

INDUSTRY-RELEVANT IMPLICIT LARGE-EDDY SIMULATION OF A HIGH-PERFORMANCE ROAD CAR VIA SPECTRAL/HP ELEMENT METHODS *

GIANMARCO MENGALDO[†], DAVID MOXEY[‡], MICHAEL TURNER[§], RODRIGO C.
MOURA[¶], AYAD JASSIM^{||}, MARK TAYLOR[#], JOAQUIM PEIRÓ[§], AND SPENCER J.
SHERWIN[§]

Abstract. We present a successful deployment of high-fidelity Large-Eddy Simulation (LES) technologies based on spectral/*hp* element methods to industrial flow problems, which are characterized by high Reynolds numbers and complex geometries. In particular, we describe the numerical methods, software development and steps that were required to perform the implicit LES of a real automotive car, namely the Elemental Rp1 model. To the best of the authors' knowledge, this simulation represents the first high-order accurate transient LES of an entire real car geometry. Moreover, this constitutes a key milestone towards considerably expanding the computational design envelope currently allowed in industry, where steady-state modelling remains the standard. To this end, a number of novel developments had to be made in order to overcome obstacles in mesh generation and solver technology to achieve this simulation, which we detail in this paper. The main objective is to present to the industrial and applied mathematics community, a viable pathway to translate academic developments into industrial tools, that can substantially advance the analysis and design capabilities of high-end engineering stakeholders. The novel developments and results were achieved using the academic-driven open-source framework **Nektar++**.

Key words. spectral/*hp* element methods; high-order CFD; high-order mesh generation; CAD integration; high Reynolds number flows; high-fidelity simulations; implicit LES; under-resolved DNS; industrial applications; automotive design.

AMS subject classifications. 68Q25, 68R10, 68U05

1. Introduction. The accurate modelling and prediction of fluid dynamics structures and aerodynamic forces around complex bodies is a classical and challenging problem that lies at the heart of modern automotive design. The increasing availability and lowering cost of computational modelling compared to experimental testing means that computational fluid dynamics (CFD) now forms a key part of the engineering design process. When considering the high Reynolds number flows around the complex bodies found in these problems, there are two key considerations that need to be taken into account: (a) creating highly accurate meshes for the geometry at hand, and (b) producing physically-faithful results, while using numerical discretisations that are necessarily unable to capture all the fluid dynamics scales involved due to the large computational costs required.

In industry, the most common practice revolved around the use of ensemble-averaged steady-state modelling, commonly using either Reynolds-averaged Navier-Stokes (RANS) or, more recently, Detached eddy Simulation (DES). Typically these

*Submitted to the editors DATE.

Funding: EPSRC under the Platform Grant PRISM: Underpinning Technologies for Finite Element Simulation (EP/L000407/1); EU Horizon 2020 project ExaFLOW (grant 671571). EPSRC and Airbus under an Industrial CASE studentship.

[†]National University of Singapore (NUS), Singapore (mpegim@nus.edu.sg)

[‡]University of Exeter, UK (d.moxey@exeter.ac.uk)

[§]Imperial College London, UK (mtturner4350@gmail.com, j.peiro@imperial.ac.uk, s.sherwin@imperial.ac.uk)

[¶]Instituto Tecnológico de Aeronáutica, Brazil (moura@ita.br)

^{||}Hewlett Packard Enterprise, UK (ayad.jassim@hpe.com)

[#]London Computational Solutions, UK (mark@lcs-fast.com)

are used in conjunction with low-order numerical discretisations, predominantly the second-order accurate finite volume method, together with linear meshes of planar-faced elements. Although this approach is computationally cheap, it can undermine the accuracy of the simulations and keep the practitioner from identifying aerodynamic design solutions that would be otherwise feasible if the CFD methodology was able to more faithfully describe the flow physics [75]. Indeed, on one hand, the use of highly diffusive numerical schemes and steady-state modeling strategies leads to suppress under-resolved scales and their feedback on the mean flow, thereby additionally corrupting the accuracy of the simulation. On the other, the use of linear meshes requires high degrees of refinement in order to capture highly-curved surfaces in the geometry, which may introduce artificial diffusion and significantly alter the flow evolution that one would otherwise observe. Addressing point (a) and (b) within the context of industrial applications is therefore of primary importance to advance the state-of-the-art of CFD, and can dramatically enhance the computational aerodynamic analysis and design capabilities of high-end engineering stakeholders.

Point (a) requires substantial theoretical and algorithmic advances in high-order mesh generation. From this perspective, the aim is to produce a methodology for generating high-order meshes that achieve levels of robustness over a range of very complex cases comparable to that found in commercial linear mesh generators. Indeed, commercial packages, such as Pointwise [66] and Centaur [8], have attempted to add high-order meshing capabilities to their products with variable degree of success.

The standard approach in generating a high-order mesh is to adopt an *a posteriori* procedure, whereby a coarse linear mesh is first generated, and is then deformed by introducing the geometric curvature into elements lying on the surface. Several research groups have been working on such an approach, see for example [13, 74, 44, 72, 70, 85], which gives an indication towards the level of interest in this area.

The challenge of this approach, however, lies in the second step of the process: introducing curvature purely at the surface will typically lead to elements becoming inverted and unsuitable for computation. This has proven a significant stumbling block in the use of high-order methods in industry, and many of the aforementioned works focus on developing techniques to regularize or heal the mesh as part of the *a posteriori* process. While this is the main route to high-order mesh generation, some alternatives exist. These include the use of the Nash theorem to map a Riemannian space onto an isometric Euclidean space of higher dimension [62, 87, 11], semi-structured approaches to block mesh generation [3], anisotropic high-order boundary-layer mesh generation [61], and PDE-based approaches based on elastic analogy [68].

In this work, we tackle the problem of high-order mesh generation by using a variational framework that uses the minimization of an energy functional. The process begins by defining a linear surface mesh that conforms with the underlying geometry, typically defined in a computer-aided design (CAD) format that follows standard ISO 10303 (informally known as standard for the exchange of product model data, or STEP) [69]. The linear surface mesh is then complemented with a linear volume mesh along with a boundary-layer mesh (if required). Finally, the linear surface mesh is made high-order to fit the curved surfaces of the underlying geometry, and the interior mesh entities (edges and surface) of the linear volume mesh are deformed to accommodate the curved boundaries imposed by the geometry. The latter step is achieved by the minimization of the energy functional mentioned above, that can be arbitrarily selected by the practitioner depending on the mesh requirements. The linear mesh is obtained beforehand, and uses a combination of the commercial packages `CADfix` and `Star-CCM+` for CAD healing and for linear and boundary-layer mesh generation,

respectively, as well as `NekMesh` [80], which is the open-source high-order meshing framework developed as part of the `Nektar++` project [63, 5, 59].

