JOURNAL OF AIRCRAFT
Vol. 45, No. 1, January–February 2008

Simulation of Flow Separation at the Wing–Body Junction with Different Fairings

Chunna Li,* Zhengyin Ye,† and Gang Wang‡
Northwestern Polytechnical University, 710072 Xi'an, People's Republic of China

DOI: 10.2514/1.26638

The numerical simulation and analyses of the flow separation at the wing–body junction with different designed fairings for a transporter are carried on by using a viscous flow solver based on hybrid unstructured meshes in this paper. In the process of generating hybrid meshes, the Pirzadeh advancing-layer method is improved to generate high-quality hybrid prismatic/pyramid/tetrahedral unstructured meshes in the boundary-layer region, and the advancing-front method is used to construct isotropic tetrahedral meshes in the residual flow region. Navier–Stokes equations are solved using a cell-centered finite volume method with the Spalart–Allmaras one-equation turbulence model. According to the Batina and Tomaro time-stepping theories, time is marched by means of an implicit Gauss–Seidel relaxation procedure, which is constructed by using the first-order linearizing of the flux vector and the maximal eigenvalue splitting of the flux Jacobian matrix. With local time stepping and implicit residual smoothing, artifice is used to accelerate convergence. The agreement of the computational results and the experimental measurements proves good capability of simulating wing–body interferences on a transporter by using a viscous flow solver based on hybrid unstructured meshes. In addition, numerical study indicates that wing–body fairing could suppress flow separation at the wing–body junction, increase the lift–drag ratio, and improve the aerodynamic performance.

Nomenclature

$\boldsymbol{A}$	=	Jacobian matrix of the inviscid flux term
C_d	=	drag coefficient
C_l	=	lift coefficient
C_p	=	pressure coefficient
$\boldsymbol{D}$	=	artificial viscosity flux
e_0	=	specific total energy of the fluid
$\boldsymbol{F}$	=	inviscid flux
$\boldsymbol{G}$	=	viscous flux
Ma	=	Mach number
$N(i)$	=	ith neighbor of the central cell i
N_p	=	number of faces connected to P
$\boldsymbol{n}$	=	unit normal vector on the surface of the element
$\boldsymbol{n}_f$	=	unit normal vector of the fth face connected to P
$\boldsymbol{n}_{i,m}$	=	unit normal vector on the public surface of the central element i and its neighbor m
P	=	pressure of the fluid
Pr	=	laminar Prandtl number taken as 0.72
Pr_t	=	turbulent Prandtl number taken as 0.9
$\boldsymbol{Q}$	=	conservative flux
$\boldsymbol{R}$	=	inviscid flux
Re	=	Reynolds number
$\boldsymbol{R}^v$	=	viscous flux
r_1, r_2	=	nonnegative constants that control the derivative of δ_n with respect to $\boldsymbol{v}_o$
S	=	surface of the bounded domain
T	=	temperature of the fluid
u_i	=	velocity of the flow in the coordinate direction x_i
u_j	=	velocity of the flow in the coordinate direction x_j
V_i	=	volume of the ith element
$\boldsymbol{v}_p$	=	surface unit vector at point P
w_f^t	=	weighting factor assigned to the fth face at iteration level t
α	=	angle of attack
$\bar{\alpha}$	=	average of α_f
α_f	=	angle formed by $\boldsymbol{v}_p$ and $\boldsymbol{n}_f$
β	=	relaxation factor defined as 1.0
γ	=	ratio of specific heats defined as 1.4
δ	=	advancing distance
δ_k	=	distance of the kth layer to the surface
δ_0	=	distance of the first layer to the surface
ε	=	smoothing coefficient taken as 0.5
η	=	semispan position
μ	=	laminar viscosity
μ_t	=	turbulence viscosity
ρ	=	density of the fluid
Ω	=	bounded domain
$\partial\Omega$	=	boundary of the bounded domain
ω	=	relaxation factor

Subscripts

i	=	ith element
L	=	left cell of the public face $S_{i,m}$
m	=	mth neighbor of the ith element
R	=	right cell of the public face $S_{i,m}$

Superscripts

k	=	number of the layer marching
n	=	time step
p	=	point on the surface

I. Introduction

IN THE simulation of the flow around aircraft, aerodynamic interference always occurs between the components, such as at the wing–body junction, the wing–pylon–nacelle junction, and the tail-fuselage junction. Under the dual influence of the boundary layer's development and the adverse pressure gradient, flow separation appears at the wing root of the supercritical and laminar

Presented as Paper 1089 at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 8–11 January 2007; received 19 July 2006; accepted for publication 25 September 2007.

*Postgraduate Student, National Key Laboratory of Aerodynamic Design and Research, P.O. Box 114; tracy-li@hotmail.com.

†Professor, National Key Laboratory of Aerodynamic Design and Research, P.O. Box 114; yezy@nwpu.edu.cn.

‡Instructor, National Key Laboratory of Aerodynamic Design and Research, P.O. Box 114; wanggang@nwpu.edu.cn.

wing, and it reduces the lift–drag ratio and causes noises and buffets of the wing. Furthermore, it could severely deteriorate the efficiency of the wing and other aerodynamic rudders. Therefore, the aircraft's stability and controllability are destroyed. So modern aircraft designers should do their best to eliminate the negative effect of wing–body interference and to obtain accurate prediction and analyses for wing–body aerodynamic interference. Because flow separation is sensitive to the configuration of the body, angle of attack, Mach number, and Reynolds number, the prediction of the aerodynamic performance under such separation is quite difficult for computationalists, although computational fluid dynamics (CFD) tools are playing an increasingly vital role in aircraft design now. Most recent research efforts on wing–body interference concentrate on a simplified configuration, such as the symmetry wing mounted perpendicularly to a flat plate [1,2]. The particular focuses of this paper are the simulation of the flow around complicated wing–body fairing configuration and the efficiency of different fairings on suppressing or retarding flow separation.

With these motivations, computational investigation about wing–body interference of a transporter is carried out by using a viscous flow solver [3,4] based on hybrid unstructured meshes. In the process of generating hybrid meshes, Pirzadeh's [5] advancing-layer method is improved to generate high-quality hybrid prismatic/pyramid/tetrahedral unstructured meshes in the boundary-layer region, and the advancing-front method is used to construct isotropic tetrahedral meshes in the other region. Navier–Stokes equations are solved using a cell-centered finite volume method with the Spalart–Allmaras one-equation [6] turbulence model. According to Batina's [7] and Tomaro's [8] theory, time is marched by means of an implicit Gauss–Seidel relaxation procedure, which is formulated by using the first-order linearizing of flux vector and the maximal eigenvalue splitting of the flux Jacobian matrix. In addition, local time-stepping and implicit residual-smoothing artifice are used to accelerate convergence. The computational results agree well with the experimental measurements. Consequently, the algorithm with good efficiency and accuracy presented in this paper has advantages in simulating the interference at the wing–body junction. At the same time, it shows that an appropriate fairing could suppress or retard flow separation at the wing–body junction efficiently.

II. Hybrid Mesh Generation

In practical engineering problems, mesh-generation techniques must satisfy the following conditions. First, the procedure of mesh generation should be direct, easy, and automatic. Second, the meshes require excellent flexibility to adapt to different complicated configurations. What is more, modification of the mesh size could actually reflect the physical scale of various flow properties.

In fact, using conventional structured grids, confined by structure and orthogonalization, it is difficult to satisfy the preceding conditions. Unstructured meshes in common use are flexible, with the disadvantage for computation in the viscous boundary-layer region. To settle the problem, we generate prismatic/pyramid hybrid meshes in the boundary layer and tetrahedral meshes in the inviscid region. The hybrid unstructured-mesh-generation process can be divided into four separate stages.

