Simulation of turbulent flows using a fully discrete explicit \( hp \)-nonconforming entropy stable solver of any order on unstructured grids

Matteo Parsani *, Radouan. Boukharfane †, Irving E. Reyna Nolasco ‡, Lisandro Dalcin §, David E. Keyes ¶
King Abdullah University of Science and Technology (KAUST), Computer Electrical and Mathematical Science and Engineering Division (CEMSE), Extreme Computing Research Center (ECRC), 23955-6900, Thuwal, Saudi Arabia

We report the numerical solution of two challenging turbulent flow test cases simulated with the SSDC framework, a compressible, fully discrete \( hp \)-nonconforming entropy stable solver based on the summation-by-parts discontinuous collocation Galerkin discretizations and the relaxation Runge—Kutta methods. The algorithms at the core of the solver are systematically designed with mimetic and structure-preserving techniques that transfer fundamental properties from the continuous level to the discrete one. We aim at providing numerical evidence of the robustness and maturity of these entropy stable scale-resolving methods for the new generation of adaptive unstructured computational fluid dynamics tools. The two selected turbulent flows are i) the flow past two spheres in tandem at a Reynolds number based on the sphere diameter of \( \text{Re}_D = 3.9 \times 10^3 \) and \( 10^4 \), and a Mach number of \( \text{Ma}_\infty = 0.1 \), and ii) the NASA junction flow experiment at a Reynolds number based on the crank chord length of \( \text{Re}_\ell = 2.4 \times 10^6 \) and \( \text{Ma}_\infty = 0.189 \).

I. Introduction

High fidelity simulation methodologies must be developed to account for more ambitious physical phenomena being modeled and the emerging heterogeneous data-centric computing hardware [1, 2]. The aerospace industry represents a very tangible example of this transformational need. The ability to simulate aerodynamic flows using computational fluid dynamics (CFD) has progressed rapidly during the past four decades and has fundamentally changed the aerospace and aeronautics design process [3]. In particular, numerous reports have emphasized the fact that advances in computational algorithms have contributed equally or more compared to advances in high-performance computing hardware over the same time frame [4, 5].

In the exascale era and beyond, numerical methods for flow simulations will rely on efficient, very compact adaptive high-order formulations for unstructured grids. Among modern, unstructured high order methods, we can mention...
discontinuous Galerkin, spectral difference, flux reconstruction methods, and residual distribution, which can produce highly accurate solutions with minimum numerical dispersion and dissipation. Although these methods are well suited for smooth solutions, numerical instabilities may occur if the flow contains under-resolved physical features (e.g., under-resolved turbulent flows) or discontinuities (e.g., shocks); see, for instance, [6–8], and the discussion in [9]. Thus, the use of most high-order accurate methods for complex flow applications is problematic, and most commercial and industrial software rely on robust nominally second-order discretizations, which are sub-optimal on current and emerging supercomputer architectures [4].

The NASA-sponsored CFD Vision 2030 study presented a bold vision for physics-based computational capabilities when it was released in 2014 [4]. In the CFD vision 2030 technology development roadmap, adaptive “Production scalable entropy stable solvers” are a key milestone that should be in part of the production workflows by year 2030. The use of entropy stable discretizations gives a high level of robustness and offers the possibility of carefully controlling the dissipation of the smallest resolved scales using physically derived subgrid-scale models rather than through excessive numerical dissipation as in the case of DG-type discretizations which use standard numerical flux functions.

Over the last decade, there have been rapid developments in spatially entropy stable high order methods, which are crucial for the construction of provably high order nonlinearly stable robust solvers for the compressible Euler and Navier–Stokes equations; see, for instance, [10–23]. High order entropy stable schemes are often based on the well known matrix-vector nodal formulation collocated at quadrature points; see, for instance, [24] for a clear introduction of nodal discontinuous Galerkin methods. Because of the approximation error induced by quadrature, we no longer have the integration by parts property and the chain rule in all cases. However, since integration by parts is an essential ingredient for stability proofs at the continuous level, it is necessary to transfer this property to the discrete level. This is precisely the design goal of summation-by-parts (SBP) operators. The methods used in this article belong to the class of collocated SBP discontinuous Galerkin discretizations [12, 13]. In particular, they are based on collocated Legendre–Gauss–Lobatto (LGL) quadrature rules in one space dimension and the corresponding discrete SBP operators on curvilinear, unstructured tensor product elements.

While SBP operators can be viewed as matrix difference operators that are mimetic of integration by parts, additional techniques are necessary to compensate for the lack of the chain rule. For example, ad hoc split form methods have been provided for some partial differential equations such as the Burgers’ and the compressible Euler equations [14]. In [12], Carpenter and co-authors demonstrated the generic logic behind the splitting [25] procedure by showing the flux differencing technique with the telescoping property, i.e., a telescoping flux form at the element level. Flux differencing is essentially a high order difference operation on Tadmor’s entropy conservative fluxes [26], and applies to any system with any given entropy function [25]. Discrete stability over the whole domain is achieved by combining the SBP operator with suitable inter-element coupling procedures and boundary conditions, e.g., the simultaneous-approximation-terms (SATs); see [13, 27, 28]. The SAT technique is also used in this work. By constructing schemes that are discretely mimetic of the continuous stability analysis, the need to assume exact integration in the stability proofs is eliminated.
HseeL §or exampleL the work o§ hughes et alN these ideas have been extended to include p- and hp-adaptive discontinuous Galerkin methods (sometimes referred to as discontinuous collocated spectral elements) [19, 30], and fully discrete entropy stable algorithm [31, 32]. Those hp-adaptive methods [19] and the relaxation Runge–Kutta schemes presented in [31], which allow constructing fully discrete entropy stable discretizations with the method of lines approach, are also used herein.

