Large-Eddy Simulation of Transonic Buffet Using Matrix-Free Discontinuous Galerkin Method

Ngoc Cuong Nguyen,* Sebastien Terrana,† and Jaime Peraire‡
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

https://doi.org/10.2514/1.J060459

We present an implicit large-eddy simulation of transonic buffet over the OAT15A supercritical airfoil at Mach number 0.73, angle of attack 3.5 deg, and Reynolds number 3×10^6. The simulation is performed using a matrix-free discontinuous Galerkin (DG) method and a diagonally implicit Runge–Kutta scheme on graphics processor units. We propose a Jacobian-free Newton–Krylov method to solve nonlinear systems arising from the discretization of the Navier–Stokes equations. The method successfully predicts the buffet onset, the buffet frequency, and turbulence statistics owing to the high-order DG discretization and an efficient mesh refinement for the laminar and turbulent boundary layers. A number of physical phenomena present in the experiment are captured in our simulation, including periodical low-frequency oscillations of shock wave in the streamwise direction, strong shear layer detached from the shock wave due to shock-wave/boundary-layer interaction and small-scale structures broken down by the shear-layer instability in the transition region, and shock-induced flow separation. The pressure coefficient, the root mean square of the fluctuating pressure, and the streamwise range of the shock wave oscillation agree well with the experimental data. The results suggest that the proposed method can accurately predict the onset of turbulence and buffet phenomena at high Reynolds numbers without a subgrid scale model or a wall model.

Nomenclature

a	=	sound speed
C_p	=	pressure coefficient, $2(p - p_\infty)/(\rho_\infty u_\infty^2)$
c	=	chord
c_p	=	specific heat at constant pressure
E	=	total energy
h	=	element size
k	=	polynomial degree
M_∞	=	freestream Mach number
Pr	=	Prandtl number
p	=	static pressure
Re_c	=	Reynolds number based on chord length
T	=	temperature
T_s	=	computed nondimensionalized time period of shock wave oscillation
t	=	physical time
u_j	=	flow velocity components
u_τ	=	friction velocity
u_∞	=	freestream velocity
x	=	chordwise distance from wing apex
y	=	vertical distance from chord line
y^+	=	distance from wall in wall units
z	=	spanwise distance
α	=	angle of attack
β	=	bulk viscosity
γ	=	ratio of specific heats
Δt	=	time step
κ	=	thermal conductivity
μ	=	dynamic viscosity
ρ	=	density
τ_{ij}	=	viscous stress tensor
Ω	=	physical domain
*	=	dimensionalized value

Received 28 December 2020; revision received 22 October 2021; accepted for publication 5 December 2021; published online 22 February 2022.

*Principal Research Scientist, Department of Aeronautics and Astronautics; cuongng@mit.edu. Senior Member AIAA.

†Research Engineer; currently CEA, DAM, DIF, F-91297 Arpajon, France; sebastien.terrana@cea.fr.

‡Professor, Department of Aeronautics and Astronautics; peraire@mit.edu. Fellow AIAA.

I. Introduction

TRANSONIC flows over an airfoil result in complex interactions between shock waves and viscous boundary layers. A particularly interesting phenomenon concerning modern supercritical airfoils is transonic buffet, whereby the flow separation induces a large-scale self-sustained motion of the shock over the surface of the airfoil. Transonic buffet can cause large-scale lift oscillations and structural vibrations that limit an aircraft's flight envelope. While the shock buffet phenomenon has been widely studied since it was first observed by Hilton and Fowler [1] in the aftermath of the Second World War, there is still no consensus on the exact mechanism governing the buffet dynamics (see reviews [2,3]). Therefore, new experimental and numerical studies are still desirable to achieve a better understanding of the phenomenon.

Among the experimental investigations of transonic buffet, the test campaign organized by ONERA on the OAT15A airfoil produced a variety of high-quality pressure and velocity measurements suitable for comparisons with numerical studies. The OAT15A airfoil was investigated experimentally by Jacquin et al. [4] at a freestream Mach number 0.73 and a chord-based Reynolds number 3×10^6. The OAT15A is a supercritical wing section with a thickness-to-chord ratio of 12.3%, a chord of 0.23 m, and a blunt trailing edge measuring $0.005c$. The airfoil model was tripped on both sides at $x/c = 0.07$ from the leading edge to trigger transition to turbulence at that location. The trip was 3 mm wide, with 0.089 mm grains on the lower side and 0.102 mm on the upper side. In the experiment, a periodic self-sustained shock wave motion with the frequency of 69 Hz was observed over the top surface of the airfoil between $x/c = 0.35$ (most upstream shock position) and $x/c = 0.65$ (most downstream shock position) when the airfoil was kept at 3.5 deg angle of attack. The flow separates from the foot of the shock as the shock moves to its most upstream position and reattaches as the shock moves to its most downstream position.

The numerical prediction of transonic buffet is challenging due to the coexistence of multiple physical phenomena such as turbulent structures in the thin turbulent boundary layer (TBL) at high Reynolds number, unsteady shock wave, shock-induced separation, acoustic wave generation and propagation, and structural vibrations. Numerical tools for predicting transonic buffet rely mostly on turbulence models including Reynolds-averaged Navier–Stokes (RANS), unstead RANS (URANS), and detached-eddy simulation (DES).

RANS is not capable of accurately predicting transonic buffet because it cannot capture the periodic motion of the shock. In the numerical simulations of transonic buffet, turbulence models have an enormous influence on the the prediction accuracy of the buffet onset. For instance, the study of Thiery and Coustols [5] showed that the baseline k-omega (BSL) [6] and Spalart–Allmaras (SA) [7] models yielded a steady solution, while the k-kl turbulence (KKL) [8] and shear-stress transport (SST) [6] models produced periodic shock oscillations. DES is a hybrid approach to simulate turbulent flows by modeling the attached boundary layer with RANS, while resolving the unsteady separated flow with large-eddy simulation (LES). Several studies of transonic buffet over the OAT15A airfoil have been performed by using the various versions of DES [9–11]. Deck [9] performed both standard DES and zonal DES (ZDES) of the transonic buffet over the OAT15A airfoil. In that study, ZDES predicted the self-sustained motion of the shock wave and overestimated the size of the shock-induced separation, while standard DES did not reproduce the shock motion. Grossi et al. [10] performed delayed DES (DDES) over the OAT15A airfoil and compared the results with URANS and ZDES. Huang et al. [11] performed the improved DDES (IDDES) with appropriate blending between DDES and wall-modeled LES functionality. However, the sensitivity of the prediction accuracy on the switching location between RANS and LES is a major drawback of the DES approach.

Wall-modeled LES (WMLES) is another turbulence simulation approach in which LES attempts to resolve turbulence features in the outer part of the boundary layer, while the unresolved turbulence in the inner part of the boundary layer is modeled. The recent work of Fukushima and Kawai [12] is the first WMLES prediction of the transonic buffet over the OAT15A airfoil. In that work, a selective mixed-scale model [13] is used to compute the turbulent eddy viscosity in the LES-resolved region, and the unresolved inner layer is modeled by solving two coupled ordinary differential equations (ODEs) for the wall-parallel velocity and the temperature. The number of total grid points in the WMLES computation is 440 million grid points, which is about two orders of magnitude higher than the previously mentioned DES simulations. The results obtained by the WMLES show good agreement with the experiments, and turbulence structures are better resolved than those of ZDES, DDES, and IDDES simulations. When LES resolves the inner part of the boundary layer, the approach is called wall-resolved LES (WRLES). WRLES is expected to produce more accurate predictions than WMLES, at a higher computational cost [14]. Garnier and Deck [15] conducted the first WRLES prediction of the transonic buffet over the OAT15A airfoil. To limit the required computational effort, the flow is computed in two-dimensional (2D) RANS mode using the SA model on the pressure side of the airfoil and in LES mode (the Selective Mixed Scales Model) on the suction side and in the wake. Despite the zonal treatment of the flow, 20.8 million cells are used in their coarser grid, and twice as many are used in their finer grid. The WRLES predictions agree reasonably well with the experiments in terms of the mean field analysis and spectral analysis.

In the LES approach, the large-scale eddies of the flowfield are resolved, and the small scales are modeled using an appropriate subgrid-scale (SGS) model. When the small-scale eddies are directly resolved by using a fine computational grid and a small time-step size, the method is called direct numerical simulation (DNS). DNS remains prohibitive for turbulent flows at high Reynolds numbers due to the enormous grid points required to resolve the small-scale eddies. When the number of grid points is not sufficient to resolve the small-scale eddies but enough to resolve the large-scale eddies, the method is called underresolved DNS (UDNS) or sometimes referred to as implicit LES (ILES) [16]. In recent years, the use of discontinuous Galerkin (DG) methods for UDNS/ILES of transitional and turbulent flows gains considerable attention from researchers in computational fluid dynamics [16–23]. It is shown in [24] that for moderate polynomial degrees (between 2 and 4) DG methods introduce numerical dissipation in underresolved computations of convection-dominated flows, which acts as an implicit filter to dissipate the unresolved turbulent features. The numerical dissipation is localized near the Nyquist wave number and applied to the smallest resolved scales, while the amount of such dissipation depends mostly on the energy in those scales. Therefore, by choosing the element size h and the polynomial degree k, the numerical dissipation can be tuned to filter the unresolved scales appropriately. Recently, Pazner et al. [25] applied a high-order DG method to the ILES simulation of transonic buffet on the OAT15A airfoil to study the effect of mesh refinement, polynomial degree, and artificial viscosity parameters.

In this paper, we propose an implicit matrix-free discontinuous Galerkin method for ILES computation of transonic buffet over the OAT15A supercritical airfoil at at Mach number 0.73, angle of attack 3.5 deg and Reynolds number 3×10^6. To avoid the Courant–Friedrichs–Lewy (CFL) limitation on the time step size, we use diagonally implicit Rung–Kutta (DIRK) schemes to discretize the time derivative. We develop a Jacobian-free Newton–Krylov (JFNK) method to solve nonlinear systems arising from our spatial and temporal discretization of the Navier–Stokes equations. These methods are implemented using CUDA C/C++ with MPI-based parallelization to harness the computational power of NVIDIA graphics processor units (GPUs). Furthermore, accurate prediction of turbulent flows at high Reynolds numbers requires a good approximation of both laminar and turbulent flows in the boundary layers. To adequately resolve the boundary layers, we employ a mesh refinement strategy that refines the grid toward the wall and keeps the cell aspect ratio sufficiently small. Our ILES method is used to predict the buffet onset, the buffet frequency, and turbulence statistics and capture various turbulence phenomena such as periodical low-frequency oscillations of shock wave in the streamwise direction, strong shear layer detached from the shock wave due to shock-wave/boundary-layer interaction (SWBLI) and small-scale structures broken down by the shear layer instability in the transition region, and shock-induced flow separation. The pressure coefficient, the rms of the fluctuating pressure, and the streamwise range of the shock wave oscillation are in good agreement with the experimental data [4].

The results reported in this paper demonstrate that our method can accurately predict the onset of turbulence and buffet phenomena at high Reynolds numbers without a subgrid-scale model or a wall model. The successful computation of transonic buffet over the OAT15A airfoil could represent an important step toward the development and application of wall-resolved simulations for turbulent flows at high Reynolds numbers. From our perspective, algorithmic advances in several fronts from discretization schemes, mesh adaptivity, and iterative solvers to efficient implementations on many-core processors are necessary to make wall-resolved simulations feasible for flows at Reynolds numbers that are beyond the reach of DNS for a foreseeable future. Wall-resolved simulations are useful for nonequilibrium three-dimensional turbulent boundary layers for which existing wall models lack of capabilities to capture the flow physics. In particular, nonequilibrium three-dimensional turbulent boundary layers have peculiar features that are challenging to model, such as the misalignment of the Reynolds shear stress and the mean shear stress.

The paper is organized as follows. In Sec. II, we describe the numerical methodology to discretize the Navier–Stokes equations. In Sec. III, we discuss and compare the obtained results with the experimental data [4] and with the available numerical experiments: ZDES [9], DDES [10], coupled RANS/LES [15], and WMLES [12]. Finally, some concluding remarks and a rationale for the success of ILES for transonic buffet prediction are presented in Sec. IV.

II. Methodology

A. Governing Equations

Let $t_f > 0$ be a final time, and let $\Omega \subset \mathbb{R}^d$, $1 \leq d \leq 3$ be an open, connected, and bounded physical domain with Lipschitz boundary $\partial\Omega$. The unsteady compressible Navier–Stokes equations in conservation form are given by

$$\boldsymbol{q} - \nabla \boldsymbol{u} = 0, \qquad \text{in } \Omega \times [0, t_f) \tag{1a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{u}, \boldsymbol{q}) = 0, \qquad \text{in } \Omega \times [0, t_f) \tag{1b}$$

$$\boldsymbol{B}(\boldsymbol{u},\boldsymbol{q}) = 0, \qquad \text{on } \partial\Omega \times [0, t_f) \tag{1c}$$

$$\boldsymbol{u} - \boldsymbol{u}_0 = 0, \qquad \text{on } \Omega \times \{0\} \tag{1d}$$

Here, $\boldsymbol{u} = (\rho, \rho u_j, \rho E)$, $j = 1, \ldots, d$ is the m-dimensional ($m = d + 2$) vector of conserved quantities, $\boldsymbol{u}_0$ is an initial state, $\boldsymbol{B}(\boldsymbol{u},\boldsymbol{q})$ is a boundary operator, and $\boldsymbol{F}(\boldsymbol{u},\boldsymbol{q})$ are the inviscid and viscous fluxes of dimensions $m \times d$,

$$\boldsymbol{F}(\boldsymbol{u},\boldsymbol{q}) = \begin{pmatrix} \rho\boldsymbol{u} \\ \rho\boldsymbol{u} \otimes \boldsymbol{u} + p\boldsymbol{I} - \boldsymbol{\tau} \\ \boldsymbol{u}(\rho E + p) - \boldsymbol{\tau}\boldsymbol{u} + \boldsymbol{f} \end{pmatrix} \tag{2}$$

where p denotes the thermodynamic pressure, $\boldsymbol{\tau}$ is the viscous stress tensor, $\boldsymbol{f}$ is the heat flux, and $\boldsymbol{I}$ is the identity tensor. For a calorically perfect gas in thermodynamic equilibrium, $p = (\gamma - 1)(\rho E - \rho|\boldsymbol{u}|^2/2)$, where $\gamma = c_p/c_v > 1$ is the ratio of specific heats and in particular $\gamma \approx 1.4$ for air; c_p and c_v are the specific heats at constant pressure and volume, respectively. For a Newtonian fluid with the Fourier law of heat conduction, the viscous stress tensor and heat flux are given by

$$\tau_{ij} = \mu_f \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \right) + \beta_f \frac{\partial u_k}{\partial x_k}\delta_{ij}, \quad f_j = -\kappa \frac{\partial T}{\partial x_j} \tag{3}$$

where T denotes temperature, μ_f is the dynamic (shear) viscosity, β_f is the bulk viscosity, $\kappa = c_p\mu/Pr$ is the thermal conductivity, and Pr is the Prandtl number; $Pr \approx 0.71$ for air, and $\beta_f = 0$ under the Stokes hypothesis.

To deal with shock waves and discontinuities, we add artificial viscosities to the physical ones as

$$\beta = \beta_f + \bar{\beta}^*, \qquad \mu = \mu_f + \bar{\mu}^* \tag{4}$$

where $\bar{\beta}^*$ and $\bar{\mu}^*$ are the artificial bulk viscosity and artificial shear viscosity, respectively. The governing equations numerically discretized herein are still Eqs. (1–3), where both μ_f and β_f in Eq. (3) are replaced with μ and β in Eq. (4), respectively. The artificial viscosities are defined in the next section.

B. Shock Capturing

In this work, we modify the physics-based artificial viscosity approach introduced in our previous work [26,27]. Specifically, we follow Fernandez et al. [27] without the artificial thermal conductivity which is designed for the strong shocks appearing in hypersonic flows. This approach relies on shock and shear sensors. The shock sensor, evaluated pointwise, is constructed such that

$$s_\beta(\boldsymbol{x}) = s_d \cdot s_{\boldsymbol{\omega}}, \quad s_d = -\frac{h}{k}\frac{\nabla\cdot\boldsymbol{u}}{a^*}, \quad s_{\boldsymbol{\omega}} = \frac{(\nabla\cdot\boldsymbol{u})^2}{(\nabla\cdot\boldsymbol{u})^2 + |\nabla\times\boldsymbol{u}|^2 + \varepsilon} \tag{5}$$

where ε is a constant of the order of the machine precision squared, k is the polynomial degree and a^* is the critical speed of sound. In Eq. (5), the dilatation sensor s_d is multiplied with Ducros's indicator [28] $s_{\boldsymbol{\omega}}$ to avoid adding artificial viscosity to vortices. The element size is taken along the direction of the density gradient

$$h(\boldsymbol{x}) = h_{\text{ref}} \frac{|\nabla\rho|}{\sqrt{\nabla\rho^T \cdot \boldsymbol{M}_h^{-1}\nabla\rho + \varepsilon}} \tag{6}$$

where $\boldsymbol{M}_h$ is the metric tensor of the mesh and h_{ref} is the size of the reference element used in the construction of $\boldsymbol{M}_h$. The shear sensor is also designed to detect underresolved features, namely, velocity gradients, and is constructed from

$$s_\mu(\boldsymbol{x}) = \frac{h_{\text{ref}}}{k}\frac{\|\boldsymbol{\mathcal{L}}(\boldsymbol{u})\|_2}{u_\infty} \tag{7}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, u_∞ is the freestream velocity, and $\boldsymbol{\mathcal{L}}(\boldsymbol{u}) = (1/2)(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)$. Note that the sensors not only detect discontinuities but also determine the amount of artificial viscosity to stabilize them.