A. Triangulate All the Boundaries and Calculate Background Information

In 3D, wall surface and various boundaries should be divided into several surfaces and plates, and the normal vectors of these faces must tilt toward the computational domain. Then all of these faces are triangulated by different methods, such as the 2D advancing-front method for plates [9], the crossing-diagonal method of structured grids for simple surfaces, and mapping the surface onto the 2D domain for complicated surfaces.

The background information is determined as follows:

1) Calculate the mean length of the edges for each unstructured triangular element, and this length will be taken as the characteristic length of the same element.

2) A reference distance D_1 and a relevant reference length S are defined, where D_1 is the distance from the element to the wall boundary, and S is the scale of the element.

3) If the minimum distance D from the advanced element to the wall boundary is larger than D_1, background information is calculated via interpolation between reference length S and characteristic length L_f, where L_f is the nearest distance from the advanced element to the far boundary; conversely, it is obtained by interpolation between S and characteristic length L_w, where L_w is the distance from the advanced element to the wall boundary.

B. Predetermine Surface Vectors of the Wall Boundary

A crucial factor for the advancing-layer method to distribute the new mesh points is the determination of surface vectors [10]. A necessary condition to prevent formation of negative cells at sharp convex corners is that the surface vectors at mesh points are visible by all faces connected to the corresponding point [11]. The approach is based on satisfaction of a surface vector at a point making equal angles with all the faces connected to this point, subject to the visibility condition; that is,

$$\boldsymbol{v}_p \cdot \boldsymbol{n}_f = \text{const} \qquad f = 1, 2, \ldots, N_p \tag{1}$$

A solution to Eq. (1) is obtained by the following iterative scheme:

$$\boldsymbol{v}_p^{\,t} = \omega \sum_{f=1}^{N_p} w_f^t \boldsymbol{n}_f + (1-\omega)\boldsymbol{v}_p^{t-1} \tag{2}$$

where w_f^t is obtained from the following expressions:

$$\bar{w}_f^{\,t+1} = w_f^t \left[1 - \frac{\bar{\alpha} - \alpha_f}{\bar{\alpha}}\right], \quad w_f^{t+1} = \bar{w}_f^{t+1} + w_f^t \left(1 - \sum_{f=1}^{F_p} \bar{w}_f^{t+1}\right) \tag{3}$$

Thus, valid surface vectors are produced after a few iterations.

C. Generate Hybrid Unstructured Meshes in the Boundary-Layer Region Using the Advancing-Layer Method

The process of the advancing-layer method [5,10] proceeds by selecting a triangulated face as the advancing front, adding new points along the surface vectors emanating from the vertices on the advancing front, recording δ of the vertices and its positions, and stopping under the condition that δ satisfies background information with the supplement that positions of the new points are quite closer to or penetrate through the opposing fronts.

Connect points on the stopped advancing front to the corresponding vertices to generate new prism cells, which are partitioned into prisms layer by layer according to the following formula:

$$\delta_k = \delta_0[1 + r_1(1 + r_2)^{k-1}]^{k-1} \tag{4}$$

Because of the complicated configuration and different positions of the points on the front, the meshes formed by the last layer may be prismatic/pyramid/tetrahedral unstructured meshes.

D. Create Isotropic Tetrahedral Meshes in the Inviscid Region with the Advancing-Front Method

In rapid sequence, the meshes on the symmetrical boundary are generated by using the 2D advancing-front method confined by the surface boundary curves, which are redefined by meshes on far boundary and symmetrical boundary. Choosing the last layer of viscous meshes, far boundary, and symmetrical boundary as the advancing fronts, isotropic tetrahedral meshes are generated in the inviscid region. When the inviscid region is full of isotropic tetrahedral meshes, we record the information of each element and its neighbors to solve Navier–Stokes equations.

The advantages of this method of generating hybrid meshes are as follows:

1) Large stretching of prismatic meshes takes on structure properties in the normal direction of the solid boundary, and these meshes could easily ensure the accuracy in discretization.

2) Triangulate the wall boundary flexibly, and this triangulation could exactly depict the configuration of the aircraft.

3) The one-to-one relationship between each mesh guarantees the numerical conservation.

III. Algorithm of Navier–Stokes Equations

A. Governing Equations

Three-dimensional unsteady Navier–Stokes equations in integral conservation form are as follows:

$$\frac{\partial}{\partial t}\iiint_{\Omega} \boldsymbol{Q}\,\mathrm{d}V + \iint_{\partial\Omega} F(\boldsymbol{Q})\cdot\boldsymbol{n}\,\mathrm{d}S = \iint_{\partial\Omega} G(\boldsymbol{Q})\cdot\boldsymbol{n}\,\mathrm{d}S \quad (5)$$

The vectors $\boldsymbol{Q}$, $\boldsymbol{F}$, and $\boldsymbol{G}$ are defined by

$$\boldsymbol{Q}=\begin{Bmatrix}\rho\\ \rho u_i\\ e_0\end{Bmatrix},\quad \boldsymbol{F}(\boldsymbol{Q})=\begin{Bmatrix}\rho u_j\\ \rho u_i u_j + p\delta_{ij}\\ (e_0+p)u_j\end{Bmatrix},\quad \boldsymbol{G}=\begin{Bmatrix}0\\ \sigma_{ij}\\ u_l\sigma_{jl}-q_j\end{Bmatrix} \quad (6)$$

$$\sigma_{ij}=(\mu+\mu_t)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{2}{3}(\mu+\mu_t)\frac{\partial u_k}{\partial x_k}\delta_{ij} \quad (7)$$

$$q_j=\frac{1}{\gamma-1}\left(\frac{\mu}{Pr}+\frac{\mu_t}{Pr_t}\right)\frac{\partial T}{\partial x_j} \quad (8)$$

$$p=(\gamma-1)(e_0-\tfrac{1}{2}\rho u_j u_j),\qquad T=\gamma p/\rho \quad (9)$$

where μ is determined by Sutherland's law (10), μ_0 denotes the viscosity at the reference temperature T_0, s is a constant taken as 110 K, and μ_t is computed using the Spalart–Allmaras one-equation turbulence model (11), closed by transport equation (12) for the working variable $\hat{\upsilon}$, solving by Gauss–Seidel relaxation procedure.

$$\mu/\mu_0=(T/T_0)^{3/2}[(T_0+s)/(T+s)] \quad (10)$$

$$\mu_T=\rho\hat{\upsilon}f_{\upsilon 1} \quad (11)$$

$$\frac{\mathrm{d}\hat{\upsilon}}{\mathrm{d}t}=C_{b1}[1-f_{t2}]\hat{S}\,\hat{\upsilon}+\frac{\nabla\cdot[(\upsilon+\hat{\upsilon})\nabla\hat{\upsilon}]+C_{b2}(\nabla\hat{\upsilon})^2}{\sigma} - C_{w1}f_w-\frac{(C_{b1}f_{t2})}{\kappa^2}\left(\frac{\hat{\upsilon}}{d}\right)^2+f_{t1}\Delta U^2 \quad (12)$$