II. The fully-discrete entropy stable discretization of the compressible Navier–Stokes equations

The compressible Navier–Stokes equations in Cartesian coordinates read

\[
\frac{\partial q}{\partial t} + \sum_{m=1}^{3} \frac{\partial f^I_{x_m}}{\partial x_m} = \sum_{m=1}^{3} \frac{\partial f^V_{x_m}}{\partial x_m}, \quad \forall (x_1, x_2, x_3) \in \Omega, \quad t \geq 0,
\]

\[
q \left( x_1, x_2, x_3, t \right) = g^B \left( x_1, x_2, x_3, t \right), \quad \forall (x_1, x_2, x_3) \in \Gamma, \quad t \geq 0,
\]

\[
q \left( x_1, x_2, x_3, 0 \right) = g^0 \left( x_1, x_2, x_3 \right), \quad \forall (x_1, x_2, x_3) \in \Omega,
\]

where the vectors \( q, f^I_{x_m}, \) and \( f^V_{x_m} \) are denoted as the conserved variables, the inviscid fluxes, and the viscous fluxes, respectively (a detailed description of these vectors is given later). The boundary data, \( g^B \), and the initial condition, \( g^0 \), are assumed to be in \( L^2(\Omega) \), with the further assumption that \( g^B \) coincides with linear, well-posed boundary conditions, prescribed in such a way that either entropy conservation or entropy stability is achieved.

The vector of conserved variables is given by

\[
q = [\rho, \rho U_1, \rho U_2, \rho U_3, \rho E]^T,
\]

where \( \rho \) denotes the density, \( U = [U_1, U_2, U_3]^T \) is the velocity vector, and \( E \) is the specific total energy. Herein, to close the system of equations (1), we use the following thermodynamic relation

\[
P = \rho R T, \quad R = \frac{R_u}{M_w},
\]

where \( P \) is the pressure, \( T \) is the temperature, \( R_u \) is the universal gas constant, and \( M_w \) is the molecular weight of the gas.

The compressible Navier–Stokes equations given in (1) have a convex extension (a redundant sixth equation constructed from a non-linear combination of the mass, momentum, and energy equations), that, when integrated over the physical domain, \( \Omega \), depends only on the boundary data and negative semi-definite dissipation terms. This convex extension depends on an entropy function, \( s \), that is constructed from the thermodynamic entropy as

\[
s = -\rho s,
\]
where $s$ is the thermodynamic entropy, and provides a mechanism for proving stability in the $L^2$ norm. The entropy variables, $\omega$, are an alternative variable set related to the conservative variables via a one-to-one mapping. They are defined in terms of the entropy function $s$ by the relation $\omega^T \equiv \partial s/\partial q$. This set of variables is extensively used in the entropy stability proofs of the algorithm presented herein; see for instance [12, 19, 33]. In addition, they simultaneously symmetrize the inviscid and the viscous flux Jacobians in all three spatial directions. Moreover, for the inviscid fluxes, the entropy variables satisfy the following conditions (sometimes referred to as integrability conditions)

$$\omega^T \frac{\partial f^I_{xm}}{\partial x_m} = \frac{\partial \mathcal{G}_{xm}}{\partial x_m}, \quad m = 1, 2, 3,$$

(4)

where $\mathcal{G}_{xm}$ are the inviscid entropy fluxes.

The entropy stability for the viscous terms in the compressible Navier–Stokes equations (1) is readily demonstrated by exploiting the symmetrizing properties of the entropy variables. Thus, viscous fluxes are recast in terms of the entropy variables as

$$f^V_{xm} = \sum_{j=1}^{3} C_{m,j} \frac{\partial \omega}{\partial x_j},$$

(5)

with the flux Jacobian matrices satisfying $C_{m,j} = C_{j,m}^T$ and $C_{m,j} \geq 0$ (see [13, 34] for their construction).

The entropy stability of the compressible Navier–Stokes equations can now be proven by using the following steps:

1) Contract (1) with the entropy variables, i.e., multiply by $\omega^T$, and integrate over the domain

$$\Omega \left( \omega^T \frac{\partial q}{\partial t} \sum_{m=1}^{3} \omega^T \frac{\partial f^I_{xm}}{\partial x_m} \right) d\Omega = \Omega \left( \sum_{m,j=1}^{3} \omega^T \frac{\partial \mathcal{G}_{xm}}{\partial x_m} \frac{\partial \omega}{\partial x_j} \right) d\Omega;$$

(6)

2) Use the conditions (4) and then integration by parts on the left- and right-hand side terms

$$\Omega \omega^T \frac{\partial q}{\partial t} d\Omega = \Omega \left( \sum_{m=1}^{3} \left( -\mathcal{G}_{xm} \omega^T C_{m,j} \frac{\partial \omega}{\partial x_j} \right) \right) d\Omega - \Omega \left( \sum_{m,j=1}^{3} \frac{\partial \omega^T}{\partial x_m} C_{m,j} \frac{\partial \omega^T}{\partial x_j} \right) d\Omega;$$