The second point, (b), is instead closely linked to the impossibility of carrying out simulations that can resolve all the scales present in the problem, that, in turn, leaves part of the flow physics marginally or under-resolved. The computational power required to perform a fully resolved flow simulation, also known as direct numerical simulation (DNS), is a function of the Reynolds number [37]. From this perspective, we can consider the number of floating point operations per second (FLOPS) that a computer can perform as an approximate measure of computational power, and we can then model the time to solution under different regimes of computational power growths. This gives a rough estimate of the number of years required to achieve DNS capabilities for different Reynolds numbers. Figure 1 depicts two different computational power growths. The red line assumes doubling flops every three years, while the blue line assumes a doubling of FLOPS every one and a half years. They represent a less and more optimistic scenario than Moore’s law, respectively. We can appreciate how, under the most optimistic power growth scenario (blue line), DNS capabilities for automotive simulations, which corresponds to Reynolds numbers of the order of 10^6 , will be reached approximately by 2040. Figure 1 is an ideal scenario, as it does not take into account additional costs incurred, and it also assumes using the top supercomputers, as soon as they become available. The latter aspect is obviously not achievable for analysis and design purposes in an engineering workflow. Therefore, the ability to perform DNS for industrial purposes at high Reynolds numbers on a regular basis is, at least, two or more decades ahead, even with current advances in computational technologies (e.g. many-core architectures, graphical processing units, and quantum computing). Given these limitations, we need to deal with under-resolution for the next several years, while pushing the boundaries of high-Reynolds number flow physics for industrial problems.

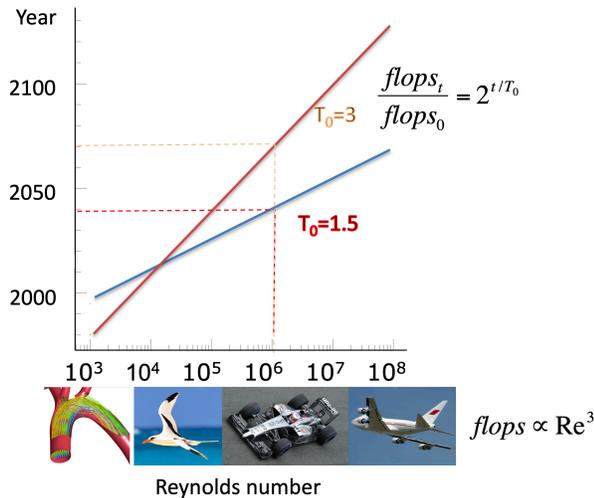


FIG. 1. Computational resources required to perform the direct numerical simulation (DNS) as a function of the Reynolds number [37]. Yellow and red lines represent projections on when DNS capabilities are due to become available as computational power increases following two different Moore law’s paths.

A significant body of work has been produced in the past several years to deal with high-Reynolds number problems. As briefly mentioned above, RANS and DES are frequently used in the context of industrial CFD, see for example [23, 86, 4], but they typically fail to capture unsteady flow features, which may be crucial for the analysis and design of a given aerodynamic solution. This aspect is especially marked when these two modeling choices are adopted in conjunction with low-order numerical schemes, that are commonly characterized by significant diffusion and dispersion, thereby causing an unphysical behaviour of the flow physics [7].

In order to overcome these issues, several research groups have been working on LES, both physical (or explicit) and implicit, over the past few decades. Explicit LES uses physical principles such as eddy viscosity to devise closure models that can approximate the under-resolved scales, while implicit LES utilizes the diffusion introduced by the numerical discretization. For a detailed review on LES methods, the interested reader can refer to Pope [67]. LES is particularly promising for automotive applications, since these regimes heavily feature flow physics that steady-state approaches struggle to capture, such as vortex separation and interaction [75]. In addition, when used in conjunction with low-diffusive and dispersive numerical schemes, the resolution power is enhanced and can produce flow simulations at unprecedented accuracy.

In this work, we use an implicit LES approach, where we treat the under-resolved scales via a novel artificial numerical regularization approach based on a so-called spectral vanishing viscosity (SVV) kernel, and we adopt robust dealiasing techniques to mitigate aliasing driven numerical instabilities that are normally present due to the low-diffusive behavior of the numerical framework. In particular, we show that the use of SVV and dealiasing strategies, along with key improvements in the computational efficiency of kernel calculations, leads to a robust (in terms of numerical stability) and accurate high-fidelity LES framework for the analysis and design of complex industrial problems. This is showcased in the successful high-fidelity simulation of the real geometry of the Rp1 automotive car at a Reynolds number of 10^6 based on the car length. All the simulations performed use the incompressible Navier-Stokes solver implemented in the open-source spectral/*hp* element framework `Nektar++`. The solver implements an operator splitting scheme [36, 29], where the spatial discretization is achieved via a continuous Galerkin method [38], and the time-discretization is obtained via a second-order implicit-explicit time-stepping strategy [83]. The numerical simulations performed were achieved with hybrid meshes containing both prisms (to discretize the boundary-layer regions in proximity to the body of the car) and tetrahedra (to discretize the farfield of the domain).

In this paper, we present the key advancement in high-order meshing and simulation technologies that we implemented in `NekMesh` and `Nektar++` in order to achieve the challenging simulation of the Rp1 car, with the aim translating academic research efforts to industrial CFD applications. In particular, section 2 describes the high-order meshing strategy we adopted, focusing on the steps required for surface and linear mesh generation as well as on the *a posteriori* high-order deformation. In section 3, we describe the novel simulation technologies implemented in terms of regularisation approaches for the under-resolved scales, dealiasing techniques for numerical stability, and computational efficiency. Section 4 presents the high-fidelity simulation of the Rp1 car, and finally, in section 5, we draw some final remarks.

2. High-order mesh generation for complex geometries. The starting point and essential prerequisite for any high-fidelity LES, within either a finite el-

ement or finite volume framework, is the generation of a geometrically-accurate and high-quality mesh. Mesh generation is a complex and involved process, particularly when the geometry under consideration is a complex, three-dimensional body, since the important regions that need to be resolved within the domain are often unknown *a priori*. The process may therefore require many iterations of meshing, simulation and analysis in order to arrive at a final mesh deemed suitable enough for accurate results.

Mesh generation tools are generally required to be both flexible and robust, leading to a wide variety of approaches. In industry, the majority of mesh generation strategies lead to a linear mesh, as this is what most simulation packages are capable of supporting. While widely adopted, the use of linear meshes could alter the flow physics in proximity of curved geometrical features, that in turn can lead to a critical reduction in simulation accuracy. The use of higher-order methods can address this loss of accuracy, as one generates boundary-conforming, curvilinear meshes which describe more accurately the underlying geometry for the purposes of simulation.

However, the generation of such meshes in a robust and efficient manner for a realistic three-dimensional geometry remains an open and challenging problem. In fact, the introduction of additional curvature within elements that are connected to the geometry – and that must conform to the boundary – will often lead to significant reduction in the quality of the mesh, or indeed deform the element to such an extent that it is self-intersecting and thus unsuitable for simulation usage. For simulations in fluid dynamics, where boundary layers of highly anisotropic elements are required to resolve the shear of the flow near the boundary, this issue becomes even more pronounced. Finding routes to mitigate and address these issues is the central challenge for the high-order mesh generation community, and forms a key bottleneck to tackle in the wider adoption of high-order methods. In this section, we outline our process to try to overcome this issue, and highlight the process within the context of realistic automotive geometries.