B. Discretization Scheme

The preceding equations are discretized with a cell-centered finite volume method, which can be rewritten in semidiscrete form as

$$V_i(\mathrm{d}\boldsymbol{Q}_i/\mathrm{d}t)=-\boldsymbol{R}_i(\boldsymbol{Q}^{n+1})+\boldsymbol{R}_i^V(\boldsymbol{Q}^{n+1})+\boldsymbol{D}_i(\boldsymbol{Q}^{n+1}) \quad (13)$$

where $\boldsymbol{R}_i(\boldsymbol{Q}^{n+1})$ and $\boldsymbol{R}_i^V(\boldsymbol{Q}^{n+1})$ are expressed as

$$\boldsymbol{R}_i(\boldsymbol{Q}^{n+1})=\sum_{m\in N(i)}\boldsymbol{F}(\boldsymbol{Q}_{i,m}^{n+1})\cdot\boldsymbol{n}_{i,m}\Delta S_{i,m},\qquad \boldsymbol{R}_i^V(\boldsymbol{Q}^{n+1})=\sum_{m\in N(i)}\boldsymbol{G}(\boldsymbol{Q}_{i,m}^{n+1})\cdot\boldsymbol{n}_{i,m}\Delta S_{i,m} \quad (14)$$

where $\boldsymbol{Q}_{i,m}$ can be obtained by interpolation between the central cell and its neighbors, based on Jameson's theory:

$$\boldsymbol{Q}_{i,m}=(\boldsymbol{Q}_i+\boldsymbol{Q}_m)/2 \quad (15)$$

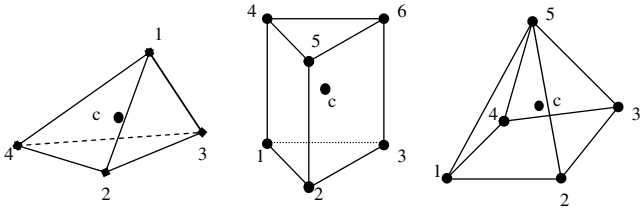


Fig. 1 Sketch map of calculation $\boldsymbol{Q}_{i,m}$

For the purpose of maintaining the accuracy of the meshes with large extension and distortion, a new method is used to calculate $\boldsymbol{Q}_{i,m}$:

$$\boldsymbol{Q}_{i,m}=(\boldsymbol{Q}_{i,m}^L+\boldsymbol{Q}_{i,m}^R)/2 \quad (16)$$

For example, $\boldsymbol{Q}_{i,m}$ can be expressed as follows, where the number denotes the node and c represents the center cell:

Tetrahedron:

$$\boldsymbol{Q}_{i,m}^L=\boldsymbol{Q}_c+[(\boldsymbol{Q}_2+\boldsymbol{Q}_3+\boldsymbol{Q}_4)/3-\boldsymbol{Q}_1]/4$$

Prism:

$$\boldsymbol{Q}_{i,m}^L=\boldsymbol{Q}_c+(\boldsymbol{Q}_1+\boldsymbol{Q}_2+\boldsymbol{Q}_3)/6-(\boldsymbol{Q}_4+\boldsymbol{Q}_5+\boldsymbol{Q}_6)/6$$

(face $S_{i,m}$ is formed by 1, 2, and 3 in Fig. 1).

$$\boldsymbol{Q}_{i,m}^L=\boldsymbol{Q}_c+(\boldsymbol{Q}_1+\boldsymbol{Q}_2+\boldsymbol{Q}_4+\boldsymbol{Q}_5)/12-(\boldsymbol{Q}_3+\boldsymbol{Q}_6)/6$$

(face $S_{i,m}$ is formed by 1, 2, 4, and 5 in Fig. 1).

Pyramid:

$$\boldsymbol{Q}_{i,m}^L=\boldsymbol{Q}_c+[(\boldsymbol{Q}_1+\boldsymbol{Q}_2+\boldsymbol{Q}_3+\boldsymbol{Q}_4)/4-\boldsymbol{Q}_5]/5$$

(face $S_{i,m}$ is formed by 1, 2, 3, and 4 in Fig. 1).

$$\boldsymbol{Q}_{i,m}^L=\boldsymbol{Q}_c+[(\boldsymbol{Q}_1+\boldsymbol{Q}_2+\boldsymbol{Q}_5)/3-(\boldsymbol{Q}_3+\boldsymbol{Q}_4)/2]/4$$

(face $S_{i,m}$ is formed by 1, 2, and 5 in Fig. 1) .

1. *Inviscid Flux*

The AUSM$^+$ scheme is used here to calculate inviscid flux $\boldsymbol{R}_i$, which is denoted as

$$\boldsymbol{R}_i=\sum_{m\in N(i)}a_{i,m}\left[M_{i,m}^+\begin{pmatrix}\rho\\ \rho u_j\\ \rho H_t\end{pmatrix}_i^L+M_{i,m}^-\begin{pmatrix}\rho\\ \rho u_j\\ \rho H_t\end{pmatrix}_m^R\right]\Delta S_{i,m} +\sum_{m\in N(i)}\begin{pmatrix}0\\ p_{i,m}n_j\Delta S_{i,m}\\ 0\end{pmatrix} \quad (17)$$

The first term on the right side is the convective flux, and the second term represents pressure flux. The other variables are defined as

$$\begin{aligned}&M_{i,m}^{\pm}=(M_{i,m}\pm|M_{i,m}|)/2,\quad M_{i,m}=M^+(M_{i,m}^L)+M^-(M_{i,m}^R),\\ &M_{i,m}^L=(\boldsymbol{n}_{i,m}\cdot\boldsymbol{V}_{i,m}^L)/a_{i,m},\quad M_{i,m}^R=(\boldsymbol{n}_{i,m}\cdot\boldsymbol{V}_{i,m}^R)/a_{i,m},\\ &a_{i,m}=\min(\tilde{a}_{i,m}^L,\tilde{a}_{i,m}^R),\quad \tilde{a}_{i,m}^L=a^{*2}/\max(|\boldsymbol{V}_{i,m}^L\cdot\boldsymbol{n}_{i,m}|,a^*),\\ &\tilde{a}_{i,m}^R=a^{*2}/\max(|\boldsymbol{V}_{i,m}^R\cdot\boldsymbol{n}_{i,m}|,a^*)\end{aligned} \quad (18)$$

The critical speed of sound evaluated is

$$a^*=\sqrt{[2(\gamma-1)H_t]/(\gamma+1)}$$

.

And the isoenergetic condition is

$$H_t=(e_0+p)/\rho=c_pT+(u^2+v^2+w^2)/2$$

.

The other variables are calculated by the following formulations:

$$M^{\pm}(M) = \begin{cases} \pm(M \pm 1)^2/4 \pm \beta(M^2-1)^2 & |M| < 1 \\ (M \pm |M|)/2 & |M| \geq 1 \end{cases} \tag{19}$$

$$p_{i,m} = \mathrm{p}^{+}(M^L_{i,m})p^L_{i,m} + \mathrm{p}^{-}(M^R_{i,m})p^R_{i,m} \tag{20}$$

$$\mathrm{p}^{\pm}(M) = \begin{cases} [1 \pm \mathrm{sign}(M)]/2 & |M| \geq 1 \\ [(M \pm 1)^2(2 \mp M)]/4 \pm \alpha M(M^2-1)^2 & |M| < 1 \end{cases} \tag{21}$$

Liou [12] proposes $\beta = 1/8$, and $\alpha = 3/16$.