(7)

3) Using the definition of the entropy function, $s$, and the chain rule on the temporal term, and entropy stable boundary conditions

$$\Omega \frac{\partial s}{\partial t} d\Omega \leq \text{Data};$$

(8)

4) Integrating in time gives a bound on the entropy which is then converted into a bound on the solution, i.e.,

$$\Omega \mathbf{q}^T \mathbf{q} d\Omega \leq \text{Data}.$$

(9)

For further details on continuous entropy analysis, see, for example, [13, 35, 36].
A. Spatial discretization

The approximation of the compressible Navier–Stokes equations proceeds by partitioning the domain $\Omega$ into $K$ non-overlapping subdomains $\Omega_\kappa$. On the $\kappa^{th}$ element, the generic entropy stable discretization of (1) reads

$$\frac{d q_\kappa}{dt} = \sum_{m=1}^{3} 2D_{m,x}^{I,\kappa} \circ F_{x_m}(q_\kappa, q_\kappa) 1_\kappa = \sum_{m,j=1}^{3} \frac{\nabla_{x_m}}{C_{m,j}} \theta_j SAT I SAT V \text{diss}^I \text{diss}^V,$$

where the vector $q_\kappa$ is the discrete solution at the mesh nodes, and the vectors $\text{diss}^I$ and $\text{diss}^V$ are added interface dissipation for the inviscid and viscous portions of the equations, respectively. To mimic the continuous entropy stability analysis, the derivatives of the inviscid fluxes are approximated in a special way, i.e.,

$$\frac{\partial f_x^I}{\partial x_m} \approx 2D_{x_m}^{I,\kappa} \circ F_{x_m}(q_\kappa, q_\kappa) 1_\kappa,$$

where $D_{x_m}^{I,\kappa}$ is a differentiation matrix for the $x_m$ direction, $\circ$ is the Hadamard product (entry-wise multiplication), $F_{x_m}(q_\kappa, q_\kappa)$ is a two-point flux function matrix [19], and $1_\kappa$ is a vector of ones.

This special construction of the approximation to the derivative of the inviscid fluxes is needed so that the volume terms mimic the entropy stability analysis at the continuous level. Namely, they discretely mimic the integration by parts property induced by the integrability condition (4), i.e., for the diagonal-norm and diagonal $E_{x_m}$ SBP operators used in SSDC solver, the following holds

$$\int_{\Omega_\kappa} w^T \frac{\partial F_{x_m}}{\partial x_m} d\Omega = \frac{1}{2} \int_{\Gamma_\kappa} (w_f^{T} f_{x_m} - \psi_{x_m}) n_{x_m} d\Gamma = \frac{1}{2} \int_{\Gamma_\kappa} (w_f^{T} f_{x_m})^T E_{x_m} - (\psi_{x_m})^T E_{x_m},$$

where $E_{x_m}$ is the scalar version of $E_{x_m}$, and $w_f^{T} f_{x_m}$ is to be understood as the vector constructed point-wise from $w^T f_{x_m}$. Moreover, on the right-hand side, the entropy potential flux, $\psi_{x_m}$, has been introduced. This quantity is related to the entropy flux via the equality $F_{x_m} = w^T f_{x_m} - \psi_{x_m}$. The critical technology here is that when the semi-discrete equations are contracted into a single scalar equation, the result is a telescoping of the entropy flux to the boundary nodes. The matrix $D_{x_m}^{I,\kappa}$ is an SBP operator constructed by discretizing the skew-symmetric split of the Cartesian derivative, i.e.,

$$\frac{\partial f}{\partial x_m} = \frac{1}{2} \sum_{l=1}^{3} \frac{\partial}{\partial \xi_l} \left( g \frac{\partial \xi_l}{\partial x_m} f \right) + \frac{\partial}{\partial x_m} \frac{\partial f}{\partial \xi_l},$$

where $g$ is the metric Jacobian. The equality in (12) holds because the metric terms satisfy the following geometric conservation laws (GCL)

$$\sum_{l=1}^{3} \frac{\partial}{\partial \xi_l} \left( g \frac{\partial \xi_l}{\partial x_m} \right) = 0, \quad m = 1, 2, 3.$$
The differentiation matrix $D_{x,m}^{I,\kappa}$ is constructed as
\[
D_{x,m}^{I,\kappa} \equiv \frac{1}{2} \sum_{l=1}^{J-1} \left( \frac{\partial}{\partial x_m} \right) \kappa \left( \begin{array}{c} J \\ \partial \xi_l \\ \partial \xi_a \\ D \xi_l \end{array} \right) .
\] (14)

The metric terms used in constructing $D_{x,m}^{I,\kappa}$ also need to satisfy a discrete version of the GCL conditions (more on this in [19, 37]). On the other hand, the differentiation matrix, $D_{x,m}^{V_{1,\kappa}}$, is constructed as
\[
D_{x,m}^{V_{1,\kappa}} \equiv J^{-1}_{\kappa} \sum_{l=1}^{J-1} D \xi_l \left( \begin{array}{c} J \\ \partial \xi_l \\ \partial \xi_a \\ D \xi_l \end{array} \right) .
\] (15)