Finally, we define artificial bulk viscosity and artificial molecular viscosity as

$$\beta^*(\boldsymbol{x}) = \hat{s}_\beta \frac{k_\beta h}{k}\sqrt{|\boldsymbol{u}|^2 + a^{*2}}, \quad \mu^*(\boldsymbol{x}) = \hat{s}_\mu \frac{k_\mu h}{k}\sqrt{|\boldsymbol{u}|^2 + a^{*2}} \tag{8}$$

Here, $k_{\mu,\kappa} = 1$, $k_\beta = 1.5$, and $(\hat{s}_\beta, \hat{s}_\mu)$ denote the smoothly bounded values of the sensors in Eqs. (5) and (7) and are given by

$$\hat{s}_\beta(\boldsymbol{x}) = \ell(s_\beta; s_{\beta,0}, s_{\beta,\max}), \qquad \hat{s}_\mu(\boldsymbol{x}) = \ell(s_\mu; s_{\mu,0}, s_{\mu,\max}) \tag{9}$$

The function ℓ represents a smooth approximation to the following limiting function: $L(s; s_0, s_{\max}) = \min\{\max\{s - s_0, 0\} - s_{\max}, 0\} + s_{\max}$. In particular, it is defined as follows:

$$\ell(s; s_0, s_{\max}) = \ell_{\min}(\ell_{\max}(s - s_0) - s_{\max}) + s_{\max} \tag{10}$$

where

$$\begin{aligned} \ell_{\max}(s) &= \frac{s}{\pi}\arctan(100s) + \frac{s}{2} - \frac{1}{\pi}\arctan(100) + \frac{1}{2}, \\ \ell_{\min}(s) &= s - \ell_{\max}(s) \end{aligned} \tag{11}$$

Here, the first parameter s_0 represents the starting point of the limiting function ℓ where it begins to increase with s, while the second parameter $s_{\max} > 0$ is the upper bound of the nonnegative variable s. The shock parameters are chosen as $s_{\beta,\max} = s_{\mu,\max} = 2$, $s_{\beta,0} = 0,01$, and $s_{\mu,0} = 1$ according to Fernandez et al. [27].

Because the original artificial viscosity fields (β^*, μ^*) are discontinuous, a node-averaging operator is applied to (β^*, μ^*) to make them $\mathcal{C}^0$ continuous. The smooth reconstruction of the artificial viscosity field $\beta^*(\boldsymbol{x})$ is done by averaging all the multiple values of $\beta^*(\boldsymbol{x})$ along the element boundaries to obtain a continuous field $\bar{\beta}^*(\boldsymbol{x})$. The proposed reconstruction is particular to the DG discretization. Let $\boldsymbol{x}_n$, $1 \le n \le n_k n_e$, be DG nodes of a high-order finite element mesh $\mathcal{T}_h$, where n_k is the number of nodes per element and n_e is the number of elements. For every node $\boldsymbol{x}_n$, $\bar{\beta}^*(\boldsymbol{x}_n) = (1/J_n)\sum_{j=1}^{J_n}\beta^*(\boldsymbol{x}_n)|_{K_j}$, where K_j, $1 \le j \le J_n$, are all the elements in which $\boldsymbol{x}_n$ is located. If a mesh node $\boldsymbol{x}_n$ is located inside an element, then $J_n = 1$, and if it is located on a face, then $J_n = 2$. If it is located on an edge or at an element vertex, then J_n is equal to the number of elements connected to that edge or that vertex, respectively. In essence, $\bar{\beta}^*(\boldsymbol{x}_n)$ is a polynomial of degree k on every element and continuous across element interfaces. Therefore, the present reconstruction is different from the elementwise linear reconstruction used in [26,27].

C. Discontinuous Galerkin Method

Let $\Omega \subseteq \mathbb{R}^d$ with $d = 3$ be a physical domain with Lipschitz boundary $\partial\Omega$. We denote by $\mathcal{T}_h$ a collection of disjoint, regular, kth degree curved elements K that partition Ω and set $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$ to be the collection of the boundaries of the elements in $\mathcal{T}_h$. Let $\mathcal{P}^k(D)$ denote the space of complete polynomials of degree k on a domain $D \in \mathbb{R}^n$, let $L^2(D)$ be the space of square-integrable functions on D, and let $\boldsymbol{\psi}_K^k$ denote the kth degree parametric mapping from the reference element K_{ref} to some element $K \in \mathcal{T}_h$ in the physical domain. We then introduce the discontinuous finite element spaces

$$\begin{aligned} \boldsymbol{\mathcal{Q}}_h^k &= \{\boldsymbol{r} \in [L^2(\mathcal{T}_h)]^{m\times d} : (\boldsymbol{r} \circ \boldsymbol{\psi}^k)|_K \in [\mathcal{P}^k(K_{\text{ref}})]^{m\times d} \forall K \in \mathcal{T}_h\}, \\ \boldsymbol{\mathcal{V}}_h^k &= \{\boldsymbol{w} \in [L^2(\mathcal{T}_h)]^m : (\boldsymbol{w} \circ \boldsymbol{\psi}^k)|_K \in [\mathcal{P}^k(K_{\text{ref}})]^m \forall K \in \mathcal{T}_h\} \end{aligned}$$

where m denotes the number of equations of the conservation law, in other words, $m = d + 2$ for the Navier–Stokes system. Next, we define several inner products associated with these finite element spaces as

$$(\boldsymbol{w}, \boldsymbol{v})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\boldsymbol{w}, \boldsymbol{v})_K = \sum_{K\in\mathcal{T}_h} \int_K \boldsymbol{w}\cdot\boldsymbol{v} \tag{12a}$$

$$(\boldsymbol{W}, \boldsymbol{V})_{\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} (\boldsymbol{W}, \boldsymbol{V})_K = \sum_{K\in\mathcal{T}_h} \int_K \boldsymbol{W}:\boldsymbol{V} \tag{12b}$$

$$\langle \boldsymbol{w}, \boldsymbol{v}\rangle_{\partial\mathcal{T}_h} = \sum_{K\in\mathcal{T}_h} \langle \boldsymbol{w}, \boldsymbol{v}\rangle_{\partial K} = \sum_{K\in\mathcal{T}_h} \int_{\partial K} \boldsymbol{w}\cdot\boldsymbol{v} \tag{12c}$$

for $\boldsymbol{w}, \boldsymbol{v} \in \boldsymbol{\mathcal{V}}_h^k$, $\boldsymbol{W}, \boldsymbol{V} \in \boldsymbol{\mathcal{Q}}_h^k$, where $\cdot$ and : denote the scalar product and Frobenius inner product, respectively.

The DG discretization of the governing equations reads as follows. Find $(\boldsymbol{q}_h(t), \boldsymbol{u}_h(t)) \in \boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k$ such that

$$(\boldsymbol{q}_h, \boldsymbol{r})_{\mathcal{T}_h} + (\boldsymbol{u}_h, \nabla\cdot\boldsymbol{r})_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{u}}_h, \boldsymbol{r}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = 0 \tag{13a}$$

$$\left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w}\right)_{\mathcal{T}_h} - (\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{q}_h), \nabla\boldsymbol{w})_{\mathcal{T}_h} + \langle \widehat{\boldsymbol{f}}_h(\boldsymbol{u}_h, \boldsymbol{q}_h), \boldsymbol{w}\rangle_{\partial\mathcal{T}_h} = 0 \tag{13b}$$

for all $(\boldsymbol{r}, \boldsymbol{w}) \in \boldsymbol{\mathcal{Q}}_h^k \times \boldsymbol{\mathcal{V}}_h^k$ and all $t \in [0, t_f)$, as well as

$$(\boldsymbol{u}_h|_{t=0} - \boldsymbol{u}_0, \boldsymbol{w})_{\mathcal{T}_h} = 0 \tag{13c}$$

for all $\boldsymbol{w} \in \boldsymbol{\mathcal{V}}_h^k$. Here, $\widehat{\boldsymbol{u}}_h$ is the numerical trace, and $\widehat{\boldsymbol{f}}_h$ is the numerical flux. For DG methods, both the numerical trace and flux must be continuous across element boundaries. The general form of the numerical trace and flux on the interior faces that satisfies the continuity requirement is given by

$$\begin{aligned}
\widehat{\boldsymbol{u}}_h &= \frac{1}{2}(\boldsymbol{u}_h^+ + \boldsymbol{u}_h^-) + (\boldsymbol{u}_h^+\boldsymbol{\beta}\cdot\boldsymbol{n}^+ + \boldsymbol{u}_h^-\boldsymbol{\beta}\cdot\boldsymbol{n}^-) + \boldsymbol{\gamma}\cdot(\boldsymbol{q}_h^+\cdot\boldsymbol{n}^+ + \boldsymbol{q}_h^-\cdot\boldsymbol{n}^-),\\
\widehat{\boldsymbol{f}}_h &= \frac{1}{2}(\boldsymbol{F}(\boldsymbol{u}_h^+, \boldsymbol{q}_h^+) + \boldsymbol{F}(\boldsymbol{u}_h^-, \boldsymbol{q}_h^-))\cdot\boldsymbol{n}^+ + \boldsymbol{\sigma}\cdot(\boldsymbol{u}_h^+ - \boldsymbol{u}_h^-)
\end{aligned} \tag{14}$$

where $\boldsymbol{\beta}$ is a vector-valued function and $\boldsymbol{\gamma}$, $\boldsymbol{\sigma}$ are a matrix-valued function. Note that $\boldsymbol{u}_h^+ = \boldsymbol{u}_h|_{F\in K^+}$ and $\boldsymbol{u}_h^- = \boldsymbol{u}_h|_{F\in K^-}$ denote the restriction of the numerical solution $\boldsymbol{u}_h$ on interior face F shared by elements K^+ and K^-. On the boundary faces, the definition of the numerical trace and flux depends on the boundary conditions. For the computation performed herein, there are two types of boundary conditions, namely, the far-field condition and adiabatic wall condition. We refer to [16,29] for the implementation of these two boundary conditions.

Different choices of the stabilization functions $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\sigma}$ result in different DG methods. The Local DG (LDG) method [30] corresponds to $\boldsymbol{\gamma} = 0$, the second Bassi–Rebay (BR2) method [31] corresponds to $\boldsymbol{\beta} = \boldsymbol{\gamma} = 0$, and the first Bassi–Rebay (BR1) method [32] corresponds to $\boldsymbol{\beta} = \boldsymbol{\gamma} = \boldsymbol{\sigma} = 0$. These stabilization functions play an important role in the stability and accuracy of the resulting DG method. Indeed, it is known that the BR1 method is not stable for elliptic problems [33]. The hybridized DG (HDG) method [16,29] does not define the numerical trace $\widehat{\boldsymbol{u}}_h$ terms of the approximate solution. In the HDG method, the numerical trace $\widehat{\boldsymbol{u}}_h$ becomes a dependent variable to be solved together with $(\boldsymbol{u}_h, \boldsymbol{q}_h)$ by introducing another equation that weakly imposes the continuity of the numerical flux. The HDG method is computationally efficient when we form and solve the matrix system because it results in smaller matrix system than the LDG method and the BR2 method. However, the HDG method is not suited to our matrix-free approach because computing the residual of the HDG method involves solving nonlinear local problems for $(\boldsymbol{q}_h, \boldsymbol{u}_h)$ in terms of $\widehat{\boldsymbol{u}}_h$. Hence, it would be computationally expensive to use the HDG method within the JFNK approach.

In this paper, the BR2 method is used to discretize the compressible Navier–Stokes equations in space because it is suited to the matrix-free solution method introduced in the next section. Herein, we choose $\boldsymbol{\beta} = \boldsymbol{\gamma} = 0$ and $\boldsymbol{\sigma} = \lambda_{\max}(\widehat{\boldsymbol{u}}_h)\boldsymbol{I}$, where $\lambda_{\max}$ denotes the maximum-magnitude eigenvalue of $\boldsymbol{A}_n(\boldsymbol{u}_h) = [\partial\boldsymbol{F}_{\text{inv}}(\boldsymbol{u}_h)/\partial\boldsymbol{u}]\cdot\boldsymbol{n}$ with $\boldsymbol{F}_{\text{inv}}$ being the inviscid part of the flux function $\boldsymbol{F}$. For general convection/diffusion problems, the stabilization term $\boldsymbol{\sigma}$ should include both the convection-stabilizing term $\boldsymbol{\sigma}_c$ and the diffusion-stabilizing term $\boldsymbol{\sigma}_d$, namely, $\boldsymbol{\sigma} = \boldsymbol{\sigma}_c + \boldsymbol{\sigma}_d$, so that the resulting scheme can be stable in both pure convection limit and pure diffusion limit. The convection-stabilizing term is usually computed by using approximate Riemann solvers such as HLL/HLLC schemes, Roe's scheme, and the Lax–Friedrich scheme, while the diffusion-stabilizing term is proportional to the diffusion coefficient. Because our particular problem presented in the next section is strongly convection dominated, the diffusion-stabilizing term $\boldsymbol{\sigma}_d \sim 1/Re$ can be neglected. See [16,29,34,35] for additional discussion on the stabilization of DG methods.

For computational efficiency, we can eliminate $\boldsymbol{q}_h$ to obtain a new system in terms of $\boldsymbol{u}_h$ only as follows. It follows from Eqs. (13a) and (14) that

$$\boldsymbol{q}_h = \nabla\boldsymbol{u}_h + \mathcal{L}(\boldsymbol{u}_h) \tag{15}$$

where $\mathcal{L}(\boldsymbol{u}_h) \in \mathcal{P}^k(K)$ is solved in an element-by-element fashion as

$$(\mathcal{L}(\boldsymbol{u}_h), \boldsymbol{r})_K = \langle \widehat{\boldsymbol{u}}_h - \boldsymbol{u}_h, \boldsymbol{r}\cdot\boldsymbol{n}\rangle_{\partial K}, \quad \forall\, \boldsymbol{r} \in \mathcal{P}^k(K) \tag{16}$$

for every element $K \in \mathcal{T}_h$. Because $\widehat{\boldsymbol{u}}_h$ depends linearly on $\boldsymbol{u}_h$, $\boldsymbol{q}_h$ also linearly depends on $\boldsymbol{u}_h$. Hence, we can substitute (15) into (13b) to obtain the following weak form. Find $\boldsymbol{u}_h(t) \in \boldsymbol{\mathcal{V}}_h^k$ such that

$$\left(\frac{\partial \boldsymbol{u}_h}{\partial t}, \boldsymbol{w}\right)_{\mathcal{T}_h} - (\boldsymbol{F}(\boldsymbol{u}_h, \mathcal{D}(\boldsymbol{u}_h)), \nabla\boldsymbol{w})_{\mathcal{T}_h} + \langle \widehat{\boldsymbol{f}}_h(\boldsymbol{u}_h, \mathcal{D}(\boldsymbol{u}_h)), \boldsymbol{w}\rangle_{\partial\mathcal{T}_h} = 0 \tag{17}$$

for all $\boldsymbol{w} \in \boldsymbol{\mathcal{V}}_h^k$, where $\mathcal{D}(\boldsymbol{u}_h) = \nabla\boldsymbol{u}_h + \mathcal{L}(\boldsymbol{u}_h)$. The given weak formulation can be written as a system of ordinary differential equations in matrix form as

$$\boldsymbol{M}\frac{d\boldsymbol{u}}{dt} + \boldsymbol{g}(\boldsymbol{u}) = 0 \tag{18}$$

where $\boldsymbol{u}$ is the vector of degrees of freedom of $\boldsymbol{u}_h$, $\boldsymbol{M}$ is the mass matrix, and $\boldsymbol{g}(\boldsymbol{u})$ is a nonlinear vector-valued function corresponding to the last two terms of Eq. (17).

Finally, the semidiscrete system (18) is further discretized in time using the L-stable DIRK schemes [36]. The use of L-stable DIRK methods for the temporal discretization is important to maintain accuracy and stability because the DG discretization of turbulent shock flows at high Reynolds number results in very stiff nonlinear ODE systems. If the use of an implicit time scheme allows us to alleviate the CFL limitation on the time-step size, the latter cannot be arbitrary large in practice. Whenever LES or DNS is the aim, the time step has to be small enough to resolve the viscous time scales (see [37] and also Sec. III.A.3). Moreover, for aeroacoustic problems, the time step also has to sample the acoustic high frequencies of interest.