2. Viscous Flux

The main differences between the process of viscous flux and inviscid flux are the derivatives of speed and temperature on the faces of the meshes. Derivatives at the center of the cells are commonly calculated by Gauss's theorem. Take u, as an example,

$$\nabla u_i = \frac{1}{V_i}\iiint_{\Omega} \nabla u_i \, \mathrm{d}V = \frac{1}{V_i}\iint_{\partial\Omega} u_i \boldsymbol{n} \, \mathrm{d}S = \frac{\sum\limits_{m\in N(i)} u_{i,m}\boldsymbol{n}_{i,m}\Delta S_{i,m}}{V_i} \tag{22}$$

C. Time-Marching Scheme

The artificial viscous term and viscous flux term are treated explicitly because of not having a fatal effect on time stepping; namely,

$$\boldsymbol{R}^V_i(\boldsymbol{Q}^{n+1}) \approx \sum_{m\in N(i)} \boldsymbol{G}(\boldsymbol{Q}^n_{i,m}) \cdot \boldsymbol{n}_{i,m}\Delta S_{i,m}, \qquad \boldsymbol{D}_i(\boldsymbol{Q}^{n+1}) \approx \boldsymbol{D}_i(\boldsymbol{Q}^n) \tag{23}$$

Based on the first-order linearizing of $\boldsymbol{R}^{n+1}$ with the ignorance of the terms of second order and higher order, then

$$\begin{aligned} \boldsymbol{R}^{n+1}_{i,m} &= \boldsymbol{R}^n_{i,m} + (\partial \boldsymbol{R}/\partial \boldsymbol{Q})^n_{i,m}(\boldsymbol{Q}^{n+1}_{i,m} - \boldsymbol{Q}^n_{i,m}) \\ &= \boldsymbol{F}(\boldsymbol{Q}^n_{i,m}) \cdot \boldsymbol{n}_{i,m}\Delta S_{i,m} + \boldsymbol{A}^n_{i,m}\Delta \boldsymbol{Q}_{i,m} \end{aligned} \tag{24}$$

where $\Delta \boldsymbol{Q}_{i,m} = \boldsymbol{Q}^{n+1}_{i,m} - \boldsymbol{Q}^n_{i,m}$, and $\boldsymbol{A}^n_{i,m} = (\partial \boldsymbol{R}/\partial \boldsymbol{Q})^n_{i,m}$ is the Jacobian matrix of inviscid flux.

Substituting Eq. (24) into Eq. (13), the semidiscrete form becomes

$$\frac{\mathrm{d}\boldsymbol{Q}_i}{\mathrm{d}t} + \frac{1}{V_i}\sum_{m\in N(i)} \boldsymbol{A}^n_{i,m}\Delta \boldsymbol{Q}_{i,m} = -\frac{1}{V_i}[\boldsymbol{R}_i(\boldsymbol{Q}^n) - \boldsymbol{R}^V_i(\boldsymbol{Q}^n) - \boldsymbol{D}_i(\boldsymbol{Q}^n)] \tag{25}$$

and the right-hand side of the preceding equation is the explicit residual $\hat{\boldsymbol{R}}_i = -(\boldsymbol{R}_i - \boldsymbol{R}^V_i - \boldsymbol{D}_i)/V_i$. Then Eq. (25) becomes

$$\frac{\mathrm{d}\boldsymbol{Q}_i}{\mathrm{d}t} + \frac{1}{V_i}\sum_{m\in N(i)} \boldsymbol{A}^n_{i,m}\Delta \boldsymbol{Q}_{i,m} = \hat{\boldsymbol{R}}_i(\boldsymbol{Q}^n) \tag{26}$$

The Jacobian matrix $\boldsymbol{A}^n_{i,m}$ has real eigenvalues and can be split into two matrices by maximal eigenvalue splitting. We get

$$\boldsymbol{A}^{+}_{i,m} = (\boldsymbol{A}^n_{i,m} + \boldsymbol{I}\beta\lambda_{i,m})/2, \qquad \boldsymbol{A}^{-}_{i,m} = (\boldsymbol{A}^n_{i,m} - \boldsymbol{I}\beta\lambda_{i,m})/2 \tag{27}$$

where $\lambda_{i,m}$ is the maximum eigenvalue of the Jacobian matrix on the public face. It is calculated from

$$\lambda_{i,m} = |\boldsymbol{V}^n_{i,m} \cdot \boldsymbol{n}_{i,m}\Delta S_{i,m}| + c_{i,m}\Delta S_{i,m}$$

and $c_{i,m}$ is the speed of sound on the face $S_{i,m}$. Now Eq. (26) becomes

$$\frac{\mathrm{d}\boldsymbol{Q}_i}{\mathrm{d}t} + \frac{1}{V_i}\sum_{m\in N(i)} [\boldsymbol{A}^{+}_{i,m}\Delta \boldsymbol{Q}_i + \boldsymbol{A}^{-}_{i,m}\Delta \boldsymbol{Q}_m] = \hat{\boldsymbol{R}}_i(\boldsymbol{Q}^n) \tag{28}$$

Note that $\boldsymbol{A}^{-}_{i,m}\Delta \boldsymbol{Q}_i$ represents the flux change associated with waves that have a negative wave speed; likewise, $\boldsymbol{A}^{+}_{i,m}\Delta \boldsymbol{Q}_i$ denotes the flux associated with waves that have a positive wave speed. Replacing the time derivative with a one-order-backward finite difference approximation, Eq. (28) becomes

$$\begin{aligned} \left[\frac{1}{\Delta t}\boldsymbol{I} + \frac{1}{V_i}\sum_{m\in N(i)} \boldsymbol{A}^{+}_{i,m}\right]\Delta \boldsymbol{Q}^k_i = \hat{\boldsymbol{R}}_i(\boldsymbol{Q}^n) - \frac{1}{V_i}\Bigg(&\sum_{m<i, m\in N(i)} \boldsymbol{A}^{-}_{i,m}\Delta \boldsymbol{Q}^k_m \\ &+ \sum_{m>i, m\in N(i)} \boldsymbol{A}^{-}_{i,m}\Delta \boldsymbol{Q}^{k-1}_m\Bigg) \end{aligned} \tag{29}$$

The Gauss–Seidel relaxation approach is used to solve Eq. (29) for each mesh element to get conservative flux at the $(n+1)$th time step, where k is the sweeping time.

D. Speed Up the Convergence

To speed up the convergence, two methods are adopted here. The first method is local time stepping. The time step that is close to the

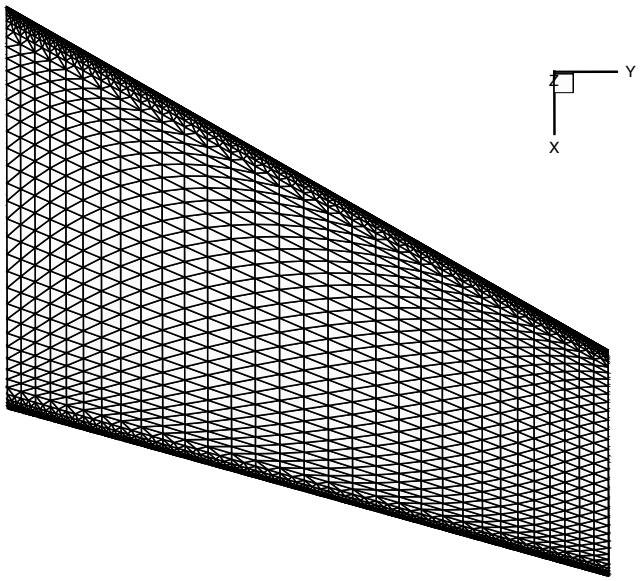


Fig. 2 Surface meshes of M6.

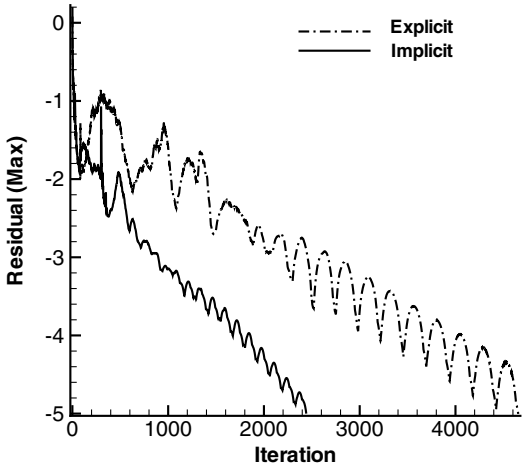


Fig. 3 Comparison of the residual between the explicit and implicit schemes.