Furthermore, the derivative of the entropy variables is approximated as
\[
\theta_a = D_{x_a}^{V_{2,\kappa}} w_{\kappa} \ SAT_{V^2} \approx \frac{\partial w}{\partial x_a} .
\] (16)

The differentiation matrix, $D_{x_j}^{V_{2,\kappa}}$, is defined as
\[
D_{x_j}^{V_{2,\kappa}} \equiv J^{-1}_{\kappa} \sum_{a=1}^{J-1} \left( \begin{array}{c} J \\ \partial \xi_a \\ \partial \xi_a \\ D \xi_a \end{array} \right) .
\] (17)

The various metric terms have been colour-coded, i.e., $J_{\kappa}$, $J_{\kappa} \left( \begin{array}{c} J \\ \partial \xi_l \end{array} \right)$, $\left( \begin{array}{c} J \\ \partial \xi_a \end{array} \right)$, to highlight that these terms can in principle, be computed in different ways. In the SSDC solver, these metrics terms are computed as described in [19, 30, 37], for conforming and $p$- and $hp$-nonconforming grids. The terms SAT are the simultaneous approximation terms that weakly couple neighboring elements or impose boundary conditions and are discussed in more detail in the subsections below.

At the beginning of the simulation, the metric terms are computed with an optimization algorithm so that they satisfy the discrete GCL conditions and approximate optimally in the $L^2$ norm the analytical metrics [37]. Then for a given Runge-Kutta stage, the right-hand side of the ordinary differential equation is computed with the following steps

1) Construct the viscous fluxes $\theta_j$;
2) Compute the inviscid terms $2 D_{x_m} \circ F_{x_m} (q_{\kappa}, q_{\kappa}) 1_{\kappa}$;
3) Compute the viscous terms $D_{x_m} \left[ C_{m,j} \right] \theta_j$;
4) Compute the SAT terms.

The presentation of the SAT terms is more involved and depends on whether the interface is conforming or nonconforming. Thus, we kindly refer the reader to [27] and [19] for conforming and nonconforming interfaces, respectively.

Thus, using the SBP-SAT operators and their equivalent telescoping form and following the discrete entropy stability
analysis presented in [19], the total entropy of the spatial discretization satisfies
\[
\frac{d}{dt} \hat{P}^T S = \frac{d\phi}{dt} = B^T - D^T \Upsilon.
\] (18)

This equation mimics at the semi-discrete level each term in (7). Here \(B^T\) is the discrete boundary term (i.e., the discrete version of the first integral term on the right-hand side of (7)), \(D^T\) is the discrete dissipation term (i.e., the discrete version of the second term on the right-hand side of (7)) and \(\Upsilon\) enforces entropy stable interface coupling and boundary conditions, [13]. For completeness, we note that the matrix \(\hat{P}\) may be thought of as the mass matrix in the context of the discontinuous Galerkin finite element method.

We note that the entropy stability statement relies on the assumption that two thermodynamic variables (e.g., density and temperature) are positive. Thus, another mechanism must be employed to bound those variables away from zero to guarantee positivity; positivity preservation will not be considered herein.

**B. Temporal discretization**

The explicit class of Runge–Kutta schemes is used to integrate in time the system of ordinary differential equations (ODEs) arising from the entropy stable spatial discretizations described in the previous section. In particular, we use the relaxation Runge–Kutta methods proposed in [31]. In the next subsection, we give an overview of these schemes.

**Runge–Kutta Methods** A general (explicit or implicit) Runge–Kutta method with \(s\) stages can be represented by its Butcher tableau [38, 39]
\[
\begin{array}{c|c}
\mathbf{c} & \mathbf{A} \\
\hline
\mathbf{b} & \mathbf{y}^T \\
\end{array}
\] (19)
where, for a scheme with \(s\) stages, \(A\) is a real square matrix of dimensions \(s \times s\), while \(b\) and \(c\) are column vectors with \(s\) real elements. For the system of ODEs (10), a step from \(q^n \approx qt_n\) to \(q^{n+1} \approx qt_{n+1}\) where \(t_{n+1} = t_n + \Delta t\) is given by
\[
y_i = q^n \Delta t \sum_{j=1}^{s} a_{ij} f t_n c_j \Delta t, y_j, \quad i = 1, \ldots, s, \quad (20a)
\]
\[
q^{n+1} = q^n \Delta t \sum_{i=1}^{s} b_i f t_n c_i \Delta t, y_i. \quad (20b)
\]

