mathbf{Q}$ at the center of its four faces. Among them, the IB faces need a special treatment. The direct BC method consists of imposing proper fluxes at the IB-face centers, marked with a black circle, in order to satisfy the Dirichlet/Neumann conditions at the wall. For each IB face, the following procedure is carried out:

1) The normal to the wall passing through the IB-face center (black circle) is traced.

2) A fluid forcing point F at a distance δ^F is set.

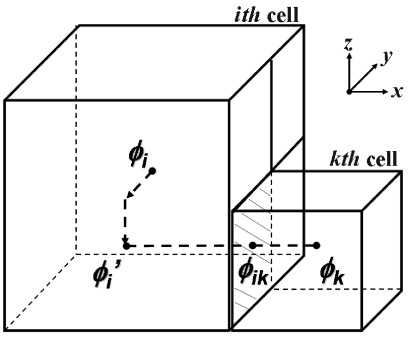


Fig. 2 Linearly exact interpolation stencil for an x-normal directed face.

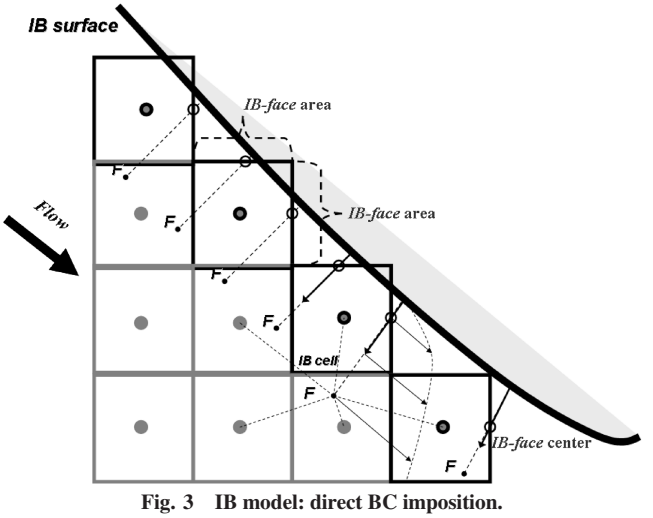


Fig. 3 IB model: direct BC imposition.

3) The flow-state vector at the F point $\mathbf{Q}^F = (\rho, \rho u, \rho v, \rho w, \rho E)^F$ is reconstructed by using a weighted least-squares (WLSQ) procedure based on the nearest cell centers (see Appendix A).

4) The flow-state vector $\mathbf{Q}$ at the IB-face center is constructed, taking into account the desired condition at the wall surface.

5) The generic convective or diffusive flux is obtained by multiplying the generic flow-state quantity with the entire IB-face area.

The geometric information necessary for the above procedure is created at the mesh generation phase. A recursive algorithm, based on the ray-tracing technique, is able to detect the foot of the normal to the wall direction passing through the IB-face center. Both the wall $\mathbf{P}_i^{IB}$ point and the corresponding $\mathbf{n}_i^{IB}$ unit normal vector are associated with a unique IB face; thus, it is sufficient to loop over the whole IB-face family to apply the correct BC.

The above procedure is different from the one initially proposed in [27] for Euler flows. In the present method, the interpolation direction is not normal to the IB face but is normal to the wall in order to facilitate the imposition of viscous BCs. The current method enforces the correct wall BC by making use of fluid points F_i. They all have the same distance δ^F from the body, and thus, on the whole, can be viewed as a body-conforming near-wall surface. Numerical experiments have shown that the forcing point F_i has to be located as near to the wall as possible and must be compatible with the numerical scheme stability. Here, the δ^F distance is set to the minimum IB-cell dimension.

In case of Euler or laminar flows, a linear interpolation, from the forcing points along the normal to the wall direction, is sufficient to satisfy the wall conditions and to carry out a second-order-accurate wall integration on feasible Cartesian meshes. Conversely, linear interpolation would require Cartesian grids, which are too fine to correctly capture the nonlinearities of turbulent wall-bounded flows. The near-wall Cartesian cells would probably have dimensions, in plus units, some order of magnitude greater than the optimal one [$\mathcal{O}(1)$]. Thus, a different strategy needs to be pursued to shift the problem from the mesh generation to the flow solver.

VI. Turbulent Wall Modeling

Linear interpolations are the basic ingredient of low Reynolds IB methods based on a discrete forcing approach. Moreover, they are effective only if the forcing points are located in the linear region of the boundary layer. As a consequence, for medium/high-Reynolds-number flows, a wall modeling is proposed. A two-layer approach, based on a decomposition of the near-wall region [31], has been developed. The basic idea is to have an external layer governed by the usual RANS equations and a near-wall zone modeled by 3-D thin boundary-layer equations:

$$\frac{\partial}{\partial x_2}\left[(\mu + \mu_t)\frac{\partial u_i}{\partial x_2}\right] = F_i, \qquad i = 1, 3 \tag{10}$$

with x_2 as the direction normal to the wall, and (x_1, x_3) as the tangential to the wall directions. In general, the right-hand side F_i is equal to sum of the unsteady, convective, and pressure gradient terms:

$$F_i = \frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) + \frac{\partial p}{\partial x_i} \tag{11}$$

Here, a simplified version of Eq. (10) is adopted, in which the unsteady and convective terms are neglected:

$$\frac{\partial}{\partial x_2}\left[(\mu + \mu_t)\frac{\partial u_i}{\partial x_2}\right] = \frac{\partial p}{\partial x_i}, \qquad i = 1, 3 \tag{12}$$

For a flat plate with zero pressure gradient, the universal law of the wall is recovered and the profiles of the boundary-layer quantities collapse in the region between the wall and the outer edge of the logarithmic layer if properly scaled with the friction velocity u_τ:

$$u^+ = \frac{u}{u_\tau}, \qquad x_2^+ = \frac{\rho u_\tau}{\mu} x_2, \qquad \mu^+ = \frac{\mu_t}{\mu}$$

$$p^+ = \frac{\mu}{\rho^2 u_\tau^3}\frac{\partial p}{\partial x_2}, \qquad k^+ = \frac{k}{u_\tau^2}, \qquad \omega^+ = \frac{\mu}{\rho u_\tau^2}\omega$$

$$g^+ = \sqrt{\frac{\rho u_\tau^2}{\mu}}\, g$$

For each IB face, a virtual subgrid starting from the forcing point F_i down to the wall is embedded (see Fig. 4). The eddy viscosity is evaluated by a mixing-length model with wall damping:

$$\frac{\mu_t}{\mu} = \kappa_a x_2^+ \left(1 - e^{-(x_2^+/A^+)}\right)^2 \tag{13}$$

where x_2^+ is the distance in wall units, $\kappa_a = 0.4$ is the von Kármán constant, and $A^+ = 19$. The analytical or numerical solution of Eq. (12) is obtained by imposing the no-slip condition at the wall and the RANS solution reconstructed at the forcing point F by using a WLSQ procedure. Moreover, Eq. (12) requires the eddy viscosity, which depends on u_τ:

$$u_\tau = \left[\left(\frac{\tau_{w1}}{\rho}\right)^2 + \left(\frac{\tau_{w3}}{\rho}\right)^2\right]^{1/4} \tag{14}$$

The velocity scale u_τ can be computed using the stresses from the previous pseudotime step or using the stresses at the current iteration step. In the latter case, the system of Eqs. (12) and (13) needs an iterative procedure to be solved in terms of wall shear stress τ_{wi}.

Neglecting the unsteady term is justified in case of steady flows. The convection is negligible with respect to diffusion in the viscous region of the boundary layer. Therefore, this assumption is acceptable as long as the resolved part of the boundary layer does not exceed the outer edge of the log layer. Numerically speaking, Eq. (10) reduces to a set of ordinary differential equations with clear advantages in terms of numerical scheme with respect to nonlinear problems.

An example of the analytical solution of Eq. (12) is proposed in [31], where the model is coupled with an incompressible LES method. Here, a numerical approach is carried out in order to allow future extensions toward more sophisticated models. The details are presented in Appendix B.

In case of body-conforming methods based on wall functions, the τ_{wi} information is sufficient to set up classic high Reynolds BCs. They prescribe the shear stress and no penetration at the wall. It is not

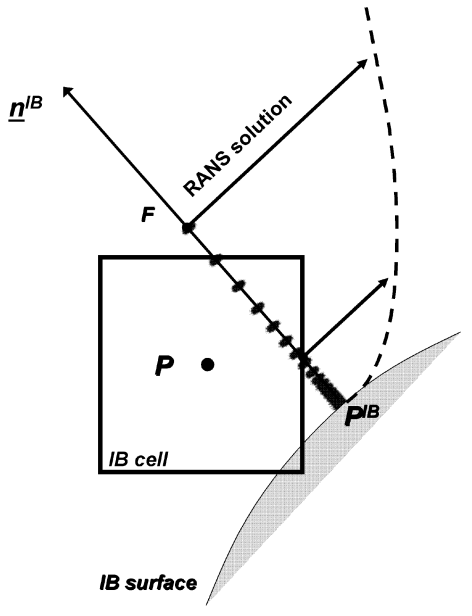


Fig. 4 Two-layer wall model.

the case of IB methods where the near-wall cell faces do not conform to the wall surface. Particularly, the problem is to translate the above high Reynolds wall conditions in feasible numerical constraints at each IB-face center.