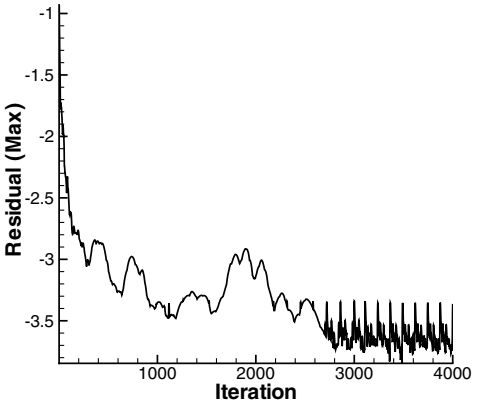


Fig. 4 Residual of the configuration with carefully designed fairing.

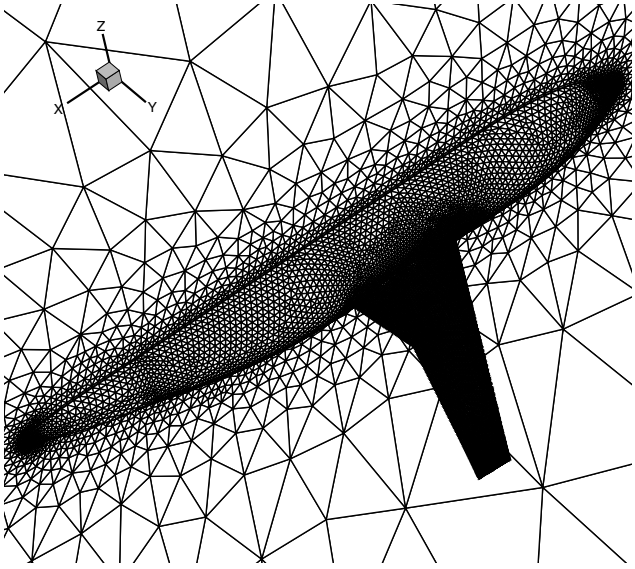


Fig. 5 Surface meshes of DLR-F6 without wing–pylon–nacelle.

limited stability at local meshes will promote the process of convergence. The time step is obtained from the following expression:

$$\Delta t_i = \frac{\mathrm{CFL}}{n}(\alpha + \beta_i) \sum_{m \in N(i)} \frac{V_i}{\lambda_{i,m} \Delta S_{i.m}} \tag{30}$$

where CFL is the Courant–Friedrichs–Lewy number, n is the number of faces around the cell i, α is a constant taken as 0.3 experientially, and β_i is a qualitative coefficient of the ith mesh element chosen from 0 to 1.0 for the tetrahedron meshes (that is, β_i is the ratio of the minimum edge over the maximum edge of the mesh for prismatic and pyramid meshes).

The implicit residual smoothing is another artifice speeding up the convergence. This artifice may restrain the fluctuation caused by odd–even irrelevancy of the cell-centered finite volume method and may obtain a large stable interval of the scheme and the maximum time step. The process proceeds as follows:

$$\hat{\boldsymbol{R}}_i' = \hat{\boldsymbol{R}}_i + \varepsilon \nabla^2 \hat{\boldsymbol{R}}_i' \tag{31}$$

where $\hat{\boldsymbol{R}}_i'$ is the smoothed residual, and the Laplacian operator ∇^2 is defined as

$$\nabla^2 \boldsymbol{Q}_i = \sum_{m \in N(i)} (\boldsymbol{Q}_m - \boldsymbol{Q}_i)$$

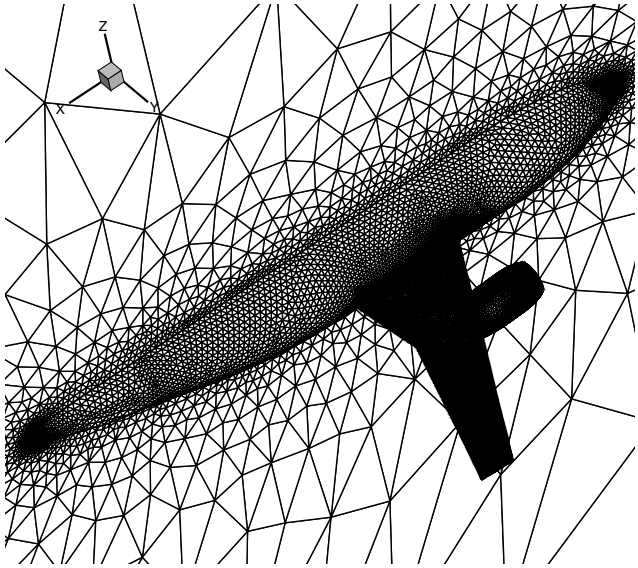


Fig. 6 Surface meshes of DLR-F6 with wing–pylon–nacelle.

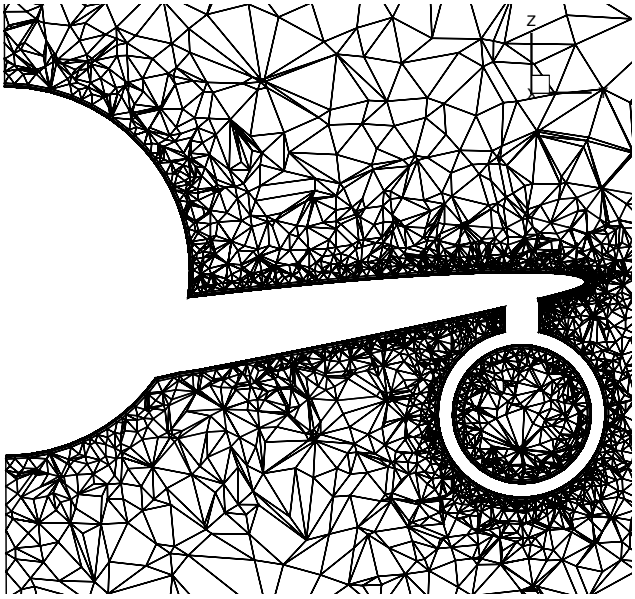


Fig. 7 Slice of DLR-F6 with wing–pylon–nacelle.

Equation (31) can be written as

$$\hat{\boldsymbol{R}}_i' = \frac{\hat{\boldsymbol{R}}_i + \varepsilon \sum\limits_{m \in N(i)} \hat{\boldsymbol{R}}_m'}{1 + n\varepsilon} \tag{32}$$

Equation (32) can be solved using the Jacobian iterative scheme, and the expression is Eq. (33):

$$\hat{\boldsymbol{R}}_i^{(k)} = \frac{\hat{\boldsymbol{R}}_i^{(0)} + \varepsilon \sum\limits_{m \in N(i)} \hat{\boldsymbol{R}}_k^{(k-1)}}{1 + n\varepsilon} \tag{33}$$

where $\hat{\boldsymbol{R}}_i^{(0)}$ is the original residual, unsmoothed.

IV. Test Case

Compared with the explicit four-stage Runge–Kutta scheme, the speed of convergence of this method based on an unstructured tetrahedral was validated by the test case of M6. The flow parameters used are $Ma = 0.8395$ and $\alpha = 3.06$ deg. Figure 2 is the

computational meshes on the surface of M6, and Fig. 3 shows the comparison between the explicit four-stage Runge–Kutta scheme and the implicit Gauss–Seidel scheme.