2.1. Approaches to high-order *a posteriori* mesh generation. Our framework to produce a viable high-order mesh, similar to other state-of-the-art methods [13, 74, 70, 32, 85], uses an *a posteriori* approach. In this setting, one first generates a coarse, linear mesh of the desired geometry. From this, high-order nodes are added within each element. Any nodes that lie on curved surfaces are projected onto the surface in such a manner as to enforce a boundary-conformal representation. In order to achieve the curvature required by the geometry being meshed, we typically adopt an isoparametric approach, in which the same basis functions that are used to represent high-order solutions within each element are also used to define the geometric coordinates within the element. Such a mapping can be expressed therefore as the summation

$$(2.1) \quad \mathbf{x} = \phi_M(\boldsymbol{\xi}) = \sum_{n=1}^N \mathbf{x}^n \ell_n(\boldsymbol{\xi}),$$

where \mathbf{x} is the spatial location of the a point within an element, $\boldsymbol{\xi}$ is a corresponding location within a reference element, \mathbf{x}^n is the location of a given node in the element, and ℓ_n is the polynomial interpolant (e.g. a Lagrange polynomial interpolant). In particular, mesh entities, namely faces and edges, on the boundary of the geometry are curved by adding high-order nodes, and then enforcing that the nodes lie on the boundary. Moreover, the position of the nodes \mathbf{x}^n must be adjusted so that the resulting mapping is a faithful polynomial representation of the geometry [74].

Although from this description this may seem a simple task, for complex geometries that can have several hundred curved surfaces, the ability to create valid high-order elements (that is elements of order greater than 2) is extremely challenging. In many cases, when simply curving the edge and/or face in proximity to a given curved geometrical feature, it is relatively easy to generate invalid elements. Figure 2 shows seven mesh elements in proximity of a curved surface, depicted in grey. The first mesh element (a) is a valid linear element that completely misses the curvature of the underlying surface. Indeed, in order to capture the curvature of the surface with a linear mesh it is necessary to add more elements around the surface than would be necessary with high-order elements; the latter benefits from a superior accuracy that allows for larger element sizes than linear elements. The second element (b) is a valid high-order element, where only the edge in proximity to the geometry is curved. The third and fourth elements, (c) and (d), are both invalid high-order elements, due to near tangent (c) or intersecting edges (d). Mesh elements (b), (c), and (d), clearly highlight the challenge of high-order mesh generation. The curvature of the geometry dictates the size of the element one can have, and in several occasions, it could lead to invalid elements. Mesh elements (e) and (f) show two possible solutions to the invalid elements (c) and (d). Mesh element (e) deforms the interior edges to accommodate the curvature of the boundary, while (f) uses a quadrilateral element to increase the permitted curvature. In both cases, there are no longer near tangent or intersecting edges. From the examples in figure 2, it should be immediately clear

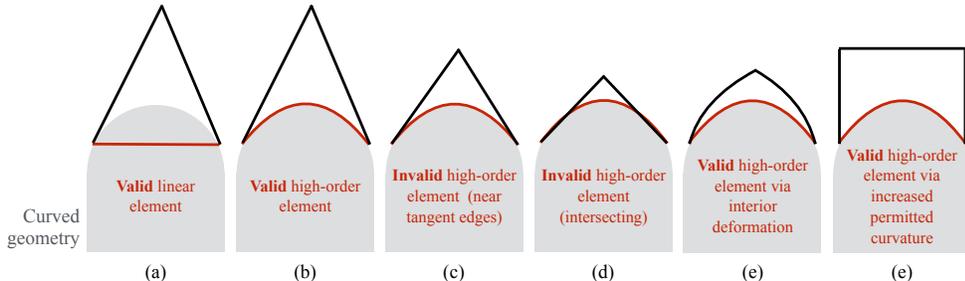


FIG. 2. *Valid and invalid high-order elements.*

that the challenge of high-order mesh generation is multifold. On the one hand, it is necessary to create curved mesh entities that conform to the underlying curved geometrical features (where the geometry is typically provided in a CAD format). On the other hand, it is essential to develop a strategy to handle invalid high-order elements and produce a valid high-quality mesh.

Our high-order meshing strategy handles the challenges just described and consists of three steps, (1) the definition of the geometry in such a way that it can be efficiently queried for curvilinear mesh generation, (2) the generation of a high-quality linear mesh, and (3) the generation of curved elements. These three crucial steps are described in detail in sections 2.2, 2.3 and 2.4, respectively. The corresponding meshing framework has been implemented in the code `NekMesh`, that is part of the open-source `Nektar++` library [5]. For a more comprehensive explanation of the mesh generation process, the interested reader can refer to Turner et al. [81].

2.2. Geometry definition. The first stage of our meshing approach is the definition of the geometry in a way that ensures its accurate representation and permits geometrical enquiries during the various mesh-generation stages. The most common

approach to define an arbitrary complex geometry is by a boundary representation (also referred to as “BRep”). The boundary representation contains surface patches and curves that link these surface patches. The surface patches and curves defining the underlying geometry are typically topologically connected, thereby forming a closed shell. These surfaces and curves are parametrised onto a lower-dimensional space, that is the one-dimensional space $\boldsymbol{x}(s)$ for curves, and the two-dimensional space $\boldsymbol{x}(s_1, s_2)$ for surfaces, where $\boldsymbol{x} = (x, y, z)$ denotes the three-dimensional coordinate of a point in the boundary representation of the geometry. The main geometrical enquiries required for mesh generation consists of three key operations:

- parametric coordinate to real space location, i.e. determining \boldsymbol{x} from (s_1, s_2) ;
- projection of a location in space onto a boundary representation entity, i.e. determining (s_1, s_2) from \boldsymbol{x} ; and
- calculation of the first- and second-order derivatives of the parametric mapping with respect to s_1 and s_2 .

In addition, in this work we do not consider the effects of incomplete or low-quality geometric representations; The boundary representation must be watertight and should represent accurately the geometry under investigation. These two requirements are crucial for high-order mesh generation, as they impact the number of high-order valid elements that is possible to generate and the quality of the mesh.

The assumption adopted in this work is that the geometry is given in a standard CAD (computer-aided design) format that supports a boundary representation description, such as STEP. In several CFD applications, this assumption is fairly common, especially among industry practitioners. The CAD object is then probed via `NekMesh`, that performs geometrical queries through interfaces to a CAD engine. To give support for different CAD engines and implementations, `NekMesh` implements a lightweight interface to the key operations above. Concrete implementations of this use `CADfix` [34] and `OpenCASCADE` [64], both of which permit reading CAD data in various formats, including STEP. In particular, the checks and queries on the CAD object were performed by `NekMesh`, while the correction of any errors was performed by `CADfix` which provides an extensive toolset for fixing geometrical inconsistencies in CAD data. `CADfix` is a commercial software, that provides extensive CAD handling tools, useful for healing and modifying the underlying CAD geometry definition. It also provides a linear mesh generator based on the medial approach that is useful for decomposing the domain into partitions. This decomposition has proven useful for designing high-quality boundary-layer meshes near the wall surfaces, aspect that was critical for the high-fidelity simulations in section 4. `OpenCASCADE` is an open-source code with reduced capabilities than `CADfix`, and represents the default CAD handler within `NekMesh`, due to its free-availability, that is at the core foundations of the `NekMesh` and `Nektar++` projects. Nevertheless, the use of `CADfix` in this work was essential to obtain a robust and efficient high-order meshing pipeline, and ultimately obtaining valid high-quality meshes. Access to `CADfix`, or some alternative approach to geometry healing and defeaturing, is a fundamental requirement for the development of a robust and successful meshing strategy.

