AIAA JOURNAL
Vol. 48, No. 6, June 2010

Robust Cartesian Grid Flow Solver for High-Reynolds-Number Turbulent Flow Simulations

Ya'eer Kidron,* Yair Mor-Yossef,† and Yuval Levy‡
Israeli Computational Fluid Dynamics Center, 38900 Caesarea Industrial Park, Israel

DOI: 10.2514/1.45817

A novel, Cartesian-grid-based flow solver is developed for predicting complex high-Reynolds-number turbulent flowfields. The Cartesian grid generator is based on the cut-cell approach using cell merge and Cartesian layer techniques. Cartesian layers imitate the structured grid approach in which the mesh is stretched gradually. Local refinement is added based on local surface curvature. As a turbulence closure model, the two equation k–ω-TNT turbulence model is successfully implemented using an unconditionally positive-convergent implicit time integration scheme. The overall flow solver's robustness and accuracy are verified using three challenging test cases. The numerical results convincingly demonstrate the robustness and accuracy of the flow solver, especially in predicting aerodynamic forces.

Nomenclature

C_d	=	drag coefficient
C_l	=	lift coefficient
C_p	=	pressure coefficient
c	=	airfoil chord
$\mathbf{E}_c$	=	mean-flow convective flux vector
$\mathbf{E}_d$	=	mean-flow diffusive flux vector
e_t	=	total energy
$\mathbf{F}_c$	=	turbulence model convective flux vector
$\mathbf{F}_d$	=	turbulence model diffusive flux vector
$\mathbf{G}$	=	mean-flow residual
h	=	specific enthalpy
h_t	=	total enthalpy
$\mathcal{I}$	=	identity matrix
k	=	turbulence kinetic energy
M_∞	=	freestream Mach number
p	=	pressure
$\mathbf{Q}$	=	mean-flow conservative vector
Q_v	=	mean-flow primitive variables
q_i	=	normal velocity
$\bar{q}_i$	=	heat conduction
$\mathbf{R}$	=	turbulence model residual
Re_∞	=	reference Reynolds number
S_i, S_j, S_M	=	signal velocities
S_{ij}	=	mean strain tensor, $\frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$
$\mathbf{S}_t$	=	turbulence model source vector
Tu	=	turbulence intensity
t	=	time
$\mathbf{U}$	=	turbulence model conservative vector
U_v	=	turbulence model primitive variables
u, v	=	Cartesian velocity vector components
α	=	angle of attack
α_ω	=	turbulence model coefficient
γ	=	ratio of specific heats
μ	=	molecular viscosity
μ_k	=	turbulent kinetic energy diffusive flux coefficient
μ_t	=	turbulent viscosity
μ_ω	=	specific turbulence dissipation rate diffusive flux coefficient
ρ	=	density
$-\rho\overline{u_i u_j}$	=	Reynolds-stress tensor
τ_{ij}	=	molecular-stress tensor
ϕ	=	limiter
ω	=	specific turbulence dissipation rate

Presented as Paper 3881 at the 19th AIAA CFD Conference, San Antonio, TX, 22–25 June 2009; received 4 June 2009; revision received 14 January 2010; accepted for publication 15 January 2010.

*Chief Technical Officer, 2 Ha'Eshel Street.
†Senior Scientist, 2 Ha'Eshel Street.
‡Chief Executive Officer, 2 Ha'Eshel Street. Senior Member AIAA.

I. Introduction

CARTESIAN-GRID-GENERATION algorithms provide the means to automatically generate a grid with very few constrains on the geometry. The principle advantage of Cartesian grids is the ease and speed with which the grids are generated and the possibility to fully automate the grid generation. The main difficulty in using Cartesian grids is in the boundary treatment of curved geometries. To alleviate this difficulty, nonboundary conforming methods are commonly used, of which there are two main classes. The first is the immersed boundary (IB) method, which enforces the wall condition indirectly through the use of force terms. The second is the Cartesian cut-cell method, which is based on the merging of cut cells with adjacent Cartesian cells to form a cell with an arbitrary shape.

The progress during the last decade in computational fluid dynamics methods on one hand, and the increase in CPU power, memory, and parallel programming techniques on the other hand, reached the point at which Cartesian grids can be considered for real, large-scale numerical simulations. Although impressive progress using a mature Cartesian-grid-based flow solver has been made (see, for example, the work by Aftosmis et al. [1]), there has been limited success in predicting turbulent flows with Cartesian grids. Most likely, the major obstacle to using Reynolds-averaged Navier–Stokes (RANS) turbulence models in conjunction with a Cartesian-grid-based flow solver lies in the lack of robust numerical methods, specifically for the turbulence model equations.

To circumvent this problem, hybrid grids are often used. Hybrid grids have an outer Cartesian mesh, while the grid near the body is body fitted; see [2,3] for examples. A different but somewhat similar approach is the *viscous* Cartesian grid method, in which a Cartesian grid that does not intersect the body is generated. The Cartesian front is then projected on to the body surface to form viscous layers. Wang and Chen [4] successfully used the *viscous* Cartesian grid method to predict turbulent flows.

Using the powerful immersed boundary method, one could keep the Cartesian-grid-generation automation advantage. Still, to alleviate the numerical difficulties that are associated with the numerical treatment of the RANS turbulence model in the near-wall region, one could use a wall function; see [5]. A successful use of the IB method

for turbulent flow simulations without the use of a wall function is presented in de Tullio et al. [6]. Another solution within the IB framework, developed by Kamatsuchi [7], is to use the idea of subgrid points to decrease the excessive number of grid cells in the boundary-layer region.

A recent development in the Cartesian-grid-based flow solver is the grid stitching approach developed by Mondal et al. [8]. In this method, the near-body Cartesian mesh nodes are slightly changed to conform to the geometry. Mondal et al. conducted turbulent flow simulations using the algebraic Baldwin–Lomax turbulence model [9].

The current work presents turbulent flow simulations using a new Cartesian-grid-based flow solver using the merged-cell Cartesian approach. As already mentioned, a pivotal issue in developing a robust Cartesian flow solver for turbulent flow simulations is the numerical treatment of the turbulence model equations, specifically, using two-equation RANS turbulence models. In particular, the end goal is a robust solver such that the RANS turbulence model could be integrated down to the wall (i.e., without the use of a wall function).

II. Governing Equations

Two-dimensional flows are governed by the compressible Favre–Reynolds-averaged Navier–Stokes equations. The unknown Favre-averaged Reynolds-stress tensor, $-\rho\widetilde{u_i u_j}$, is modeled in the current paper via a linear two-equation k–ω turbulence model. In a compact form, the RANS equations may be expressed in Cartesian coordinates as follows:

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{H}}{\partial x_j} = \mathbf{S} \tag{1}$$

The vector $\mathbf{W}$ denotes the dependent variable vector:

$$\mathbf{W} \equiv \{\mathbf{Q}^T, \mathbf{U}^T\}^T = \{(\rho, \rho u, \rho v, e_t), (\rho k, \rho\omega)\}^T \tag{2}$$

where $\mathbf{Q}$ is the vector of the mean-flow conservative variables, and $\mathbf{U}$ is the vector of the turbulence model's conservative variables. The fluid density is denoted by ρ, the Cartesian velocity vector components are denoted by u and v, and the total energy is denoted by e_t. The turbulence kinetic energy is denoted by k, whereas ω represents the specific turbulence dissipation rate. The flux vector $\mathbf{H}$ is defined as $\mathbf{H} \equiv (\mathbf{H}_c - \mathbf{H}_d)$, where $\mathbf{H}_c$ is the convective flux vector and $\mathbf{H}_d$ is the diffusive flux vector. The convective flux vector is given by

$$\mathbf{H}_c \equiv \{\mathbf{E}_c^T, \mathbf{F}_c^T\}^T = \{(\rho u_j, \rho u_j u_i + p\delta_{ij}, \rho u_j h_t), (\rho u_j k, \rho u_j \omega)\}^T \tag{3}$$

where $\mathbf{E}_c$ is the mean-flow convective flux vector and $\mathbf{F}_c$ is the turbulence model convective flux vector. The pressure is denoted by p, the total enthalpy is denoted by $h_t = h + \frac{u_i u_i}{2}$ with h being the specific enthalpy, and the Kronecker delta function is denoted by δ_{ij}. The diffusive flux vector is given by

$$\mathbf{H}_d \equiv \{\mathbf{E}_d^T, \mathbf{F}_d^T\}^T = \left\{(0, \tau_{ij}, u_j\tau_{ij} - \bar{q}_j), \left(\mu_k \frac{\partial k}{\partial x_j}, \mu_\omega \frac{\partial \omega}{\partial x_j}\right)\right\}^T \tag{4}$$

where $\mathbf{E}_d$ is the mean-flow diffusive flux vector, and $\mathbf{F}_d$ is the turbulence model diffusive flux vector. The stress tensor, τ_{ij}, can be written as the sum of the molecular-stress tensor and the Reynolds-stress tensor:

$$\tau_{ij} = \tau_{ij}^L - \rho\widetilde{u_i u_j} \tag{5}$$

where the molecular-stress tensor is given by

$$\tau_{ij}^L = \mu\left[2S_{ij} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right] \tag{6}$$

where μ is the molecular viscosity, and $S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ is the mean strain tensor. The term $\bar{q}_j$ denotes the heat conduction given by

$$\bar{q}_j = -\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial h}{\partial x_j} \tag{7}$$

where Pr is the molecular Prandtl number (set to $Pr = 0.72$), Pr_t is the turbulent Prandtl number (set to $Pr_t = 0.9$), and μ_t represents the turbulent viscosity. The turbulence diffusive flux vector coefficients, μ_k and μ_ω, are defined as

$$\mu_k = \mu + (\mu_t/\sigma_k) \tag{8}$$

$$\mu_\omega = \mu + (\mu_t/\sigma_\omega) \tag{9}$$

where σ_k and σ_ω are the turbulence closure diffusion coefficients. The mean-flow equations are closed using the equation of state for a perfect gas:

$$p = (\gamma - 1)\left[e_t - \tfrac{1}{2}\rho(u^2 + v^2)\right] \tag{10}$$

where γ is the ratio of specific heats (c_p/c_v) (set to $\gamma = 1.4$). In the present work, the source vector, denoted by $\mathbf{S}$, arises only due to the two-equation turbulence model; therefore, it may be represented by

$$\mathbf{S} \equiv \{0, \quad 0, \quad 0, \quad 0, \quad \mathbf{S}_t^T\}^T \tag{11}$$

where $\mathbf{S}_t$ is the turbulence model's source vector.

In the present work, the k–ω turbulence model developed by Kok [10] is used (also known as the TNT model). The TNT model is designed to relieve the freestream dependency of k–ω turbulence models. A clear advantage of using the TNT model in conjunction with a Cartesian grid flow solver is its topology-free feature. The TNT model source vector takes the general form:

$$\mathbf{S}_t = \begin{Bmatrix} P_k - \beta^*\rho k\omega \\ \alpha_\omega \frac{\omega}{k} P_k - \beta\rho\omega^2 + \sigma_d \frac{\rho}{\omega}\max\left[\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, 0\right] \end{Bmatrix} \tag{12}$$

where the production term, P_k, is given by

$$P_k = \left\{\mu_t\left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right] - \frac{2}{3}\rho k\delta_{ij}\right\}\frac{\partial u_i}{\partial x_j} \tag{13}$$

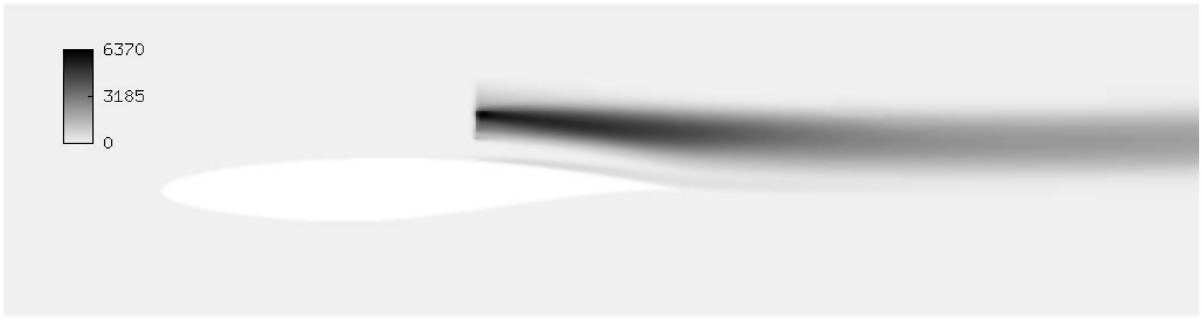
The turbulence model diffusion coefficients are $\sigma_\omega = 2.0$ and $\sigma_k = 1.5$. The remaining model constants are $\beta = 0.075$, $\beta^* = 0.09$, and $\alpha_\omega = \frac{\beta}{\beta^*} - \frac{\kappa^2}{\sigma_\omega\sqrt{\beta^*}}$, with $\kappa = 0.41$ and $\sigma_d = 0.5$. The turbulent viscosity is defined as

$$\mu_t = \rho k/\omega \tag{14}$$

Assume that the k–ω turbulence model is advanced in time using a positive scheme; ω may still reach low level values. Extremely low values of ω result in excessively high, nonphysical values of the turbulent viscosity, which in turn may lead to a breakup of the simulation. Zheng and Liu [11] showed that, by enforcing a realizability condition, a lower bound of ω can be deduced to prevent a nonphysical buildup of the turbulence viscosity. The lower bound of ω suggested by Zheng and Liu is given as

$$\omega_{\min} = \alpha_\omega\sqrt{2S_{ij}S_{ij}} \tag{15}$$

In the present work, it was found necessary to use the lower bound of ω to keep a robust simulation. Specifically, it was found that ω reached extremely low values just ahead of normal shocks, resulting an excessive turbulent viscosity buildup. Figure 1a shows an example of the nonphysical generation of turbulent viscosity behind the transonic shock in the flow about the RAE 2822 airfoil. By closely following the evolution of the numerical simulation, it was observed that the buildup process starts when the shock region approaches steady-state conditions and the shock arrives at its steady-state location. This numerical behavior starts by a decrease of ω values just ahead of the shock. An immediate result is the rapid buildup of turbulent viscosity. These excessive values propagate downstream.



a)

b)

Fig. 1 Effect of limiter on turbulent viscosity: a) no realizability constraint, and b) with realizability constraint. Grayscale levels are based on $\frac{\mu_t}{\mu}$.

Moreover, it was found that the excessive buildup process is stable and finite. Namely, the high values of turbulent viscosity reach steady-state values. Figure 1b shows the turbulent viscosity behavior for a limited solution. Note the disappearance of the high values behind the shock and the sizeable reduction in maximum levels.

III. Numerical Method

A conservative finite volume methodology to discretize the governing equations is employed. The computational domain is discretized into N_{cv} nonoverlapping control volumes. A control volume, C_v, is defined by a grid volume element, where $\partial\Gamma$ is the volume control surface, and $\mathbf{n} = [n_x, n_y]^T$ is the outward-pointing, unit normal vector to $\partial\Gamma$. The integral form of Eq. (1) for a control volume C_v can then be expressed as follows:

$$\frac{\partial}{\partial t}\int_{C_v} \mathbf{W}\,\mathrm{d}V + \int_{\partial\Gamma} (\mathbf{H}_c - \mathbf{H}_d)\cdot\mathbf{n}\,\mathrm{d}A = \int_{C_v} \mathbf{S}\,\mathrm{d}V \tag{16}$$

A. Spatial Discretization Method

The semidiscrete form of Eq. (16) for a nondeforming grid for cell i is given by

$$V_i \frac{\mathrm{d}\mathbf{W}_i}{\mathrm{d}t} = -\sum_{j\in N(i)} (\mathbf{H}_{c_{ij}} - \mathbf{H}_{d_{ij}})A_{ij} + \mathbf{S}_i V_i \equiv \mathbf{RES}_i \tag{17}$$

where $\mathbf{W}_i$ is the vector of cell-averaged conservative variables, and $\mathbf{S}_i$ is the cell source. The terms $\mathbf{H}_{c_{ij}}$ and $\mathbf{H}_{d_{ij}}$ are the convective and diffusive fluxes, respectively, normal to the interface ij shared by the cell i and its neighboring cell j; V_i is the cell area; and t is the time. The term A_{ij} is the face length of the interface ij, and $N(i)$ is the set of cell i neighbors (direct face neighbors). The vector $\mathbf{RES}_i$ signifies the right-hand side (residual) of the equation set:

$$\mathbf{RES}_i = \{\mathbf{G}^T, \mathbf{R}^T\}_i^T \tag{18}$$

where $\mathbf{G}$ is the residual of the mean-flow equations, and $\mathbf{R}$ is the residual of the turbulence model equations.

1. Mean-Flow Convective-Diffusive Method

The convective flux vector of the mean-flow equations is computed at the cell interface using the AUSM$^+$-up scheme developed by Liou [12]. The left and right state vectors of the convective flux are evaluated using second-order accuracy. The left and right states of the mean-flow primitive variables, $Q_v = (\rho, u, v, p)$, are calculated as follows:

$$(Q_v)_l = (Q_v)_i + \phi_i \nabla(Q_v)_i \cdot \mathbf{dr}_{ij} \tag{19}$$

where ϕ_i is the limiter proposed by Venkatakrishnan [13] to suppress oscillations in the solution. The vector $\mathbf{dr}_{ij}$ is the distance vector between the center of the edge ij and the center of cell i. The cell primitive variables gradient, $(\nabla Q_v)_i$, is estimated by linear reconstruction using Green's theorem. Consider a typical Cartesian stencil as shown in Fig. 2. The cell primitive gradient, for which the neighbors are denoted as $j = \{N_1, N_2, N_3, N_4, N_5, N_6\}$, is evaluated as

$$(\nabla Q_v)_i = \frac{1}{2V_i}\sum_j A_{ij}((Q_v)_j + (Q_v)_i)\mathbf{n}_{ij} \tag{20}$$

The mean-flow conservative variable vector, $\mathbf{Q}_i$, is then constructed from the mean-flow primitive variables, $(Q_v)_i$.