D. Solution Method

Our solution method aims to find good initial guesses for both nonlinear and linear systems in order to reduce the number of iterations. Newton's method is used to solve the nonlinear system of equations, $\boldsymbol{R}(\boldsymbol{u}^n) = 0$, resulting from the temporal discretization of the system (18), where $\boldsymbol{u}^n \in \mathbb{R}^{n_{\text{dof}}}$ is the vector of degrees of freedom of $\boldsymbol{u}_h^n$ and $\boldsymbol{R}(\cdot) \in \mathbb{R}^{n_{\text{dof}}}$ is the residual vector. We note that $n_{\text{dof}} = (d+2)n_k n_e$, where n_k is the number of DG nodes per

element and n_e is the number of elements. Furthermore, the time-stage step is given by $n = (n_t - 1)n_s + s$, where n_t is the n_tth time-step, n_s is the number of stages for a DIRK scheme, and s is the sth stage. To reduce the number of Newton iterations, we propose to compute the initial guess $\boldsymbol{u}^{n,0}$ as

$$\boldsymbol{u}^{n,0} \coloneqq \sum_{j=1}^{n_\alpha} \alpha_j \boldsymbol{u}^{n-j} \tag{19}$$

where the $\boldsymbol{u}^{n-j}$ are the previously computed solutions of the nonlinear systems and the coefficients α_j are found by solving the following least-squares problem

$$(\alpha_1, \ldots, \alpha_{n_\alpha}) = \arg \min_{(\beta_1, \ldots, \beta_{n_\alpha}) \in \mathbb{R}^{n_\alpha}} \left\| \boldsymbol{R}\left(\sum_{j=1}^{n_\alpha} \beta_j \boldsymbol{u}^{n-j} \right) \right\| \tag{20}$$

This optimization problem is solved by using the Levenberg–Marquardt algorithm [38] in which the gradient vectors $\partial \boldsymbol{R}/\partial \beta_j$ are approximated by finite difference and thus require n_α residual evaluations. This procedure introduced in [16] aims to reduce the number of Newton iterations compared to the standard initialization using the previous solution as the initial guess. For the computation presented in this paper, the number of Newton iterations required to converge for a specified tolerance 10^{-8} is equal to 2 for $n_\alpha = 5$ and 3 for $n_\alpha = 1$. Hence, the initialization (19) reduces the number of Newton iterations from 3 to 2, compared to the standard initialization that uses the previous solution as the initial guess. This can substantially reduce the computational cost because the number of GMRES iterations per Newton iteration is about 30 for our simulation of the buffeting phenomena described in the next section.

For every mth iteration of the Newton method, we use GMRES to solve the resulting linear system

$$\boldsymbol{J}(\boldsymbol{u}^{n,m}) \delta \boldsymbol{u}^{n,m} = -\boldsymbol{R}(\boldsymbol{u}^{n,m}) \tag{21}$$

for the Newton increment $\delta \boldsymbol{u}^{n,m}$, where $\boldsymbol{J}(\boldsymbol{u}^{n,m}) = \partial \boldsymbol{R}(\boldsymbol{u}^{n,m})/\partial \boldsymbol{u}$ is the Jacobian matrix. In what follows, we shall drop the superscript m to simplify the notation. Hence, the superscript n should be understood as the superscript n, m in the remainder of this section. Following the same idea applied to the Newton method, we construct a reduced basis (RB) approximation [39–41] to $\delta \boldsymbol{u}^n$ and use it as an initial guess in order to accelerate the GMRES method. Given a reduced basis $\boldsymbol{W}_n = \text{span}\{\delta \boldsymbol{u}^{n-j}, 1 \le j \le n_{\text{rb}}\}$ consisting of n_{rb} previous solution vectors of the linear systems, the RB approximation to $\delta \boldsymbol{u}^n$ is computed as the best least-squares solution of the following problem:

$$\delta \boldsymbol{u}^n_{\text{rb}} = \arg \min_{\delta \boldsymbol{w} \in \boldsymbol{W}_n} \| \boldsymbol{J}(\boldsymbol{u}^n) \delta \boldsymbol{w} + \boldsymbol{R}(\boldsymbol{u}^n) \| \tag{22}$$

It thus follows that $\delta \boldsymbol{u}^n_{\text{rb}} = \boldsymbol{W}_n \boldsymbol{a}^n_{\text{rb}}$, where $\boldsymbol{a}^n_{\text{rb}} \in \mathbb{R}^{n_{\text{rb}}}$ is the solution of the RB system, $\boldsymbol{J}_{\text{rb}}(\boldsymbol{u}^n) \boldsymbol{a}^n_{\text{rb}} = -\boldsymbol{R}_{\text{rb}}(\boldsymbol{u}^n)$, with $\boldsymbol{J}_{\text{rb}}(\boldsymbol{u}^n) = (\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n)^T (\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n)$ and $\boldsymbol{R}_{\text{rb}}(\boldsymbol{u}^n) = (\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n)^T \boldsymbol{R}(\boldsymbol{u}^n)$. Forming the RB system requires $\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n$, which is approximately computed by the finite difference (28). Typically, we set $n_{\text{rb}} = 5$, and the cost of inverting the RB system is thus negligible. The RB approximation $\delta \boldsymbol{u}^n_{\text{rb}}$ is used as the initial guess in the GMRES method that solves (21) for $\delta \boldsymbol{u}^n$. We thus obtain $\delta \boldsymbol{u}^n = \delta \boldsymbol{v}^n + \delta \boldsymbol{u}^n_{\text{rb}}$, where $\delta \boldsymbol{v}^n$ is the best least-squares solution of the following problem:

$$\delta \boldsymbol{v}^n = \arg \min_{\delta \boldsymbol{w} \in \mathcal{K}_r(\boldsymbol{J}(\boldsymbol{u}^n), \boldsymbol{b}(\boldsymbol{u}^n))} \| \boldsymbol{J}(\boldsymbol{u}^n) \delta \boldsymbol{w} + \boldsymbol{b}(\boldsymbol{u}^n) \|,$$

$$\text{with} \quad \boldsymbol{b}(\boldsymbol{u}^n) = \boldsymbol{R}(\boldsymbol{u}^n) + \boldsymbol{J}(\boldsymbol{u}^n) \delta \boldsymbol{u}^n_{\text{rb}} \tag{23}$$

Here, $\mathcal{K}_r(\boldsymbol{J}(\boldsymbol{u}^n), \boldsymbol{b}(\boldsymbol{u}^n)) = \text{span}\{\boldsymbol{b}(\boldsymbol{u}^n), \boldsymbol{J}(\boldsymbol{u}^n)\boldsymbol{b}(\boldsymbol{u}^n), \ldots, (\boldsymbol{J}(\boldsymbol{u}^n)^r \boldsymbol{b}(\boldsymbol{u}^n)\}$ is the Krylov subspace at the rth iteration. Therefore, it follows from Eqs. (22) and (23) that our method is similar to the restarted GMRES method. The main difference between our method and the restarted GMRES method is that the restarted GMRES method uses the Krylov subspace $\mathcal{K}_r(\boldsymbol{J}(\boldsymbol{u}^n), \boldsymbol{R}(\boldsymbol{u}^n))$, whereas our method uses the RB space $\boldsymbol{W}_n$ for the first outer iteration.

Furthermore, we make use of the reduced basis $\boldsymbol{W}_n$ to construct a matrix-free preconditioner to be used in the GMRES method for solving (23). The idea lies in the construction of an approximation to the Jacobian matrix $\boldsymbol{J}(\boldsymbol{u}^n)$ through a suitable low-rank approximation. In particular, the preconditioner has the form

$$\boldsymbol{P}_n = \boldsymbol{M} + \boldsymbol{V}_n \boldsymbol{D}_n^{-1} \boldsymbol{W}_n^T \tag{24}$$

where $\boldsymbol{M}$ is a matrix whose inverse is inexpensive to compute, while $\boldsymbol{V}_n$ and $\boldsymbol{D}_n$ are chosen to satisfy the following condition:

$$\boldsymbol{P}_n \boldsymbol{W}_n = \boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n \tag{25}$$

It thus follows that we obtain $\boldsymbol{D}_n = \boldsymbol{W}_n^T \boldsymbol{W}_n$ and $\boldsymbol{V}_n = \boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n - \boldsymbol{M} \boldsymbol{W}_n$. Using the Sherman–Morrison–Woodbury formula, we can compute the inverse of the preconditioner $\boldsymbol{P}_n$ as

$$\boldsymbol{P}_n^{-1} = \boldsymbol{M}^{-1} - \boldsymbol{M}^{-1} \boldsymbol{V}_n (\boldsymbol{D}_n + \boldsymbol{W}_n^T \boldsymbol{M}^{-1} \boldsymbol{V}_n)^{-1} \boldsymbol{W}_n^T \boldsymbol{M}^{-1} \tag{26}$$

The preconditioner of the form (24) is similar to the BFGS update [42–45] with a distinctive feature that our approach allows for arbitrary-rank approximation, whereas the BFGS update is only a rank-2 approximation of the Jacobian matrix. In this paper, we use the mass matrix to form the matrix $\boldsymbol{M}$. Note that the mass matrix and its inverse are block diagonal and computed only once. It follows from Eq. (26) that the product of $\boldsymbol{P}_n^{-1}$ with any given vector $\boldsymbol{y}$ is given by

$$\boldsymbol{P}_n^{-1} \boldsymbol{y} = \boldsymbol{z} - \boldsymbol{s} \tag{27}$$

where $\boldsymbol{z} = \boldsymbol{M}^{-1} \boldsymbol{y}, \boldsymbol{w} = \boldsymbol{W}_n^T \boldsymbol{z}, \boldsymbol{p} = \boldsymbol{H}_n \boldsymbol{w}, \boldsymbol{d} = \boldsymbol{V}_n \boldsymbol{p}, \boldsymbol{s} = \boldsymbol{M}^{-1} \boldsymbol{d}$ with $\boldsymbol{H}_n = (\boldsymbol{D}_n + \boldsymbol{W}_n^T \boldsymbol{M}^{-1} \boldsymbol{V}_n)^{-1}$. As a result, we compute and store $\boldsymbol{W}_n \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{rb}}}$, $\boldsymbol{V}_n \in \mathbb{R}^{n_{\text{dof}} \times n_{\text{rb}}}$, and $\boldsymbol{H}_n \in \mathbb{R}^{n_{\text{rb}} \times n_{\text{rb}}}$. Because the RB dimension n_{rb} is very small, computing Eq. (27) is considerably less expensive than evaluating the residual vector.

Our solution method requires the computation of $\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n$, which can be expensive if we have to form the Jacobian matrix $\boldsymbol{J}(\boldsymbol{u}_n)$ and perform matrix/matrix multiplication. Instead, the product of the Jacobian matrix with any vector $\boldsymbol{y}$ can be approximately computed by the Taylor expansion as

$$\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{y} \approx \frac{\boldsymbol{R}(\boldsymbol{u}^n + \epsilon \boldsymbol{y}) - \boldsymbol{R}(\boldsymbol{u}^n)}{\epsilon} \tag{28}$$

for small enough ϵ. We see that computing $\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n = [\boldsymbol{J}(\boldsymbol{u}^n) \delta \boldsymbol{u}^{n-1} \boldsymbol{J}(\boldsymbol{u}^n) \delta \boldsymbol{u}^{n-2} \ldots \boldsymbol{J}(\boldsymbol{u}^n) \delta \boldsymbol{u}^{n-n_{\text{rb}}}]$ requires n_{rb} residual evaluations. In actual practice, we replace $\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n$ with $\boldsymbol{U}_n = [\boldsymbol{J}(\boldsymbol{u}^n) \delta \boldsymbol{u}^{n-1} \boldsymbol{J}(\boldsymbol{u}^{n-1}) \delta \boldsymbol{u}^{n-2} \ldots \boldsymbol{J}(\boldsymbol{u}^{n+1-n_{\text{rb}}}) \delta \boldsymbol{u}^{n-n_{\text{rb}}}]$. Because only the first column of $\boldsymbol{U}_n$ has to be computed, while the remaining columns were already computed and stored, only one residual evaluation is required to form $\boldsymbol{U}_n$. By using $\boldsymbol{U}_n$ in place of $\boldsymbol{J}(\boldsymbol{u}^n) \boldsymbol{W}_n$ to compute the GMRES initial guess and construct the preconditioner, we reduce the number of residual evaluations from n_{rb} to 1. Henceforth, both the GMRES initial guess and the preconditioner add little to the overall cost because its computational cost can be far smaller than the residual evaluations required during the GMRES iterations.

A goal of the JFNK approach [46] is to avoid forming the Jacobian matrix and construct an effective preconditioner to reduce the number of GMRES iterations. As discussed in great detail in [46], there is a wide variety of preconditioning techniques from incomplete LU (ILU) factorizations, multigrid methods, Schwarz-based domain decomposition methods, physics-based preconditioning, and matrix-free preconditioning methods. A number of preconditioning techniques such as ILU still form matrices that are reduced in complexity as compared to the full Jacobian. Storage and memory bandwidth limitations provide a motive for preconditioning approaches that do not require the formation of any matrix. We emphasize that the solution method proposed in the paper is completely Jacobian free because

it does not require any part of the Jacobian matrix in the solution process. The method makes use of the previously computed solutions of time-dependent systems to compute the initial guesses and construct the preconditioner. While the main ingredients of the method are not new, the application of the method for solving nonlinear semidiscrete systems is first considered in this paper.

E. Residual Calculation

As computing the residual vector is the most expensive operation in our solver, we vectorize the residual calculation to allow for efficient GPU implementation. The residual vector is assembled from the element residual vector $\boldsymbol{R}_e(\boldsymbol{u}^n)$ resulting from the element integral $(\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{q}_h), \nabla \boldsymbol{w})_{\mathcal{T}_h}$, the face residual vector $\boldsymbol{R}_f(\boldsymbol{u}^n)$ resulting from the face integral $\langle \widehat{\boldsymbol{f}}_h(\boldsymbol{u}_h, \boldsymbol{q}_h), \boldsymbol{w}\rangle_{\partial\mathcal{T}_h}$, and the mass residual vector resulting from the discretization of the time derivative. First, we compute $\boldsymbol{q}^n$, the vector of degrees of freedom of the approximate gradient $\boldsymbol{q}_h^n \in \boldsymbol{\mathcal{Q}}_h^k$, by solving the equation

$$(\boldsymbol{q}_h^n, \boldsymbol{r})_{\mathcal{T}_h} = \langle \widehat{\boldsymbol{u}}_h^n, \boldsymbol{r}\cdot\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} - (\boldsymbol{u}_h^n, \nabla\cdot\boldsymbol{r})_{\mathcal{T}_h}, \quad \forall\, \boldsymbol{r} \in \boldsymbol{\mathcal{Q}}_h^k \tag{29}$$

where $\hat{\boldsymbol{u}}_h^n$ is computed from $\boldsymbol{u}_h^n$ as described earlier. This step requires us to compute the inverse of the mass matrix and the right-hand side in Eq. (29). Note that the inverse of the mass matrix is precomputed and stored. The right-hand side in Eq. (29) is computed by using Gauss quadrature for all elements at once. Once $\boldsymbol{q}^n$ is obtained, we can compute the element residual and the face residual as described in the following.

For tensor product elements, the solution vector $\boldsymbol{u}^n$ and the residual vector $\boldsymbol{R}(\boldsymbol{u}^n)$ are stored as three-dimensional arrays of size $(k+1)^3 \times (d+2) \times n_e$. They can be viewed as matrices of $(k+1)^3$ rows and $(d+2)n_e$ columns. Let (ξ, η, η) be the coordinates on the master element $[0, 1]^3$. The element residual vector is computed by transforming the integral on physical elements to the one on the master element and using Gauss quadrature as follows:

$$\begin{aligned}\boldsymbol{R}_e(\boldsymbol{u}^n) = {} & (\boldsymbol{D}^\xi \otimes \boldsymbol{S}^\eta \otimes \boldsymbol{S}^\zeta)\boldsymbol{H}_e^\xi(\boldsymbol{u}^n, \boldsymbol{q}^n) + (\boldsymbol{S}^\xi \otimes \boldsymbol{D}^\eta \otimes \boldsymbol{S}^\zeta)\boldsymbol{H}_e^\eta(\boldsymbol{u}^n, \boldsymbol{q}^n) \\ & + (\boldsymbol{S}^\xi \otimes \boldsymbol{S}^\eta \otimes \boldsymbol{D}^\zeta)\boldsymbol{H}_e^\zeta(\boldsymbol{u}^n, \boldsymbol{q}^n)\end{aligned} \tag{30}$$

Here, $\boldsymbol{S}^\alpha, \boldsymbol{D}^\alpha, \alpha = \xi, \eta, \zeta$ are matrices of size $(k+1) \times \ell$, which store values of the shape functions and their derivatives at the Gauss quadrature points on the unit interval $[0, 1]$, respectively. Note that $\boldsymbol{S}^\xi = \boldsymbol{S}^\eta = \boldsymbol{S}^\zeta$ and $\boldsymbol{D}^\xi = \boldsymbol{D}^\eta = \boldsymbol{D}^\zeta$ because the same polynomials are used along each direction and that ℓ is the number of quadrature points on the unit interval and chosen to be equal to the number of nodes on every element, namely, $\ell = k+1$, to integrate the mass matrix exactly. Furthermore, $\boldsymbol{H}_e^\alpha, \alpha = \xi, \eta, \zeta$ are matrices of size $\ell^3 \times (d+2)n_e$ and obtained by computing the flux function at the quadrature points for all elements, as well as the mapping between physical elements and the master element. Note that the same shape functions are used to represent the approximate solution and the physical curved elements. Furthermore, we use the tensor product with sum factorization [47] to compute (30), thereby effectively reducing the computational complexity from $O((k+1)^6(d+2)n_e)$ to $O((k+1)^4(d+2)n_e)$.