Once the CAD representation of the geometry is satisfactory, it is possible to complete the definition of the computational domain, including the external boundaries, and proceed to the next meshing stage, that is generating a valid linear mesh. This step is described next.

2.3. Linear mesh. The generation of a high-quality linear mesh from the CAD representation of the geometry is a demanding task. Indeed, it is necessary to have a

sufficiently coarse initial mesh that serves as a starting point for the high-order meshing process. This initial mesh is typically obtained via a user-driven manipulation of the mesh spacing definition at all points within the computational domain to achieve satisfying geometric accuracy, coarseness and mesh gradation. This procedure is typically time-consuming and challenging, possibly involving a large number of iterations before obtaining a linear mesh that satisfies the quality required by the practitioner.

In order to automate this step as much as possible, we combined the advanced linear meshing capabilities of the commercial package **Star-CCM+** with the high-order meshing procedures of **NekMesh**. This allowed us to produce an automatic industrial linear meshing pipeline, that required minimal intervention from the user. The main obstacle that we encountered when using **Star-CCM+**, and that is common to encounter when using commercial CFD software, was the lack of information relating the linear mesh to the original CAD definition. Often the CAD surfaces are replaced by a fine triangulation that represents the “true” boundary of the geometry within a given tolerance. The linear mesh is then built using this approximate representation and the original CAD information discarded. This is a significant issue since inaccuracies in the definition of the geometry could lead to errors in the high-order mesh generation process, that in turn could lead to inaccurate flow simulation results.

In order to tackle this problem, we used the projection of the linear mesh onto the CAD surfaces to reconstruct the missing CAD information. In particular, we imported the linear mesh generated by **Star-CCM+** into **NekMesh**. The import was performed by incorporating curvature-driven mesh size restrictions such that the boundary of the surface mesh is a reasonably close representation of the underlying CAD model. Once the import of the linear mesh into **NekMesh** is complete, the CAD model is processed in order to obtain parametric coordinates in the CAD representation that will be used to enhance the adherence of the linear mesh to the CAD model. From this perspective, the first step is to linearise each surface of the CAD model via an “auxiliary” triangulation, so that each CAD surface has two representations, the CAD model itself, and the “auxiliary triangulation”. The second step is to define a bounding box (inflated by 5% in each direction), for each CAD surface, and store these boxes into a k -dimensional ($k-d$) tree data structure [1]. Once the two CAD representations are available and the bounding boxes are defined for each CAD surface, the surface mesh nodes are processed to obtain their CAD parametric coordinates. More specifically, we associate each node of the surface mesh that was generated in **Star-CCM+** to a list of potential CAD parent surfaces by querying the bounding box $k-d$ tree. If a node is within a bounding box, the surface is added to the list of potential parents surfaces. The use of the $k-d$ tree makes this process particularly efficient and reduces the number of potential parent surfaces to just a few candidates, typically on the order of 2 or 3. The surface mesh node is then projected onto the CAD surface for each of the parent surface candidates. This is achieved by identifying the nearest vertex in the “auxiliary triangulation” of the CAD surface, and then using this vertex as an initial guess for the nonlinear optimization problem associated to node-to-surface projection. The parent surface is finally identified as the closest surface to the projected node.

We note that, since the “auxiliary triangulation” of the CAD surfaces is not directly related to the **Star-CCM+** linear mesh, the surface node of the linear mesh might not lie on the CAD surface. Therefore, once the parent CAD surface associated to the mesh node is identified, the mesh node is moved to the CAD surface it belongs to, thereby improving the adherence of the linear mesh to the underlying CAD surfaces. The node movement just described is always performed, except in two cases: if the required displacement of the node is greater than 10% of the length of the edges in

the local mesh, and if the movement of the node induces inverted (i.e. invalid) elements. In these two instances, the node is added to an “exclusion set” with a view to preventing deterioration of mesh quality or generation of invalid elements. The final product of the projection process just described is a set of nodes that are accurately placed onto the original CAD surfaces, and a set of nodes that are placed into an “exclusion set”. Each node, regardless if in the “exclusion set” or not, will have all the parametric information required to generate the final high-order curving of the surface mesh, with the exception that any mesh entity (edge or face) that has a node in the “exclusion set” will be left linear. The approach outlined has been named *projection curving* [80], as it involves the projection of the surface mesh generated via **Star-CCM+** onto an “auxiliary triangulation” of the CAD surfaces. This approach is the one used in the final industrial meshing pipeline, that was adopted for the last design stage, D3, of the road car in section 4. A boundary-layer mesh was also created by **Star-CCM+**, and consisted of a prism layer in proximity of the surfaces of the geometry.

We finally point out that we initially attempted two additional alternatives, which were however discarded because they became too time-consuming and user-driven. These are named *analytic curving*, and were used for the first two designs, D1 and D2, of the road car presented in section 4. In both the *analytic curving* strategies, all the parametric information associated with the linear surface mesh was known, hence high-order curving of the surface was a relatively simple task, unlike in the *projection curving* approach. The first *analytic curving* strategy used the linear meshing capabilities of **NekMesh** to create the surface mesh that was later exported to **Star-CCM+** for the generation of the volume mesh, and of the boundary layer mesh (constituted by a near-wall macro prism layer). This approach was however extremely time-consuming as it involved dozens of iterations between **NekMesh**, **Star-CCM+**, and the original CAD model. The second *analytic curving* strategy tried to address the bottlenecks of the first by allowing the **Star-CCM+** linear mesh generator to create the surface mesh from a linearised CAD surface obtained using **NekMesh**. The surface mesh was then forced to explicitly conform to the boundaries of the CAD surfaces. This is critical in an industrial meshing pipeline, as it is typically impractical to clean complex CAD models with thousands of surfaces, so as to guarantee a clean enough CAD to enforce the surface mesh to obey to the CAD surfaces.