Different from the body-fitted methods, not only diffusive but also convective fluxes are required to assure the correct balance inside the IB cells. The diffusive fluxes at IB-face centers are evaluated by using the shear stress computed through Eq. (12). Convective fluxes need some attention. The adopted FV method is nominally second-order accurate in space; that is to say, a Cartesian cell near the wall can see only linear variations of the flow variables. Thus, to avoid numerical instabilities, a linear approximation inside the near-wall cells has to be guaranteed. A linear behavior is assumed, locally, as for the tangential and for the normal to the wall velocity components. The tangential ones are linearly extrapolated from the forcing point values down to the IB-face center:

$$u_i = u_i^F + \left(\frac{\partial u_i}{\partial x_2}\right)^F (x_2 - x_2^F), \qquad i = 1, 3 \tag{15}$$

where $(\partial u_i/\partial x_2)^F$ are obtained by Eq. (12). The nonpenetration condition at the wall is enforced by imposing

$$u_2 = u_2^F \frac{x_2}{x_2^F} \tag{16}$$

where x_2^F and x_2 are wall distances of the forcing point and the IB-face center, respectively. The temperature is obtained by the Crocco–Busemann equation coupled with the adiabatic wall condition:

$$T = T^F + \frac{1}{2c_p} Pr^{1/3}[(|\mathbf{u}_t^F|)^2 - (|\mathbf{u}_t|)^2] \tag{17}$$

where $|\mathbf{u}_t| = (u_1^2 + u_3^2)^{1/2}$ is the module of the tangential velocity vector. The pressure is extrapolated by using the WLSQ gradient, and the density is recovered from the gas state law:

$$p = p^F + (\nabla p|^F \cdot \mathbf{x}_2)(x_2 - x_2^F), \qquad \rho = \frac{p}{RT} \tag{18}$$

Some considerations are necessary regarding the turbulent quantities. It is probable that the coarse IB-fluid cell volume contains a mix of log layer and viscous sublayer. No satisfactory interpolation formula exists for the turbulent kinetic energy. In [17], it is observed that, for the simplest case of flat plate boundary layer, the zero gradient condition $\mathrm{d}k^+/\mathrm{d}x_2^+ = 0$ is only valid at the wall and in the logarithmic region. Nevertheless, a blending of the universal viscous sublayer and log-layer laws is usually adopted for the second variable g or ω. In case of the k-g model, the following value is imposed at the IB-face center:

$$g = \sqrt{\frac{\mu}{\rho u_\tau^2}[(g_{\text{vis}}^+)^2 + (g_{\text{log}}^+)^2]} \tag{19}$$

where

$$g_{\text{vis}}^+ = x_2^+ \sqrt{\frac{\beta}{6C_\mu}}, \qquad g_{\text{log}}^+ = \sqrt{\frac{\kappa x_2^+}{\sqrt{C_\mu}}}$$

For the k-ω model, it is

$$\omega = \rho \frac{u_\tau^2}{\mu}[(\omega_{\text{vis}}^+)^2 + (\omega_{\text{log}}^+)^2]^{\frac{1}{2}} \tag{20}$$

where

$$\omega_{\text{vis}}^+ = \frac{6}{\beta(x_2^+)^2}, \qquad \omega_{\text{log}}^+ = \frac{1}{\kappa x_2 + \sqrt{C_\mu}}$$

The turbulent kinetic energy is directly recovered by the definition of the eddy viscosity for which the profile is assigned by Eq. (13). The above IB-face-center reconstructions are then used for the balance of the fluxes. It should be noted that the values derived do not have a true physical meaning but only serve the purpose of ensuring the correct fluxes out of the near-wall Cartesian cells.

Particularly critical are the flow stagnation, separation, and reattachment zones where, by definition, p^+ goes to infinity. By integrating along the wall normal direction x_2 and writing in plus units, Eq. (12) becomes

$$(1 + \mu_t^+)\frac{\partial u_i^+}{\partial x_2^+} = 1 + x_2^+ p^+, \qquad i = 1, 3 \tag{21}$$

The source term $x_2^+ p^+$ causes Eq. (21) to become stiff, and instabilities arise due to an incorrect evaluation of the assumed eddy viscosity profile Equation (13) too. In this case, u_τ is replaced by an alternative velocity scale u_τ^*:

$$u_\tau^* = C_\mu^{0.25}\sqrt{k} \tag{22}$$

It is clear that the equilibrium stress balance assumption, made by many wall-function methods, is valid in the limit of vanishing $x_2^+ p^+$. Nevertheless, it is commonly recognized that the presence of streamwise pressure gradient blended into a wall model can alleviate the grid dependence in computing the boundary-layer properties and should improve predictions in case of pressure-induced separations.

VII. Flow-Based Mesh Refinement

Usually, a numerical flow solution is the result of multiple simulations on differently refined meshes and several numerical scheme adjustments. This is particularly true for Cartesian methods for which the basic driver consists of obtaining acceptable results by using meshes that are as coarse as possible. Often, a first attempt is carried out on coarse meshes and, if necessary, an improvement can be achieved by using flow-based mesh adaptation.

A tool has been developed that is able to detect zones of appreciable flow activity by using sensors linked to characteristic flowfield quantities. Velocity divergence and pressure gradient are examples of possible refinement variables for compressible flow simulations. The criteria adopted make use of the following cell-center sensor:

$$\varphi_i = \phi_i \cdot \Delta_i^{3/2} \tag{23}$$

where ϕ_i is a flow parameter at the ith cell center, and $\Delta_i = \sqrt[3]{\Omega}$ is the cube root of the cell volume. The variance of the sensor over the whole flowfield is

$$\sigma_\phi = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \varphi_i^2} \tag{24}$$

where N is the total number of cells. A Cartesian cell is tagged for refinement when

$$\varphi_i > \epsilon \cdot \sigma_\phi \tag{25}$$

where ϵ is a threshold value (usually $\epsilon = 1$) determining the total number of refined cells. Five sensors are considered based on the following flow parameters: 1) velocity divergence, $\phi_i = |\nabla \cdot \mathbf{u}|_i$; 2) pressure gradient, $\phi_i = |\nabla p|_i$; 3) vorticity, $\phi_i = |\nabla \times \mathbf{u}|_i$; 4) gradient of the turbulent kinetic energy, $\phi_i = |\nabla k|_i$; and 5) gradient of the second turbulent variable, $\phi_i = |\nabla g|_i$ or $\phi_i = |\nabla \omega|_i$.

Each fully fluid cell is initially set as not to be refined. A sweep over the five sensors is carried out and, if at least one of them satisfy the refinement criteria (25), the cell is tagged for refinement. After the cell is split, the flow-state vector at the cell center is recomputed by using the linear reconstruction of Eq. (7). Thus, the new refined mesh along with the reconstructed flow quantities are passed to the flow solver for a renewed simulation.

Table 2 Characteristics of the computational meshes around the flat plate

	Mesh		
	Coarse	Medium	Fine
$\triangle_{far}/c$	0.2	0.2	0.2
Refinement levels	6	7	8
$\triangle_{wall}/c$	$3.1250 \cdot 10^{-3}$	$1.5625 \cdot 10^{-3}$	$7.8125 \cdot 10^{-4}$
Total number of cells	10,406	22,086	43,304
$x_2^+\|_{t.e.}$	96.25	46.97	23.04

Table 3 Characteristics of the computational meshes around the NACA 0012 airfoil

	Mesh		
	Coarse	Medium	Fine
$\triangle_{far}/c$	2.0	2.0	2.0
Refinement levels	9	11	12
$\triangle_{wall}/c$	$3.9062 \cdot 10^{-3}$	$9.7656 \cdot 10^{-4}$	$4.8828 \cdot 10^{-4}$
Total number of cells	10,206	37,866	74,274
$x_2^+\|_{max}$	853.2	240.2	96.72

VIII. Results and Discussion

The existing Euler flow solver [27] was upgraded to enable Navier–Stokes simulations to be performed. A validation campaign examined laminar flows for which the results are not reported here. Subsequently, the solver was modified with the addition of a wall model for high-Reynolds-number applications. The following sections present different 2-D/3-D numerical cases and comparisons with body-conforming simulations and/or experimental data.

The fully turbulent computations were carried out by using the κ-ω TNT model coupled with the two-layer approach (12). A three-stage Runge–Kutta scheme with a single evaluation of the viscous terms was adopted while cell-center gradients were recomputed at each stage. Matrix artificial dissipation and residual smoothing were switched on. The OpenMP version of the code was used to run on four to eight CPUs of the CIRA NEC-SX6 vector machine.

A procedure was developed to convert a structured surface mesh into an unstructured surface triangulation in order to assure the equivalence between the starting CAD geometries for both the body-fitted and the IB methods.

Body-conforming numerical data were obtained by using the CIRA-ZEN flow solver [32], which is based on the classical Jameson–Schmidt–Turkel scheme with scalar artificial dissipation. Fully turbulent body-fitted data were obtained by using the same κ-ω TNT model and without the use of wall functions. The freestream turbulence intensity and the ratio between turbulent and molecular viscosities were set to 0.1% and 0.1, respectively, for both codes.