To quantify the computational cost, a case of the configuration with carefully designed fairing was tested, and the residual is shown in Fig. 4. The flow parameters used are $Ma = 0.76$, $\alpha = 0$ deg, and $Re = 1.5 \times 10^7$, and uniform flow is used as the incoming flow. Each time step costs about 30 s; consequently, 26 h are needed to arrive at the convergence.

The DLR-F6 wing–body configuration with wing–pylon–nacelle is tested to validate the accuracy of the solver. Figure 5 shows the surface meshes of the configuration without wing–pylon–nacelle,

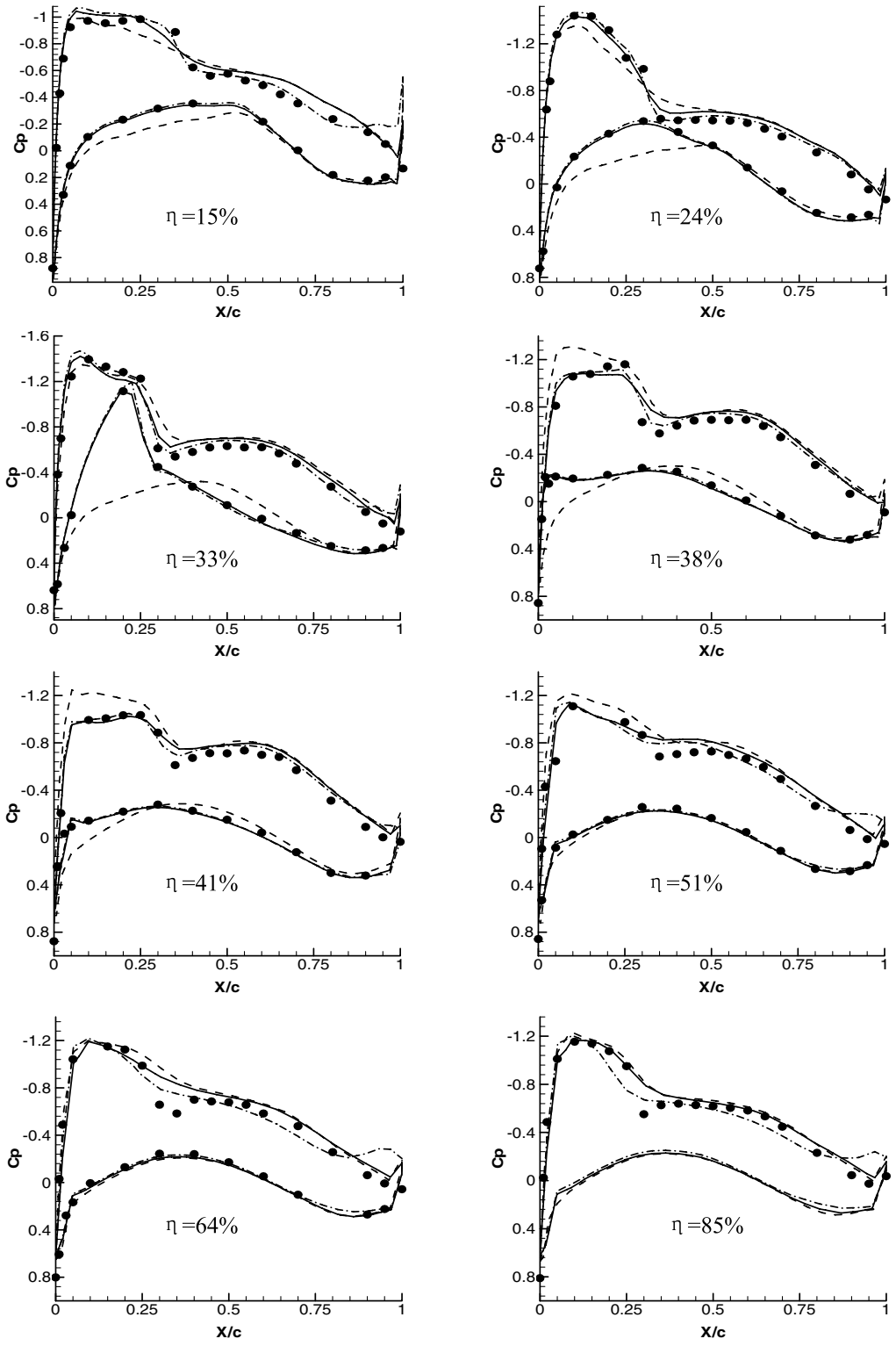
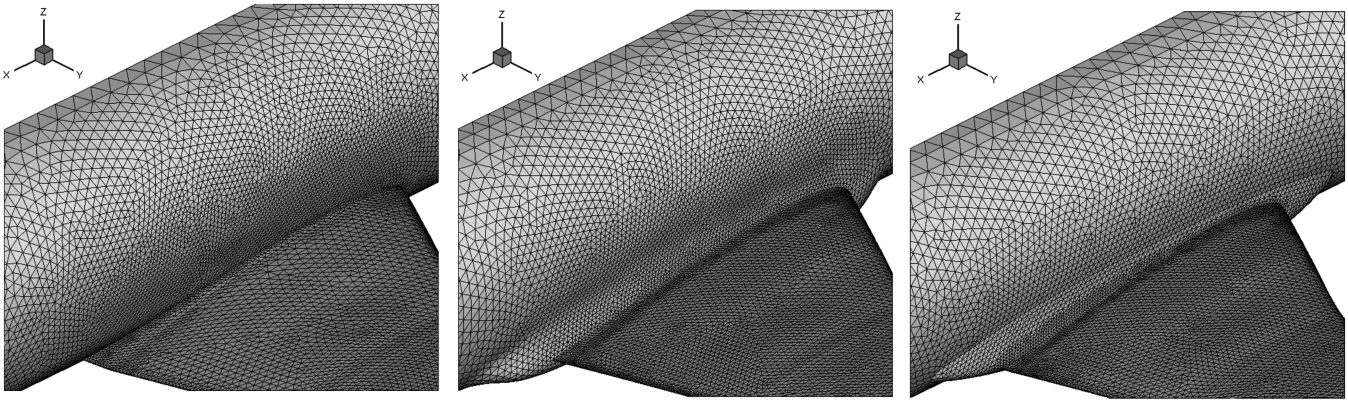
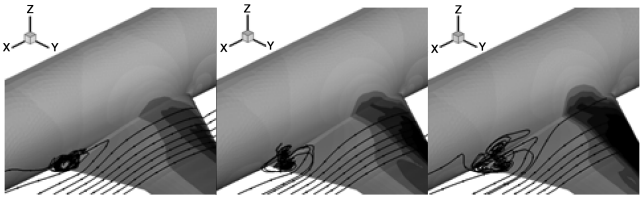


Fig. 8 Comparison of pressure coefficient distributions at different semispans for DLR-F6.