2.4. Generation of high-order elements. The third stage is the generation of the curved surfaces, where the coarse linear mesh generated in the previous step and described in section 2.3 is deformed to accommodate the curvature of the boundary. We already mentioned in section 2.3, that the creation of curved boundary mesh entities that conform to curved geometric features can be handled via two different strategies, *projection curving* and *analytic curving*, and we found the former to be the optimal approach for complex industrial CAD models. While curving the boundaries attached to the geometry is a relatively simple task once the parametric coordinates of the surface mesh are known and accurate, the creation of valid high-order elements remains a challenge, as depicted in figure 2. Indeed, one of the key research themes in *a posteriori* high-order mesh generation is to find suitable deformations of the interior mesh entities. The ability to deform interior mesh entities allows for better mesh quality and for a significant reduction of invalid elements, see figure 2-(e). To obtain suitable deformation of interior mesh entities, there exist a certain number of alternatives, including methods based on partial differential equations [65, 85, 60, 22, 30] and on the minimization of an energy functional [78, 24]. In this work we pursued

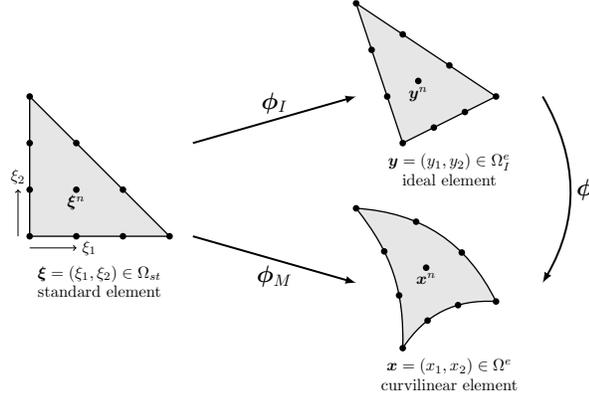


FIG. 3. Deformation of a straight-sided triangular element, referred to as standard element, onto an ideal element via mapping $\phi_I : \Omega_{st} \rightarrow \Omega_I^e$, and onto a curvilinear element via mapping $\phi_M : \Omega_{st} \rightarrow \Omega^e$. The mapping of the deformation ϕ from the ideal element and the curvilinear element is defined as the composition of the two mappings, $\phi : \phi_M \circ \phi_I^{-1}$. Figure taken from Turner et al. [81]

the latter, and developed a variational framework able to encapsulate arbitrary energy functionals in a unified high-order meshing framework that has been implemented in **NekMesh**.

The variational formulation begins with a coarse linear mesh that was created as described in section 2.3. Following [80], we consider a linear mesh with N_{el} straight-sided elements, that is $\Omega_I = \bigcup_{e=1}^{N_{el}} \Omega_I^e$, where each element is provided with a high-order polynomial basis function based on standard Lagrange polynomials. The mapping between the straight-sided mesh, Ω_I , and the curvilinear mesh, Ω , is denoted by $\phi : \Omega_I \rightarrow \Omega$.

With reference to figure 3, we denote the coordinates of the standard (also referred to as reference) element as $\xi \in \Omega_{st}$, those of the curvilinear element as $x \in \Omega^e$, and those of the ideal element as $y \in \Omega_I^e$. The mappings between the elements in figure 3 are constructed in an isoparametric fashion, such that the nodes ξ^n defining the Lagrange basis functions on the standard element map to y^n under ϕ_I and to x^n under ϕ_M . The example reported in figure 3 is for triangular elements but it can be readily generalized to any element shape, in both two and three dimensions.

Once the nomenclature for the mesh is specified, we can define the energy functional adopted throughout this work, that is

$$(2.2) \quad \mathcal{E}(\nabla\phi) = \int_{\Omega_I} W(\nabla\phi) d\mathbf{y},$$

where W depends on the deformation gradient tensor $\nabla\phi(\mathbf{y}) = \frac{\partial\phi}{\partial\mathbf{y}}$ and its determinant $J = \det[\nabla\phi]$, also referred to as the Jacobian. The functional W can assume various forms depending on the associated energy adopted. We have implemented functionals associated to linear elasticity energy, isotropic hyperelasticity energy, Winslow equation energy, and a distortion-based energy. Details of these functionals can be found in Turner et al. [81] and in Turner [80], and are not reported here for the sake of brevity.

Since the ideal mesh is a union of many elements, the functional can subsequently

be defined element by element

$$(2.3) \quad \mathcal{E}(\nabla\phi) = \sum_{e=1}^{N_{el}} \int_{\Omega_I^e} W(\nabla\phi) d\mathbf{y} = \sum_{e=1}^{N_{el}} W[\nabla\phi_M(\boldsymbol{\xi})\nabla\phi_I^{-1}(\phi_I(\boldsymbol{\xi}))] \det(\nabla\phi_I) d\boldsymbol{\xi},$$

where we used the composition $\phi = \phi_M \circ \phi_I^{-1}$. To fully define equation (2.3), we need to specify the expression of ϕ_M and ϕ_I . The first, ϕ_M , assumes the usual form of an isoparametric mapping commonly found in nodal spectral element methods and specified in equation (2.1). The second, ϕ_I , assumes an analytical form that is a combination of linear finite element basis functions and that depends on the element shape. For instance, in the case of a triangle with vertices \mathbf{v}^1 , \mathbf{v}^2 , and \mathbf{v}^3 , the mapping is given by

$$(2.4) \quad \phi_I = (\mathbf{v}^2 - \mathbf{v}^1)\xi_1 + (\mathbf{v}^3 - \mathbf{v}^1)\xi_2 + \mathbf{v}^1.$$

The minimization of the functional is then formally defined as follows

$$(2.5) \quad \text{find } \min_{\phi} \mathcal{E}(\phi) = \min_{\phi} \int_{\Omega_I} W(\nabla\phi) d\mathbf{y},$$

where we seek the configuration of nodes of the linear mesh, denoted by $\mathbf{N} = [\mathbf{n}^1, \dots, \mathbf{n}_{\text{nodes}}^N]$, that minimizes the energy functional in equation (2.5). In practice, this means to start from the initial distribution of nodes in the linear mesh, \mathbf{N}^0 , in which the curvature of the boundary mesh entities is imposed, regardless of whether it causes self-intersections. Then, the minimization proceeds in iteratively updating \mathbf{N} towards lower energy configurations, that is: $\mathbf{N}^k \rightarrow \mathbf{N}^{k+1}$, such that $\mathcal{E}^k \rightarrow \mathcal{E}^{k+1} < \mathcal{E}^k$. The minimization is achieved via a gradient- and Hessian-based method, and utilizes a relaxation strategy, whereby a local optimization problem is solved in place of a global one. More specifically, the minimization in equation (2.5) is performed for a subset of elements that are influenced by a change in the position of the node under consideration. This requires the solution of a smaller and less computationally expensive problems, compared to the global problem that would be otherwise necessary to solve. The minimization of equation (2.5) can then be recast into the minimization of a set of energies, each of which belongs to a node and the subset of elements the node influences. The stopping criterion used to find the global minimum based on the set of local energies, is $\|\mathbf{N}^{k+1} - \mathbf{N}^k\|_{\infty}/L < \epsilon$, where we used $\epsilon = 10^{-6}$, and $\|s\|_{\infty} = \max\{|s_1|, \dots, |s_n|\}$ is the uniform norm.

This process also requires the computation of the gradient of $\mathcal{E}(\phi)$, with respect to the movement of node positions. As a first approach, these gradients can be approximated by use of finite difference stencils. However, our testing indicates that this approach is both expensive and sensitive to the size of the stencil. Instead, we make use of an analytic expression for the desired gradients. Ultimately, applications of the chain rule therefore require the same derivatives for ϕ_I and ϕ_M . Although the ideal mapping ϕ_I readily admits an analytic expression, more care is required for ϕ_M , but by exploiting properties of the Lagrangian expansion (2.1), analytic forms of the derivatives can be obtained, which are given in the appendix of [81].

Figure 4 shows the high-order surface mesh of the Rp1 car, resulting from our meshing process. It is possible to appreciate the high-order elements tessellating the CAD surface. We note that the spokes within the wheel shown in figure 4 were not included in the simulations presented in section 4 to avoid the requirement for a moving mesh.