A. Flat Plate

A 2-D boundary layer without pressure gradient at $Re_\infty = 1 \times 10^6$ and $\text{Mach}_\infty = 0.1$ was numerically investigated. A sharp flat plate of chord length c was placed on the lower boundary of a computational domain for which the inflow and outflow were located at the leading edge and trailing edge, respectively. The far-field boundary was placed $1.0c$ away from the plate surface. A Cartesian uniform mesh, having a root cell dimension of $0.2c$, was recursively refined near the wall to obtain three different meshes for which the main characteristics are shown in Table 2. Converged solutions were obtained on every mesh, as shown in Fig. 5a. As the mesh was refined, the skin-friction coefficient approximated the power law (Fig. 5b), and the velocity profiles in plus units at $x/c = 0.9$ followed the log law (Fig. 5c). The turbulent kinetic energy profiles in plus units collapse on the finest mesh data too.

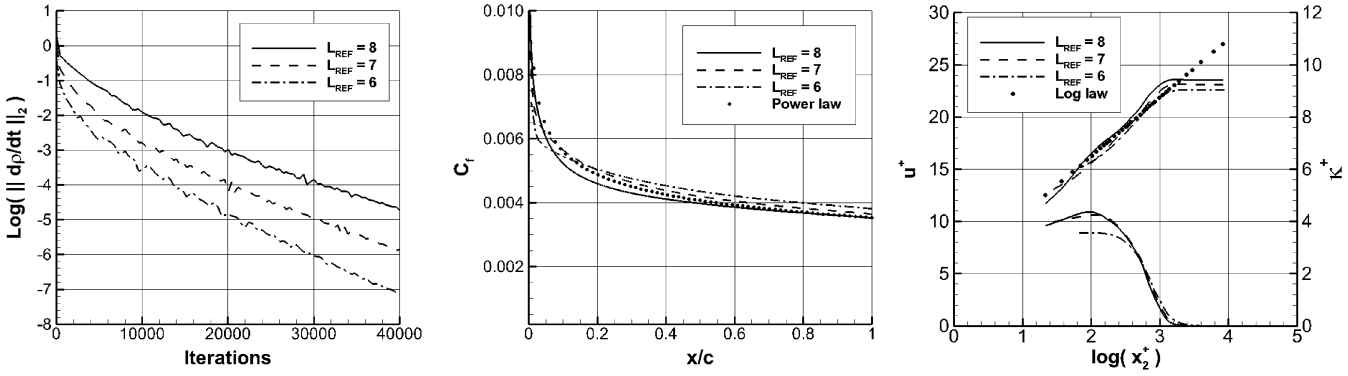
B. NACA 0012 Airfoil

The ability of the wall model to reproduce a 2-D boundary layer subject to pressure gradient effects was explored. A low-subsonic turbulent flow around the NACA 0012 airfoil of chord c at $Re_\infty = 3 \times 10^6$, $\text{Mach}_\infty = 0.3$, and zero angle of incidence was simulated. A computational domain extending $30c$ away from the body was initially discretized using cells of uniform dimensions ($\Delta x = \Delta z = 2.0c$). Three different refinement levels were considered of 9, 11, and 12, corresponding to x_2^+ maximum values of 853.2, 240.2, and 96.72, respectively, at the leading edge (see Table 3).

Mesh-converged numerical results, obtained by applying Eq. (12) without the pressure gradient term, are shown in Figs. 6a and 6b in terms of pressure and friction coefficients. The finest mesh C_P and C_f coefficients are aligned with the body-conforming data. The corresponding Cartesian nondimensional velocity profiles in log scale at different x stations are in good agreement with the body-conforming numerical data as shown in Fig. 7. Nondimensional temperature and turbulent kinetic energy comparisons demonstrate the ability of the model to correctly capture nonseparated boundary layers.

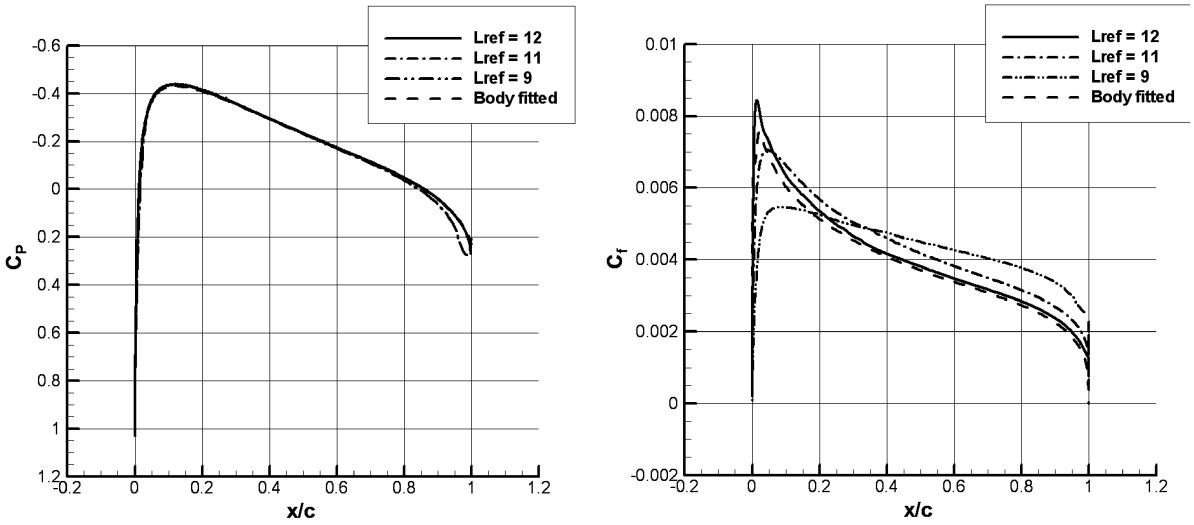
C. RAE 2822 Airfoil

The wall model was tested on more challenging 2-D flows around the RAE 2822 transonic airfoil. Both numerical and experimental data are available due to a detailed studies carried out in the framework of the European initiative on validation of CFD codes (EUROVAL) project [33]. Results of the flow conditions, named as case 9 and case 10 in the literature (see Table 4), are presented. Both flows are characterized by strong shock/boundary-layer interactions and represent severe tests for any numerical method. An initial uniform mesh of 30×30 cells extending $30c$ away from the body was processed with nine levels of geometry-based anisotropic refinements. Buffer layers of uniform cells were forced inside the boundary layer, and a mesh containing a total number of 26,531 cells was generated (see Fig. 8a).



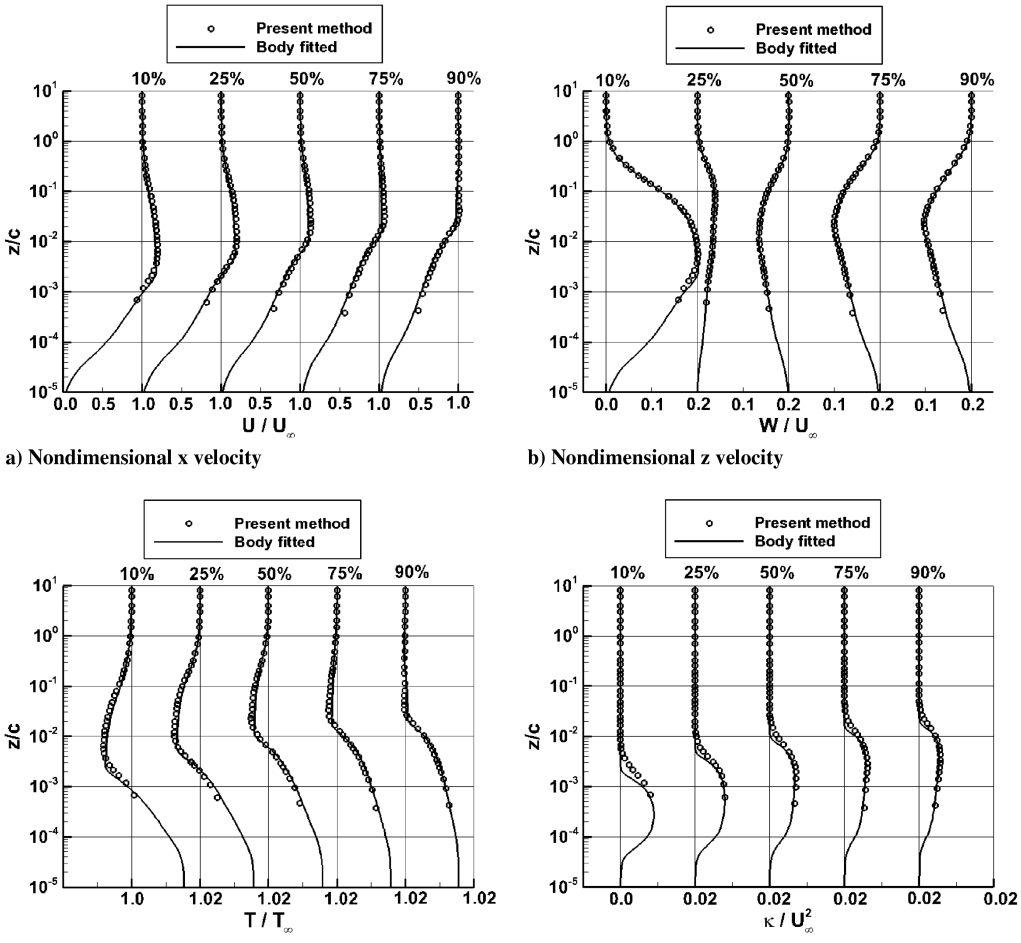
a) Continuity equation: L^2-norm residual error (plotted every 40 iterations) **b) Skin-friction coefficient** **c) Velocity and eddy viscosity profiles in plus units at $x/c = 0.9$**