Here, \(y_i\) are the stage values of the Runge–Kutta method. For simplicity, we will make use of the shorthand
\[
f_i = f t_n c_i \Delta t, y_i, \quad f_0 = f t_n, q^n. \quad (21)
\]
Relaxation Runge–Kutta Methods  Following [31], the basic idea to make a given Runge–Kutta method entropy stable is to scale the weights \( b_i \) by a real value parameter \( \gamma_n \), i.e., to use

\[
q^{n+1}_\gamma = q^n \gamma_n \Delta t \sum_{i=1}^s b_i f_i
\]

instead of \( q^{n+1} \) in (20b) as the new value after one time step. The generalization to entropy stability is to enforce the condition

\[
\eta q^{n+1}_\gamma - \eta q^n = \gamma_n \Delta t \sum_{i=1}^s b_i \langle \eta' y_i, f_i \rangle,
\]

by finding a root \( \gamma_n \) of

\[
r \gamma = \eta (q^n \gamma \Delta t \sum_{i=1}^s b_i f_i) - \eta q^n - \gamma \Delta t \sum_{i=1}^s b_i \langle \eta' y_i, f_i \rangle,
\]

where \( \eta = \Omega \cdot d\Omega \) and \( \langle \cdot, \cdot \rangle \) denote the inner product inducing the norm \( \| \cdot \| \). Note that the direction

\[
d^n = \sum_{i=1}^s b_i f_i,
\]

and the estimate of the entropy change

\[
e = \Delta t \sum_{i=1}^s b_i \langle \eta' y_i, f_i \rangle
\]

can be computed on the fly during the computation of the Runge–Kutta method and are not influenced by \( \gamma_n \). Hence, existing low-storage Runge–Kutta implementations can be used.

III. Numerical results

The curvilinear, unstructured grid SSDC solver developed in the Advanced Algorithms and Numerical Simulations Laboratory (AANSLab) [40] is used to perform the numerical simulations. The SSDC solver is built on top of the Portable and Extensible Toolkit for Scientific computing (PETSc) [41], its mesh topology abstraction (DMplex) [42], and scalable ordinary differential equation (ODE)/differential algebraic equations (DAE) solver library [43].

The 3(2) Runge–Kutta pair proposed by Bogacki and Shampine [44], with an adaptive time-stepping based on signal processing [45, 46], is used to integrate the numerical solution in time for all the numerical results provided in the paper. The relative and absolute tolerances used for the time adaptivity are set to \( 10^{-6} \) for all the test cases. The parameter \( \gamma_n \) of the relaxation Runge–Kutta schemes is computed from eq. (24) to machine precision using the bisection method which, for efficiency, is implemented directly in the SSDC solver.

All the grids are constructed using the commercial software Pointwise V18.3 released in September, 2019.
A. Flow past two spheres in tandem

We investigate the flow past two equally sized spheres with a diameter $D$, held fixed in a rectangular domain located at a separation distance of $10D$, at the moderate Reynolds numbers of $Re_D = U_{\infty}D/\nu = 3.9 \times 10^3$ and $10^4$, and a Mach number $Ma_{\infty} = 0.1$. The Prandtl number for both simulations is set to $Pr = 0.7$. Herein, the quite upstream flow conditions of the first sphere are used to define the similarity parameters, e.g., $U_{\infty}$ is the free-stream velocity.

The case corresponding to $Re_D = 3.9 \times 10^3$ is part of benchmarks proposed for the $5^{th}$ International Workshop on High–Order CFD Methods (HiOCFD5) [47]. Despite the simplicity of the geometry, capturing the flow at this regime is quite challenging. The importance of such flows around multiple bodies, and in particular around two spheres, is considered from many practical problems to be of great interest, since it provides, for example, a better understating of the effect of the wake behind a leading bluff-body on the flow around a trailing one. A non-exhaustive list of important applications ranging from industrial fluidized beds to bio-reactors, to the combustion of aerosols, could be liquid-gas two-phase flows [48], suppression of icing on the solid surface [49], and oil droplets [50].

We perform the numerical simulations using one of the grids $\text{TandemSpheresHexMesh2Pm}$ provided by Steve Karman of Pointwise for the HiOCFD5 [47]. In this study, this mesh is used in combination with two solution polynomial order: $p = 8$, and $10$ for both Reynolds’ numbers. The integer $p$ refers to the solution polynomial degree of the SSDC scheme. The two simulations have respectively $\approx 1.432 \times 10^7$ and $\approx 2.616 \times 10^7$ DOFs. The time-averaged collection of data are computed at non-dimensionalized convective time scale $tU_{\infty}D \in 100, 200$.

To assess the quality of the grid resolution, we perform a posteriori analysis by comparing the grid size, $h \equiv \Delta x \times \Delta y \times \Delta x^{13}$, to the Kolmogorov length scale, $\eta_k$, based on Kolmogorov’s hypothesis of local isotropy. Such a hypothesis is obtained from the local rate of dissipation of turbulence kinetic energy per unit mass of fluid, $\epsilon$, and the kinematic viscosity of the fluid, $\nu$, as,

$$\eta_k = \left( \frac{\nu^3}{\epsilon} \right)^{14}.$$

In fig. 1, we present the ratio of $h\eta_k$ along the tandem spheres centerline at both Reynolds numbers obtained with $p = 8$ and $p = 10$. We note that for $Re_D = 3.9 \times 10^3$, the average ratio between the grid size and the Kolmogorov length scales is less than 2 in the near wake zone and up to a distance of $x_1 D \sim 16$ from the second sphere. The case $Re_D = 1.0 \times 10^4$ exhibits a similar profile, with a higher peak value. Thus, we can conclude that the grid density used for both Reynolds numbers and solution polynomial degrees can be considered sufficiently fine for resolving the smallest flow scales in the critical region of the near wake zone of both spheres [51].