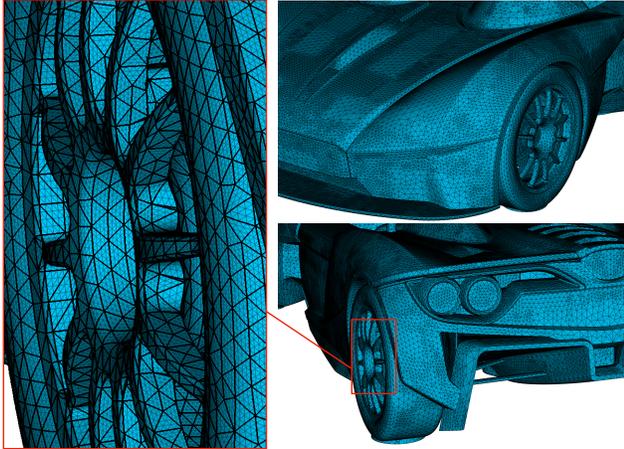


FIG. 4. Details of the $Rp1$ high-order mesh produced using the high-order meshing workflow presented in section 2. We note that the spokes in the wheel were not included in the final simulations to reduce complexity.

As a final remark, we point out that, all those nodes on the boundary surface mesh that belonged to the “exclusion set” are left straight-sided. This leads to a small but not insignificant proportion of the surface mesh entities that will be straight-sided as opposed to curved, and could have significant impact on the geometric accuracy of the mesh and hence the solution. However, in the results that we present in section 4, we observe little noticeable impact, since often these excluded elements were in locations where multiple CAD surfaces met but not typically in places of interest to the flow physics. Obviously this cannot be always guaranteed to be true. A full investigation into this has not been conducted yet, but this work proves that high-order meshes can be produced on industrial models and sensible results obtained with little meshing effort. Indeed the final projection meshing pipeline only required one execution of the linear mesh generation and `NekMesh`, with no need for repetition with either system or the CAD.

3. Numerical methodology. The model underlying the simulations carried out in this study is constituted by the incompressible Navier-Stokes equations for a Newtonian fluid

$$(3.1a) \quad \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}$$

$$(3.1b) \quad \nabla \cdot \mathbf{u} = 0,$$

where $\mathbf{u} = (u, v, w)$ is the velocity vector, p denotes the pressure, ν is the kinematic viscosity, and \mathbf{f} is a generic forcing term. This set of equations are particularly suited to accurately model the flow around a road car, as compressibility effects are deemed to be unimportant at the speeds reached. For diffusers and other heated parts of the car, the compressible Navier-Stokes equations are better suited. This was however out of the scope of the current investigation, which was mainly focussed on the external aerodynamics of the car.

The system of equations (3.1) and its numerical discretization has been implemented in the open-source spectral/ hp element library `Nektar++`. The equations are

solved via an operator splitting scheme (sometimes referred to as a velocity correction scheme) [36, 29] in combination with a consistent boundary condition for the pressure Poisson equation. In particular, following [29], we recast problem (3.1) into a weak pressure Poisson problem by taking the inner product over the solution domain Ω of equation (3.1a) with respect to the gradient of the test function, ∇v , i.e.

$$(3.2) \quad \int_{\Omega} \nabla v \cdot \frac{\partial \mathbf{u}}{\partial t} + \int_{\Omega} \nabla v \cdot \mathcal{N}(\mathbf{u}) = - \int_{\Omega} \nabla v \cdot \nabla p + \int_{\Omega} \nabla v \cdot \nu \nabla^2 \mathbf{u},$$

where $\mathcal{N}(\mathbf{u}) = \mathbf{u} \cdot \nabla \mathbf{u}$ and we set the forcing term \mathbf{f} to zero. Using the identity $\nabla^2 \mathbf{u} = -\nabla \times \nabla \times \mathbf{u} + \nabla(\nabla \cdot \mathbf{u})$, we can enforce the divergence to be zero by setting the last term to zero. By integrating the first, second and last term in equation (3.2) by parts we get the weak pressure equation

$$(3.3) \quad \begin{aligned} \int_{\Omega} \nabla v \cdot \nabla p^{n+1} &= \int_{\Omega} v \nabla \cdot \left(\frac{\partial \mathbf{u}^{n+1}}{\partial t} + \mathcal{N}(\mathbf{u})^{n+1} \right) \\ &\quad - \int_{\partial \Omega} v \left(\frac{\partial \mathbf{u}^{n+1}}{\partial t} + \mathcal{N}(\mathbf{u})^{n+1} + \nu \nabla \times \nabla \times \mathbf{u}^{n+1} \right) \cdot \mathbf{n}, \end{aligned}$$

where $\partial \Omega$ is the boundary of the domain, \mathbf{n} indicates the outward pointing normal to the domain Ω and the superscript $(n+1)$ denotes the time step that aim to compute. Note that in equation (3.3) we have also used the identity $\nabla \cdot (\nabla \times \nabla \times \mathbf{u}) = 0$. By using a backward approximation of the time derivative,

$$(3.4) \quad \frac{\partial \mathbf{u}^{n+1}}{\partial t} \simeq \frac{\gamma_0 \tilde{\mathbf{u}}^{n+1} - \hat{\mathbf{u}}}{\Delta t}$$

along with a consistent extrapolation for the non-linear terms $\mathcal{N}^{*,n+1}$, where $\tilde{\mathbf{u}}^{n+1}$ is an intermediate velocity, γ_0 is a constant, and $\hat{\mathbf{u}}$ is an extrapolation of the velocity, we can finally calculate the pressure at step $n+1$. We can then solve for the velocity at step $n+1$ by using the pressure at $n+1$ just calculated. This leads to solving the following Helmholtz problem

$$(3.5) \quad \left(\Delta - \frac{\gamma_0}{\nu \Delta t} \right) \mathbf{u}^{n+1} = - \left(\frac{\gamma_0}{\nu \Delta t} \right) \hat{\mathbf{u}} + \frac{1}{\nu} \nabla p^{n+1}.$$

The velocity correction scheme just outlined (equations (3.3),(3.5)) is spatially discretized using a high-order continuous Galerkin formulation [38]. Here, one typically expresses the solution in terms of global basis functions $\Phi_k(\mathbf{x})$ as follows

$$(3.6) \quad \mathbf{u} \approx \mathbf{u}_h = \sum_k \mathbf{u}_k(t) \Phi_k(\mathbf{x}),$$

where \mathbf{u}_k are the weights or coefficients of the polynomial expansion, and we introduced the superscript δ to denote the spatial discretization of \mathbf{u} . Equation (3.6) can be decomposed into local boundary and interior function, such that equations(3.3) and (3.5) can be written in terms of element-wise contributions (omitted here for the sake of brevity). The elemental polynomial expansions can be written as

$$(3.7) \quad \mathbf{u}_{\Omega_e} \approx \mathbf{u}_{\Omega_e}^{\delta} = \sum_{j=0}^P \ell_k(t) \phi_k(\mathbf{x}),$$

where the elemental basis functions $\phi_k(\mathbf{x})$ used are *modified Jacobi* or *Lagrange* polynomials defined on a set of Gauss-Lobatto-Legendre quadrature points, ℓ_k are the elemental coefficients of the expansion, and we introduced the element Ω_e of our domain discretization. The global-wise continuity of the solution required by the continuous Galerkin method is then enforced by an assembly procedure. The interested reader can refer to [36, 29] as well as to [38] (Chapter 4 and 8). The semi-discrete form of the incompressible Navier-Stokes equations that is obtained using the velocity correction scheme and the continuous Galerkin method as spatial discretization and that was just introduced is advanced in time via a second-order implicit-explicit time-stepping strategy [83].