Fig. 5 Flat plate: mesh convergence study: fine mesh (solid lines), medium mesh (dashed lines), and coarse mesh (dashed–dotted lines).



a) Pressure coefficient b) Skin-friction coefficient

Fig. 6 NACA 0012 airfoil: mesh convergence study. Present method on the fine (solid lines), medium (dashed–dotted lines), and coarse (dashed–dotted–dotted lines) meshes, and body-conforming reference method (dashed lines).



a) Nondimensional x velocity b) Nondimensional z velocity

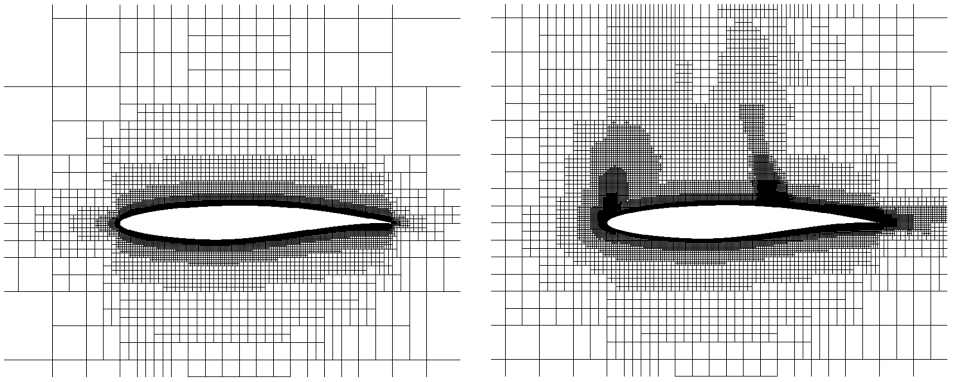
c) Nondimensional temperature d) Nondimensional turbulent kinetic energy

Fig. 7 NACA 0012 airfoil. Present method (symbols) and body-conforming reference method (solid lines).

The case 9 flow is characterized by a shock wave near 60% chord location that induces an incipient flow separation. A residual-converged solution was obtained by applying Eq. (12) without the pressure gradient term. The corresponding flow solution is represented by the curves in Figs. 9a and 9b. The C_P distribution obtained is comparable with both the experimental and body-conforming data, even if a slight underprediction of the expansion peak is visible at the

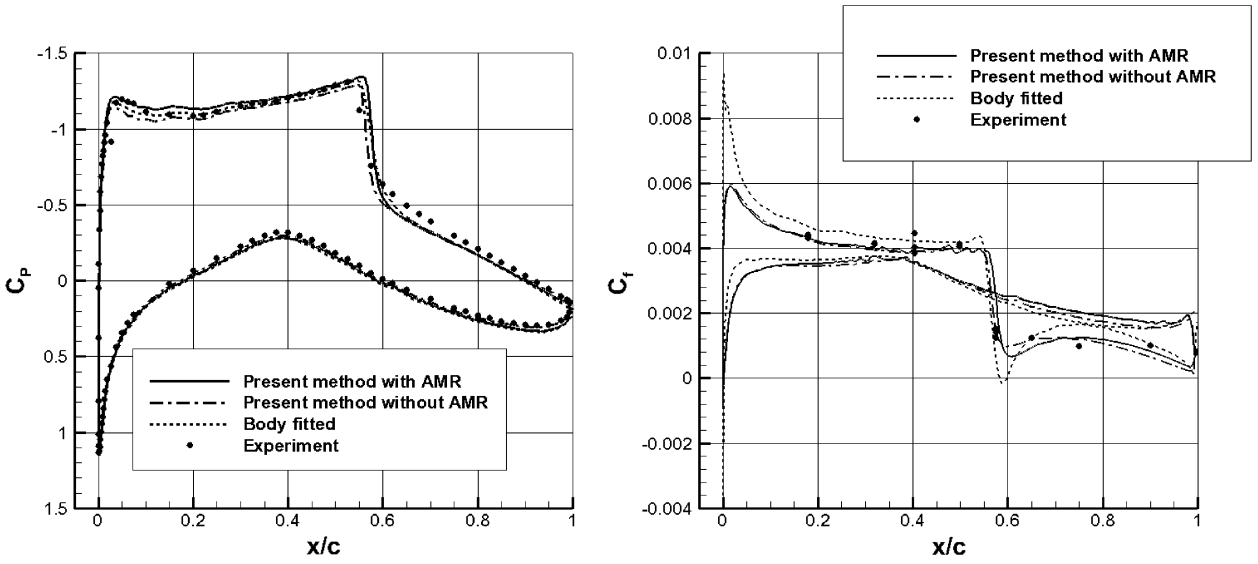
Table 4 RAE 2822 airfoil: flow conditions

RAE 2822 airfoil	Re_∞	Mach_∞	α
Case 9	6.2×10^6	0.734	2.79°
Case 10	6.2×10^6	0.754	2.57°



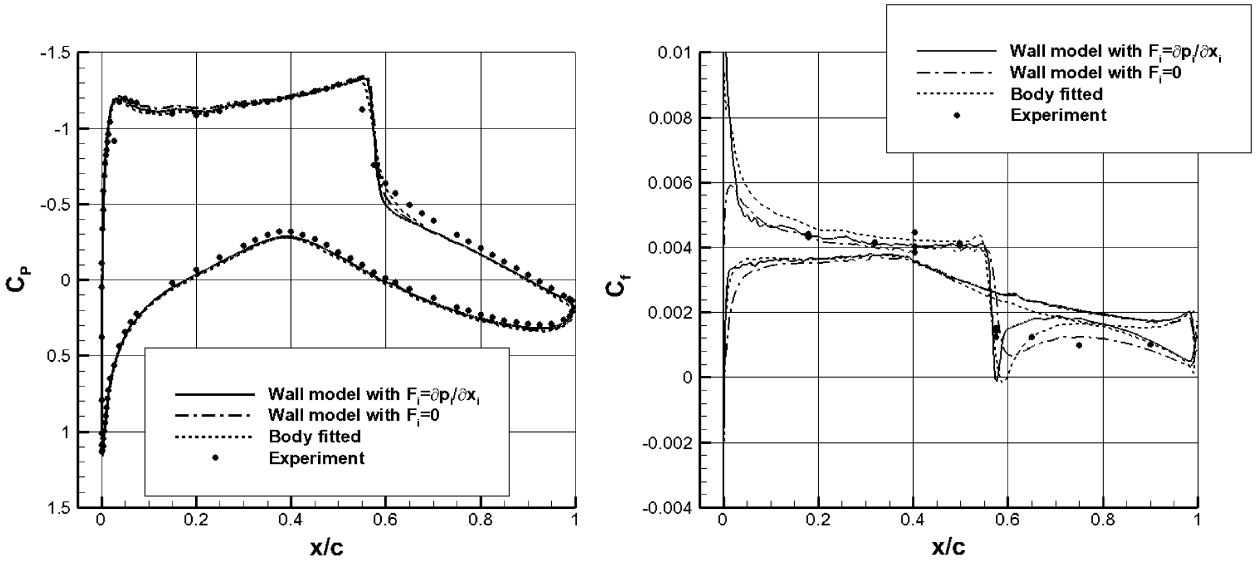
a) Geometry-adapted mesh b) Flow-adapted mesh (AMR)

Fig. 8 Cartesian meshes around the RAE 2822 airfoil.