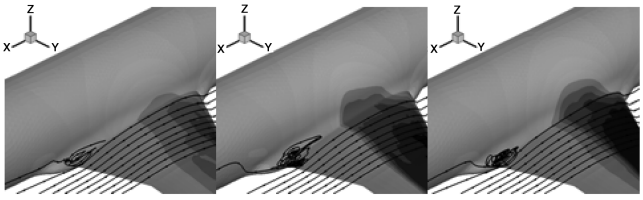


a) Original configuration **b) Configuration I** **c) Configuration II**

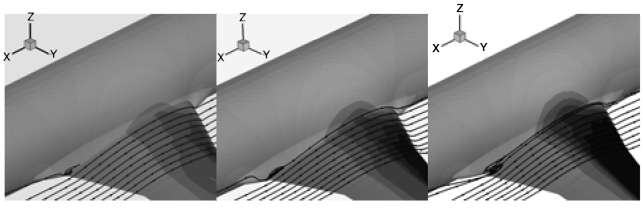
Fig. 9 Different wing–body configurations with and without fairing.



a) Original configuration

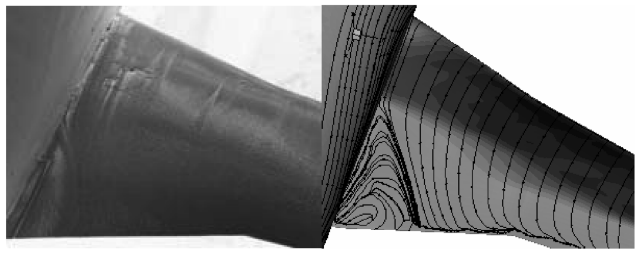


b) Configuration I

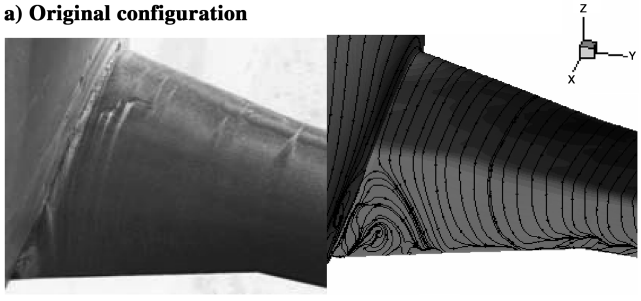


c) Configuration II

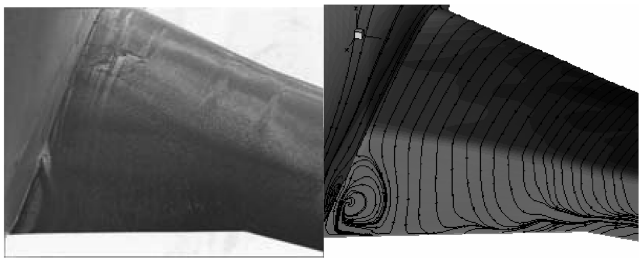
Fig. 10 Spatial streamlines at the wing–body junction of different configurations with $Re = 1.5 \times 10^7$, $Ma = 0.76$, and $\alpha = 1, 2, 3$ deg.



a) Original configuration



b) Configuration I



c) Configuration II

Fig. 11 Comparison of the surface streamlines and oil photos.

and Fig. 6 shows the surface meshes of the configuration with wing–pylon–nacelle. The slice of spatial meshes of the configuration with wing–pylon–nacelle is shown in Fig. 7. The minimum height of the layer partitioned is 2.0×10^{-5}, and the reference length is the mean aerodynamic chord. Flow parameters used are $Ma = 0.75$, $\alpha = 1$ deg, and $Re = 3.0 \times 10^6$. The comparison of the pressure coefficient distributions are indicated in Fig. 8.

V. Numerical Results and Discussion

Because of the aerodynamic interference at the wing–body junction, conventional design at the wing–body junction will cause a large separation region there. But a suitable fairing could suppress this separation. Because of the complex flow structure that occurs near the wing–body junction, the configuration of the fairing traditionally depends on experience and trials. Currently, because of the accuracy and the efficiency of CFD, it has become an effective means of designing fairing configuration.

By using the preceding viscous flow solver, two different wing–body fairing configurations are investigated. A transporter's wing–body configuration is chosen as the original configuration. And based on it, a roughly designed fairing and a carefully designed fairing are investigated. The wing–body configurations with and without fairing are shown in Fig. 9.

Figure 9a is the original configuration; Fig. 9b is configuration I with the roughly designed fairing; and Fig. 9c is configuration II with the carefully designed fairing. Flow at the wing–body junction is predicted using numerical approaches described previously. The number of hybrid meshes is approximately one million, and the minimum height of the layer partitioned is 2.0×10^{-5}. The parameters are $\alpha = -2$–3 deg, $Re = 1.5 \times 10^7$, $Ma = 0.76$ and 0.80, and the reference length is taken as the mean aerodynamic chord. The computational results are compared with the experimental measurements in this section.

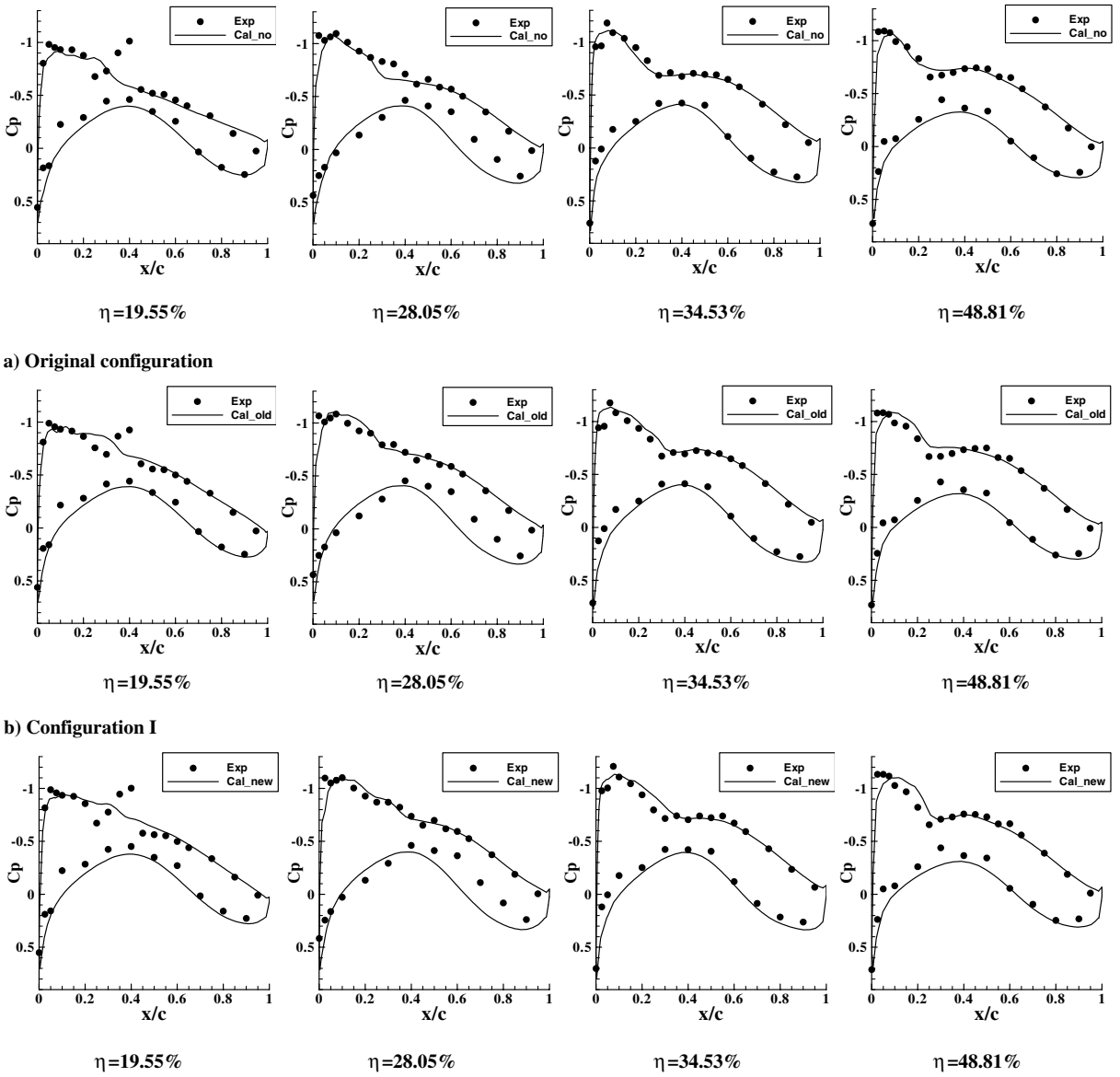


Fig. 12 Pressure coefficient distributions for three configurations with $Re = 1.5 \times 10^7$, $Ma = 0.76$, and $\alpha = 2$ deg at four semispans.