In order to successfully achieve the challenging high-Reynolds number LES described in section 4, we needed to significantly enhance the numerics implemented in *Nektar++*, especially in terms of (i) numerical stability, (ii) regularization techniques for under-resolved scales, and (iii) computational efficiency. These three key building blocks are discussed in the following.

3.1. Numerical stability. Numerical stability was a key aspect that we needed to carefully take into account in order to successfully simulate the road car presented in section 4, at the Reynolds numbers required by industry. To achieve this, we adopted polynomial dealiasing via consistent integration [48], combined with spectral vanishing viscosity (SVV) [35, 40, 50]. The former helps suppressing aliasing-driven instabilities that are due to inexact numerical integration of the equations, while the latter targets all those flow scales that are necessarily under-resolved in high-Reynolds number simulations. While SVV primary target was not the numerical stability of the simulations, it was necessary the use of both, dealiasing and SVV, to achieve stable (i.e. non-crashing) computations. More details on the SVV adopted are reported in section 3.2, where we discuss its use to accurately treat under-resolved scales. In this section, we focus on polynomial dealiasing that was a key component without which it was otherwise not possible to obtain stable simulations.

Indeed, the main issue when dealing with element-wise integrations typical of spectral/*hp* methods, is the inexact numerical integration of the discrete terms of the equations. In fact, it is well known that there is a minimum number of quadrature points Q_{\min} , that is required to produce exact results for integrals of non-rational polynomials of any given degree. This minimum number of quadrature points depends on the type of numerical quadrature adopted and on the set of quadrature points used. In our case, we employed Gaussian quadrature on a set of Gauss-Lobatto-Legendre quadrature points [38]. This type of numerical integration guarantees that the minimum number of quadrature points Q_{\min} necessary to exactly integrate any given order P polynomial $u(\xi) \in \mathcal{P}_P$ up to machine precision is $Q_{\min} = (P + 3)/2$, where \mathcal{P} is the space of polynomials. It is then possible to calculate the number of quadrature points necessary to exactly integrate different nonlinearities that might arise in the equations, as reported in table 1. We can observe how Q_{\min} increases as the degree of the nonlinearity increases.

Polynomial order P	Q_{\min}
$[u(\xi)]^2 \in \mathcal{P}_{2P}$	$Q \geq P + 3/2$
$[u(\xi)]^3 \in \mathcal{P}_{3P}$	$Q \geq 3P/2 + 3/2$
$[u(\xi)]^4 \in \mathcal{P}_{4P}$	$Q \geq 2P + 3/2$

TABLE 1

Number of GLL quadrature points for the GLL quadrature to be exact up to machine precision as a function of the polynomial order of the integrand

Let us now consider the incompressible Navier-Stokes equations (3.1), and split them into linear and nonlinear terms as follows:

$$(3.8) \quad \frac{d\mathbf{u}^\delta}{dt} = \mathbf{M}^{-1} (\mathcal{L} + \mathcal{N}) ,$$

where \mathbf{M} is the mass matrix that arises from the numerical discretization, $\mathcal{L} = -\nabla p^\delta + \nu \nabla^2 \mathbf{u}^\delta + \mathbf{f}^\delta$ contains the linear terms, $\mathcal{N} = -\mathbf{u}^\delta \cdot \nabla \mathbf{u}^\delta$ contains the convective term, and superscript δ denotes the spatial discretization. By noting that the convective term is a quadratic nonlinearity, we can identify the Q_{\min} required for exact integration. In particular, given that the spatial discretization we employ is a continuous Galerkin method, we are interested in the integration of L^2 inner products of two polynomials (u_p, u_q). On the one hand, to calculate the linear terms, it is necessary to calculate $[u(\xi)]^2 \in \mathcal{P}_{2P}$ (i.e. the inner product of the solution with respect to the polynomial basis). Therefore, for exact integration, we need $Q_{\min} = P + 2$ Gauss-Lobatto-Legendre quadrature points in one-dimension and a tensor product of points in higher dimensions. On the other hand, to calculate the nonlinear terms, we wish to calculate $[u(\xi)]^3 \in \mathcal{P}_{3P}$, that corresponds to $Q_{\min} = \frac{3}{2}(P + 1)$ Gauss-Lobatto-Legendre quadrature points. We note that using other sets of quadrature points, Q_{\min} changes. For instance, the use of Gauss-Legendre points guarantees exact integration up to $Q_{\min} = P + 1/2$, for any given order P polynomial $u(\xi) \in \mathcal{P}_P$. While it is typically possible to exactly integrate the nonlinearities \mathcal{N} present in the equations when dealing with mesh elements that are straight-sided parallelograms, it is more complicated if not impossible when dealing with curved mesh elements. In fact, in the case of straight side parallelograms, the Jacobian of the mapping that defines the coordinates of the element is affine with constant determinant. For curved elements however, this mapping is an isoparametric polynomial expansion, which when incorporated into integrands leads to an additional source of aliasing error, called geometrical aliasing [48]. Geometrical aliasing enters into the discrete formulation via the inverse of the mass matrix \mathbf{M}^{-1} , as well as from both the linear and nonlinear terms in equation (3.8), and it is typically in the form of a rational polynomial. It is well known that rational polynomials cannot be exactly integrated within the functional space used to approximate the problem. Therefore, numerical stability cannot be formally guaranteed. We also note that geometrical dealiasing could couple with the underlying equation nonlinearities, thereby exacerbating numerical instabilities. While, for problems that employ unstructured meshes with curvilinear elements like the ones described in this paper, numerical stability cannot be guaranteed, it can be substantially improved by adopting dealiasing strategies that mitigate aliasing-driven instabilities.

The dealiasing approach adopted in this work is the so-called *Global* dealiasing [48], that employs a larger and identical number of quadrature points on each term

of the numerical discretization (3.8) of (3.1), that is

$$(3.9) \quad \frac{d\mathbf{u}^\delta}{dt} = \mathbf{M}^{-1} \Big|_Q \left(\mathcal{L} \Big|_Q + \mathcal{N} \Big|_Q \right),$$

The subscript Q in each term of equation (3.9) denotes a higher number of quadrature points than required for exact integration of quadratic linear terms, that is $Q > P + 3/2$.

This dealiasing strategy mitigates aliasing-driven instabilities induced by equation nonlinearities, curved mesh elements, and the coupling between equation nonlinearities and curved mesh elements. Exact integration is however not guaranteed, except for an asymptotically infinite number of quadrature points. Hence, marginal aliasing errors can still be present and contribute to numerical instabilities. In this work, we always used a quadrature order consistent with the nonlinearities of the governing equations, often with a safety margin to account for curved elements, that is $Q > 3P/2 + 3/2$. We note that for simulations that are adequately resolved (e.g. laminar flows), aliasing usually does not affect numerical robustness. However, for under-resolved turbulence computations, this aliasing effect leads to a significant build-up of error in high-frequency solution modes and usually causes the simulation to abruptly diverge.