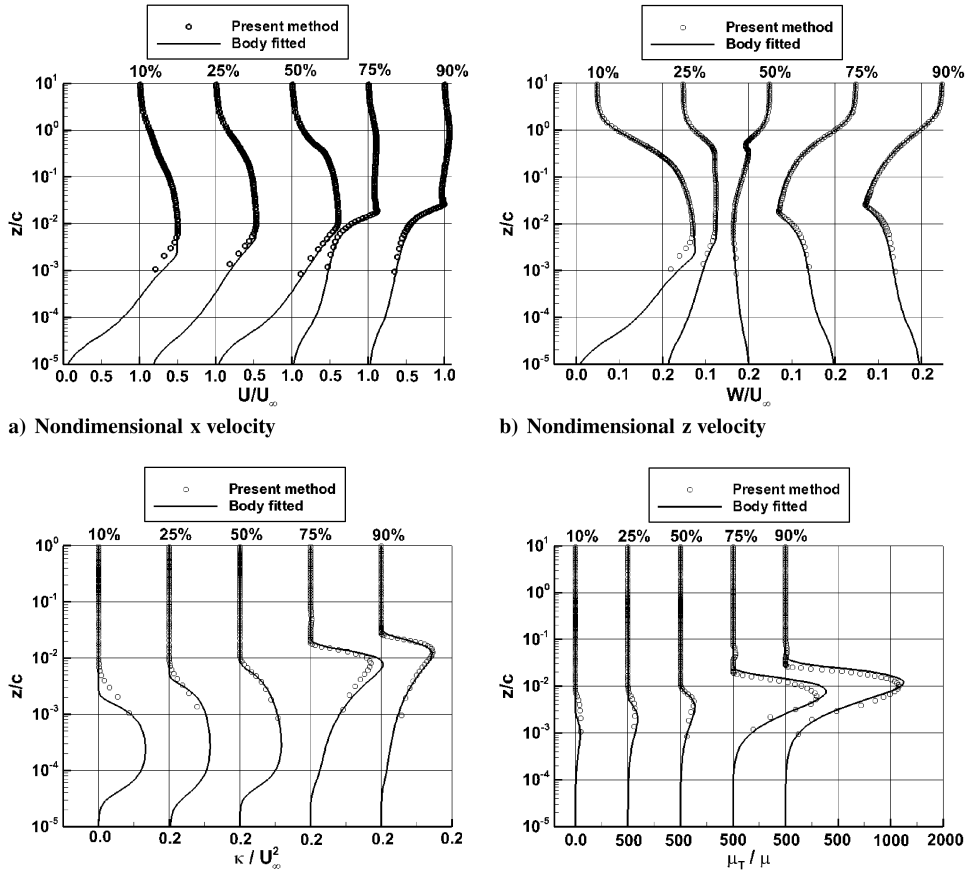
a) Pressure coefficient b) Skin-friction coefficient

Fig. 9 RAE 2822 airfoil: case 9 flow. Present method on the flow-adapted mesh (solid lines), geometry-adapted mesh (dashed–dotted lines), body-conforming reference method (dashed lines), and experimental data (symbols).



a) Pressure coefficient b) Skin-friction coefficient

Fig. 10 RAE 2822 airfoil: case 9 flow. Present method on the flow-adapted mesh. Wall model with pressure gradient (solid lines), without pressure gradient (dash–dotted lines), body-conforming reference method (dashed lines), and experimental data (symbols).



a) Nondimensional x velocity

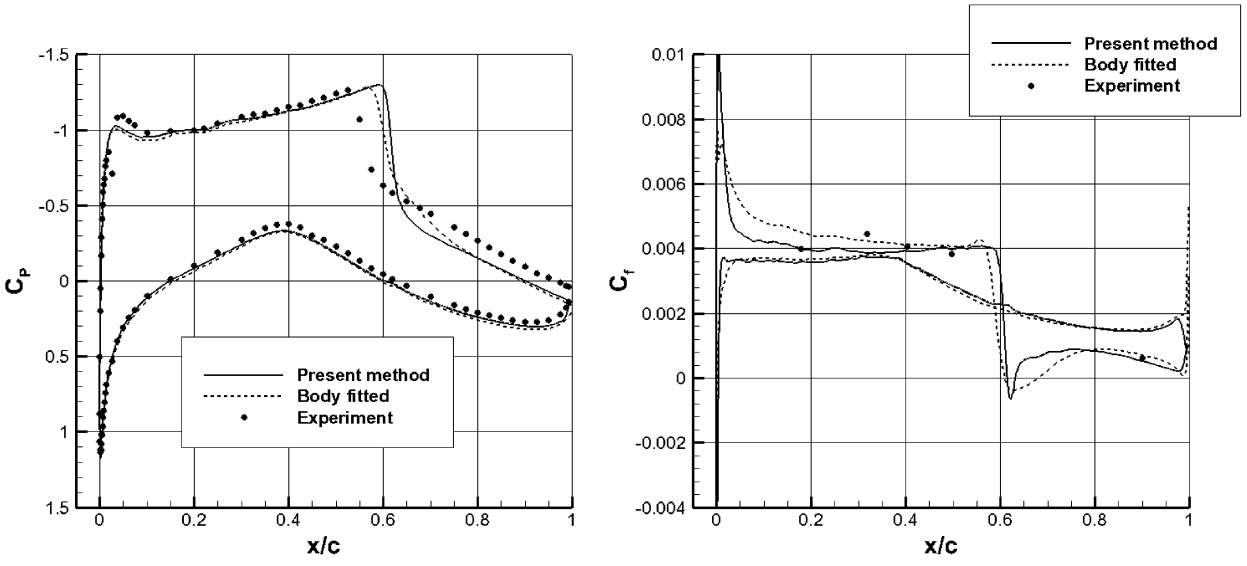
b) Nondimensional z velocity

c) Nondimensional turbulent kinetic energy

d) Nondimensional eddy viscosity

Fig. 11 RAE 2822 airfoil: case 9 flow. Wall model with pressure gradient (symbols), and body-conforming reference method (solid lines).

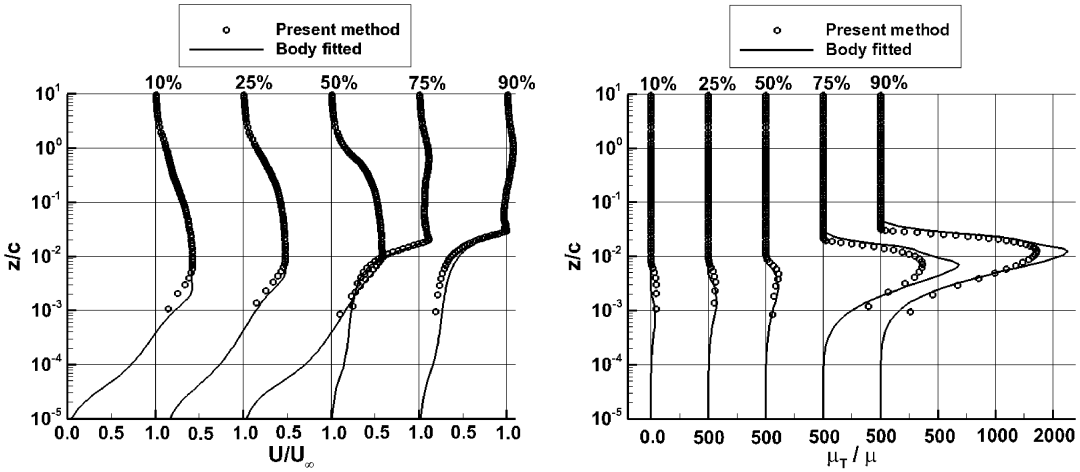
airfoil leading edge. The behavior of the skin-friction coefficient is acceptable upstream of the shock. Moreover, the shock boundary-layer interaction seems weaker than that of the body-fitted solution. A successive flow-adapted mesh refinement (AMR), consisting of 60,427 cells and $\triangle_{\text{wall}}/c = 9.7656 \cdot 10^{-4}$, was generated based on the previous flowfield data (Fig. 8b). The velocity divergence, vorticity, and pressure gradient were used as sensors. The corresponding flow solution is represented by the curves in Figs. 9a and 9b. In this case, the pressure expansion peak was recovered and the skin-friction behavior in the shock region was better than the previous computation but with less inflexion than that of the body-fitted solution. The same flow-adapted mesh was used to apply the wall model (12) with the pressure gradient term activated. Comparisons between the two versions of the wall model are shown in Fig. 10. As expected, the wall model has almost no effect on the C_P distribution (see Fig. 10a), but a reasonable improvement in the skin-friction coefficient is observed in Fig. 10b. Indeed, in spite of an x_2^+ of $\mathcal{O}(10^2)$, the position of the incipient separation and the



a) Pressure coefficient

b) Skin-friction coefficient

Fig. 12 RAE 2822 airfoil: case 10 flow. Present method on the flow-adapted mesh. Wall model with pressure gradient (solid lines), body-conforming reference method (dashed lines), and experimental data (symbols).



a) Nondimensional x velocity b) Nondimensional eddy viscosity

Fig. 13 RAE 2822 airfoil: case 10 flow. Wall model with pressure gradient (symbols), and body-conforming reference method (solid lines).

aftershock skin-friction level are in good agreement with both the body-conforming mesh solution and the experimental data. The profiles shown in log scale of various flow quantities are compared with the body-fitted data at different x stations; see Fig. 11. The nondimensional x and z velocities compare well with body-fitted data, whereas the shear layer is slightly underpredicted near the shock wave.

Flow case 10 is characterized by a stronger shock boundary-layer interaction than case 9. A zone of flow separation with a reattachment is present at the foot of the shock wave for which the intensity and position are often not captured by body-conforming simulations. Indeed, the present method and the body-fitted flow solver both slightly underpredict the expansion peak, thus shifting the shock downstream (see Fig. 12a) compared with the experimental data [33]. The skin-friction distributions, shown in Fig. 12b, are in acceptable agreement with each other, except for a different estimation of the flow separation.

The log-scale profiles of the nondimensional x velocity compare well with the body-fitted data at different x stations, as shown in Fig. 13. Similar to case 9, the aftershock turbulent mixing is a slightly lower with the present method than the body-fitted data, as indicated by the nondimensional eddy viscosity profiles shown in Fig. 13b.

The Cartesian mesh used was the same flow-adapted one as for case 9. The streamwise pressure gradient term inside the wall model (12) was activated.