The mean-flow diffusive flux vector is discretized by employing central differencing. The first derivative of the primitive vector is required at the edge ij midpoints. One may have used a scheme with simple arithmetic averaging of the two adjacent cell gradients. However, this may lead to odd–even decoupling modes [14], especially when quadrilateral cells are used [15]. The modified average, suggested by Weiss et al. [16], prevents decoupling and, therefore, is adopted in the current work. Hence, the edge ij diffusive gradient is evaluated by

$$(\nabla Q_v)_{ij} = \frac{(Q_v)_j - (Q_v)_i}{|\mathbf{d}_{ij}\cdot\mathbf{n}_{ij}|}\mathbf{n}_{ij} + \left[(\overline{\nabla Q_v})_{ij} - \frac{(\overline{\nabla Q_v})_{ij}\cdot\mathbf{d}_{ij}}{|\mathbf{d}_{ij}\cdot\mathbf{n}_{ij}|}\mathbf{n}_{ij}\right] \tag{21}$$

where $\mathbf{d}_{ij}$ is the displacement vector of cell j relative to cell i (see Fig. 2) and $(\overline{\nabla Q_v})_{ij}$ is defined as follows:

$$(\overline{\nabla Q_v})_{ij} = \tfrac{1}{2}[(\nabla Q_v)_i + (\nabla Q_v)_j] \tag{22}$$

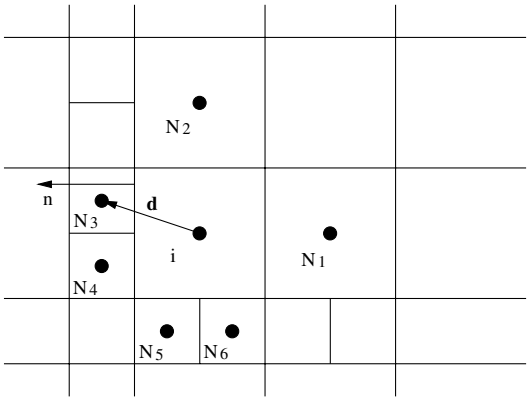


Fig. 2 Schematic Cartesian computational grid; reconstruction stencil map.

2. Turbulence Model Convective-Diffusive Method

Special attention is required for the discretization methods of the turbulence model convective and diffusive flux vectors. In the context of this work, two requirements are critical when choosing the appropriate discretization methods: 1) monotonicity, and 2) compact stencil.

Monotonicity of the discrete turbulence model equation fluxes is pivotal to the turbulence model equation solver stability, which, in turn, is the key to overall flow solver robustness. Monotonicity of the convective flux may be achieved using an upwind scheme. Enforcing monotonicity is more difficult when the diffusive flux is considered, especially on unstructured or Cartesian grids (see Lipnikov et al. [17]). The compact stencil requirement stems from the use of the unconditionally positive-convergent time integration scheme for two-equation turbulence models [18] (briefly introduced in Sec. II). The implicit operator of the unconditionally positive-convergent scheme considers *all* of the residual operator stencil. Namely, high-order methods with increasing stencil would result in a larger bandwidth of the implicit operator. To keep the implicit operator compact, the discrete form of the convective and diffusive flux vector is limited to direct face neighbors only. Using first-order accurate upwind schemes to discretize the convective flux satisfies the above two requirements. Consequently, the diffusive flux is discretized using the thin-layer approximation. The thin-layer approximation satisfies the compactness of the stencil and it is monotone when applied to the Laplace equation. In general, it is not monotone, but it is considered a practical compromise between the two requirements.

In this work the convective flux vector of the turbulence model equations is computed based on the passive scalar approach [19] within an HLLC numerical framework [20]. Using first-order accuracy, the convective flux vector of the turbulence model equations, normal to the edge ij, is evaluated as follows:

$$\mathbf{F}_{c_{ij}} = \begin{cases} q_i \mathbf{U}_i, & S_i > 0 \\ \dfrac{S_i - q_i}{S_i - S_M} S_M \mathbf{U}_i, & S_i \le 0 < S_M \\ \dfrac{S_j - q_j}{S_j - S_M} S_M \mathbf{U}_j, & S_M \le 0 \le S_j \\ q_j \mathbf{U}_j, & S_j < 0 \end{cases} \tag{23}$$

where q is the normal velocity to the interface ij, $q = un_x + vn_y$. The signal velocities S_i, S_j, and S_M are computed according to [20]:

$$S_i = \min(q_i - a_i, \hat{q}_i - \hat{a}_i), \qquad S_j = \max(q_j + a_j, \hat{q}_j + \hat{a}_j) \tag{24}$$

$$S_M = \frac{\rho_j q_j (S_j - q_j) - \rho_i q_i (S_i - q_i) + p_i - p_j}{\rho_j (S_j - q_j) - \rho_i (S_i - q_i)} \tag{25}$$

where a is the speed of sound and $\hat{a}$ and $\hat{q}$ are the Roe speed of sound and Roe normal velocity to the interface, respectively. It should be emphasized that the above choice of the HLLC scheme to approximate the turbulence convective flux vector is due to its positive characteristics. The turbulence model diffusive flux vector is evaluated according to the thin-layer approximation. Let $U_v = (k, \omega)$ denote the primitive variables of the turbulence model equations; then their gradient on an edge ij, $(\nabla U_v)_{ij}$, is evaluated as follows:

$$(\nabla U_v)_{ij} = \frac{(U_v)_j - (U_v)_i}{|\mathbf{d}_{ij} \cdot \mathbf{n}_{ij}|} \mathbf{n}_{ij} \tag{26}$$

The primitive variables and the laminar and turbulent viscosity appearing in the diffusive flux vector are obtained by simple averaging of the two adjacent cell center values.

B. Time Integration Method

Implicit time integration of the mean-flow and the turbulence equations is used in a loosely coupled manner. Because the current work concerns steady-state flows only, a first-order backward Euler time integration method is employed.

1. Mean-Flow Equations' Time Integration Method

The algebraic set of the mean-flow equations may read as

$$\left[\frac{V}{\Delta t}\mathcal{I} - \frac{\partial \mathbf{G}}{\partial \mathbf{Q}}\right]^n \Delta \mathbf{Q}^n = \mathbf{G}^n \tag{27}$$

where $\mathcal{I}$ is the identity matrix, and the Δ operator is the increment between time levels n and $n+1$. Equation (27) is solved using a point block symmetric Gauss–Seidel (SGS) method. The evaluation of the exact Jacobian, $\frac{\partial \mathbf{G}}{\partial \mathbf{Q}}$, of the high-order, nonlinear explicit operator, $\mathbf{G}$, is very complicated. To alleviate this difficulty, common practice is to approximate the convective Jacobian based on the spatial lower-order accuracy of the explicit operator. That means that the approximated Jacobian of the convective part is based on a first-order spatial accuracy of the convective explicit operator. Analytically, the diffusive flux vector may be written in a general form as $\mathbf{E}_d = \mathbf{E}_d(\mathbf{Q}, \nabla \mathbf{Q})$. Traditionally, the diffusive Jacobian is derived based on the analytical form of the flux. Alternatively, one can derive the diffusive Jacobian directly based on the discrete form of the diffusive flux. In other words, because the discrete form of the diffusive flux vector may be written as $\mathbf{E}_{d_{ij}} = \mathbf{E}_{d_{ij}}(\mathbf{Q})$, the diffusive implicit operator may be derived as follows:

$$\frac{\partial \mathbf{E}_{d_{ij}}}{\partial \mathbf{Q}} \Delta \mathbf{Q} = \underbrace{\frac{\partial \mathbf{E}_{d_{ij}}}{\partial \mathbf{Q}_{d_{ij}}} \frac{\partial \mathbf{Q}_{d_{ij}}}{\partial \mathbf{Q}_i} \Delta \mathbf{Q}_i + \frac{\partial \mathbf{E}_{d_{ij}}}{\partial \mathbf{Q}_{d_{ij}}} \frac{\partial \mathbf{Q}_{d_{ij}}}{\partial \mathbf{Q}_j} \Delta \mathbf{Q}_j}_{\text{direct face}} + \sum_m \frac{\partial \mathbf{E}_{d_{ij}}}{\partial \mathbf{Q}_{d_{ij}}} \frac{\partial \mathbf{Q}_{d_{ij}}}{\partial \mathbf{Q}_m} \Delta \mathbf{Q}_m \tag{28}$$

where m denotes any cell that belongs to the support of the diffusive flux stencil such that $m \neq i$, j, and $\mathbf{Q}_d = \{u, v, w, T\}^T$. As already mentioned, to keep a compact form of the implicit operator, only direct face neighbors are considered. Therefore, the third term on the right-hand side of Eq. (28) is neglected.