The complexity of the flow fields is illustrated in fig. 2 by means of the iso-contours of the second invariant of the velocity gradient tensor, $Q U_{\infty} D^2 = 2$, colored by the magnitude of the vorticity, $\|\omega\| = \|\nabla \times U\|$. According to the definition of $Q$-criterion, a vortical structure is identified in a region with positive $Q$, i.e., a region where the vorticity overcomes the strain. In both cases, we observe an axisymmetric laminar boundary layer that i) separates nearly from the equator of the first sphere, and ii) remains laminar up to a certain distance from the blunt-body and then it becomes
unstable. The flow becomes fully three-dimensional and results in a transition to turbulence.

Furthermore, for \( \text{Re}_D = 3.9 \times 10^3 \), we observe that the vortex rings and the hairpin-like structures which appear behind the first sphere and form regular vortices that move downstream in a helical-like fashion are similar to those observed numerically by Dorschner et al. [52]. The unsteady wake impinges on the downstream cylinder and generates a more chaotic structures and a faster transition to turbulence. In addition, the instantaneous iso-contour of the \( Q \)-criterion shows that the separating shear layers are longer in the simulations with a finer resolution (\( i.e., p = 10 \)) than those
resolved in the coarser grid (i.e., $p = 8$). Apart from the differences in the vortical structures shed into the wake, a higher Reynolds number corresponds to a faster separation. Specifically, at $Re_D = 3.9 \times 10^3$, the separation occurs past the apex of both spheres at a location close to $\theta_{sep} = 100^\circ$ whereas, for $Re_D = 1.0 \times 10^4$, it occurs close to the sphere apex. This flow feature is well known in literature, and it persists in the sub-critical regime up until the drag crisis occurs [53]. The co-existence of a thin shear layer that develops from the front sphere, the laminar-transition of the wake of the first sphere that impacts and interacts with the flow past the second sphere, the complex unsteady multi-scale flow features at these low Mach and transitional Reynolds numbers possess various numerical difficulties that have to be carefully handled and resolved by both the spatial and temporal integration algorithms.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Front mean $C_D$</th>
<th>Back mean $C_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present results for $p = 8$</td>
<td>0.372</td>
<td>0.389</td>
</tr>
<tr>
<td>Present results for $p = 10$</td>
<td>0.372</td>
<td>0.424</td>
</tr>
<tr>
<td>5.2M DOFs $P_2$, [54]</td>
<td>0.367</td>
<td>N/A</td>
</tr>
<tr>
<td>12.3M DOFs $P_3$, [54]</td>
<td>0.374</td>
<td>N/A</td>
</tr>
<tr>
<td>Adaptive $P_3$, FEniCS-HPC, [55]</td>
<td>$\approx 0.384$</td>
<td>$\approx 0.370$</td>
</tr>
<tr>
<td>Mesh 3 $P_3$, FEniCS-HPC, [55]</td>
<td>$\approx 0.390$</td>
<td>$\approx 0.340$</td>
</tr>
<tr>
<td>Mesh 4 $P_3$, FEniCS-HPC, [55]</td>
<td>$\approx 0.440$</td>
<td>$\approx 0.445$</td>
</tr>
<tr>
<td>Mesh 5 $P_3$, FEniCS-HPC, [55]</td>
<td>$\approx 0.425$</td>
<td>$\approx 0.445$</td>
</tr>
<tr>
<td>69.4M DOFs, Mesh 4 $P_2$ SDIRK4 $\Delta t = 0.03$ [56]</td>
<td>0.411</td>
<td>0.439</td>
</tr>
<tr>
<td>21.5M DOFs, Mesh 3 $P_2$ SDIRK4 $\Delta t = 0.03$ [56]</td>
<td>0.457</td>
<td>0.458</td>
</tr>
</tbody>
</table>

Table 1 Comparison of the mean drag coefficient, $C_D$, for the tandem spheres at $Re_D = 3.9 \times 10^3$ (adapted from Martinelli and Lohry [56]).

In Table 2, we compare the present mean drag coefficients and those reported during the HiOCFD5 workshop [47] for $Re_D = 3.9 \times 10^3$. The mean drag results for $Re_D = \times 10^4$ are instead shown in Table 1. Unsurprisingly, the predicted $C_D$ between $p = 8$ and $p = 10$ is similar for the front sphere because of the nominally steady and laminar flow passing over it. A slight difference can be observed in the back sphere drag coefficient, since the second sphere experiences unsteady flow effects.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Front mean $C_D$</th>
<th>Back mean $C_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present results for $p = 8$</td>
<td>0.405</td>
<td>0.434</td>
</tr>
<tr>
<td>Present results for $p = 10$</td>
<td>0.399</td>
<td>0.402</td>
</tr>
<tr>
<td>LES results for one sphere [57]</td>
<td>0.402</td>
<td>-</td>
</tr>
<tr>
<td>Experimental result [58]</td>
<td>0.4</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2 Comparison of the mean drag coefficient, $C_D$, for the tandem spheres at various resolutions for $Re_D = \times 10^4$.