D. RAE M2155 Wing

The performance of the wall model was analyzed by simulating a strongly 3-D transonic flow around the RAE M2155 swept wing at high Reynolds number. The wing was tested at the Defence Evaluation Research Agency transonic wind tunnel [34] and was the object of a numerical simulation campaign aimed at validating various turbulence models within the Advanced Viscous Flow Simulation Tools for Complete Civil Transport Aircraft Design project [32,35]. For the case 2 flow conditions ($\text{Mach}_\infty = 0.806$, $Re_\infty = 4.1 \times 10^6$) based on geometric mean chord and $\alpha = 2.5°$, a triple shock wave is present on the upper side of the wing along with shock-induced separations.

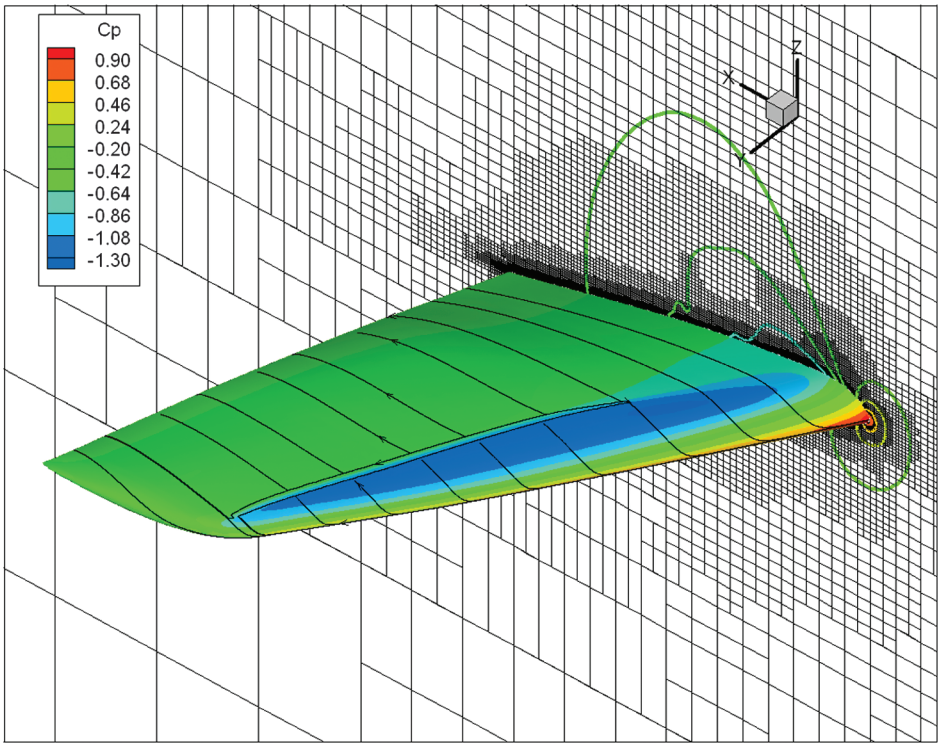
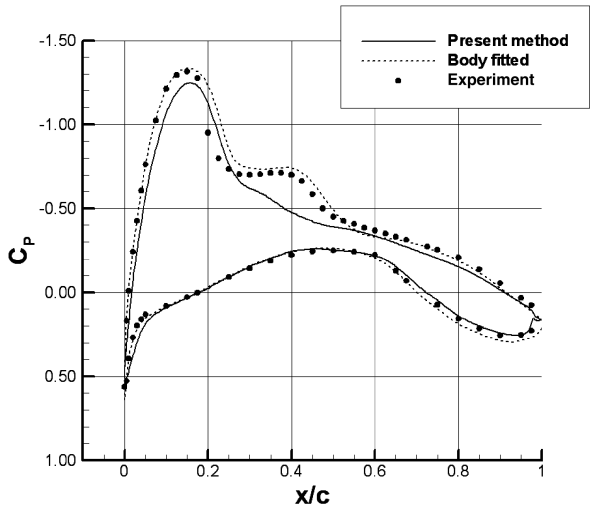
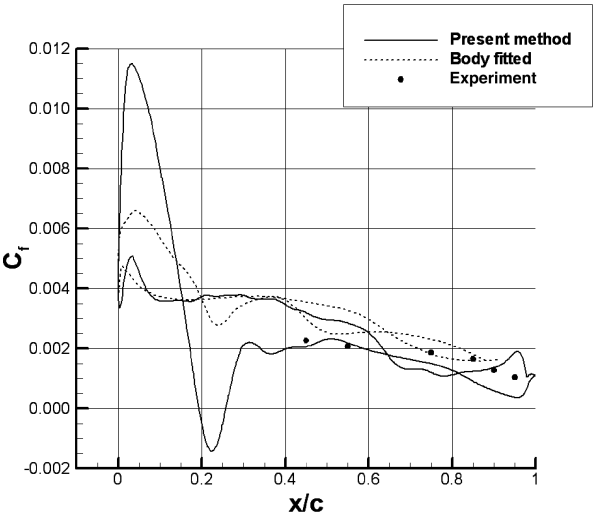


Fig. 14 RAE M2155 wing: case 2 flow on the fine mesh. Pressure contours and skin-friction lines.

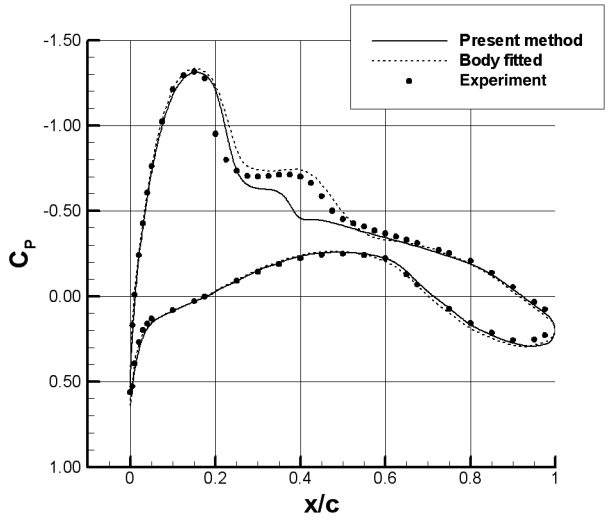


a) Pressure coefficient at $y/b = 0.2$

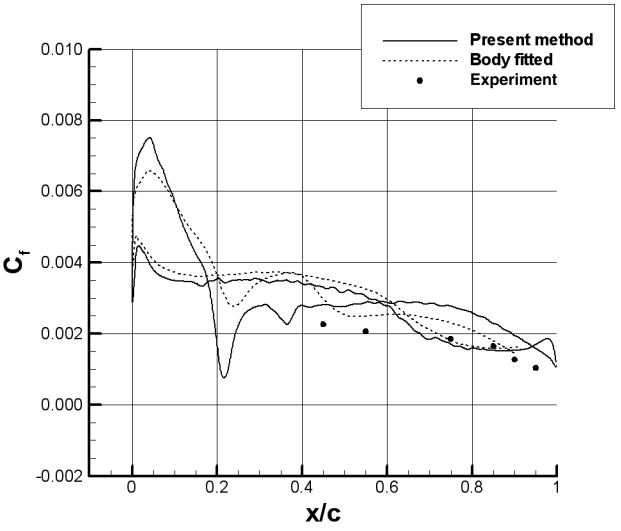


b) Skin friction coefficient at $y/b = 0.2$

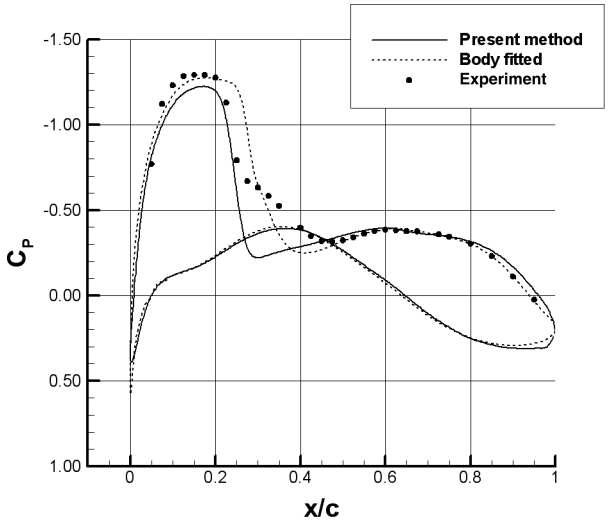
Fig. 15 RAE M2155 wing: case 2 flow on the coarse mesh. Present method (solid lines), body-conforming reference method (dashed lines), and experimental data (symbols).