The calculation of the face residual vector is similar to that of the element residual vector and thus omitted to save space. Note that the face residual vector must be assembled into the element residual vector to form the full residual vector. To avoid memory-access conflicts on GPUs, the coloring scheme [48] is used to assemble the face residual vector into the element residual vector. Specifically, faces that does not have the common degrees of freedom of $\boldsymbol{u}_h$ are marked as the same color, and the face residual on the same-coloring faces can be assembled together into the element residual at once. As the mass residual vector is the product of the mass matrix and the temporal discretization of the time derivative of the conserved variables, it is much faster to compute than the element and face residual vectors.

F. Implementation

The present discretization and solution methods have been implemented in an open-source software, Exasim [49], which generates DG codes to numerically solve a wide variety of partial differential equations (PDEs). Exasim combines high-level languages with low-level languages to allow users to define PDE models and obtain high-performance C++ codes that can run on both CPU and GPU clusters. The kernels for the discretization and solution methods are written in C++ with MPI for distributed-memory systems and CUDA/OpenMP for shared-memory processors, while the code generators and preprocessors are written in Julia, Python, and MATLAB® to produce C++ codes that handle fluxes, source terms, boundary conditions, and initial solutions for a particular PDE system. Exasim have been thoroughly validated through a large collection of examples for a wide variety of partial differential equations including the compressible Navier–Stokes equations. The underresolved direct numerical simulation of transonic buffet flows over the OAT15A airfoil at Mach 0.73 and Reynolds number 3×10^6 presented in the next section was performed using Exasim. The computed solution and source code for this problem are available on Exasim's GitHub website.[§]

III. Results and Discussions

A. Computation Description

1. Computational Mesh

The mesh used for this study is made of 1.12 million quadratic isoparametric hexahedra elements; in other words, second-order polynomials are used both to parameterize the elements geometry and to approximate the solution. Therefore, the method can be potentially third order accurate [50,51]. For a discontinuous Galerkin method, it represents a total of approximately 30 million nodes using tensor-product (27-nodes) elements. The three-dimensional mesh is obtained by extruding a 2D mesh over $0.065c$ in the z direction, by using 32 elements, in other words, 65 nodes. The spanwise domain size $0.065c$ is equivalent to the time-averaged local boundary-layer thickness at $x/c = 0.075$ in the experiment. This spanwise domain size is also sufficient to capture the separation size induced by the SWBLI (see [12]). The nodes on every element are uniformly distributed.

As shown in Fig. 1, the 2D mesh consists of both structured and unstructured grids, which are designed to resolve the boundary layers and the shock motion over the airfoil top surface. In particular, the unstructured grid is refined four times toward the wall, yielding five structured blocks. The first structured block (i.e., the most outer layer) has zero refinement. The second structured block is the first refinement, while the third (respectively, fourth and fifth) structured block is the second (respectively, third and fourth) refinement. The refinement is done by subdividing an element into two smaller elements in the streamwise direction, as shown in the bottom right part of Fig. 1. The aspect ratio of the 2D elements is being kept less than 20 on the top surface and less than 100 on the bottom surface. The unstructured grid is generated by using the Gmsh software [52] and has two separate blocks. The first unstructured block resides on top of the structured grid and extends from $x/c = 0.3$ to $x/c = 0.7$. It is carefully constructed to capture the shock motion of the buffet phenomenon. The second unstructured block occupies the remaining computational domain, which extends over $50c$ around the airfoil in the (x, y) plane; see Fig. 1.

The mesh has been designed such that $\Delta y^+ = \delta y/d_v < 1$, where δy denotes the distance from the wall to the first high-order node along the wall-normal direction and $d_v = \nu/u_\tau$ is the viscous distance based on the local friction velocity u_τ. Δx^+ and Δz^+ are similarly constructed by using nodes spacings in the streamwise and

[§]Data available online at https://github.com/exapde/Exasim [retrieved 7 September 2020].

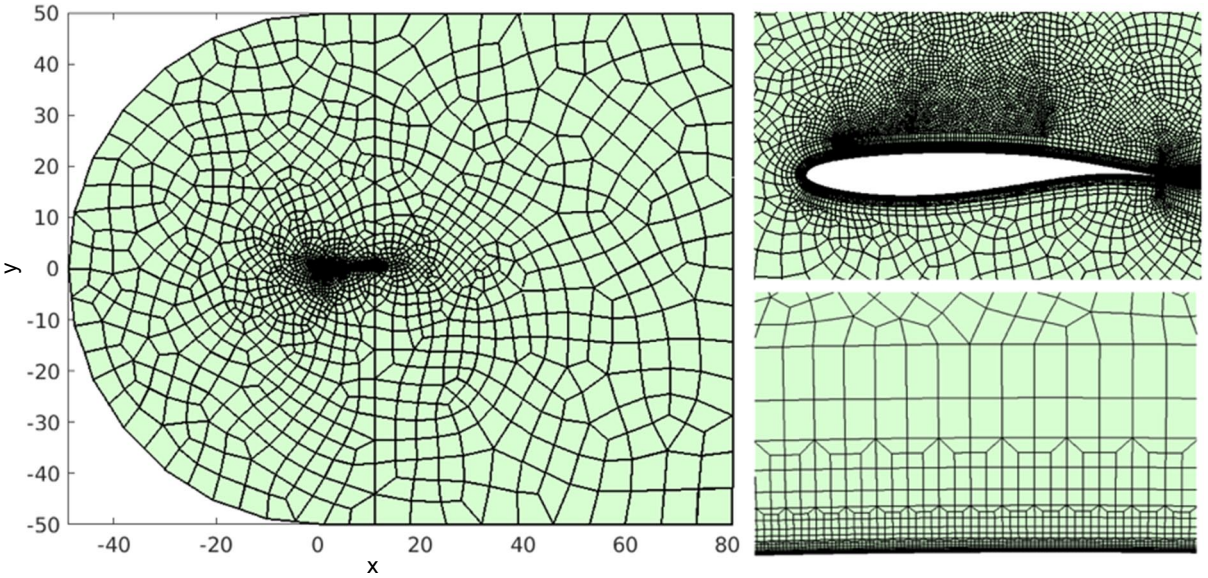


Fig. 1 Section of the mesh used for the present paper. The mesh comprises an unstructured part far from the boundary layer and a more structured part to resolve the boundary layer. The top right picture on the right shows the zoom of the mesh near the airfoil, while the bottom right picture shows the zoom of the top right picture near the wall.

spanwise directions, respectively. Note that the high-order mesh nodes are used, not the element sizes. Figure 2 shows the distributions of Δy^+, Δx^+, and Δz^+ postprocessed from our numerical results, where d_v is both spanwise averaged and time averaged over one buffet cycle. Except near the leading edge, $\Delta y^+ < 1$ on the lower side, and $\Delta y^+ < 0.5$ on the upper surface (Fig. 2, left). Moreover, as we use a stretching coefficient 1.33 for the element sizes in the wall normal direction, we ensure that typically $20 \sim 25$ high-order nodes sample the inner boundary layer $0 \leq y^+ \leq 100$, following common practice in wall-resolving LES [53]. Moreover, the average thickness of the boundary layer at $x/c = 0.28$, 0.35, 0.55 and 0.75 is sampled with, respectively, 38, 38, 54, and 58 high-order nodes, in the wall-normal direction. In the streamwise direction, $\Delta x^+ < 40$ on the upper side, which is finer than the maximum grid spacing $\Delta x^+ \approx 100$ usually recommended for WRLES ([53,54]). However, the grid spacing in the spanwise direction is too coarse to meet the wall-resolved LES requirements, in other words, $\Delta z^+ \approx 20$ according to [53,54]. Note that these requirements apply for unshocked attached TBL on flat plates, so they may actually be too permissive for the flow studied here. On the lower side, as the mesh is coarser, neither Δx^+ nor Δz^+ complies with the customary grid size criteria for WRLES.

Note that, if we had used at least twice more elements in the spanwise direction, our 30 million nodes mesh would count 60 million nodes, and it would have been somewhat comparable to the 40 million cells mesh used for the LES zone in the coupled RANS/LES simulations [15]. Interestingly, both our ILES and the RANS/LES [15] make use of meshes significantly lighter than the WMLES [12] with 440 million grid points, which seems counterintuitive because the WMLES approach is expected to save grid points. The main reason is that the WMLES uses many more points in the spanwise direction (565 for WMLES vs 280 for RANS/LES and 65 for the present ILES). The other reason is that the WMLES grid is fully structured on the whole computational domain, while only the suction side and the wake are finely meshed by using multiple structured blocks in [15]. As we use a succession of structured blocks with five levels of refinement, combined with an unstructured mesh far from the wing, our meshing strategy is more flexible, and it allows us to save many points, mainly in the streamwise direction.

2. *Transition Trips*

The mesh geometry models the transition trips as steps (see Fig. 3), with dimensions equal to the width and the height of the experimental trips. The trips are located at $x/c = 0.07$, and they are $0.013c$ wide in the chord direction. On the lower surface, the trip thickness is $3.87 \times 10^{-4}c$. On the upper surface, the trip thickness is $4.43 \times 10^{-4}c$.[¶]

We emphasize the importance of the transition trips. When performing the simulation without modeling them, we do not observe the buffet phenomena. Instead, we observe that the shock is stationary around $x/c = 0.6$ and that transition from laminar to turbulence occurs near the shock foot. Without the transition trips, the boundary layer on the top surface is stable all the way to the shock location because it is a supersonic boundary layer as the local Mach number at the edge of the boundary layer is greater than 1. The presence of the transition trips makes the flow transition from laminar to turbulence near $x/c = 0.07$, as shown in Fig. 3. Therefore, the transition trips have a very significant impact on the onset of turbulence and buffet phenomena.

Note that previous work [9–12] did not consider transition trips to trigger transition. Instead, transition was prescribed by using trip terms of the turbulence model [9–11] or modifying the eddy viscosity at the transition location [12].

3. *Computational Time*

The numerical study is performed using the DIRK(2,2) method to discretize the time derivative with a nondimensional time step $\Delta t = \Delta t^* u_\infty^*/c^* = 10^{-4}$. This time step corresponds to a CFL number of 10 because the the minimum grid spacing at the wall is $10^{-5}c$, and it ensures that the viscous time scale $\Delta t^+ = u_\tau^2 \Delta t/\nu$ is kept below 0.4 on all the airfoil, apart from a small area near the leading edge ($x/c < 0.04$). The WMLES computation of Fukushima and Kawai [12] is carried out using the third-order total variation diminishing explicit Runge–Kutta scheme with a nondimensional time-step size $\Delta t^* u_\infty^*/c^* = 1.095 \times 10^{-5}$ and the minimum grid spacing at the wall of $1.15 \times 10^{-4}c$. Therefore, the DIRK scheme allows a much larger time step than the explicit Runge–Kutta method.

The simulation is run for a total of 95 chord-based time units tu_∞/c. Starting from an extruded 2D solution, the shock oscillations start after $tu_\infty/c \approx 20$. Discarding the initial transient flow, we can record up to five full periodic oscillations of the shock, which are used to perform the statistical data presented in the following. Note that one periodic shock oscillation is defined as the time period in which the shock wave moves from the most upstream position to the downstream and reaches the most upstream position again. One

[¶]This information was obtained from private communication with L. Jacquin, P. Molton, S. Deck, B. Maury, and D. Soulevant, the authors of the experiment paper [4], July, 25 2019.

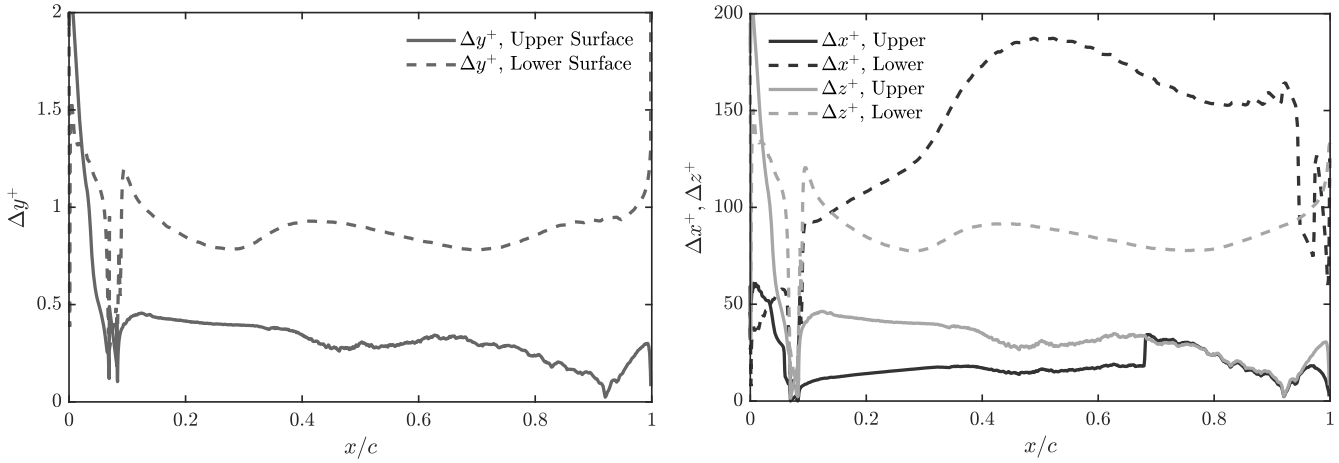


Fig. 2 Distribution of the wall-unit high-order nodes spacing in the wall-normal direction (left) and in the streamwise and spanwise directions (right).

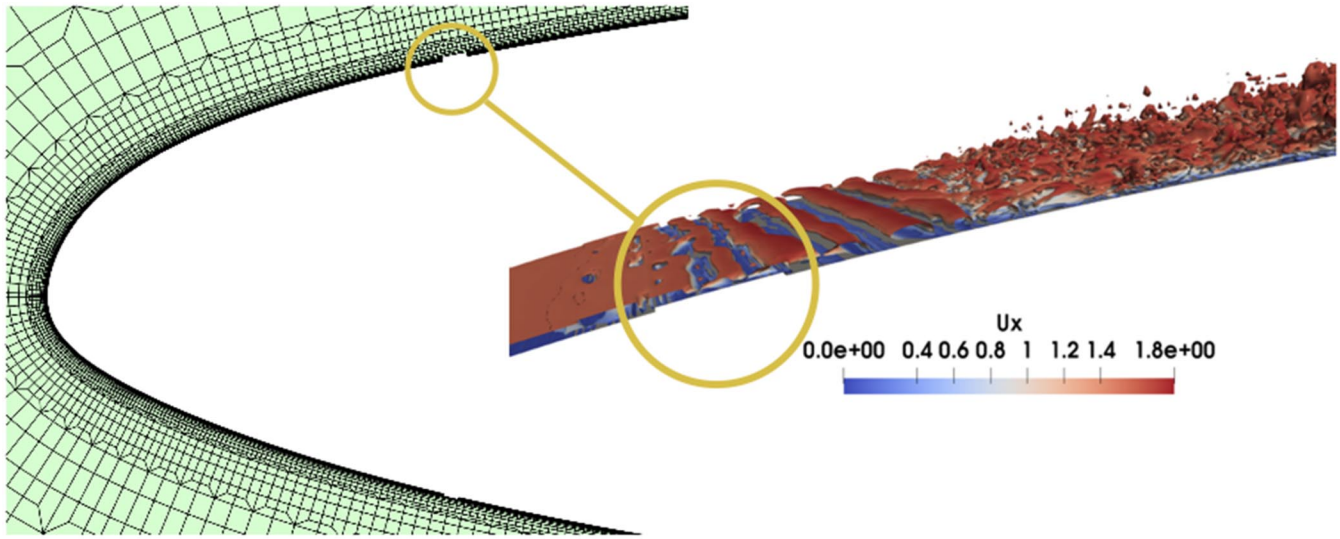


Fig. 3 Zoom of the mesh near the transition trips and the flow structure near the upper transition trip. It can be seen that the flow instability is triggered by the presence of the trip and that transition from laminar to turbulence occurs within a short distance (0.015c) downstream of the trip. The numerical transition trips were modeled by using the dimensions of the real trips used in the experiment [4].

periodic shock oscillation takes slightly less than 14 chord-based time units. The tolerance for the residual norm for the convergence of Newton method is 10^{-8}. It typically takes two Newton iterations to converge. The tolerance for GMRES is set to 10^{-3} relatively to the norm of the residual, and it takes more or less 30 GMRES iterations per linear solve. Without preconditioning, it would take about 130 GMRES iterations to converge to the same tolerance. We observe through our experiences that the number of GMRES iterations tends to scale linearly with the time-step size. The whole computation was performed using 32 NVIDIA V100 GPUs at the Barcelona Supercomputer Center for approximately 700 run-time hours.