Within the framework of a direct face neighbor implicit operator, the implicit delta form of the mean-flow equations may be written as

$$\left[\frac{V}{\Delta t}\mathcal{I} + \sum_{j \in N(i)} \left(\frac{\partial \mathbf{E}_{c_{ij}}}{\partial \mathbf{Q}_i} - \frac{\partial \mathbf{E}_{d_{ij}}}{\partial \mathbf{Q}_i}\right)\right]^n \Delta \mathbf{Q}_i^n = \mathbf{G}_i^n - \sum_{j \in N(i)} \left[\left(\theta \frac{\partial \mathbf{E}_{c_{ij}}}{\partial \mathbf{Q}_j} - \frac{\partial \mathbf{E}_{d_{ij}}}{\partial \mathbf{Q}_j}\right) \Delta \mathbf{Q}_j\right]^n \tag{29}$$

In the current work, the approximated Jacobian of the mean-flow convective part is evaluated using the van Leer Jacobian [21]. To enhance flow solver robustness, the off-diagonal convective Jacobian elements were multiplied by a scalar θ. Based on numerical experiments the optimum value of θ was found to be $\theta = 0.85$. Thanks to this simple implicit operator modification, the Courant–Friedrichs–Lewy (CFL) number limit is significantly increased.

To improve convergence to steady state, the B2 scheme proposed by Batten et al. [20], is used. The B2 scheme is a modified variant of the backward Euler time integration method. Denoting the time integration given in Eq. (27) by

$$\Delta\mathbf{Q}^n = \mathrm{B1}(\mathbf{Q}^n, \Delta t) \Rightarrow \mathbf{Q}^{n+1} = \mathbf{Q}^n + \Delta\mathbf{Q}^n \tag{30}$$

the B2 scheme is given via two successive modified B1 steps as follows:

$$\begin{aligned} &\text{First step: } \Delta\mathbf{Q}^* = B1(\mathbf{Q}^n, \Delta t/2) \Rightarrow \mathbf{Q}^* = \mathbf{Q}^n + \Delta\mathbf{Q}^* \\ &\text{Second step: } \overline{\Delta\mathbf{Q}} = B1(\mathbf{Q}^*, \Delta t) \Rightarrow \mathbf{Q}^{n+1} = \mathbf{Q}^* + \overline{\Delta\mathbf{Q}}/2 \end{aligned} \tag{31}$$

As pointed out in [20], the B2 scheme may alleviate convergence difficulties that are associated with the high-frequency fluctuations of limiters. In particular, it was found in the current work that the B2 scheme considerably alleviates the mismatch of the implicit and explicit operators.

2. Turbulence Model Equations' Time Integration

Similar to the algebraic set of the mean-flow equations, the algebraic set of the turbulence model equations is given as

$$\left[\frac{V}{\Delta t}\mathcal{I} - \frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right]^n \Delta\mathbf{U}^n = \mathbf{R}^n \tag{32}$$

In the current work, the same point block SGS time integration method that is employed for solving the mean-flow equations is used to advance the turbulence model equations in time. It should be noted that the B2 scheme is not used for the time integration of the turbulence model equations; therefore, Eq. (32) is employed only in the second step of the B2 scheme of the mean-flow equations.

A straightforward implementation of the turbulence model equations' exact Jacobian, $\frac{\partial\mathbf{R}}{\partial\mathbf{U}}$, usually leads to an unstable scheme that exhibits convergence and positivity preserving difficulties. These numerical difficulties may have a strong impact on the turbulence model's numerical stability as well as on overall flow solver robustness. Therefore, a stable time integration method of the turbulence model is necessary. The unconditionally positive-convergent implicit time integration scheme for two-equation turbulence models developed by Mor-Yossef and Levy [22] is adopted here.

For completeness, the general outline of their scheme is described herein. Let a matrix $\mathcal{M}$ be an approximation to $-\frac{\partial\mathbf{R}}{\partial\mathbf{U}}$ such that it fulfills the following two conditions: 1) $\mathcal{M}$ is an M matrix, and 2) $\mathbf{R} + \mathcal{M}\mathbf{U}$ is a nonnegative vector (i.e., all its entries are nonnegative).

Substituting the matrix $-\frac{\partial\mathbf{R}}{\partial\mathbf{U}}$ in Eq. (32) with the matrix $\mathcal{M}$ yields

$$\left[\frac{V}{\Delta t}\mathcal{I} + \mathcal{M}\right]^n \Delta\mathbf{U}^n = \mathbf{R}^n \tag{33}$$

In the limit of $\Delta t \to \infty$, Eq. (33) becomes

$$\mathcal{M}^n \Delta\mathbf{U}^n = \mathbf{R}^n \tag{34}$$

Because $\mathcal{M}$ is an M matrix, classical numerical theory [23] shows that a system of the form of Eq. (34) is an unconditionally convergent scheme (because the spectral radius of an M matrix is lower than one). In a similar manner, the unconditional positivity of the scheme may be proved as follows. The delta form given in Eq. (34) may be split as follows:

$$\mathcal{M}^n \mathbf{U}^{n+1} = \mathbf{R}^n + \mathcal{M}^n \mathbf{U}^n \tag{35}$$

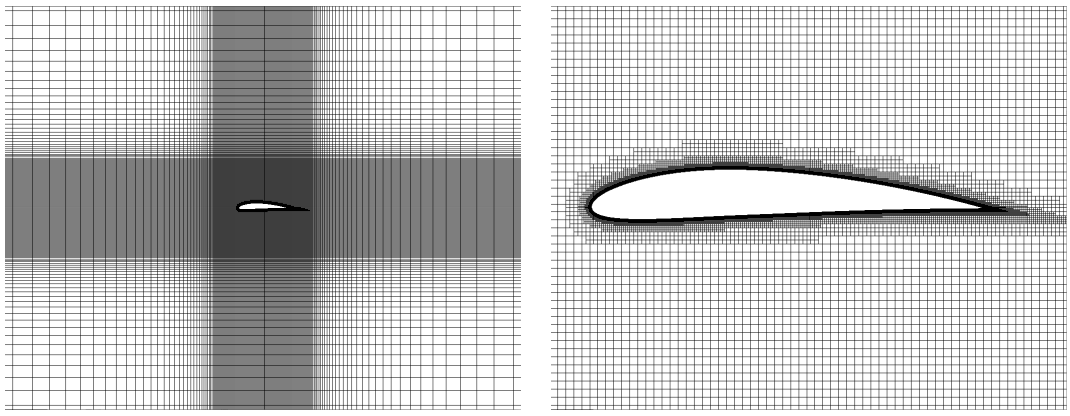
Note, based on condition 2, the right-hand side of Eq. (35) is a nonnegative vector. Furthermore, because the inverse of $\mathcal{M}$ is a nonnegative matrix, the positivity of the vector $\mathbf{U}^{n+1}$ is guaranteed; therefore, the unconditional positivity nature of Eq. (34) is proven. One should be aware that the first condition, namely, that $\mathcal{M}$ is an M matrix, guarantees the convergence of the system of equations but not its positivity. Namely, the system of equations may converge to a negative solution. Only the addition of the second condition guarantees a nonnegative solution. A detailed description of the scheme and the construction of the matrix $\mathcal{M}$ may be found in Mor-Yossef and Levy [18].

IV. Cartesian Grid Generation

Cartesian grid generation can be conducted in various fashions. Ideally, an isotropic Cartesian grid should be used. However, such a grid would consist of an overwhelming number of cells compared to contemporary computer power. This is particularly true for turbulent flow simulations in which the cells near the wall must be sufficiently small to resolve the boundary layer. Typically, Cartesian grids are locally refined. Grid generators differ by the initial Cartesian cell or grid before the refinement, and the refinement method, for example, static refinement as opposed to dynamic refinement during the simulation, isotropic or anisotropic refinement, etc.

The first step in generating a Cartesian grid is the definition of the initial, body-penetrating grid. The present work does not require an initial uniform grid, and it does not limit the initial grid to a single "root" cell. On the contrary, the initial body-penetrating grid can be anisotropic, thereby helping to refine the cells around the body before the refinement step. As can be seen in Fig. 3a of a NACA 4412 airfoil, the initial stretched grid has the same characteristics of a structured grid. The cells in the near field around the body are very small, whereas the cells in the far field are large. In the NACA 4412 case shown, the initial "core" Cartesian grid is isotopic (see Fig. 3b).

One of the advantages of body-fitted structured grids is that the extrusion of the grid generates adjacent cells for which the volumes differ slightly. Cells gradually increase from their minute size near the wall to a large size far away from the body. However, for Cartesian grids, no matter how close or far from the body, when cells are locally



a) b)

Fig. 3 Cartesian grid about the NACA 4412 airfoil: a) far-field grid, and b) isotropic center grid.