In fig. 3, we look at the convergence of the mean average pressure coefficient, $C_P = 2 \bar{P} - \bar{P}_\infty \rho_\infty U^2_\infty$, on the surface of both spheres. This pressure coefficient is plotted as a function of the angle $\theta$ (measured from the upstream stagnation
point. First, the distribution of the pressure on the surface of the sphere is symmetric at both Reynolds’ numbers, since the incoming flow and the problem geometry are both symmetric. Second, we observe that by increasing the solution polynomial degree, $p$, the coefficient $C_P$ in the $x_1$-$x_2$ passing through the centerline converges to a well-defined distribution for both Reynolds numbers. We also observe that the angular position of the minimum value of $C_P$ is captured well for both polynomial degrees, while the back pressure is slightly underestimated by the solution computed with $p = 8$. The angular position of the pressure minimum is the same at both Reynolds’ numbers. In addition, we observe that the pressure distribution is quite similar at both Reynolds’ numbers except for some differences in the back region, i.e., in the plateau regions between $-90^\circ$ and $90^\circ$. Furthermore, it appears that the flow past the second sphere requires more time to reach a converged $C_P$ distribution in the back region, especially in the $x_2$-$x_3$ plane and for a higher Reynolds value number.

In fig. 4, we show the variation of the mean stream-wise velocity, $U_1$, the mean density, $\bar{\rho}$, the mean temperature, $\bar{T}$, the diagonal components of the Reynolds tensor, $U_1^2$, $U_2^2$, and $U_3^2$ along the centerline for $p = 8$ and 10. We observe that as the polynomial degree increases, the first-order quantities converge to a well-defined distribution. In particular, for $p = 8$, the minimum velocity, as well as the size of the recirculation zone, matches the results computed

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3.png}
\caption{Time average of the surface pressure coefficient, $C_P$, at $Re_D = 3.9 \times 10^3$ (solid lines) and $Re_D = 1.0 \times 10^4$ (dashed lines).}
\end{figure}
with the most refined solutions, i.e., $p = 10$ on the finest grid. The negative values of the stream-wise velocity in the recirculation bubble are higher for i) the second sphere compared to the first sphere and ii) for the higher Reynolds number, i.e., $Re_D = \times 10^4$. This means that, when the regime goes from sub-critical to fully turbulent, the velocity deficit in the wake increases as the size of the bubble decreases (as can be seen from fig. 2). The size of the recirculation bubble in the wake is drastically reduced from the first to the second wake at both Reynolds’ numbers. Note also the absence of the inflection point in the mean-stream-wise profile, $\overline{U_1}$, near the first sphere when the Reynolds’ number becomes higher. The profile of both density and temperature are well converged at both Reynolds’ numbers, and they differ only in the wake of the first sphere. In regards to the diagonal components of the Reynolds tensor, we observe that the three components i) exhibit a maximum near $3.6D$ for $Re_D = 3.9 \times 10^3$ and $2.1D$ for $Re_D = \times 10^4$ and rapidly
decreases, and ii) present the same peak at \( \sim 11D \) as the flow passes through the second sphere. The turbulence intensity level in the wake of the second sphere is i) lower than the one observed in the wake of the first sphere, and ii) is in the same order of magnitude for both \( Re_D \), meaning that independently of the value of Reynolds number, the flow passing through the second sphere could be classified according to the same regime.

Figure 5  Normalized diagonal Reynolds stresses \( \overline{U_1'U_1'^2} \) on transverse sections \( x_1D = 2, 4, 6, 8, 12, 14, 16 \) and 18, for the flow past two spheres in tandem. Each plot is separated by a horizontal offset of 0.08

The wake mean flow normalized diagonal Reynolds stresses \( \langle U_1'U_1'^2 \rangle \), \( \langle U_2'U_2'^2 \rangle \), and \( \langle U_3'U_3'^2 \rangle \) are shown in fig. 5. The convergence in \( p \) is more pronounced in the lower regime of turbulence, while it is more difficult to reach in the higher turbulence regime. The fluctuations for \( Re_D = 10^4 \) are larger within the recirculation zone of the first sphere and comparable with the wake of the second sphere. The three components of the Reynolds tensor are positive for \( x_1D > 5 \) but near zero, since this position corresponds to the end of the recirculation bubble, and the location is closer to the aft sphere, \( i.e., x_1D - 10D \sim 2 \). The cross stress distributions presented in fig. 6, reveal the near-wake structure associated with the changes in the base-suction coefficient. In particular, they highlight that the recirculation region of the first sphere is relatively large while a smaller region is present behind the second sphere, for which the flow regime is
completely turbulent. For $p = 10$, peaks in the stresses appear at a distance < 4D, and their levels are higher in the wake of the second sphere. This is consistent with the higher level of turbulence intensity passing through the second sphere.

**B. NASA juncture flow experiment**

In this section, we perform the simulation of the NASA juncture flow experiment [59], which is a new effort specifically designed to collect validation data in the juncture region of a wing-body configuration [60]. In fact, very few experimental details are available for such flows, in part because of the difficulties inherent in measuring data in the flow field very close to the walls. Some efforts have aimed at improving knowledge of these flow physics [61], but
significant research is still needed. Because multiple juncture flows are present on practically all civilian and military air vehicles, there is strong motivation to improve CFD’s capabilities to predict them. Current turbulence models routinely employed by Reynolds-averaged Navier–Stokes CFD solvers are inconsistent in their prediction of corner flow separation in aircraft juncture regions, so experimental data in the near-wall region of such a configuration are useful both for assessment as well as for turbulence model improvement [60].