4. *Artificial Viscosities*

The Fig. 4 shows the artificial viscosities when the shock is reaching its most upstream location. At that stage of the buffet cycle, the turbulent shear layer is thick and fully separated (see Sec. III.C.2), which triggers the maximum amount of artificial viscosities. As expected, the artificial bulk viscosity $\bar{\beta}^*$ is activated principally at the shock location (see Fig. 4, top left) and grows along the shock, as the mesh becomes coarser, away from the upper surface. To a lesser extent, the bulk viscosity sensor is also activated by the two expansion waves generated by the transition trip and in the area of the detached shear layer (Fig. 4, top right). The bottom row of Fig. 4 shows that the artificial shear viscosity added by the method is limited to the upper part of the turbulent shear layer. There, the detached shear layer enters an area where the mesh is abruptly coarsened (see Fig. 1). As the velocity gradient becomes underresolved, both the bulk and shear sensors activate the artificial viscosities. Note that they are not activated closer to the wall, in the lower part of the shear layer. They are applied neither on the bottom side of the wing nor in the TBL upstream of the shock, except in a few isolated elements. Therefore, in the present simulation, the artificial viscosities stabilize both the shock and the outer part of the shocked shear layer.

B. Flow Statistics and Comparisons to Experiment

1. *Mean and Variance of Pressure Fields*

The pressure coefficiens C_p displayed in Figs. 5 and 6 are spanwise averaged and time averaged over five buffet oscillations cycles. The pressure standard deviations P_{rms} are also spanwise averaged. Comparisons with experimental data [4] as well as numerical other numerical simulations (ZDES [9], DDES [10], LES [15], WMLES [12], and ILES [25]) are shown in Fig. 6. The pressure coefficient and fluctuations obtained by the ILES agree very well with the experimental data. The pressure coefficient on the suction side decreases rapidly over a short distance from the leading edge to the transition location $x/c = 0.07$ and flattens out from the transition location to $x/c = 0.35$. It can be seen from Fig. 5 that there is no sharp variation of the averaged pressure coefficient. A sharp pressure variation would have been the hallmark of a steady shock. Instead, the averaged pressure smoothly increases over $x/c \in [0.35, 0.55]$ due to the shock

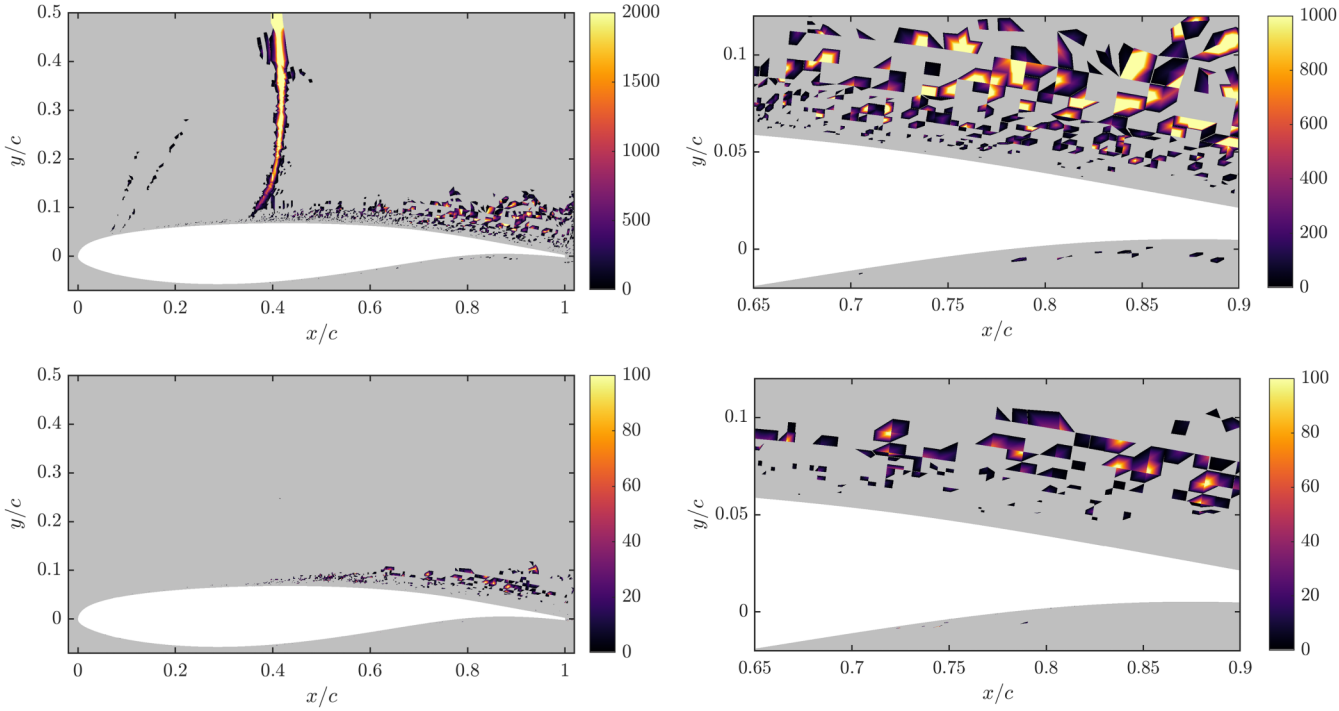


Fig. 4 2D snapshots of the instantaneous artificial viscosities at $z = 0.0325c$, when the shock reaches its most upstream location. Top row: normalized artificial bulk viscosity β^*/μ. Bottom row: normalized artificial shear viscosity μ^*/μ. Right column: zooms into the fully separated turbulent shear layer. The gray background indicates regions in which no viscosity is added. Note that these instantaneous artificial viscosities are discontinuous. They are made C^0 continuous by using a node-averaging operator described in Sec. II.B.

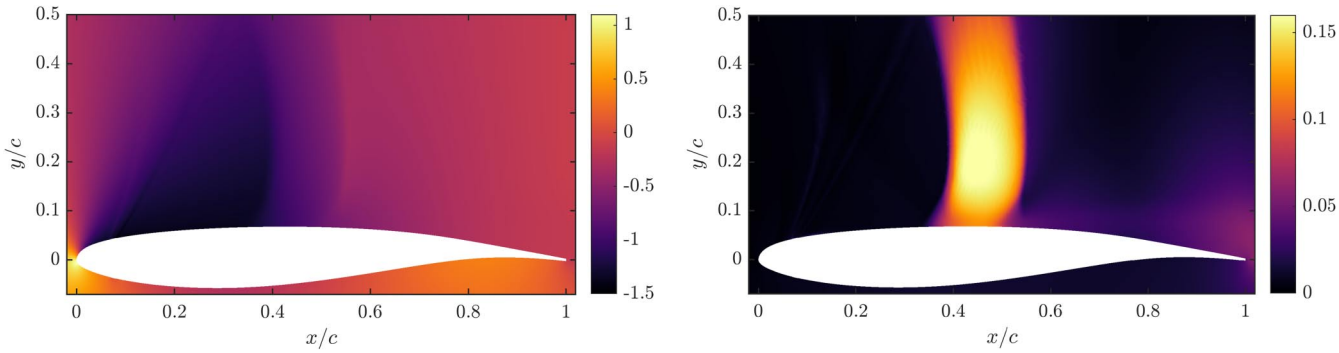


Fig. 5 Averaged pressure and its fluctuation near the airfoil for the present ILES: spanwise-and-time-averaged pressure coefficient distribution (left) and pressure fluctuation distribution (right).

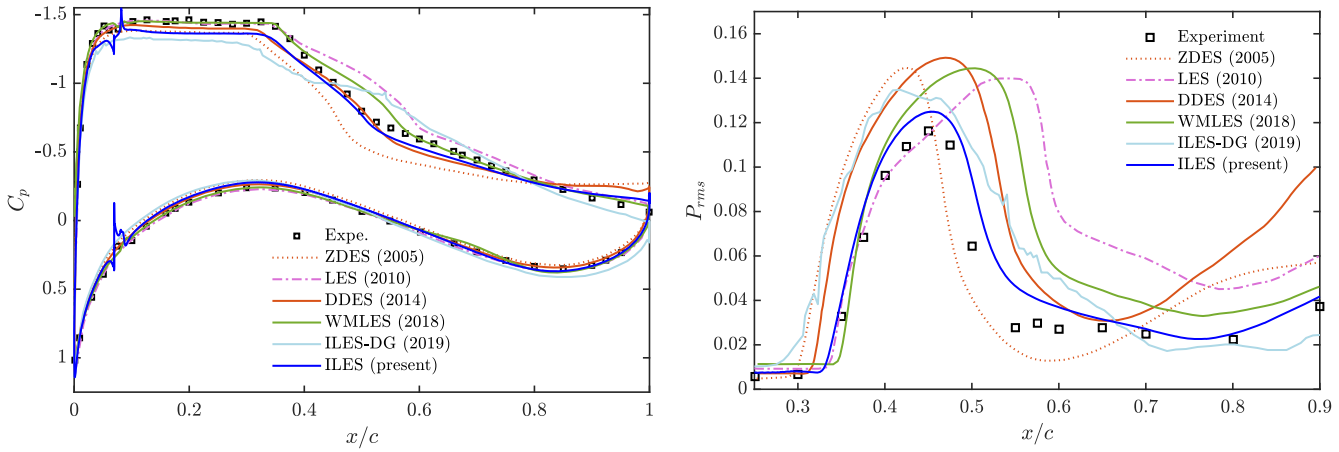


Fig. 6 Comparisons of the spanwise-and-time-averaged pressure coefficient on the airfoil surface (left) and wall pressure fluctuations on the suction surface (right). The ILES predictions agree well with the experimental data [4]. The overshoots of the pressure coefficient around $x/c = 0.07$ obtained by the ILES are due to the transition trips.

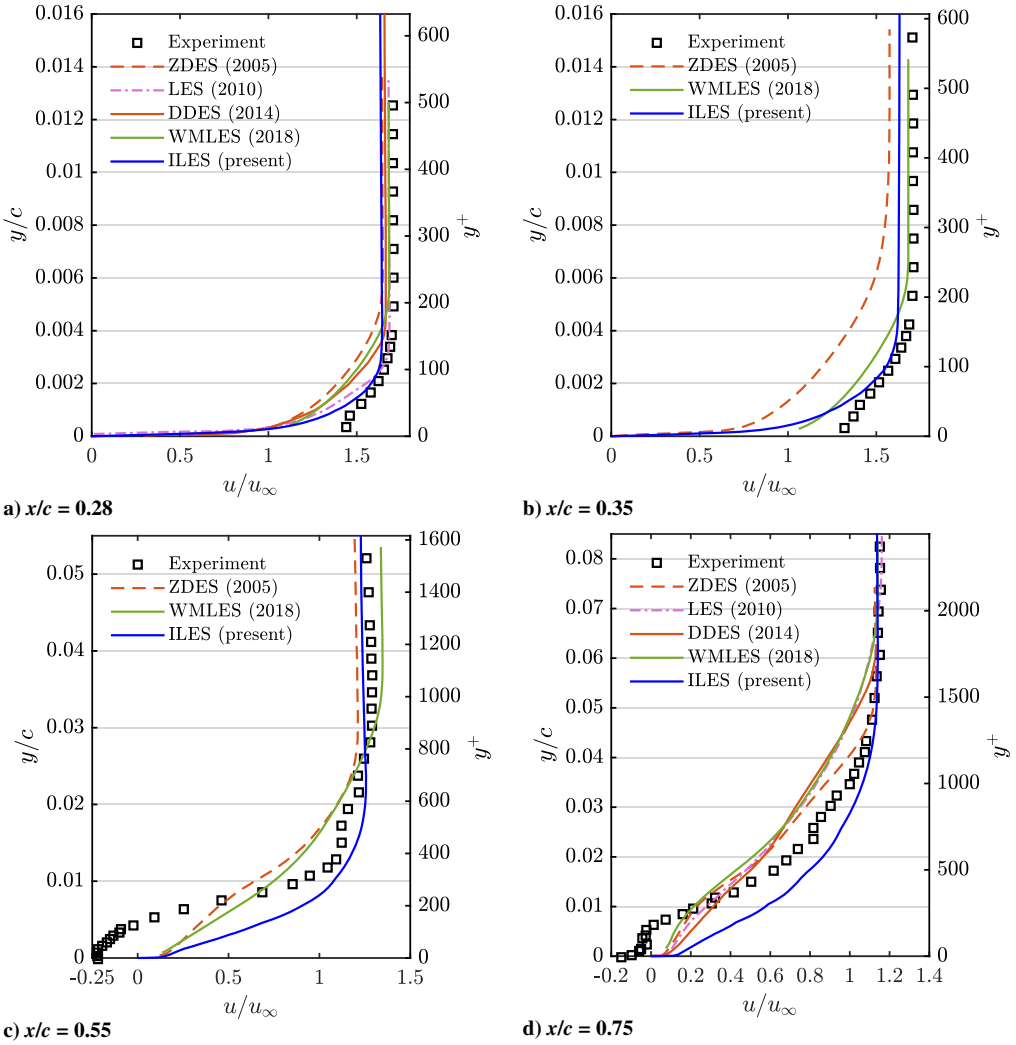


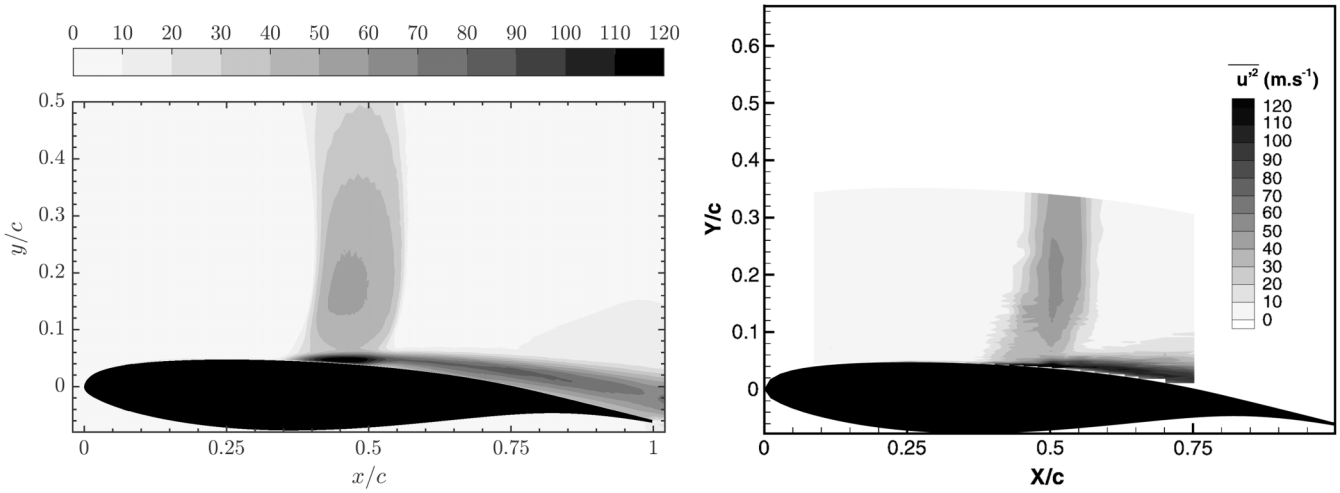
Fig. 7 Spanwise-and-time-averaged streamwise velocity profiles on the suction surface compared with experimental data and others numerical experiments. Note that *y* is the distance from the wall in the vertical direction.

motion in this range. Finally, it slowly increases over a longer distance from $x/c = 0.55$ to the trailing edge. The average location of the shock is very precisely captured, and the averaged pressure in the turbulent boundary layer is slightly overestimated upstream of the shock wave (see Fig. 6). Behind the shock wave ($x/c \geq 0.6$), the agreement is also fairly good, suggesting that the turbulent reattachment behind the shock wave is correctly predicted by the ILES computation. Note that the pressure coefficient obtained by the ILES has small overshoots around $x/c = 0.07$, while those obtained by other computations are smooth there. The overshoot is due to the transition trips in the ILES computation, whereas other computations do not employ such trips. The computed pressure fluctuations are high in the shock motion range $x/c \in [0.35, 0.55]$, with a maximum value of 0.12 at $x/c = 0.46$ (see the right parts of Figs. 5 and 6). The fluctuations tend to level off downstream of the shock area, for $x/c \in [0.6, 0.8]$, and then increase again while approaching the trailing edge, for $x/c > 0.8$.

When compared to other published numerical experiments (Fig. 6), the ILES accurately predicts the pressure distribution on the whole upper side of the wing. The DDES, LES, and WMLES can also predict a fairly accurate distribution of C_p. However, the two latter simulations tend to predict the region of the shock wave oscillation slightly downstream. The ILES can locate that region more accurately, as illustrated by the peak of pressure fluctuations in the region of $x/c \in [0.35, 0.55]$; see Fig. 6. Moreover, the amplitude of the fluctuations agrees well with the experiment and shows that the strength of the shock wave is also better predicted by the ILES approach. Thus, the pressure field statistics in the range [0.35, 0.55], which are typical of transonic buffet, are all accurately modeled by the ILES. Finally, the accurate prediction of the pressure fluctuations in the region of the attached turbulent boundary layer upstream of the shock seems to indicate that the fluctuations of the inner-layer turbulence are resolved well enough with ILES. Interestingly, the ILES in [25] makes use of a numerical setup similar to the present study (i.e., DG with quadratic hexahedral elements, artificial viscosity, and without wall model), but it predicts the pressure distribution less accurately. This is likely due to the use of a coarser mesh of 0.28 million elements vs 1.2 million for the present study.