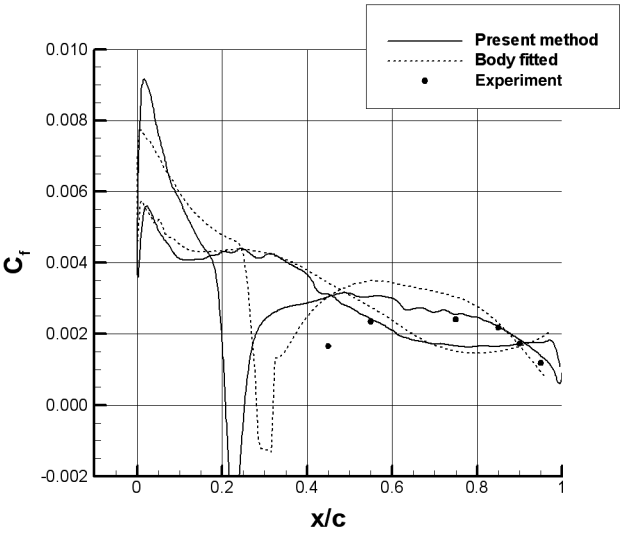
a) Pressure coefficient at $y/b = 0.2$



b) Skin-friction coefficient at $y/b = 0.2$



c) Pressure coefficient at $y/b = 0.7$



d) Skin-friction coefficient at $y/b = 0.7$

Fig. 16 RAE M2155 wing: case 2 flow on the fine mesh. Present method (solid lines), body-conforming reference method (dashed lines), and experimental data (symbols).

Using the present method, the wing, having span b, was fixed to a root boundary of a computational domain that extended a distance $4b$ in all Cartesian directions. Starting from a cell root dimension of $1b$, eight levels of geometry-based refinements were carried out to obtain a near-wall cell dimension of $\triangle_{\text{wall}}/b = 3.9062 \cdot 10^{-3}$ and a total number of 1,265,759 cells. Euler wall BCs were enforced at the upper, lower, and lateral domain-boundary faces. Classical inflow/outflow BCs based on characteristic conditions were applied. Because of the flow complexity, the pressure gradient source term inside the wall model was maintained.

The upper-surface pressure contours are shown in Fig. 14 along with a partial view of the mesh at the symmetry plane. Note the shock wave structure and the separation/reattachment zones where skin-friction lines collapse. Since huge 3-D meshes were generated, the fully vectorized and parallelized version of the code was run on the CIRA NEC-SX6 machine to shorten the CPU time resources.

A preliminary converged solution was obtained, and a successive flow-based refinement was adopted to create a new mesh of 1,596,597 cells having the same $\triangle_{\text{wall}}/b$. The pressure and skin-friction coefficients shown in Figs. 15a and 15b reveal the coarseness of the mesh when compared with the body-fitted flow solution and experimental data at $y/b = 0.2$ station.

Subsequently, a new refined mesh containing 5,147,767 cells was generated that guaranteed $\triangle_{\text{wall}}/b = 1.9531 \cdot 10^{-3}$. When compared with the previous coarse solution, a substantial improvement was obtained. In particular, at $y/b = 0.2$ station (see Fig. 16a), the expansion peak is captured and a more pronounced aftershock pressure plateau is visible but overestimated when compared with the body-fitted and experimental data. The skin-friction behavior is globally improved but, as expected, the aftershock values are not in as good agreement with body-fitted data, as shown in Fig. 16b.

This latter aspect could be due to the inability of the wall model to reproduce the adequate turbulent mixing behind the shock. Consider that the forcing points located in the log layer upstream of the shock may lie in the viscous sublayer downstream. As a result, the mesh coarseness at the foot of the shock does not enable the wall model to correctly capture the growth of the boundary layer.

Indeed, an inspection of the solution at a $y/b = 0.7$ station, shown in Fig. 16c, reveals the usual leading-edge mesh deficiency. This latter deficiency is responsible for an incorrect expansion with a subsequent shift of the shock position. Thus, even if a separation was captured by the model, its position does not agree with the body-fitted data that also have discrepancies relative to the experimental data (see Fig. 16d). It is evident that the simulation of the complex interactions between the fully 3-D boundary layer and strong adverse pressure gradients is not a trivial task, even for a body-conforming method that uses low-Reynolds BCs. To the author's knowledge, there is little evidence in the literature of IB applications to this kind of wing flow.

IX. Conclusions

A wall model for IB methods has been developed and tested on some high-Reynolds-number flows. The 2-D cases prove the ability of the model to treat attached and mild separated flows by using relatively coarse Cartesian meshes. Ad hoc flow-based adaptive meshes improved the solution quality. A 3-D flow with shock-induced separations around a transonic wing was simulated. Acceptable agreement with body-conforming and experimental data was obtained when the mesh was sufficiently refined. In this case, the wall model appeared able to capture global 3-D flow effects. Discrepancies were observed in zones of poor meshing. Further investigations are needed to establish the limits of the model in treating complex 3-D flows.

At the moment, few applications of the IB method to this kind of problem are referenced in literature. Indeed, the major drawbacks of general wall-function formulations based on the universal law of wall assumption arise where the flow equilibrium hypothesis is not satisfied. Stagnation points, strong pressure gradients, and regions of separated flows are typical examples of failure zones. It is not expected that the inclusion of pressure gradient effects in the wall model are able to resolve these problems, but the proposed turbulent IB method attempts to at least mitigate them. In conclusion, the simulation of high-Reynolds-number flows by using IB approaches seems affordable but remains a challenging field of research. Future efforts are planned to improve the wall-model performances by using, for example, turbulence models based on transport equations instead of a simple mixing-length model.

Appendix A: Weighted Least-Squares Reconstruction

The flow-state vector $\mathbf{Q}^F$ at the generic forcing F point is used by the wall-model equation (12) in order to properly apply BCs at the edge of the inner layer. A reconstruction procedure is needed, as the F point does not necessarily overlap with any of the surrounding cell centers. A cloud of N cell centers is defined in the vicinity of F in order to have a discrete representation of each flow quantity $\phi(x, y, z)$. A least-squares approach is carried out to search for an approximating linear function in the form

$$f(x, y, z) = b_0 + b_1 x + b_2 y + b_3 z = \sum_{i=0}^{3} \psi_i b_i = \Psi^T \mathbf{b} \qquad \text{(A1)}$$

where $\Psi^T = [1, x, y, z]$ is a basis vector, and $\mathbf{b}$ is the vector for which the components are the coefficients of the linear function. The values of the four unknowns b_i are obtained by minimizing the following functional:

$$G(f) = G(b_0, b_1, b_2, b_3) = \sum_{n=1}^{N} w_n [f(P_n) - \phi(P_n)]^2 \qquad \text{(A2)}$$

where $w_n = 1/|\mathbf{r}_n - \mathbf{r}^F|$ is the inverse distance between the forcing F point and the surrounding nth cell center P_n. Setting the derivatives $\partial G/\partial b_i$ equal to 0, the following system of equations is obtained:

$$\sum_{j=0}^{3} A_{ij} b_j - c_i = 0 \qquad i = 0, \ldots, 3 \qquad \text{(A3)}$$

where the matrix A_{ij} is defined as

$$A_{ij} = \sum_{n=1}^{N} w_n \psi_i(P_n) \psi_j(P_n) \qquad \text{(A4)}$$

and the known terms are

$$c_i = \sum_{n=1}^{N} w_n \psi_i(P_n) \phi(P_n) \qquad \text{(A5)}$$

Finally the coefficients of the linear function can be computed by inverting matrix A_{ij}. A minimum number of three ($n = 3$) and four ($n = 4$) cell centers are required for 2-D and 3-D computations, respectively. They have to be noncollinear in 2-D and noncoplanar in 3-D to avoid a singular least-squares matrix.

The following criteria that guarantees robustness is applied. The host cell containing the forcing F point is first determined. A cloud of nine cells ($n = 9$) in 2-D and 27 cells ($n = 27$) in 3-D are chosen among the nearest ones. A threshold distance is adopted to avoid possible zero divisions in computing the weight of the host cell.

The adopted WLSQ approximation is second-order accurate in space like the solver basic scheme. Only the cell-center quantities, and not the wall values (these are not-conserved variables), are involved in the reconstruction in order to enforce the conservative character of the whole method. Note that the inverse of matrix equation (A4) and coefficients equation (A5) depend only on metrics terms, and therefore can be computed and stored once and for all the forcing F_i points at the preprocessing phase.

Appendix B: Notes on Numerical Integration of the Wall Model

The system of equations (12) and (13) is spatially integrated along the normal to the wall direction by using a finite volume method. Dirichlet BCs are applied at the top of the inner layer for which the values are obtained from the exterior full RANS solution. At the wall, the no-slip condition is imposed. The streamwise pressure gradient is assumed constant along the normal direction and equal to the outer flow WLSQ reconstruction ($\partial p/\partial x_i = \partial p/\partial x_i|^F$).

The embedded mesh is obtained by a hyperbolic distribution of N volumes with a major clustering in the near-wall zone. In general, mesh-converged solutions are obtained in the range of cells $N \in [30, 50]$. A number of $N = 40$ cells is proven satisfactory for the test problems shown in the paper.

The diffusive fluxes are approximated by a second-order accurate in space one-dimensional CDS. Integration is obtained by a Runge–Kutta marching method that is coupled with the main flow equations. The coupling strategy is necessary to obtain a solution of the whole flowfield in a reasonable amount of time. In fact, an uncoupled approach would require a subiteration loop for each of the embedded meshes until convergence is achieved. On the other hand, the coupled approach requires the storage of flow quantities at the last time step and for each embedded mesh along the wall. Currently, the wall model retains about 20% of the whole computational cost (i.e., memory and CPU time).