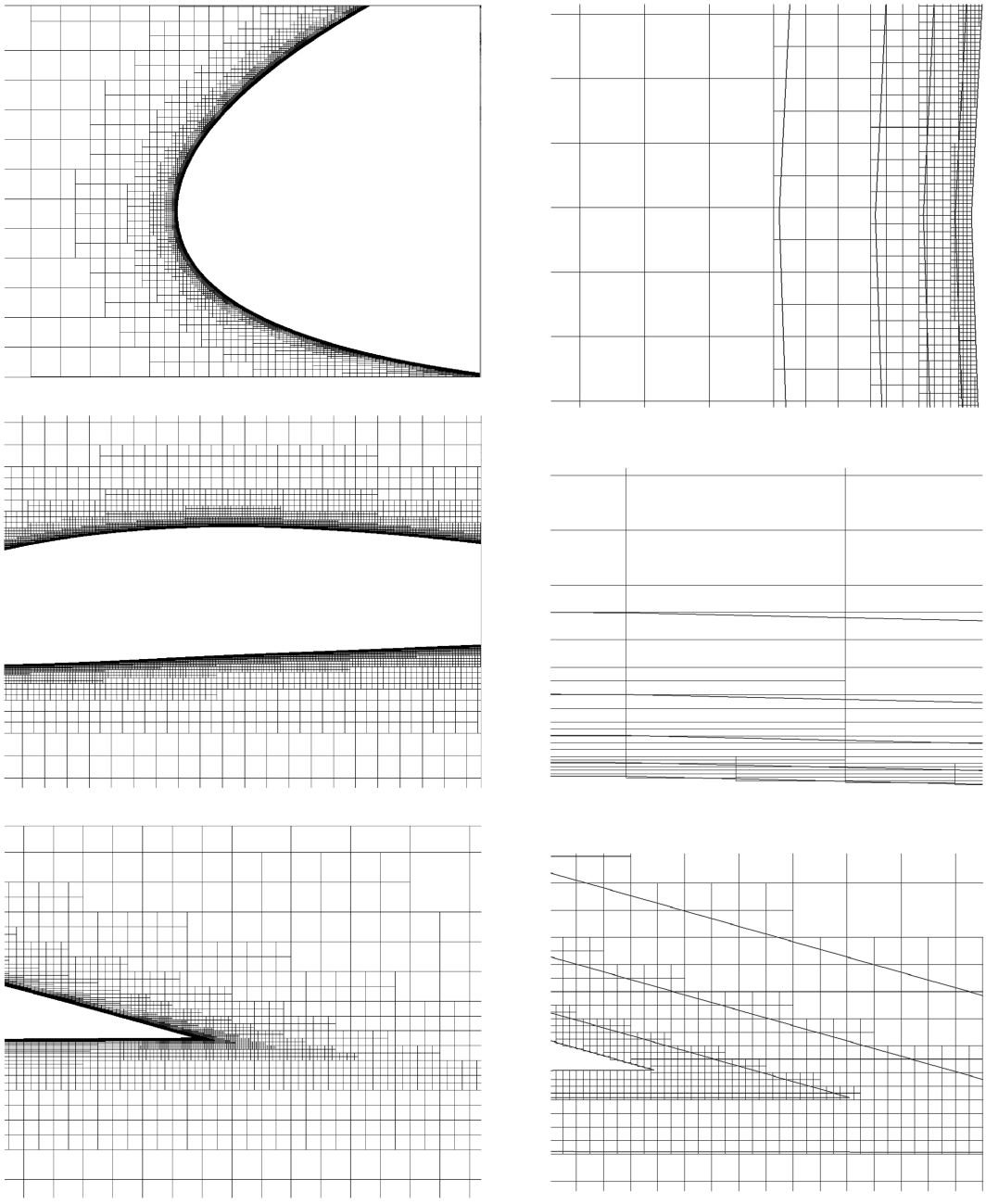


Fig. 4 Cartesian grid about the NACA 4412 airfoil.

refined there is a leap of at least twice their volume. This is especially significant near walls at which a considerable change in cell dimensions hampers the solution accuracy and, in turn, harms the quality of the flow simulation. The advantage of body-fitted structured grids is harnessed in the present work, in an attempts to generate Cartesian grids that mimic the characteristics of a similar structured grid. To this end, virtual layers are formed in an increasing distance from the wall with the aim of obtaining a desired ratio between neighboring cell dimensions, outside and inside of each layer. These layers are also responsible for increasing the distance between successive leaps, resulting in more gradual cell size changes. An example of such layers on a Cartesian grid about a NACA 4412 airfoil are shown in Fig. 4, superimposed on the final grid.

The local refinement of each layer may be different. It can be isotropic, meaning that a cell is divided in all dimensions, even if the original cell is not uniform, as shown in Fig. 4. Near walls, isotropic refinement may result in an excessive number of cells. In the current work, cell refinement of inner layers is also based on the body local curvature and the cell orientation toward the body (see Fig. 4). The leading- and trailing-edge regions of the airfoil have high curvature, whereas other areas have very small curvature; therefore, the cells in the inner levels around the edges are finer.

In effect, the layers define enclosures that are not limited to the body shape. One could define additional enclosures to better resolve a shock or a wake, thus making the refinement process nearly automatic and greatly improved. When the local refinement step is complete, neighboring cells may still vary in size in more than the allowable ratio (e.g., a factor of 2). This may happen when, for example, the single face of a cell shares an interface with more than two other cells' faces. For such cases, an additional step is conducted and these cells are further divided to obtain an improved grid. Throughout the process, body cells are eliminated, and cut cells are formed around the body. After the final grid is refined, the grid generator examines the cut cells. The cut cells can be in arbitrary shapes. Particularly, they can have a very high aspect ratio or their volume may differ greatly from their neighbors. These properties may hinder convergence or stop it altogether. To minimize that risk, as a final step the generator merges these cells with adjacent cells to

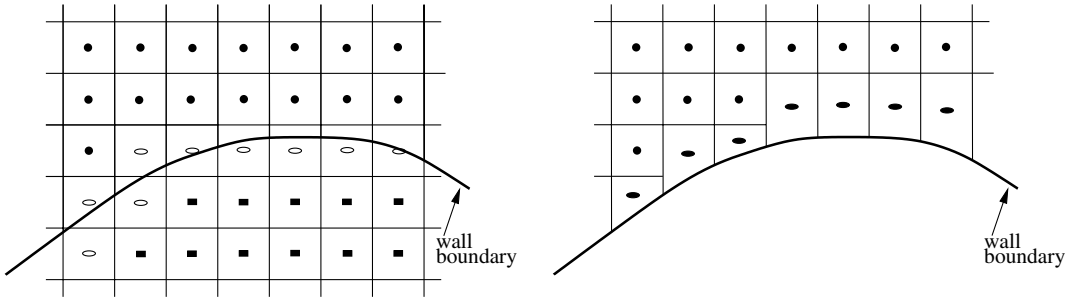


Fig. 5 Near-wall Cartesian grid cells: a) before merging, and b) after merging.

form arbitrarily shaped cells. Figure 5 illustrates the idea of merging cut cells. Cells that are marked by filled circles at the Cartesian cell center are completely outside the body (flow region), whereas cells that are marked by filled squares are entirely inside the body. Cells that are marked by unfilled ellipses represent cut cells before the merging process, whereas cells that are marked by filled ellipses represent merged cut cells of arbitrary shapes. Note that the filled ellipses mark the cell center of the merged cells. There are two main criteria to merge cut cells. The first is designed to prevent high-aspect-ratio cells. It is conducted by merging all cells for which the Cartesian cut-cell center resides inside the body. The second is designed to prevent high volume ratio between neighboring cells. The small cell is combined with the neighboring cell for which the interface with the merged cell has the largest length.

V. Results and Discussion

Three well-known test cases are examined in this paper. The cases include the transonic flow about the RAE 2822 airfoil, the high incidence flow about the NACA 4412 airfoil, and the flow about the multi-element NHLP-2D airfoil at two angles of attack. The initial conditions of the mean-flow equations and of the turbulence model equations are uniform and are based on the freestream conditions. In all conducted simulations, the mean-flow equations were advanced in time using a CFL number of 100, whereas an infinite CFL is used for the turbulence model equations' time integrations. Namely, the turbulence model equations were advanced in time using Eq. (34).

The boundary conditions considered in this work are set as follows. On the solid wall, a no-slip condition is imposed by setting the velocity components to zero. The pressure is extrapolated from the interior cell. The turbulent kinetic energy is set to $k = 0$, whereas the turbulent dissipation rate at the wall is set to $\omega = \frac{60\nu}{\beta(\Delta y_1)^2}$, where Δy_1 is the distance of the first cell center neighboring the wall from the wall (determined according to [24]). The far-field boundary conditions of the mean-flow variables are based on simplified one-dimensional characteristic boundary conditions. The inflow turbulent kinetic energy is evaluated according to the relation $k = \frac{3}{2}(Tu \cdot U_\infty)^2$, where Tu is the turbulence intensity. For all of the simulations the turbulence intensity was set to $Tu = 0.1\%$. The inflow turbulence specific dissipation rate is set so that the inflow turbulent viscosity is equal to $\mu_{t\infty} = 0.01\mu_\infty$. At the outflow boundary, the turbulence variables are extrapolated from the interior.