2. *Mean and Variance of Velocity Fields*

Figure 7 displays spanwise-and-time-averaged streamwise velocity profiles at $x/c = 0.28, 0.35, 0.55$, and 0.75 for several DES- and LES-based computations in comparison with the experimental data. The wall units are based on the viscous distance d_v postprocessed from the ILES results. At the upstream of the shock wave $x/c = 0.28$ in the attached TBL region, the ILES predicts remarkably well the experimental mean bulk velocity in the log-law region $30 < y^+ < 120$, whereas the ZDES, DDES, and WMLES all underpredict the bulk velocity. The ILES slightly underpredicts the experimental data both below ($y^+ < 30$) and above ($y^+ > 150$) the log-law region. The same observations apply for $x/c = 0.35$, where the ZDES computation predicts streamwise velocities even further from the experiments. It is worth noting that the WMLES predictions of the velocity profiles are not even valid below $h_{wm} \approx 6 \times 10^{-4} c$, the



a) ILES results **b) Experimental results (taken from deck [9])**

Fig. 8 Streamwise velocity fluctuations $\sqrt{\overline{u_1^{*\prime}u_1^{*\prime}}}$ obtained by ILES and corresponding experimental results.

interface height between the LES and the wall model. Behind the shock wave at $x/c = 0.55$, the ILES predicts the bulk velocity well for $y/c > 0.02$, while the WMLES shows slightly higher mean bulk velocity than the experiments. Interestingly, none of the computations captures the average flow reversal for $y/c < 0.01$. Although the ILES predicts a reverse flow at some stage of the buffet cycle (see Sec. III.C.2), that reversal is not strong or long enough to reverse the averaged velocity profile. At the downstream location $x/c = 0.75$, the results obtained by the ILES and all the other computations show relatively good agreement with the experiments.

Figure 8 shows the dimensionalized streamwise velocity fluctuations $u_{\text{rms}} = \sqrt{\overline{u_1^{*\prime}u_1^{*\prime}}}$ over five shock buffet cycles and the experimental result [4]. We observe that the ILES prediction matches well with the experimental result. The distribution of the streamwise velocity fluctuations is quite similar to that of the pressure fluctuations. However, the velocity fluctuations are larger in the separation region near the wall, while the pressure fluctuations are larger in the buffet region above the wall. These results are also qualitatively similar to those of the WMLES [12] and the ZDES [9].

Figure 9 shows the standard deviations of the nondimensional streamwise velocity profiles u_{rms}/u_∞ on the suction surface. In the region of the fully developed attached TBL, at $x/c = 0.28$, only the WMLES results reproduce the experiments with some accuracy, while the LES overestimates the velocity fluctuations, the ILES underestimates them (i.e., the BL is not turbulent enough), and the DES-based computations predict almost no fluctuation close to the wall. At $x/c = 0.35$, both the WMLES and ILES predict the fluctuations very well, while they are severely overestimated by the ZDES computation. This is likely related to the accurate prediction of the range of the shock motion. At $x/c = 0.55$, the ILES gives good predictions near the wall $y/c \leq 0.007$ or far from it $y/c \geq 0.035$. In contrast, WMLES overestimates the velocity fluctuations inside and above the shear layer. This last observation may be explained by the WMLES prediction of the shock location slightly downstream of the experiments. Behind the SWBLI at $x/c = 0.75$, all the computations quantitatively predict the experiments reasonably well. Overall, we observe some significant differences in the standard deviations of the velocity profiles among the simulation results.

As the velocity profiles at the intermediate locations ($x/c = 0.35$ and 0.55) were not published in the LES study [15], we could not include them on Figs. 7 and 9. For the sake of completeness, it should be noted that Garnier and Deck [15] investigated three different LES configuration, and only one of them, the simulation LES-B3, is reported in the present study. Another LES configuration could compute velocity profiles in excellent agreement with the experimental data, while it also predicted inaccurate pressure distributions. Interestingly, Garnier and Deck concluded that the accurate resolution of the attached TBL does not guarantee the right prediction of the shock motion range.

Overall, we see that the LES-based results match the experimental data better than the DES-based ones, probably owing to the fact that LES/ILES resolves the turbulent boundary layer more accurately than DES. However, very close to the wall in the attached TBL, our ILES predictions of both velocity and pressure are less accurate than the other LES-based computations. We put forward a hypothesis to explain these inaccuracies. As the velocity fluctuations are significantly underpredicted in the attached TBL, some turbulent features may be underresolved due to the large Δz^+ of our mesh. The comparison with the mesh used by other authors points in the same direction: at least four time more points are used by the RANS/LES and the WMLES simulations. Moreover, as Pazner at al. [25] use a mesh even coarser in the spanwise direction and predict the wall pressures even less accurately in the attached TBL, this gives an indication of the sensibility of the DG-ILES results to the spanwise spacing. Unfortunately, because of the lack of computational resources, we were unable to verify that hypothesis.

C. Dynamics of the Transonic Buffet

1. *Instantaneous Flow Structures*

To visualize the instantaneous vortex structures of the flow, Fig. 10 displays the isosurfaces of Q, the second invariant of the velocity gradient tensor given by $Q = (1/2)(\text{tr}(\nabla \boldsymbol{u})^2 - \text{tr}(\nabla \boldsymbol{u}^2))$. On Fig. 10, the isosurfaces are colored by the streamwise velocity, and a spanwise cross-section of the Mach number field is also displayed in the background. At $x/c = 0.07$, the transition trip generates an expansion wave, visible on the Mach number plots (Figs. 10a and 10b). Two-dimensional spanwise-coherent vortices are induced by the trip and immediately break down to smaller three-dimensional structures (Fig. 10b). Quasi-streamwise vortices and irregular lambda-shape vortices are visible in the lower part of the boundary layer, where the two-dimensional vortices breaks down at $x/c \approx 0.10$. The transition happens very fast, and the boundary layer seems fully turbulent for $x/c > 0.12$. Packets of hairpinlike vortices are visible in the shock foot area, and they grow toward the downstream (Fig. 10c). As expected, the turbulent boundary layer significantly thickens after passing through the foot shock. From Fig. 10d, it is clear that during the upstream excursion of the shock the shear layer is thicker and completely separated downstream of the lambda shock, while during the downstream shock excursion, the shear layer is thinner and remains largely attached to the wall, except very locally at the trailing edge.

All these observations are consistent with the experiment [4], with the Q-criteria analysis of the WMLES [12] computation, and with the LES [15] results. Therefore, the ILES seems to resolve most of the medium-to-small-scale vortices that are found over the airfoil during the shock buffet cycle, although the smallest scales are probably

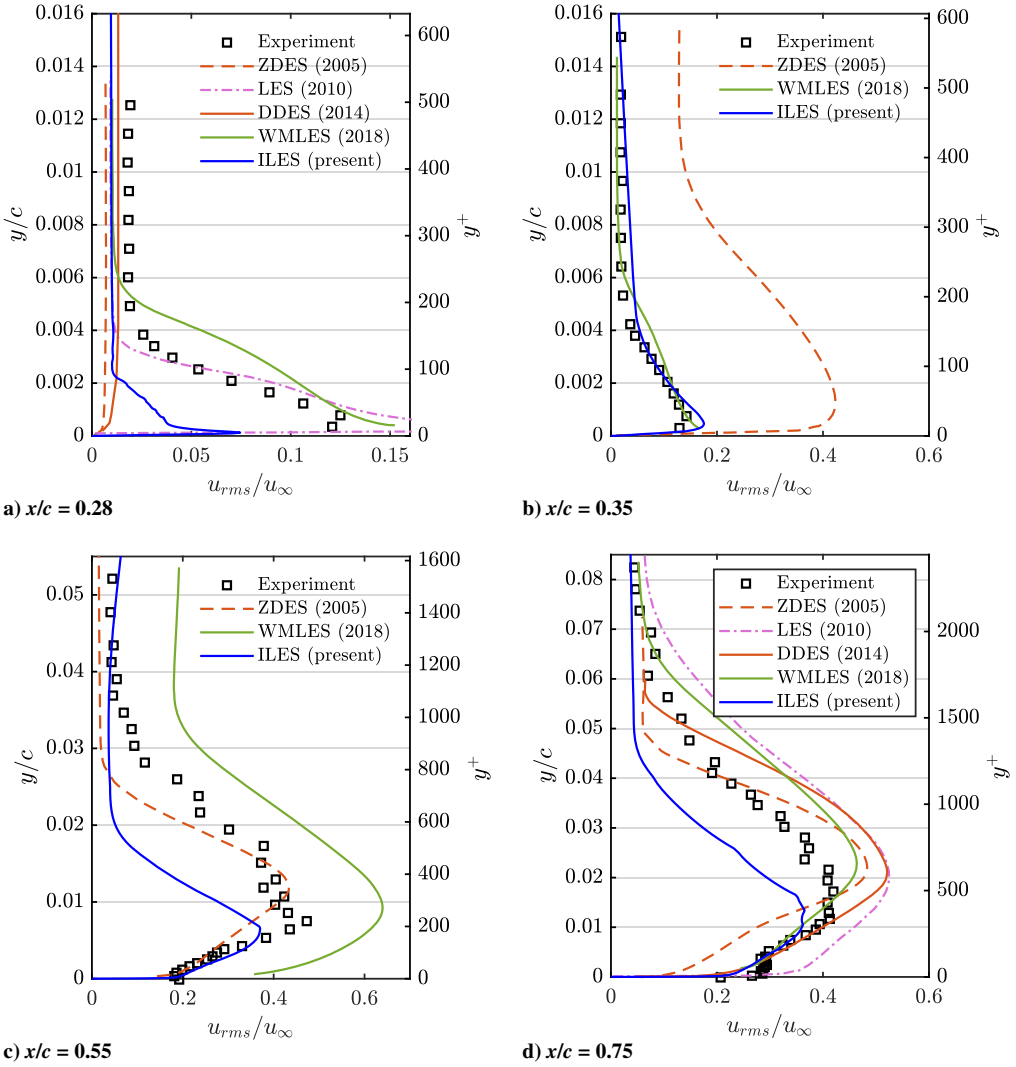


Fig. 9 Standard deviations of the streamwise velocity profiles on the suction surface compared with experimental data ans other numerical experiments. Note that y is the distance from the wall in the vertical direction.

underresolved (see Sec. III.B.2). In contrast, ZDES [9], IDDES [11], and DDES [10] computations are unable to resolve the small vortex structures, and their predicted flow remains essentially two dimensional even far downstream from the lambda shock.

2. *Dynamics of Shock-Wave/Boundary-Layer Interaction*

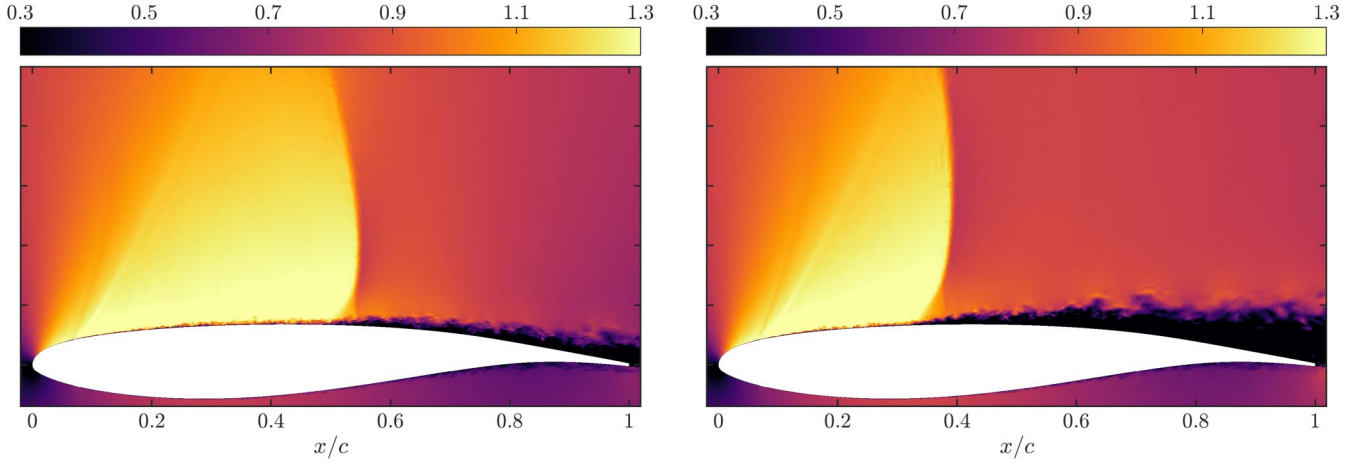
Figure 11 shows instantaneous streamwise velocity u_1 and instantaneous magnitude of the density gradient $\|\nabla\rho\|$ at different stages of the buffet cycle for the ILES simulation. In the streamwise velocity snapshots (Figs. 11a–11f), the white areas indicate the separated flow regions where $u_1 < 0$.

The lambda structure of the shock foot is clearly observable in the density gradient snapshots (Figs. 11g–11l), and the vortices in the turbulent boundary layer noticeably grow as the they pass through the shock foot. During the downstream excursion of the shock (first four rows of Fig. 11), as the effective shock Mach number is reduced, the shock is weaker, and the flow remains largely attached. The turbulent shear layer stays near the wall, as shown on the velocity snapshots (Figs. 11a–11d). Moreover, the rear shock of the lambda structure has a larger density gradient than the upstream oblique compression wave. During the upstream shock excursion (last three rows of Fig. 11), the effective Mach number is higher, and the shock is stronger, inducing a large separation of the turbulent shear layer (Figs. 11d–11f). The lambda shock foot widens, the rear shock becomes weaker, and the oblique shock is strengthened (Figs. 11j–11l). Furthermore, the shock travels upstream faster than it travels downstream, which was also predicted by [12].

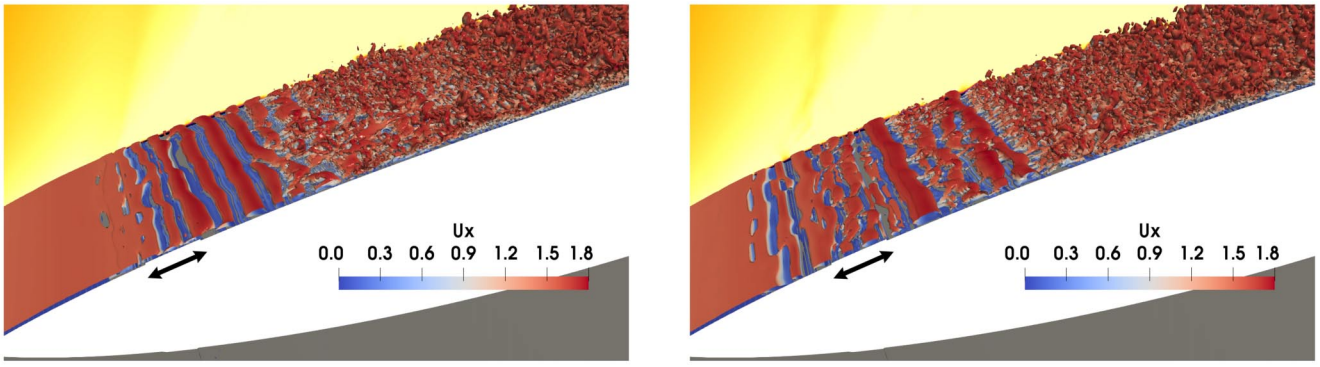
All these predictions are qualitatively similar to the experimental schlieren pictures [4] and to the WMLES results [12] and are typical of a shock buffet cycle [3]. Contrary to the WMLES, the ILES does not predict any compression wave train downstream of the shock foot. However, Jacquin et al. [4] suggest that the wave trains observed during the experiment may be due to three-dimensional effects at the side walls of the wind tunnel, which are not modeled in the present study.