Here, we use the NASA juncture flow geometry with a wing based on the DLR-F6 and a leading-edge horn to mitigate the effect of the horseshoe vortex over the wing-fuselage juncture. A general view of the geometry is shown in fig. 7b. The model crank chord (chord length that the wing platform break) is $\ell = 557.1$ mm, the wingspan is $77.89\ell$ and the fuselage length is $f = 8.69\ell$. The origin of the frame of reference is placed at the tip of the nose of the fuselage. The $x_1$ axis is aligned with the fuselage centerline, the $x_2$ axis denotes span-wise direction, and the $x_3$ axis is the vertical direction. The wing leading-edge horn meets the fuselage at $x_1 = 3.45\ell$, and the wing root trailing-edge is located at $x_1 = 5.31\ell$. In the wind tunnel, the model was mounted on a sting aligned with the fuselage axis [59]. The sting was attached to a mast that emerged from the wind tunnel floor. We performed all calculations in free air conditions, and the sting and mast are ignored. The computational set-up is shown in fig. 7. The Reynolds number is $Re_\ell = 2.4 \times 10^6$ and the free-stream Mach number is $Ma_\infty = 0.189$. The angle of attack is $AoA = -2.5^\circ$. We impose a uniform flow at the inflow boundary conditions, far-field boundary conditions at the outlet, entropy stable inviscid wall boundary conditions in the lateral boundaries [13], and on the model surfaces we apply the entropy stable adiabatic no-slip wall boundary conditions presented in [13, 28].

The grid is subdivided into three blocks, as shown in fig. 7b. Each block corresponds to a different solution polynomial degree, $p$. In particular, we use $p = 1$ in the far-field region, $p = 3$ in the region that surrounds the model, and $p = 2$ in the remaining part. The total number of hexahedral elements and DOFs are $\approx 6.762 \times 10^5$ and $\approx 4.091 \times 10^7$, respectively. The number of DOFs is very close to the coarse mesh used for the same problem in [62, 63]. We highlight that the boundary layer thickness over the fuselage for $x = 1,000$-2,000 mm is about $16$ mm, while it is about $20$ mm over the wing upstream of the separation bubble [63]. In the present simulation, we use 89 LGL points in the boundary layer thickness, $\delta_{99}$.

The simulation is started from scratch using a uniform flow as the initial condition and is run for 14.8 (transient) + 14.8 (statistics). As the first simulation of this kind with SSDC, we chose to limit to CFL $\approx 1.0$. However, current ongoing simulations indicate that we can use larger CFL values. The computational cost for the simulation is calculated as the number of processors $\times$ number of hours $= 4,096 \times 7 = 28,716$ core hours, to run the simulation for one $\tau U_\infty$ period of time.

In fig. 8, we depict the vortical features captured by the grid using iso-contours of the $Q$-criterion colored by the normalized instantaneous horizontal velocity contours, $U_1 U_\infty$. Near the wing surface, unsteady vortical features are observed after $\sim 34\ell$, indicating that the flow has tripped to turbulence. To provide more insights into the topology of the flow, mainly the separation zone, we provide in fig. 9 a visual impression of the temporal average stream-wise velocity
$U_1$ in a $x_1 - x_3$-plane located at $x_2 = 236.9$ mm. In this contour plot, we observe a separation bubble, corresponding to negative values of the stream-wise velocity (i.e., the dark blue region on top of the surface).

In fig. 10, we compare the pressure distributions obtained with the SSDC solver, the experimental data of Kegerise and Neuhart [59], and the numerical results of Lee and Pulliam [64] (the free air to wind tunnel sting-mast results). The latter numerical results are obtained using Overflow 2.2N, which feature a third-order accurate Roe upwind scheme for the convective fluxes, an implicit solve done using the ARC3D Beam–Warming scalar pentadiagonal scheme, and a low-Mach number preconditioning. From fig. 10, we observe a good agreement with the measured data and, virtually, no difference between the simulation results, confirming that the present low grid resolution featuring no wall-modeling is sufficient for providing reliable results.
Figure 8  Vortical features for NASA junction experiment visualized using iso-contours of the $Q$-criterion colored by normalized instantaneous velocity $U_1 U_\infty$.

Figure 9  2D dimensionless slice of the temporal average streamwise velocity field at $x_2 = 236.9$ mm close to the trailing edge for the NASA junction experiment.

IV. Conclusion

In computational fluid dynamics, the demand for increasingly multidisciplinary reliable simulations, for both analysis and design optimization purposes, requires transformational advances in individual components of future solvers. Here, we provided numerical evidence of competitiveness and adequacy of provably entropy-stable, adaptive, high-order accurate scale-resolving methods for the new generation of CFD tools with a good level of maturity. This work indicates that, although incremental improvements to existing algorithms will continue to improve overall capabilities, the development of novel robust numerical techniques such as $hp$-adaptive fully–discrete entropy-stable schemes and their
Figure 10  Mean pressure coefficient, $C_P$, distribution at several cross sections of the NASA juncture experiment wing; AoA = $-2.5^\circ$, $Re_c = 2.4 \times 10^6$, and $Ma_\infty = 0.189$.

Acknowledgments

The research reported in this paper was funded by King Abdullah University of Science and Technology. We are thankful for the computing resources of the Supercomputing Laboratory and the Extreme Computing Research Center at King Abdullah University of Science and Technology.
References