Surface pressure distributions that are computed from Cartesian grid solutions suffer from the well-known staircase phenomenon. To alleviate that, for the purpose of postprocessing only, the following procedure is devised and followed. The surface pressure distribution along the airfoil in this work is determined based on the initial geometry database. The pressure is derived by summing up the forces on a complete database body polygon, from all the cut cells that share that polygon, and dividing the summation by the database body polygon length. The surface pressure is assigned to the center of the polygon; thereby, a complete pressure profile around the airfoil can be drawn.

A. Transonic Flow About the RAE 2822 Airfoil

The transonic turbulent flow simulation past the supercritical RAE 2822 airfoil is conducted as a test case involving shock/boundary-layer interaction. The flow conditions that are used in the current work are a Mach number of $M_\infty = 0.734$, a Reynolds number of $Re_\infty = 6.5 \times 10^6$, and an incidence angle of $\alpha = 2.54$ deg. These conditions are the corrected conditions of case 9 from the experiment by Cook et al. [25].

The computational grid contains roughly 500,000 cells, extended to approximately 30 chord lengths. The Cartesian grid is composed of 11 layers of variable resolution surrounding the airfoil (see Fig. 6; for the sake of clarity of the presentation, the grid that is presented is coarser than the grid used in the computations). Note that the curved lines drawn in Fig. 6b signify the layer barriers, demonstrating the *wave-frontlike* structure of the grid. The first grid point is placed at a distance of 1×10^{-5} chord lengths above the airfoil surface. This results in $y^+ \leq 3.5$.

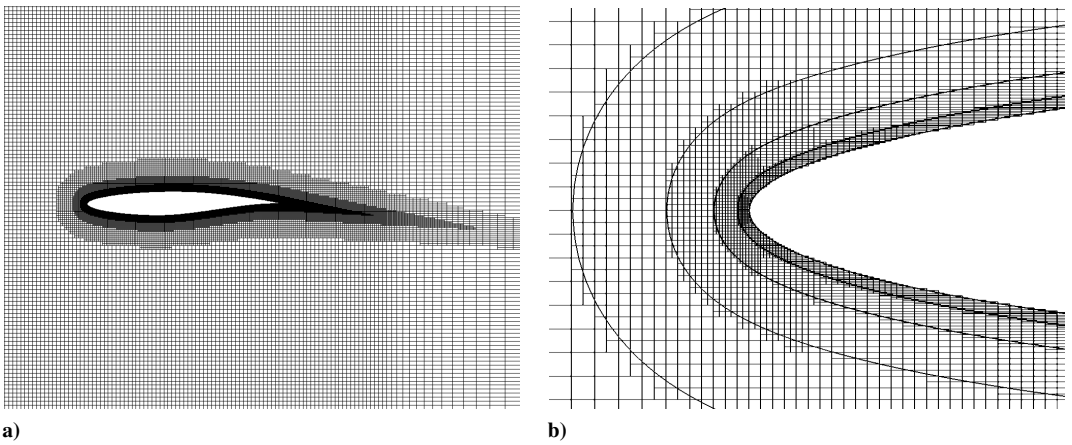


Fig. 6 Typical Cartesian grid used for RAE 2822 airfoil: a) far-field view, and b) leading-edge near-field view.

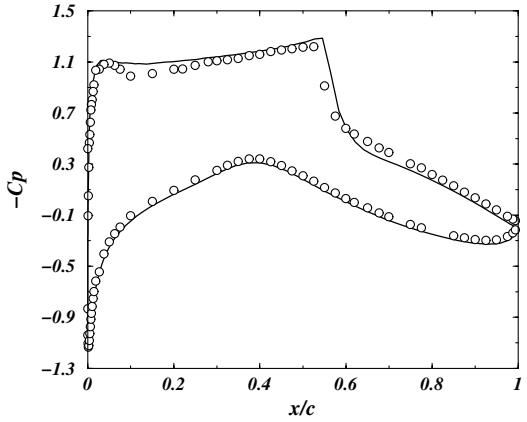


Fig. 7 Comparison of surface pressure coefficient distribution along the RAE 2822 airfoil. Line: calculation, circles: experiment [25].

Figure 7 shows a comparison of the surface pressure coefficient distribution along the airfoil between the computational and experimental results [25]. An excellent agreement is obtained. Specifically, the shock is sharply captured with a very high resolution, pointing to the high accuracy of the flow solver and the method as a whole. Figure 8 shows the convergence time history of the mean-flow equations and the turbulence model equations. The mean-flow equations' residual drops nearly 5 orders of magnitude, whereas the turbulence model equations' residual drops approximately 11 orders of magnitude. Note that all residual plots start from zero and that the turbulence equations residual has an immediate drop that is followed by an additional drop as the mean-flow solution progresses. The mean-flow convergence history pattern is characterized by small oscillations with an increasing number of spikes. Most of the spikes originate from the shock surroundings; therefore, it is assumed that these spike are caused by the limiter function. Some of the other spikes are observed in arbitrary grid cells. These may be explained by either limiter fluctuations or by the nonpositivity effect of the diffusive operator. It is important to note that all spikes reflect localized limited errors that do not hinder the global convergence in any manner. It should be emphasized that the turbulence model's dependent variables, namely, ρk and $\rho\omega$, maintained positive values throughout the simulation duration. This is a direct result of the positivity preserving convergent scheme.

B. Separated Flow About the NACA 4412 Airfoil

The NACA 4412 test case is a well-known test case for high-lift separated flows. In the current work, a single incidence of $\alpha = 13.87$ deg at a Reynolds number of $Re_\infty = 1.52 \times 10^6$ and a Mach number of $M_\infty = 0.2$ is simulated and compared with the experimental results reported by Coles and Wadcock [26]. At this incidence, a steady trailing-edge separation is present.

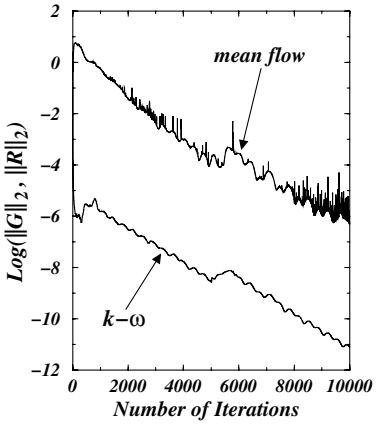


Fig. 8 Convergence history of the mean-flow and turbulence model equations for the RAE 2822 airfoil.

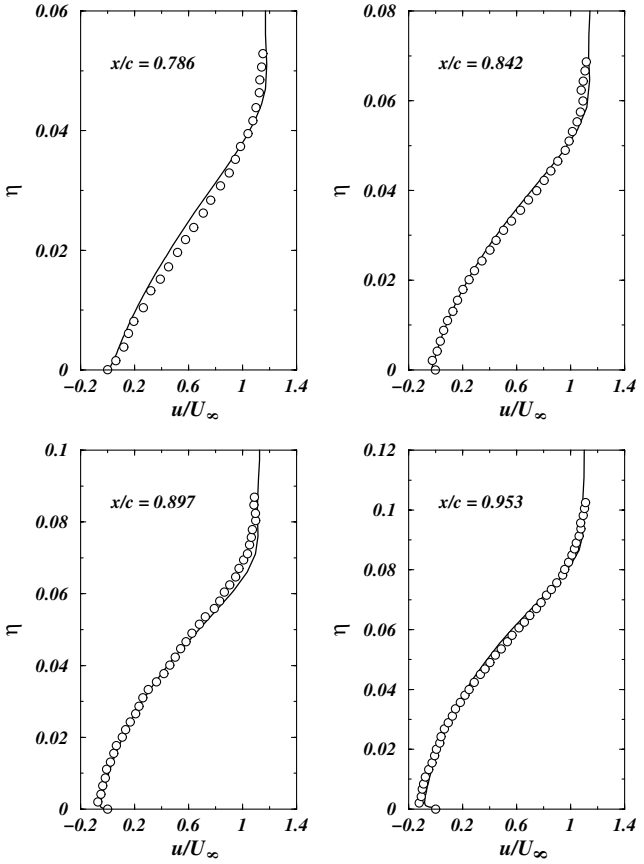


Fig. 9 Comparison of the streamwise velocity for the NACA 4412. Line: calculation, circles: experiment [26]. η is the coordinate perpendicular to the airfoil upper surface.