A. Influences on the Flow Separation

The spatial streamlines at the wing–body junction of three configurations with α from 1.0 to 3.0 deg are compared in Fig. 10 at $Ma = 0.76$. For the original configuration at $\alpha = 10$ deg, there is a large separation region. The separation region of configuration I is smaller, and that of configuration II is not obvious. Along with the increasing angle of attack, the separate regions of the three configurations expand. Relatively, the flow separation of the original configuration (for which the spatial streamlines have already undergone chaos) is the most severe. It proves that configurations I and II can stabilize the boundary layer and retard flow separation at the wing–body junction. What delights the designers most is that configuration II presents a little better efficiency.

Comparison between the separated streamlines and oil photos is shown in Fig. 11. The parameters are $Re = 1.5 \times 10^7$, $\alpha = 2$ deg, and $Ma = 0.80$. Figure 11a is the original configuration. It shows that surface streamlines separate severely at the wing–body junction. That's because the curvature of the top wing at the trailing edge constricts faster, which results in flow separation easily in the process of recovering the pressure. Figures 11b and 11c are, respectively, configurations I and II. Flow separation also occurs at the wing–body junction; however, the separated region of configuration II is smaller than configuration I. The numerical results agree well with the experimental measurements. It shows that configuration II is more efficient.

B. Comparison of the Pressure Coefficient Distributions

The C_p profiles are presented in Fig. 12 for three configurations with $Ma = 0.76$, $Re = 1.5 \times 10^7$, and $\alpha = 2$ deg at four different semispans: $\eta = 19.55$, 28.05, 34.53, and 48.81%. In the figure legends, no represents the original configuration, old represents configuration I, and new indicates configuration II.

The y axis is C_p, and the x axis is the semispan. The points in the figure are experimental measurements. It shows that the pressure gradient after the peak of the two fairing configurations becomes smoother than the original configuration. It indicates that the pressure recovery slows down. These changes will suppress and retard flow separation at the wing–body junction. The most satisfactory result is that the C_p curves are basically in agreement with the position of the shock wave. But a shortcoming is that computed results could not resolve the secondary suction peak in the inboard region, which may result in bad prediction in drag.

For the differences between computational and experimental results on the underside of the wing at the 28.05% span, we think that

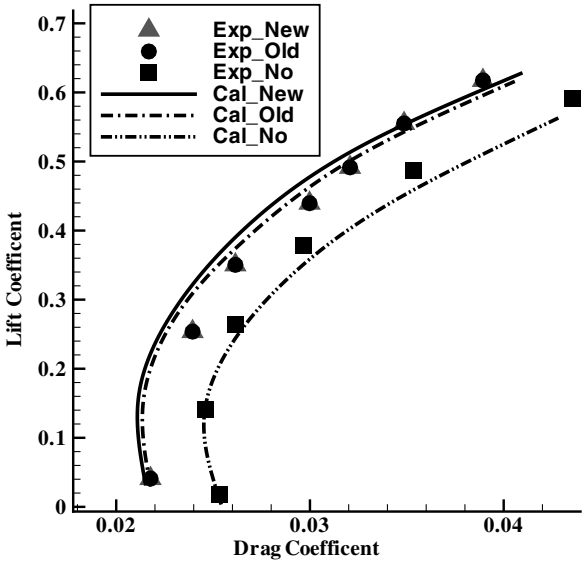


Fig. 13 Comparison of drag polar for different configurations with $Ma = 0.76$.

there was some error in the experimental results, for we have found the same phenomena of the computational results with different Mach number and angle of attack at the 28.05% span.

C. Effects on Lift and Drag

The comparison of the drag polar of three different configurations is presented in Fig. 13. The x axis is the drag coefficient, and the y axis is the lift coefficient.

The curves in the figure show that the drag of the original configuration is higher under the same lift compared with the other two configurations. The drag polar of configurations 1 and II is quite close to each other. The experimental measurements are also shown to evaluate the competence of the preceding algorithm. In all, numerical results of the flow simulation basically agree with the measurements. But we fail to accurately predict the increment in drag between the three geometries. Certainly, it is quite difficult to predict drag accurately in 3D, especially for complicated configurations. So we will make efforts on prediction of the drag in the following work. The results shown in the figures indicate that the lift/drag ratio of the original configuration could be remarkably increased. It can also be observed that configuration II is a little more effective than configuration I.

VI. Conclusions

It is known to all that flow simulation (especially, flow separation) is considerably tough for complicated configuration. But the prediction of flow separation that will greatly influence the performance of the aircraft is rather important. Fortunately, from the numerical results of flow simulation in this paper, it can be concluded that a basically accurate simulation of the flow at the wing–body junction by using an unstructured viscous flow solver is feasible. The algorithm in this paper takes on properties of good precision and efficiency.

Simultaneously, the numerical results indicate that the flow separation at the wing–body junction for conventional transporters is severe. With appropriate fairing of the junction, the separation can be greatly suppressed, and the aerodynamic properties of the transporter are obviously improved. Consequently, this algorithm is capable of simulating flow separation and helping designers design an upstanding fairing at the wing–body junction or the junctions of other aerodynamic components.

Acknowledgments

This work was supported by the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of the Ministry of Education, People's Republic of China. The author would also like to thank Yang Yongnian.

References

[1] Paciorri, R., Mascio, A. D., and Favini, B., "A Comparative Study of Turbulence Models for Junction Flows," AIAA Paper 2002-2964, 2002.

[2] Krishnamurthy, R., Cagle, C. D., and Chandra, S., "A Numerical Simulation of Wing–Body Junction Flow," AIAA Paper 98-1054, 1998.

[3] Wang, G., and Ye, Z. Y., "Generation of Three Dimensional Mixed and Unstructured Grids and Its Application in Solving Navier–Stokes Equations," *Acta Aeronautica et Astronautica Sinica*, Vol. 24, No. 5, 2003, pp. 385–390.

[4] Wang, G., and Ye, Z. Y., "Mixed Element Type Unstructured Grid Generation and its Application to Viscous Flow Simulation," International Council of the Aeronautical Sciences, Paper ICAS 2004-2.4 (St.).R.3, Sept. 2004.

[5] Pirzadeh, S., "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layer Method," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 43–49.

[6] Spalart, P. R., and Allmaras S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, 1992.

[7] Batina, J. T., "Implicit Upwind Solution Algorithms for Three-Dimensional Unstructured Meshes," *AIAA Journal*, Vol. 31, No. 5, 1993, pp. 801–805.

[8] Tomaro, R. F., Strang, W. Z., and Sankar, L. N., "An Implicit Algorithm for Solving Time Dependent Flow on Unstructured Grids," AIAA Paper 97-0333, 1997.

[9] Nakahashi, K., and Sharov, D., "Direct Surface Triangulation Using the Advancing Front Method," AIAA Paper 95-1686, 1995.

[10] Pirzadeh, S., "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layer Method," AIAA Paper 94-0417, Jan. 1994.

[11] Kallinderis, Y., and Ward, S., "Prismatic Grid Generation with an Efficient Algebraic Method for Aircraft Configurations," AIAA Paper 92-2721, 1992.

[12] Liou, M. -S., "A Sequel to AUSM: AUSM$^+$," *Journal of Computational Physics*, Vol. 129, No. 2, 1996, pp. 364–382.