3. *Power Spectral Analysis and Pressure Waves*

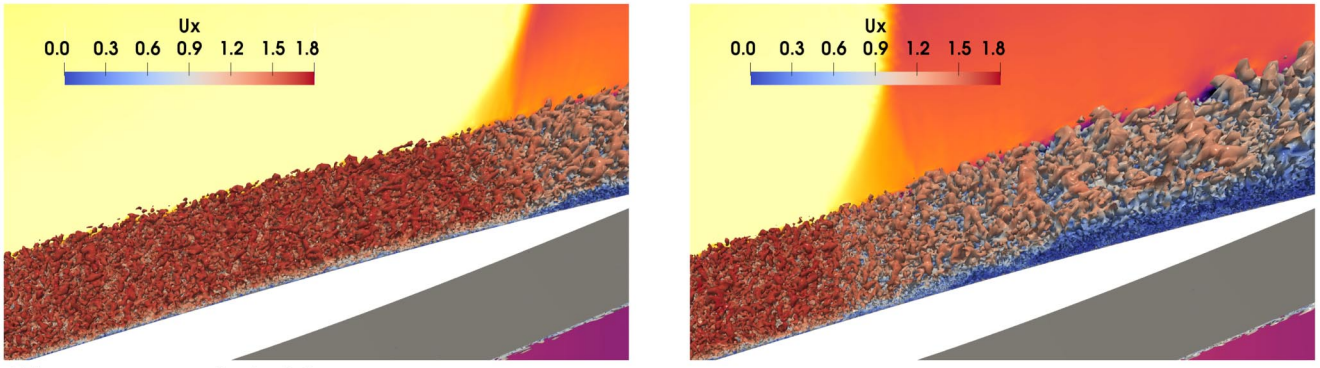
Several theories have been proposed to explain the mechanisms governing the transonic shock buffet (see the review by Giannelis et al. [3]). Some popular explanations involve an aeroacoustic feedback loop between the shock and the trailing-edge pressure perturbations. Lee [55] suggested that the fluctuations of the shock foot generate hydrodynamic instability waves propagating downstream in the turbulent shear layer. When these perturbations reach the sharp trailing edge, they generate acoustic waves traveling upstream above the shear layer. Finally, the upstream pressure waves exchange energy with the shock wave, which completes the feedback loop. Jacquin et al. [4] enriched the previous feedback mechanism considering also acoustic waves propagating along the lower surface of the wing, turning around the leading edge, and hitting the shock from upstream. Hartmann et al. [56] proposed another acoustic feedback loop, where the shock motion drives the strength of the vortices generated at the shock foot. As these vortices



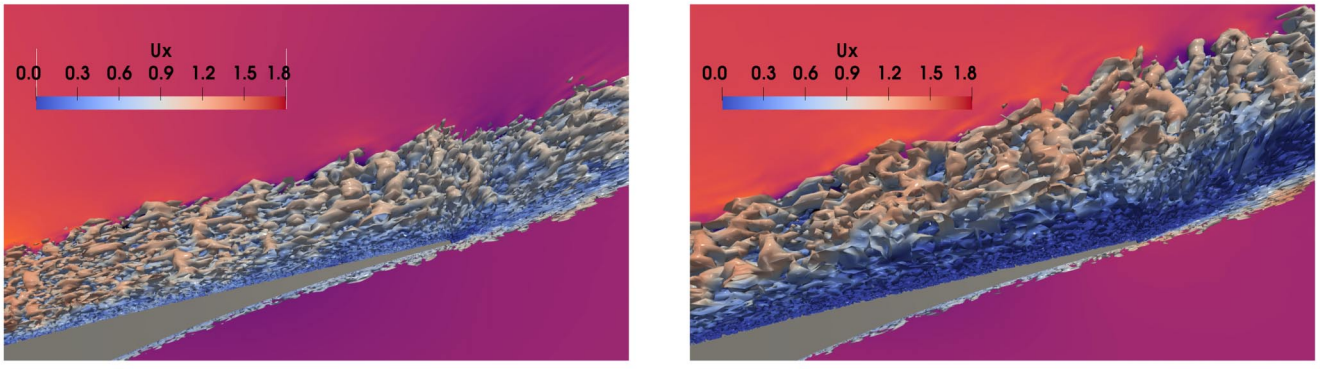
a) Side view of the flow, only the instantaneous mach number is displayed



b) Flow structure near the transition trip

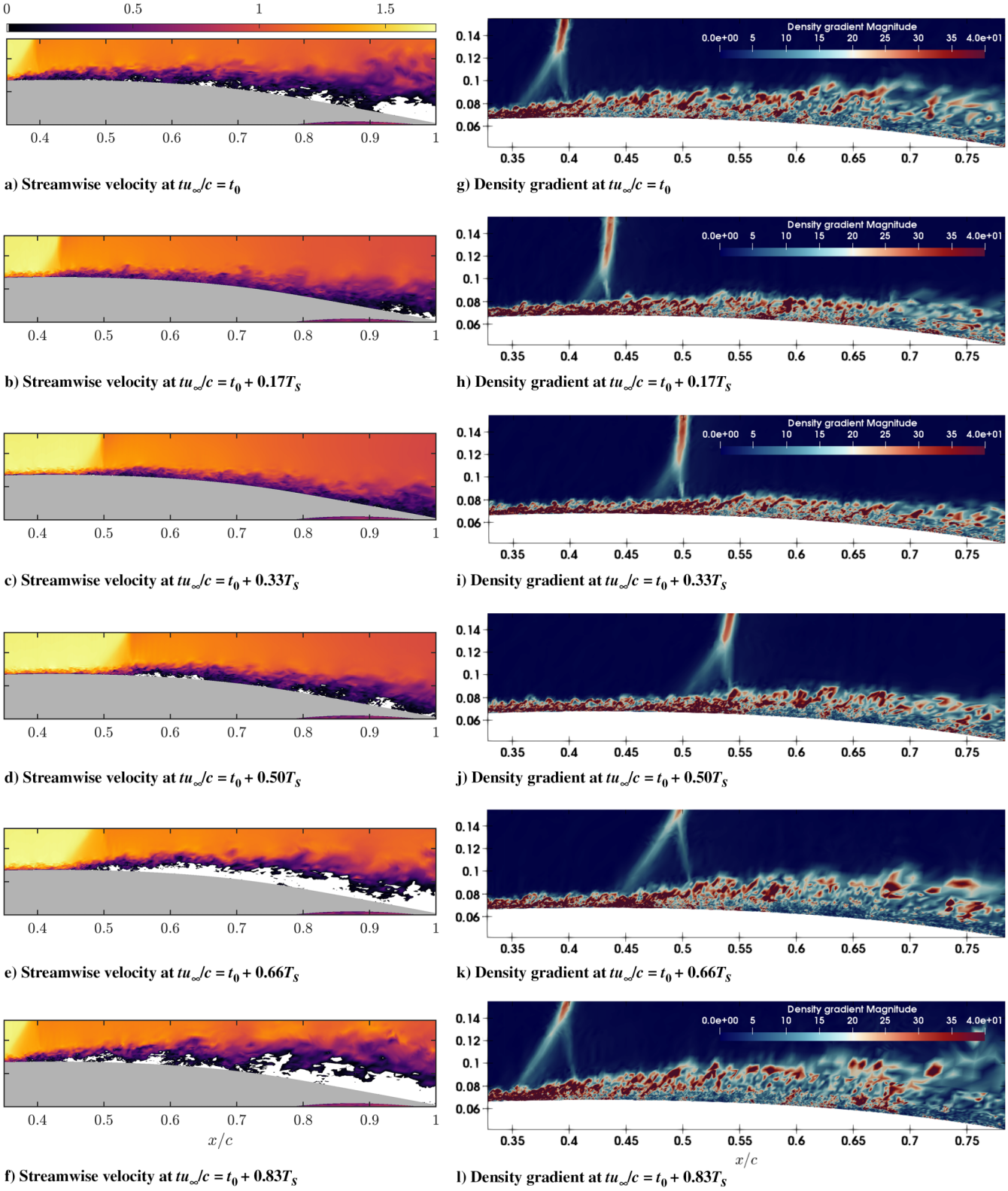


c) Flow structure near the shock foot



d) Flow structure near the trailing edge

Fig. 10 Instantaneous Mach number at $z = 0$ and isosurfaces of Q-criterion ($Q/M_\infty^2 = 200$) colored by the streamwise velocity obtained by ILES. Left column: maximal downstream shock location ($tu_\infty/c = t_0$). Right column: maximal upstream shock location ($t = t_0 + 0.5T_s$). In part b, the trip width is indicated with an arrow.



a) Streamwise velocity at $tu_\infty/c = t_0$

g) Density gradient at $tu_\infty/c = t_0$

b) Streamwise velocity at $tu_\infty/c = t_0 + 0.17T_S$

h) Density gradient at $tu_\infty/c = t_0 + 0.17T_S$

c) Streamwise velocity at $tu_\infty/c = t_0 + 0.33T_S$

i) Density gradient at $tu_\infty/c = t_0 + 0.33T_S$

d) Streamwise velocity at $tu_\infty/c = t_0 + 0.50T_S$

j) Density gradient at $tu_\infty/c = t_0 + 0.50T_S$

e) Streamwise velocity at $tu_\infty/c = t_0 + 0.66T_S$

k) Density gradient at $tu_\infty/c = t_0 + 0.66T_S$

f) Streamwise velocity at $tu_\infty/c = t_0 + 0.83T_S$

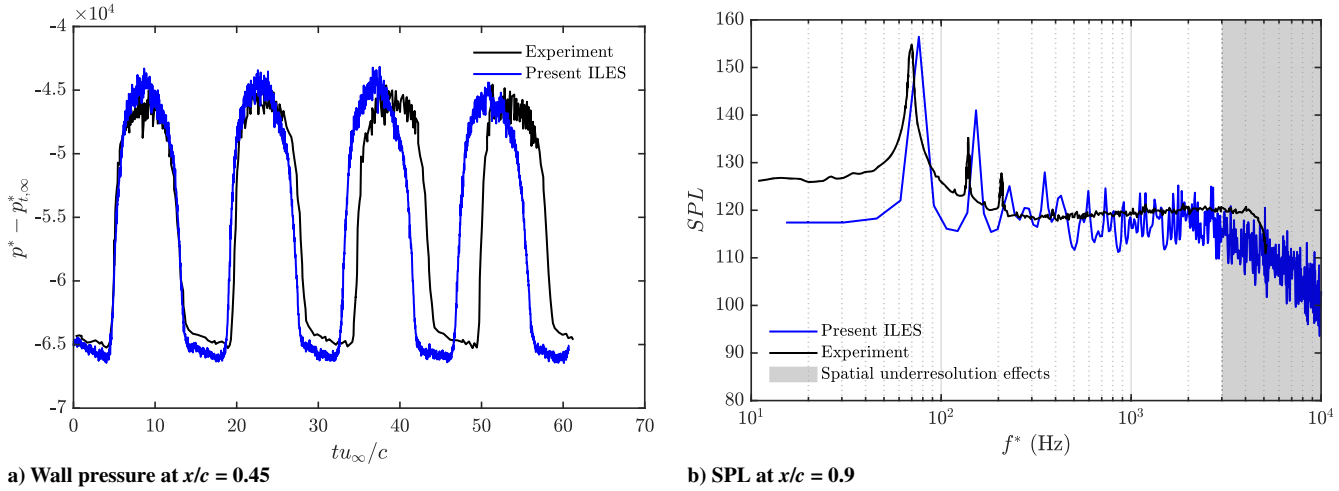
l) Density gradient at $tu_\infty/c = t_0 + 0.83T_S$

Fig. 11 Left column: instantaneous streamwise velocity u_1 at different stages of the buffet cycle. White areas indicates $u_1 < 0$. Right column: instantaneous magnitude of the density gradient $\|\nabla\rho\|$ at the same stages.

impinge on the trailing edge, they generate acoustic waves with varying sound pressure levels which propagate upstream before interacting with the shock. In this feedback loop, the shock motion is driven by the sound pressure levels at the trailing edge. The influence of the the acoustic waves on the shock dynamics was experimentally investigated by Alshabu and Olivier [57], supporting the findings of Hartmann et al.

Therefore, it seems that a numerical simulation has to accurately solve the pressure wave field in order to deliver accurate shock buffet predictions. The present subsection assesses the ILES ability to model all the pressure perturbations involved in the buffet feedback mechanism.

Figure 12a displays the time history of the wall pressure computed by the ILES at $x/c = 0.45$, compared with the experiment. The

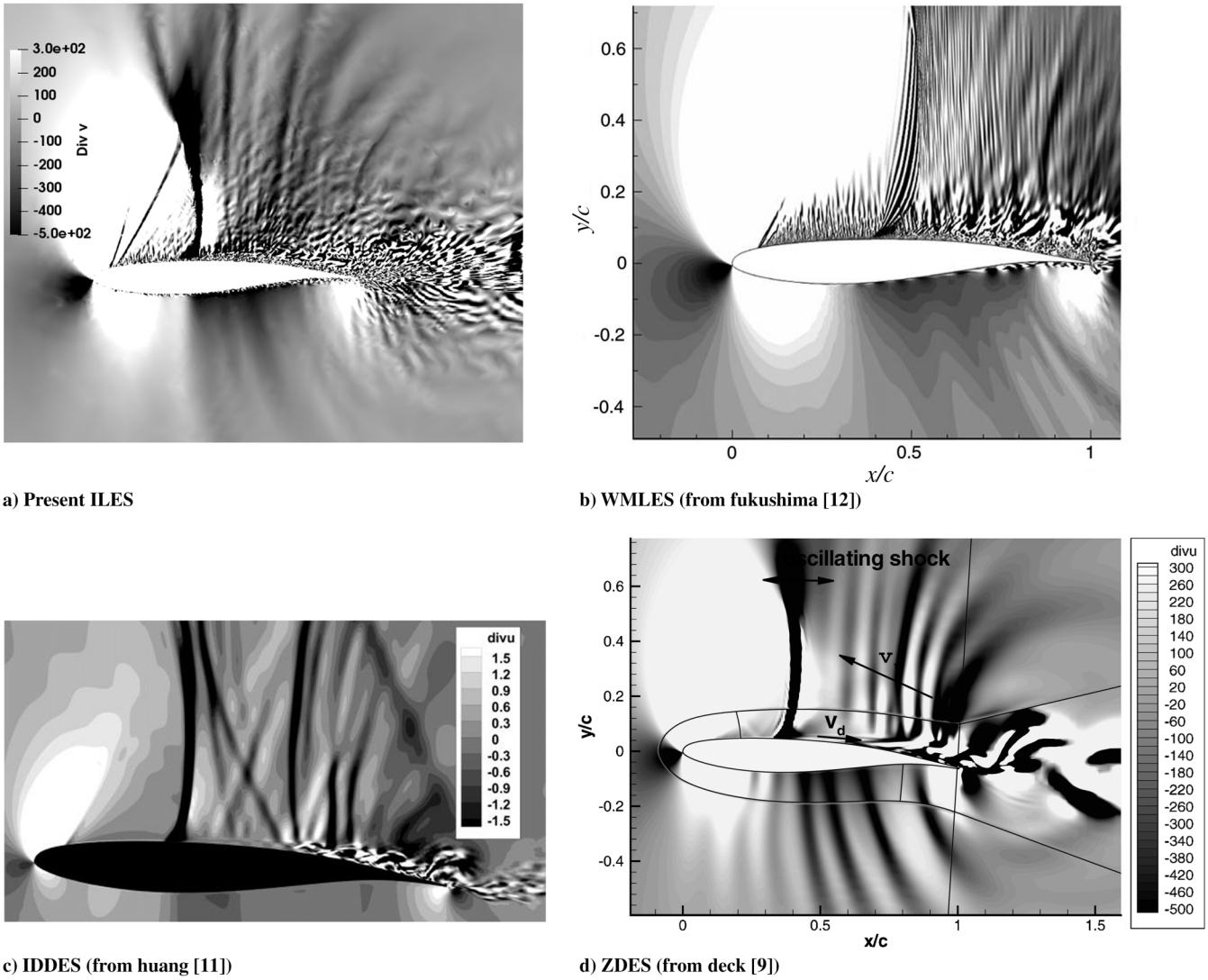


a) Wall pressure at *x/c* = 0.45 **b) SPL at *x/c* = 0.9**

Fig. 12 Time history of dimensionalized wall pressure in the shock region (left) and power spectral density of wall pressure fluctuation near the trailing edge (right).

low-frequency large-amplitude oscillations are due to the periodic shock motion. While the amplitude is correct, the buffet frequency is overpredicted. As observed in the experiment, when the shock location is downstream of the pressure sensor, there is little pressure fluctuation, and when the shock is upstream of the sensor, the pressure fluctuations are stronger. It indicates that the ILES predicts the previously mentioned pressure perturbations generated at the shock foot and propagating downstream in the TBL.

An autoregressive power spectral density (PSD) estimate of the wall pressure has been computed, via Burg's method, as advised by



a) Present ILES **b) WMLES (from fukushima [12])**

c) IDDES (from huang [11]) **d) ZDES (from deck [9])**

Fig. 13 Instantaneous dilatation fields $\nabla \cdot \boldsymbol{u}$.

Deck [9]. Figure 12b shows the PSD of the wall pressure fluctuations at $x/c = 0.9$ expressed in terms of sound pressure level (SPL), in decibels. The main peak corresponds to the main frequency of the shock periodic motion. The ILES slightly overpredicts the buffet frequency (76 Hz instead of 69 Hz for the experiment). However, such overprediction is quite common and was also observed with ZDES [15], IDDES [11], DDES [10], and to a lesser extent with the coupled RANS/LES [15]. On the other hand, both ZDES [9] and WMLES [12] could predict the correct buffet frequency. The ILES correctly captures the three first harmonics but also predicts a fourth one, which is not present in the experiment, suggesting that the ILES may overestimate the periodicity of the flow. Below 100 Hz, the lower-frequency spectrum is underpredicted, probably because the pressure signal is not long enough to compute an accurate PSD estimate in this frequency band. In the high-frequency region, the predicted PSD agrees reasonably well with the experiment. The rather flat PSD at high frequency (400 to 3000 Hz) is reproduced by the computation. As the flat high-frequency spectrum is due to the various scales of the turbulent structures, these results suggest that a large range of turbulence scales are resolved by the ILES.

Although the fine mesh in the shear layer can resolve all the acoustic frequencies of interest (up to 5000 Hz), it is no longer the case above the shear layer ($y/c > 0.1$) and downstream of the shock, where the acoustic waves travel from the trailing edge to the shock. There, the mesh size $h \lesssim 0.02c$ is larger due to the aggressive mesh coarsening. From this mesh size and from the velocity $\gtrsim 0.2u_\infty$ of the upstream propagating pressure waves, the highest resolved dimensional frequency $f^*_{\max} \approx 3 \times 10^3$ Hz can be estimated for that area. Therefore, the aeroacoustic feedback mechanism is likely to be affected by the mesh resolution for frequencies $f^* > f^*_{\max}$, which may explain the overprediction of the buffet frequency by the ILES. The affected frequency band is represented by a shaded area on Fig. 12b, and it corresponds to a drop of the predicted SPL, while the measured drop of SPL happens after 4000 Hz due to experimental low-pass filtering. Also, there is no temporal resolution effect on the acoustic predictions, as the time step is based on the turbulent time scales, which are several order of magnitudes lower than the acoustic periods of interest.