The computational grid contains roughly 250,000 cells, extended to approximately 30 chord lengths. The Cartesian grid is composed of 11 layers of variable resolution surrounding the airfoil. The first grid point is placed at a distance of 1×10^{-5} chord lengths above the airfoil surface. This results in $y^+ \leq 2.35$. Figure 9 shows a comparison of the calculated streamwise velocity profiles with the experimental data at four stations along the upper airfoil surface, one before the separation point and three after the separation point. The calculated results are in excellent agreement with the experiments, demonstrating the ability of the flow solver to correctly predict separated flows. The convergence time history of the NACA 4412 cases is presented in Fig. 10. Similar to the RAE 2822 case, good and clear convergence is exhibited. The mean-flow equations' residual converged approximately 5 orders of magnitude, whereas the turbulence equations' residual dropped almost 7 orders of magnitude.

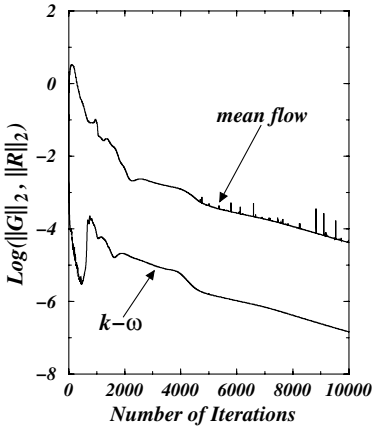


Fig. 10 Convergence time history of the mean-flow and turbulence model equations for the NACA 4412 case.

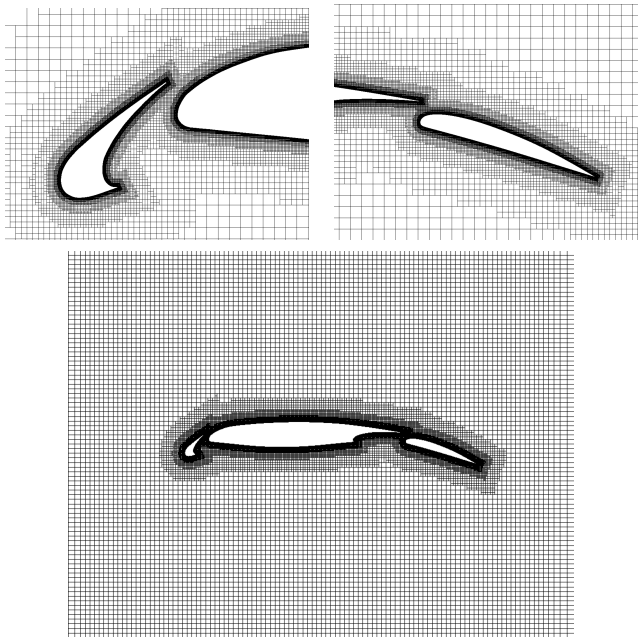


Fig. 11 NHLP-2D airfoil Cartesian grid.

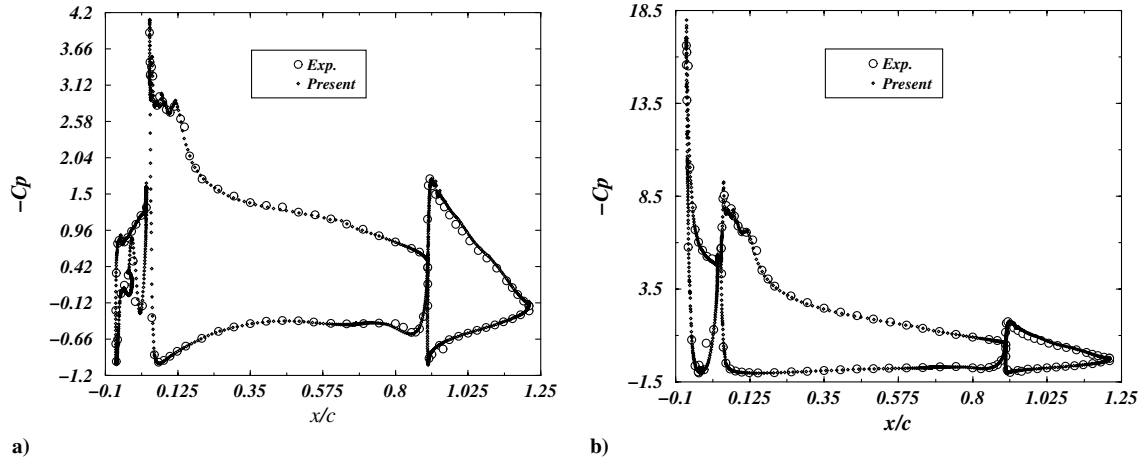
C. Multi-Element NHLP-2D Airfoil

The flow physics about multi-element wings involve highly complex flow details as compared to a single-element wing. The confluent and merging boundary layers, the wakes, and the mixing layers are only some of the reasons for the complex flow structure. Far-reaching insight into the flow physics that govern high-lift aerodynamics were summarized by Smith [27] in his landmark classical paper. Accurate flow predictions about multi-element wings using computational fluid dynamics tools are very challenging, especially for predicting aerodynamics forces (see the review paper by Rumsey and Ying [28] and the references therein).

The multi-element NHLP-2D airfoil in takeoff configuration is chosen in the present work. In this configuration, the slat deflection is $\delta_s = 25$ deg and the flap deflection is $\delta_f = 20$ deg. There are two different ways of defining the configuration: the vertical sense, as defined in the experiment [29], and the orthogonal sense, as adopted by most researchers. (For the definitions of vertical and orthogonal sense, the reader is referred to Balaji et al. [30].) The current paper adopts the definition of the original experiment (i.e., the vertical sense). Because of grid-generation considerations, it is common practice to sharpen the trailing edges of the three elements. In contrast, the exact experimental geometry is reproduced in the current work and the trailing edges are of finite thickness.

Multi-element wing flowfields are characterized by complex flow details. To correctly capture all hosted details, a fine-resolution grid is vital for an accurate solution. Beyond the high grid resolution that is required near the wall and in wake regions of all airfoil elements, care should be taken when treating the mesh in the far-field boundary region. The computational mesh should be generated so that it has a reasonable resolution near the outer boundary due to the high circulation that may exist at certain regions of the far-field boundary. Results show that lack of far-field boundary resolution may result in a significant loss of accuracy, especially in the prediction of the pressure drag [31].

Fig. 12 NHLP-2D; Mach number grayscale map; $\alpha = 20.18$ deg.



a) b)

Fig. 13 Comparison between computed and experimental surface pressure coefficient distribution: a) $\alpha = 4.01$ deg, and b) $\alpha = 20.18$ deg.

Table 1 Aerodynamics forces at $\alpha = 4.01$ and 20.18 deg, $Re_\infty = 6.5 \times 10^6$, $M_\infty = 0.734$

Contribution	α, deg	C_l	C_d
Present	4.01	2.1206	0.0275
[32]	4.01	2.1211	0.0284
Experiment set 1	4.01	2.1952	0.0306
Experiment set 2	4.01	2.1206	0.0311
Present	20.18	4.06	0.0729
[32]	20.18	4.074	0.0720
Experiment set 1	20.18	4.1099	0.0691
Experiment set 2	20.18	4.0748	0.0720

The Cartesian grid is composed of 11 layers at different heights in which the cell height near the body is not more than 1×10^{-5} chord lengths, resulting in $y^+ \leq 4.5$. It should be emphasized that the maximum value of y^+ is confined to a small region near the leading edge of the slat, whereas the value of $y^+ < 2$ is found at the rest of the near-body grid points. Near-field and close-up views of the slat, flap, and the cove regions of the main element are shown in Fig. 11. The grid contains about 1.2×10^6 cells. The strength of Cartesian grids is noted when dealing with complex geometries, as the grid-generation process may be automatically conducted. In the present simulations, the flow conditions are chosen to match those of the experiment. The freestream Mach number is set to $M_\infty = 0.197$ and the Reynolds number is set to $Re = 3.52 \times 10^6$. The simulations include flow prediction for two angles of attack, $\alpha = 4.01$ and 20.18 deg.

Figure 12 shows a grayscale map of the local Mach number for the angle of attack of $\alpha = 20.18$ deg. One can learn from the image the rich flow features that are exhibited by this case. The flow around the slat region reaches values of close to Mach 1.0 and this is captured by the simulation. Also captured are all the wakes and mixing layers that are present in this case [28].

The accuracy with which this flow is predicted is reflected in the comparison of the calculated surface pressure coefficient with the experimental results, as presented in Fig. 13 (the experiment included two sets from two tunnel runs; the experimental results in the figure are taken from the second set). The agreement between the computational and experimental results is excellent in this case as well. Note that the good agreement presented is also evident at the slat trailing edge. This agreement of the calculated pressure coefficient with the experiment in the slat trailing edge could not have been achieved while using the commonly practiced sharpened trailing edge (see, for example, [30]).

To further examine the accuracy of the present flow solver the aerodynamic force coefficients are integrated and compared to the experimental data. They are presented in Table 1. Once again, the present calculated aerodynamics forces are in very good agreement with the experimental data, thus demonstrating the high accuracy of the flow solver for real, complex, turbulent flowfields.