A final comment regarding the computational strategy adopted to obtain a solution with the streamwise pressure gradient activated. The source term nature of $F_i = \partial p/\partial x_i$ causes robustness issues during the computation especially at the run starting phase. A remedy is to use, as startup flow, the converged solution obtained with $F_i = 0$. Thereafter, the flow is adapted to the presence of the streamwise pressure gradient by progressively augmenting its intensity.

Acknowledgments

The research activity was partially supported by the Italian Ministry of Education, University and Research in the framework of the Advanced Computational Aerodynamic Design Environment for Multidisciplinary Integrated Analysis project. The author wishes to thank Pietro Catalano for providing body-conforming data and Ainslie French for revising this manuscript.

References

[1] Coirier, W. J., and Powell, K. G., "Solution Adaptive Cartesian Cell Approach for Viscous and Inviscid Flows," *AIAA Journal*, Vol. 34, 1996, pp. 938–945. doi:10.2514/3.13171

[2] Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," *Lecture notes for 28th Computational Fluid Dynamics Lecture Series*, von Kármán Inst. for Fluid Dynamics, Rhode-Saint-Genèse, Belgium, March 1997.

[3] Peskin, C. S., "The Fluid Dynamics of Heart Valves: Experimental, Theoretical and Computational Methods," *Annual Review of Fluid Mechanics*, Vol. 14, 1982, pp. 235–259. doi:10.1146/annurev.fl.14.010182.001315

[4] Mohd-Yusof, J., "Combined Immersed-Boundary/B-Spline Methods for Simulations of Flow in Complex Geometries," *Annual Research Briefs*, Center for Turbulence Research, Stanford, CA, 1997, pp. 317–328.

[5] Mittal, R., and Iaccarino, G., "Immersed Boundary Methods," *Annual Review of Fluid Mechanics*, Vol. 37, 2005, pp. 239–261. doi:10.1146/annurev.fluid.37.061903.175743

[6] Clarke, D., Salas, M., and Hassan, H., "Euler Calculations for Multielement Airfoils Using Cartesian Grids," *AIAA Journal*, Vol. 24, 1986, pp. 353–358. doi:10.2514/3.9273

[7] Keats, W. A., and Lien, F. S., "Two-Dimensional Anisotropic Cartesian Mesh Adaptation for the Compressible Euler Equations," *International Journal for Numerical Methods in Fluids*, Vol. 46, 2004, pp. 1099–1125. doi:10.1002/fld.780

[8] Dadone, A., and Grossman, B., "Further Developments in the Three-Dimensional Cartesian-Grid Ghost-Cell Method," Reno, NV, AIAA Paper 2006-1085, 2006.

[9] Ye, T., Mittal, R., Udaykumar, H. S., and Shyy, W., "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries," *Journal of Computational Physics*, Vol. 156, 1999, pp. 209–240. doi:10.1006/jcph.1999.6356

[10] Ham, F. E., Lien, F. S., and Strong, A., "A Cartesian Grid Method with Transient Anisotropic Adaptation," *Journal of Computational Physics*, Vol. 179, 2002, pp. 469–494. doi:10.1006/jcph.2002.7067

[11] Tseng, Y. H., and Ferziger, J. H., "A Ghost-Cell Immersed Boundary Method for Flow in Complex Geometry," *Journal of Computational Physics*, Vol. 192, 2003, pp. 593–623. doi:10.1016/j.jcp.2003.07.024

[12] de Tullio, M., Palma, P. D., Iaccarino, G., Pascazio, G., and Napolitano, M., "An Immersed Boundary Method for Compressible Flows using Local Grid Refinement," *Journal of Computational Physics*, Vol. 225, 2007, pp. 2098–2117. doi:10.1016/j.jcp.2007.03.008

[13] Kalitzin, G., and Iaccarino, G., "Turbulence Modeling in an Immersed Boundary RANS Method," *CTR Annual Briefs*, Center for Turbulence Research, NASA Ames Research Center, Stanford University, Stanford, CA, 2002, pp. 415–426.

[14] Lee, J., and Ruffin, S. M., "Development of a Turbulent Wall-Function Based Viscous Cartesian-Grid Methodology," Reno, NV, AIAA Paper 2007-1326, 2007.

[15] Roman, F., Armenio, V., and Froelich, J., "A Simple Wall-Layer Model for Large Eddy Simulation with Immersed Boundary Methods," *Physics of Fluids*, Vol. 21, 2009, Paper 101701. doi:10.1063/1.3245294

[16] Ghosh, S., Choi, J., and Edwards, J. R., "Numerical Simulation of Effects of Micro Vortex Generators Using Immersed-Boundary Methods," *AIAA Journal*, Vol. 48, 2010, pp. 92–103. doi:10.2514/1.40049

[17] Kalitzin, G., Medic, G., Iaccarino, G., and Durbin, P., "Near-Wall Behavior of RANS Turbulence Models and Implications for Wall Functions," *Journal of Computational Physics*, Vol. 204, 2005, pp. 265–291. doi:10.1016/j.jcp.2004.10.018

[18] Kalitzin, G., and Iaccarino, G., "Toward Immersed Boundary Simulations of High Reynolds Number Flows," *CTR Annual Briefs*, Center for Turbulence Research, NASA Ames Research Center, Stanford University, Stanford, CA, 2003, pp. 369–378.

[19] Piomelli, U., "Wall-Layer Models for Large Eddy Simulations," *Progress in Aerospace Sciences*, Vol. 44, 2008, pp. 437–445. doi:10.1016/j.paerosci.2008.06.001

[20] Tessicini, F., Iaccarino, G., Wang, M., and Verzicco, R., "Wall Modelling for Large Eddy Simulation using an Immersed Boundary Method," *CTR Annual Briefs*, Center for Turbulence Research, NASA Ames Research Center, Stanford University, Stanford, CA, 2002, pp. 181–187.

[21] O'Rourke, J., *Computational Geometry in C*, 2nd ed., Cambridge Univ. Press, New York, 1998, pp. 257–342.

[22] Delanaye, M., Aftosmis, M., Berger, M., Liu, Y., and Pulliam, T., "Automatic Hybrid-Cartesian Grid Generation for High-Reynolds Number Flows Around Complex Geometries," Reno, NV, AIAA Paper 1999-0771, 1999.

[23] Berger, M. J., and Aftosmis, M. J., "Aspects (and Aspect Ratios) of Cartesian Mesh Methods," *Proceedings of the 16th International Conference on Numerical Methods in Fluid Dynamics*, Springer–Verlag, Heidelberg, Germany, 1998, pp. 1–12.

[24] Capizzano, F., "Algorithm for the Generation of Euler/RANS Cartesian Grids," Centro Italiano Ricerche Aerospaziali Rept. Cira-cf-07-1121, Capua, Italy, 2007.

[25] Bonet, J., and Peraire, J., "An Alternating Digital Tree (ADT) Algorithm for 3-D Geometric Searching and Intersection Problems," *International Journal for Numerical Methods in Engineering*, Vol. 31, 1991, pp. 1–17. doi:10.1002/nme.1620310102

[26] Kok, J. C., "Resolving the Dependence on Freestream Values for the κ-ω Turbulence Model," *AIAA Journal*, Vol. 38, 2000, pp. 1292–1295. doi:10.2514/2.1101

[27] Capizzano, F., "A Compressible Flow Simulation System Based on Cartesian Grids with Anisotropic Refinements," 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2007-1450, 2007.

[28] Mavriplis, D. J., "Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes," National Institute of

Aerospace Rept. 2003-06, Hampton, VA, 2003.
[29] Swanson, R., "On Central-Difference and Upwind Schemes," *Journal of Computational Physics*, Vol. 101, 1992, pp. 292–306. doi:10.1016/0021-9991(92)90007-L
[30] Durbin, P. A., and Reif, B. A. P., *Statistical Theory and Modeling for Turbulent Flows*, Wiley, New York, 2003, pp. 136–138.
[31] Wang, M., and Moin, P., "Dynamic Wall Modeling for Large Eddy Simulation of Complex Turbulent flows," *Physics of Fluids*, Vol. 14, 2002, pp. 2043–2051. doi:10.1063/1.1476668
[32] Catalano, P., and Amato, M., "An Evaluation of RANS Turbulence Modeling for Aerodynamic Applications," *Aerospace Science and Technology*, Vol. 7, No. 7, 2003, pp. 493–590. doi:10.1016/S1270-9638(03)00061-0
[33] Haase, W., Bradsma, F., Elsholz, E., Leschziner, M., and Schwamborn, D., "EUROVAL: An European Initiative on Validation of CFD Codes," *Notes on Numerical Fluid Mechanics*, Vol. 42, 1993, pp. 127–184.
[34] Firmin, M., and McDonald, M., "Measurement of the Flow over a Low Aspect Ratio Wing in the Mach Number Range 0.6 to 0.87 for the purpose of Validation of Computational Methods," Defence Evaluation Research Agency TR 92016, Farnborough, England, U.K., 1992.
[35] Gould, A., Courty, J., Sillen, M., Elsholz, E., and Abbas, A., "The AVTAC Project: A Review of European Aerospace CFD," *Proceedings of the ECCOMAS 2000 European Congress on Computational Methods in Applied Sciences and Engineering* [CD-ROM], Barcelona, 2000.

J. Sahu
Associate Editor