To visualize the complete pressure wave field, the instantaneous divergence fields of velocity $\nabla \cdot \boldsymbol{u}$ obtained by the present ILES, WMLES, IDDES, and ZDES are displayed in Fig. 13. All simulations predict the low-frequency upstream propagating acoustic waves generated when the shear-layer instabilities impinge on the trailing edge. Both ILES and ZDES predict acoustic waves propagating along the lower surface. However, with ILES, we cannot see these waves rounding the leading edge and interacting with the shock from upstream, as speculated by Jacquin et al. [4]. Both the ILES and WMLES also show small-scale structures and high-frequency acoustic waves, some of which may be numerical artifacts like the vertical wave train predicted by WMLES or the noisy wake predicted by ILES. Indeed, our mesh is aggressively coarsened in the immediate wake area, where the elements are very stretched. When the small turbulent structures enter this area, they become severely underresolved (see Fig. 10d) and generate high-frequency oscillations and possibly spurious pressure waves before being tamed by the artificial viscosity and the natural DG dissipation. Figure 13a shows that ILES predicts upstream pressure waves originating from both the trailing edge and from the noisy wake.

IV. Conclusions

In this paper, the authors performed an underresolved direct numerical simulation of transonic buffet phenomena over the OAT15A supercritical airfoil using the implicit discontinuous Galerkin method. They propose a Jacobian-free Newton/GMRES method to solve the nonlinear systems of the implicit DG discretization by forming the reduced basis to compute good initial guesses and construct a matrix-free preconditioner. It makes the implementation of the implicit DG method efficient on GPUs because only the residual vectors are calculated. As a result, one can afford to resolve the boundary layers at high Reynolds numbers to capture flow transition and onset of turbulence. In the authors' simulation, the flow transition was triggered by using the strips that have the same dimensions as the real strips used in the experiment. Making use of neither subgrid scale models nor wall models, the ILES method successfully predicts the buffet onset, the buffet frequency, and turbulence statistics. Various turbulence phenomena are predicted and demonstrated, such as periodic low-frequency oscillations of shock wave in the streamwise direction, strong shear layer detached from the shock wave due to shock-wave/boundary-layer interaction and small-scale structures broken down by the shear layer instability in the transition region, and shock-induced flow separation. The pressure coefficient, the rms of the fluctuating pressure, and the streamwise range of the shock wave oscillation agree well with experimental data. The ILES computation represents a different approach to this problem, as the previous computations use a turbulence model or a subgrid scale model.

Acknowledgments

The authors gratefully acknowledge NASA (under grant number NNX16AP15A) and the Air Force Office of Scientific Research (under grant number FA9550-16-1-0214) for supporting this work. Also, the authors thank the Barcelona Supercomputing Center, which has provided access to its graphics processor unit cluster. The authors appreciate the reviewers for their constructive comments that help improve the paper.

References

[1] Hilton, W., and Fowler, R., *Photographs of Shock Wave Movement*, National Physical Lab., London, U.K., 1947.

[2] Lee, B., "Self-Sustained Shock Oscillations on Airfoils at Transonic Speeds," *Progress in Aerospace Sciences*, Vol. 37, No. 2, 2001, pp. 147–196.

[3] Giannelis, N. F., Vio, G. A., and Levinski, O., "A Review of Recent Developments in the Understanding of Transonic Shock Buffet," *Progress in Aerospace Sciences*, Vol. 92, 2017, pp. 39–84. https://doi.org/10.1016/j.paerosci.2017.05.004

[4] Jacquin, L., Molton, P., Deck, S., Maury, B., and Soulevant, D., "Experimental Study of Shock Oscillation over a Transonic Supercritical Profile," *AIAA Journal*, Vol. 47, No. 9, 2009, pp. 1985–1994. https://doi.org/10.2514/1.30190

[5] Thiery, M., and Coustols, E., "Numerical Prediction of Shock Induced Oscillations over a 2D Airfoil: Influence of Turbulence Modelling and Test Section Walls," *International Journal of Heat and Fluid Flow*, Vol. 27, No. 4, 2006, pp. 661–670. https://doi.org/10.1016/j.ijheatfluidflow.2006.02.013

[6] Menter, F. R., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1598–1605.

[7] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale*, Vol. 1, Grenoble, France, 1994, pp. 5–21. https://doi.org/10.2514/6.1992-439

[8] Daris, T., and Bezard, H., "Four-Equation Models for Reynolds Stress and Turbulent Heat Flux Predictions," *International Heat Transfer Conference Digital Library*, Begel House, 2002.

[9] Deck, S., "Numerical Simulation of Transonic Buffet over a Supercritical Airfoil," *AIAA Journal*, Vol. 43, No. 7, 2005, pp. 1556–1566. https://doi.org/10.2514/1.9885

[10] Grossi, F., Braza, M., and Hoarau, Y., "Prediction of Transonic Buffet by Delayed Detached-Eddy Simulation," *AIAA Journal*, Vol. 52, No. 10, 2014, pp. 2300–2312. https://doi.org/10.2514/1.J052873

[11] Huang, J., Xiao, Z., Liu, J., and Fu, S., "Numerical Investigation of Shock Buffet on an OAT15A Airfoil and Control Effects of Vortex Generators," *50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0044, 2012. https://doi.org/10.2514/6.2012-44

[12] Fukushima, Y., and Kawai, S., "Wall-Modeled Large-Eddy Simulation of Transonic Airfoil Buffet at High Reynolds Number," *AIAA Journal*, Vol. 56, No. 6, 2018, pp. 2372–2388. 10.2514/1.J056537

[13] Lenormand, E., Sagaut, P., and Ta Phuoc, L., "Large Eddy Simulation of Subsonic and Supersonic Channel Flow at Moderate Reynolds Number," *International Journal for Numerical Methods in Fluids*, Vol. 32, No. 4, 2000, pp. 369–406.

[14] Piomelli, U., "Wall-Modeled Large-Eddy Simulations: Present Status and Prospects," *Direct and Large-Eddy Simulation VII*, Springer-Verlag, 2010, pp. 1–10.

[15] Garnier, E., and Deck, S., *Large-Eddy Simulation of Transonic Buffet over a Supercritical Airfoil*, Springer-Verlag, Heidelberg, Germany, 2010, pp. 135–141.

[16] Fernandez, P., Nguyen, N. C., and Peraire, J., "The Hybridized Discontinuous Galerkin Method for Implicit Large-Eddy Simulation of Transitional Turbulent Flows," *Journal of Computational Physics*, Vol. 336, 2017, pp. 308–329. https://doi.org/10.1016/j.jcp.2017.02.015.

[17] Beck, A. D., Bolemann, T., Flad, D., Frank, H., Gassner, G. J., Hindenlang, F., and Munz, C.-D., "High-Order Discontinuous Galerkin Spectral Element Methods for Transitional and Turbulent Flow Simulations," *International Journal for Numerical Methods in Fluids*, Vol. 76, No. 8, 2014, pp. 522–548. https://doi.org/10.1002/fld.3943

[18] Frere, A., Hillewaert, K., Chivaee, H. S., Mikkelsen, R. F., and Chatelain, P., "Cross-Validation of Numerical and Experimental Studies of Transitional Airfoil Performance," *33rd Wind Energy Symposium, AIAA SciTech Forum*, AIAA, Reston, VA, 2015. https://doi.org/10.2514/6.2015-0499

[19] Gassner, G. J., and Beck, A. D., "On the Accuracy of High-Order Discretizations for Underresolved Turbulence Simulations," *Theoretical and Computational Fluid Dynamics*, Vol. 27, 2013, pp. 221–237. https://doi.org/10.1007/s00162-011-0253-7

[20] Murman, S. M., Diosady, L., Garai, A., and Ceze, M., "A Space-Time Discontinuous-Galerkin Approach for Separated Flows," *54th AIAA Aerospace Sciences Meeting*, AIAA Paper 2016-1059, 2016. https://doi.org/10.2514/6.2016-1059

[21] Renac, F., de la Llave Plata, M., Martin, E., Chapelier, J. B., and Couaillier, V., *Aghora: A High-Order DG Solver for Turbulent Flow Simulations*, Springer International Publishing, Cham, Switzerland, 2015, pp. 315–335. https://doi.org/10.1007/978-3-319-12886-3_15

[22] Uranga, A., Persson, P.-O., Drela, M., and Peraire, J., "Implicit Large Eddy Simulation of Transition to Turbulence at Low Reynolds Numbers Using a Discontinuous Galerkin Method," *International Journal for Numerical Methods in Engineering*, Vol. 87, No. 1-5, 2011, pp. 232–261. https://doi.org/10.1002/nme.3036

[23] de Wiart, C. C., and Hillewaert, K., *Development and Validation of a Massively Parallel High-Order Solver for DNS and LES of Industrial Flows*, Springer International Publishing, Cham, 2015, pp. 251–292. https://doi.org/10.1007/978-3-319-12886-3_13

[24] Fernandez, P., Moura, R. C., Mengaldo, G., and Peraire, J., "Non-Modal Analysis of Spectral Element Methods: Towards Accurate and Robust Large-Eddy Simulations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 346, 2019, pp. 43–62. https://doi.org/10.1016/J.CMA.2018.11.027

[25] Pazner, W., Franco, M., and Persson, P.-O., "High-Order Wall-Resolved Large Eddy Simulation of Transonic Buffet on the OAT15A Airfoil," *AIAA Scitech 2019 Forum*, AIAA, Reston, VA, 2019. https://doi.org/10.2514/6.2019-1152

[26] Moro, D., Nguyen, N. C., and Peraire, J., "Dilation-Based Shock Capturing for High-Order Methods," *International Journal for Numerical Methods in Fluids*, Vol. 82, No. 7, 2016, pp. 398–416. https://doi.org/10.1002/fld.4223

[27] Fernandez, P., Nguyen, N. C., and Peraire, J., "A Physics-Based Shock Capturing Method for Unsteady Laminar and Turbulent Flows," *56th AIAA Aerospace Sciences Meeting*, AIAA Paper 2018-0062, 2018.

[28] Ducros, F., Ferrand, V., Nicoud, F., Weber, C., Darracq, D., Gacherieu, C., and Poinsot, T., "Large-Eddy Simulation of the Shock/Turbulence Interaction," *Journal of Computational Physics*, Vol. 152, No. 2, 1999, pp. 517–549.

[29] Nguyen, N. C., and Peraire, J., "Hybridizable Discontinuous Galerkin Methods for Partial Differential Equations in Continuum Mechanics," *Journal of Computational Physics*, Vol. 231, No. 18, 2012, pp. 5955–5988. https://doi.org/10.1016/j.jcp.2012.02.033

[30] Cockburn, B., and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1998, pp. 2440–2463. https://doi.org/10.1137/S0036142997316712

[31] Bassi, F., Crivellini, A., Rebay, S., and Savini, M., "Discontinuous Galerkin Solution of the Reynolds-Averaged Navier-Stokes and k-ω Turbulence Model Equations," *Computers & Fluids*, Vol. 34, No. 4, 2005, pp. 507–540.

[32] Bassi, F., and Rebay, S., "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations," *Journal of Computational Physics*, Vol. 131, No. 2, 1997, pp. 267–279.

[33] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems," *SIAM Journal on Numerical Analysis*, Vol. 39, No. 5, 2002, pp. 1749–1779.

[34] Vila-Pérez, J., Giacomini, M., Sevilla, R., and Huerta, A., "Hybridisable Discontinuous Galerkin Formulation of Compressible Flows," *Archives of Computational Methods in Engineering*, Vol. 28, No. 2, 2021, pp. 753–784. https://doi.org/10.1007/s11831-020-09508-z

[35] Terrana, S., Nguyen, N. C., Bonet, J., and Peraire, J., "A Hybridizable Discontinuous Galerkin Method for Both Thin and 3D Nonlinear Elastic Structures," *Computer Methods in Applied Mechanics and Engineering*, Vol. 352, 2019, pp. 561–585. https://doi.org/10.1016/j.cma.2019.04.029

[36] Alexander, R., "Diagonally Implicit Runge-Kutta Methods for stiff ODEs," *SIAM Journal on Numerical Analysis*, Vol. 14, 1977, pp. 1006–1021.

[37] Georgiadis, N. J., Rizzetta, D. P., and Fureby, C., "Large-Eddy Simulation: Current Capabilities, Recommended Practices, and Future Research," *AIAA Journal*, Vol. 48, No. 8, 2010, pp. 1772–1784.

[38] Marquardt, D. W., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *SIAM*, Vol. 11, No. 2, 1963, pp. 431–441.

[39] Grepl, M. A., Maday, Y., Nguyen, N. C., and Patera, A. T., "Efficient Reduced-Basis Treatment of Nonaffine and Nonlinear Partial Differential Equations," *ESAIM: Mathematical Modelling and Numerical Analysis*, Vol. 41, No. 3, 2007, pp. 575–605. https://doi.org/10.1051/m2an:2007031

[40] Rozza, G., Huynh, D. B. P., and Patera, A. T., "Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations: Application to Transport and Continuum Mechanics," *Archives Computational Methods in Engineering*, Vol. 15, No. 4, 2008, pp. 229–275.

[41] Nguyen, N. C., "A Multiscale Reduced-Basis Method for Parametrized Elliptic Partial Differential Equations with Multiple Scales," *Journal of Computational Physics*, Vol. 227, No. 23, 2008, pp. 9807–9822.

[42] Broyden, C. G., "The Convergence of a Class of Double-Rank Minimization Algorithms 1. General Considerations," *IMA Journal of Applied Mathematics*, Vol. 6, No. 1, 1970, pp. 76–90. https://doi.org/10.1093/imamat/6.1.76

[43] Fletcher, R., "A New Approach to Variable Metric Algorithms," *Computer Journal*, Vol. 13, No. 3, 1970, pp. 317–322. https://doi.org/10.1093/comjnl/13.3.317

[44] Goldfarb, D., "A Family of Variable-Metric Methods Derived by Variational Means," *Mathematics of Computation*, Vol. 24, No. 109, 1970, pp. 23–23. https://doi.org/10.1090/S0025-5718-1970-0258249-6

[45] Shanno, D. F., "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computation*, Vol. 24, No. 111, 1970, pp. 647–647. https://doi.org/10.1090/S0025-5718-1970-0274029-X

[46] Knoll, D. A., and Keyes, D. E., "Jacobian-Free Newton-Krylov Methods: A Survey of Approaches and Applications," *Journal of Computational Physics*, Vol. 193, 2004, pp. 357–397. https://doi.org/10.1016/j.jcp.2003.08.010

[47] Pazner, W., and Persson, P. O., "Approximate Tensor-Product Preconditioners for Very High Order Discontinuous Galerkin Methods," *Journal of Computational Physics*, Vol. 354, 2018, pp. 344–369. https://doi.org/10.1016/j.jcp.2017.10.030

[48] Cecka, C., Lew, A. J., and Darve, E., "Assembly of Finite Element Methods on Graphics Processors," *International Journal for Numerical Methods in Engineering*, Vol. 85, No. 5, 2011, pp. 640–669. https://doi.org/10.1002/nme.2989

[49] Nguyen, N. C., and Peraire, J., "EXASIM—Generating Discontinuous Galerkin Codes For Extreme Scalable Simulations," 2020.

[50] Peraire, J., Nguyen, N., and Cockburn, B., "A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2010-0363, 2010.

[51] Nguyen, N. C., Peraire, J., and Cockburn, B., "An Implicit High-Order Hybridizable Discontinuous Galerkin Method for the Incompressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 230, No. 4, 2011, pp. 1147–1170.

[52] Geuzaine, C., and Remacle, J. F., "Gmsh: A 3-D Finite Element Mesh Generator with Built-In Pre- and Post-Processing Facilities,"

International Journal for Numerical Methods in Engineering, Vol. 79, No. 11, 2009, pp. 1309–1331.
https://doi.org/10.1002/nme.2579

[53] Choi, H., and Moin, P., "Grid-Point Requirements for Large Eddy Simulation: Chapman-Estimates Revisited," *Physics of Fluids*, Vol. 24, No. 1, 2012, Paper 011702.

[54] Chapman, D. R., "Computational Aerodynamics Development and Outlook," *AIAA Journal*, Vol. 17, No. 12, 1979, pp. 1293–1313.

[55] Lee, B., "Oscillatory Shock Motion Caused by Transonic Shock Boundary-Layer Interaction," *AIAA Journal*, Vol. 28, No. 5, 1990, pp. 942–944.

[56] Hartmann, A., Feldhusen, A., and Schröder, W., "On the Interaction of Shock Waves and Sound Waves in Transonic Buffet Flow," *Physics of Fluids*, Vol. 25, No. 2, 2013, Paper 026101.

[57] Alshabu, A., and Olivier, H., "Unsteady Wave Phenomena on a Supercritical Airfoil," *AIAA Journal*, Vol. 46, No. 8, 2008, pp. 2066–2073.

J. Larsson
Associate Editor