VI. Conclusions

A new Cartesian-grid-based flow solver for turbulent flow predictions is presented. The flow solver is based on the Cartesian cut-cells approach. A unique algorithm is developed for the grid generator. The new algorithm provides the means to construct a grid that mimics some of the ideas behind structured grids; therefore, an adequate grid can be easily generated. The flexibility and efficiency of the grid generator was especially demonstrated by modeling the multi-element airfoil.

Emphasis is placed on designing an overall robust and highly stable implicit scheme, especially for complex turbulent flow simulations. A pivotal part of achieving the high robustness of the flow solver is the use of the unconditionally positive-convergent implicit scheme for two-equation turbulence models [22]. Moreover, a simple modification of the convective Jacobian facilitates the use of high CFL numbers, hence keeping the advantage of the positive-convergent implicit scheme of using any time step.

Numerical simulations were conducted to examine the flow solver robustness and accuracy. The numerical results are found to be in excellent agreement with the experimental data, especially in the accurate predictions of aerodynamic forces. Moreover, a simple modification of the mean-flow convective Jacobian facilitates the use of high CFL numbers, hence keeping the advantage of the absolute stability of the turbulence model implicit time-marching scheme.

Although the successful concept of using a Cartesian grid for high-Reynolds-number turbulent flows has been demonstrated, extending the use of the method and the code to realistic three-dimensional cases requires further development. In particular, an efficient and appropriate grid adaptation algorithm is necessary. In addition, convergence acceleration algorithms should be developed and implemented. These issues are left for future research.

References

[1] Aftosmis, M. J., Berger, M. J., and Melton, J. E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 952–960. doi:10.2514/2.464

[2] Karman, S. L., "SPLITFLOW: A 3-D Unstructured Cartesian/Prismatic Grid CFD Code for Complete Geometries," AIAA Paper 95-0343, Jan. 1995.

[3] Delanaye, M., Aftosmis, M. J., Berger, M. J., Y. Liu, and Pulliman, T. H., "Automatic Hybrid-Cartesian Grid Generation for High-Reynolds Number Flows Around Complex Geometries," AIAA Paper 99-0777, Jan. 1999.

[4] Wang, Z. J., and Chen, R. F., "Anisotropic Solution-Adaptive Viscous Cartesian Grid Method for Turbulent Flow Simulation," *AIAA Journal*, Vol. 40, No. 10, 2002, pp. 1969–1978. doi:10.2514/2.1558

[5] Lee, J. D. , and Ruffin, S. M., "Development Of A Turbulent Wall-Function Based Viscous Cartesian Grid Methodology," AIAA Paper 2007-1326, Jan. 2007.

[6] De Tullio, M. D., De Palma, P., Iaccarino, G., Pascazio, G., and Napolitano, M., "An Immersed Boundary Method for Compressible Flows Using Local Grid Refinement," *Journal of Computational Physics*, Vol. 225, No. 2, 2007, pp. 2098–2117. doi:10.1016/j.jcp.2007.03.008

[7] Kamatsuchi, T., "Turbulent Flow Simulation Around Complex Geometries with Cartesian Grid Method," AIAA Paper 2007-1459, Jan. 2007.

[8] Mondal, P., Munikrishna, N., and Balakrishnan, N., "Cartesian-Like Grids Using a Novel Grid-Stitching Algorithm for Viscous Flow Computations," *Journal of Aircraft*, Vol. 44, No. 5, 2007, pp. 1598–1609. doi:10.2514/1.27854

[9] Baldwin, B. S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Jan. 1978.

[10] Kok, C. J., "Resolving the Dependence on Freestream Values for the k–ω Turbulence Model," *AIAA Journal*, Vol. 38, No. 7, 2000, pp. 1292–1295. doi:10.2514/2.1101

[11] Zheng, X., and Liu, F., "Staggered Upwind Method for Solving Navier–Stokes and k–ω Turbulence Model Equations," *AIAA Journal*, Vol. 33, No. 6, 1995, pp. 991–998. doi:10.2514/3.12808

[12] Liou, M.-S., "A sequel to AUSM, part II: AUSM$^+$-Up for All Speeds," *Journal of Computational Physics*, Vol. 214, No. 1, 2006, pp. 137–170. doi:10.1016/j.jcp.2005.09.020

[13] Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *Journal of Computational Physics*, Vol. 118, No. 1, 1995, pp. 120–130. doi:10.1006/jcph.1995.1084

[14] Liu, F. and Zheng, X., "A Strongly Coupled Time-Marching Method for Solving the Navier–Stokes and k–ω Turbulence Model Equations with Multigrid," *Journal of Computational Physics*, Vol. 128, No. 2, 1996, pp. 289–300. doi:10.1006/jcph.1996.0211

[15] Haselbacher, A. and Blazek, J., "Accurate and Efficient Discretization of Navier–Stokes Equations on Mixed Grids," *AIAA Journal*, Vol. 38, No. 11, 2000, pp. 2094–2102. doi:10.2514/2.871

[16] Weiss, J. M., Maruszewski, J. P., and Smith, W. A., "Implicit Solution of Preconditioned Navier–Stokes Equations Using Algebraic Multigrid,"

AIAA Journal, Vol. 37, No. 1, 1999, pp. 29–36.
doi:10.2514/2.689

[17] Lipnikov, K., Shashkov, M., Svyatskiy, D., and Vassilevski Y., "Monotone Finite Volume Schemes for Diffusion Equations on Unstructured Triangular and Shape-Regular Polygonal Meshes," *Journal of Computational Physics*, Vol. 227, 2007, pp. 492–512.
doi:10.1016/j.jcp.2007.08.008

[18] Mor-Yossef, Y., and Levy, Y., "The Unconditionally Positive-Convergent Implicit Time Integration Scheme for Two-Equation Turbulence Models: Revisited," *Computers and Fluids*, Vol. 38, No. 10, 2009, pp. 1984–1994.
doi:10.1016/j.compfluid.2009.06.005

[19] Toro, E. F., *Riemann Solvers and Numerical Methods for Fluid Dynamics; A Practical Introduction*, 2nd ed., Springer–Verlag, Berlin/Heidelberg, 1999.

[20] Batten, P., Leschziner, M. A., and Goldberg, U. C., "Average-State Jacobians and Implicit Methods for Compressible Viscous and Turbulent Flows," *Journal of Computational Physics*, Vol. 137, No. 1, 1997, pp. 38–78.
doi:10.1006/jcph.1997.5793

[21] Van Leer, B., *Flux-Vector Splitting for the Euler Equations*, Vol. 170, Lecture Notes in Physics, Springer–Verlag, New York, 1982, pp. 507–512.

[22] Mor-Yossef, Y., and Levy, Y., "Unconditionally Positive Implicit Procedure for Two-Equation Turbulence Models: Application to k–ω Turbulence Models," *Journal of Computational Physics*, Vol. 220, No. 1, 2006, pp. 88–108.
doi:10.1016/j.jcp.2006.05.001

[23] Berman, A. and Plemmons, R. J., *Nonnegative Matrices in the Mathematical Sciences*, Computer Science and Applied Mathematics, Academic Press, New York, 1979.

[24] Menter, F. R., "Zonal Two-Equation k–ω Turbulence Models for Aerodynamic Flows," AIAA Paper 93-2906, July 1993.

[25] Cook, P. H., McDonald, M. A., and Firmin, M. C. P., "Aerofoil RAE2822—Pressure Distributions, Boundary Layer and Wake Measurements," AGARD Rept. AR 138, 1979.

[26] Coles, D., and Wadcock, A. J., "Flying-Hot-Wire Study of Flow Past an NACA 4412 Airfoil at Maximum Lift," *AIAA Journal*, Vol. 17, No. 4, 1979, pp. 321–329.
doi:10.2514/3.61127

[27] Smith, A. M. O., "High Lift Aerodynamics," *Journal of Aircraft*, Vol. 12, No. 6, 1975, pp. 501–530.
doi:10.2514/3.59830

[28] Rumsey, C. L., and Ying, S. X., "Prediction of High Lift: Review of Present CFD Capabilities," *Progress in Aerospace Sciences*, Vol. 38, 2002, pp. 145–180.
doi:10.1016/S0376-0421(02)00003-9

[29] Moir, I., "Measurements on a Two-Dimensional Airfoil with High Lift Devices: A Selection of Experimental Test Cases for the Validation of CFD Codes," AGARD Rept. AR-303, 1994.

[30] Balaji, R., Bramkamp, F., Hesse, M., and Ballmann, J., "Effect of Flap and Slat Riggings on 2-D High-Lift Aerodynamics," *Journal of Aircraft*, Vol. 43, No. 5, Sept.–Oct. 2006, pp. 1259–1271.
doi:10.2514/1.19391

[31] Murayama, M., Yamamoto, K., and Kobayashi, K., "Validation of Flows on High-Lift Configuration by Structured- and Unstructured-Mesh Method," AIAA Paper 2005-1226, Jan. 2005.

[32] Mor-Yossef, Y., and Levy, Y., "Structured and Unstructured Code Predictions for High Lift Airfoil Flow," *47th Israel Annual Conference on Aerospace Sciences*, Feb. 2007.

Z. Wang
Associate Editor