There are additional tools to suppress aliasing-driven instabilities, that have not been explored in this work. These include the use of skew-symmetric forms of the convective term of the underlying equations [2, 25, 84], polynomial filtering [27, 19, 21, 31], variational multiscale schemes [33, 46], and exponential based modal filtering [26]. As mentioned above, we note that SVV could, in principle, also be used to suppress aliasing-driven instabilities [35]. However, the diffusion required would be significantly high, and could suppress important flow features, thereby undermining the accuracy of the simulation. Additionally, because the SVV operator is relatively isotropic, whereas dealiasing is anisotropic, the two do not completely overlap. Hence, a reduced amount of regularisation can be achieved with dealiasing, thereby better preserving the solution accuracy.

The use of the dealiasing techniques described in this section removes numerical integration errors that could lead to numerical instabilities in under-resolved simulations; however, it does not address the under-resolved scales that are ubiquitous in high-Reynolds number simulations. These under-resolved scales, similarly to aliasing errors, can be one of the main drivers of numerical instabilities and solution inaccuracies, and they exacerbate aliasing errors leading to the possible failure of the simulation. The handling of under-resolved scales to prevent numerical instabilities and maintain solution accuracy is described next.

3.2. Regularization for under-resolved scales. In order to treat under-resolved scales commonly present in high-Reynolds number simulations, such as the one in section 4, we use a regularization technique based on a novel SVV operator. This new operator has been carefully designed to incorporate insights obtained from recent dispersion/diffusion eigenanalysis [57, 49, 47, 50], and results in well-balanced diffusion characteristics that allow for the accurate simulation of high Reynolds number flow problems. Indeed, when stabilising a high-order formulation, such as the continuous Galerkin scheme adopted in this study, it is critical to achieve an adequate balance between numerical diffusion and solution accuracy, especially for turbulent flow computations.

The general idea behind SVV is that its diffusion can be designed to affect mainly

the element-wise polynomial modes of highest order within the spectral/ hp element discretization. These polynomial modes comprise the most oscillatory part of the solution and represent the smallest turbulent scales captured by the degrees of freedom employed. The SVV technique was first proposed by Tadmor [76] as a stabilisation strategy for classical spectral methods in Fourier space. Essentially, if the solution in Fourier space is

$$(3.10) \quad \mathbf{u}(x, t) = \sum_k \hat{\mathbf{u}}_k \exp(ikx),$$

then SVV amounts to an artificial diffusion term added to the governing equations in the form

$$(3.11) \quad \nu_{svv} \nabla(\mathcal{Q} \star \nabla \mathbf{u}) = -\nu_{svv} \sum_{\kappa} \kappa^2 \hat{\mathcal{Q}}_{\kappa} \hat{\mathbf{u}}_{\kappa} \exp(i\kappa x),$$

where ν_{svv} is a constant representing SVV's overall diffusion magnitude, κ is the wavenumber, $\hat{\mathbf{u}}_{\kappa}$ are the Fourier coefficients of the solution, $\hat{\mathcal{Q}}_{\kappa}$ are the Fourier coefficients of the SVV kernel that define how diffusion is to be distributed across Fourier modes, and $i = \sqrt{-1}$. We note that by setting $\hat{\mathcal{Q}}_{\kappa} \equiv 1$, the standard diffusion (Laplacian) operator is formally recovered. Traditional SVV kernel functions grow monotonically from zero, achieving $\hat{\mathcal{Q}}_{\kappa} = 1$ only at the highest order mode. For that solution component, the physical kinematic viscosity ν effectively becomes $\nu + \nu_{svv}$.

Since Tadmor's original work on spectral methods, SVV has been adapted to continuous spectral/ hp element methods and used in particular to stabilise under-resolved turbulence computations based on the continuous Galerkin method [35, 40, 41, 43, 71, ?]. In this approach the filtering properties of the SVV operator applied to the continuous Galerkin (also referred to as CG hereafter) method are used in lieu of an explicit turbulence model. The use of techniques originally designed to promote numerical stability in place of a turbulence model is already a common practice, for example, in model-free turbulence simulations via discontinuous Galerkin (or DG, hereafter) methods, where upwind-type dissipation is introduced through the Riemann fluxes employed in compressible formulations [12, 82, 54]. This setting, where artificial numerical features are used in place of an explicit turbulence model, is commonly referred to as implicit LES, and it has been investigated in the past few decades with various degrees of success [45, 56, 9, 10, 52].

The key potential issues when modeling under-resolved scales using the diffusion properties of the underlying numerical discretization or through an SVV-like operator are lack of robustness and/or accuracy of the resulting implicit LES framework, and unclear connections between the artificial numerical filters and the flow physics. In this work, we attempted to address these challenges by encapsulating novel results from the so-called eigensolution analysis (ESA) mentioned previously [57, 49, 47, 50] to build a new SVV kernel. In this context, we sought for solutions in which the element-wise coefficients arising from the numerical discretization are related to wave-like solutions of the form

$$(3.12) \quad \mathbf{u} \propto \exp[i(\kappa x - \omega t)]$$

through projection (see also [51] (Chapter 2)).

Initially, we used numerical dispersion and diffusion estimates for spectral/ hp element methods in wavenumber/frequency space provided by the so-called *temporal* ESA [57] to construct an SVV kernel \mathcal{Q} that enforces SVV diffusion to be distributed

monotonically across wavenumber space for the CG method. We note in fact that monotonic behavior in wavenumber space is not always guaranteed due the lack of one-to-one correspondence between Fourier and polynomial modes¹. This resulted in what we called “power-law” CG-SVV kernel

$$(3.13) \quad \tilde{Q}(p) = \left(\frac{p}{P}\right)^{P_{svv}}, \quad 0 \leq \frac{p}{P} \leq 1,$$

where p is the polynomial mode index, P is the overall polynomial order, and P_{svv} is a user-defined parameter that, when increased, further confines CG-SVV damping to the smaller scales. This kernel shape is motivated by hyper-viscosity operators [cite the paper that the reviewer mentions by Avrin and Chang]. We note that the operator $\tilde{Q}(p)$ in equation (3.13) is formally different from the kernel defined in equation (3.11), as this arises from the projection of the element-wise coefficients in equation (3.7) onto the wave-like solution in equation (3.12). This kernel also guarantees that raising P increases the resolution power per degree of freedom which is something expected from high-order methods, as discussed in detail in [57]. We highlight that this first study was based on the temporal ESA framework, which is relevant to temporally evolving simulations instead of spatially evolving ones, where inflow and outflow boundary conditions are typically used. While the “power-law” kernel defined in equation (3.13) solved a number of issues, still some potentially undesirable characteristics were present. These undesirable aspects were two-fold. On the one hand, non-smooth behavior of the diffusion and dispersion curves was observed for either too small or too large values of ν_{svv} . This could lead to unphysical representation of the flow physics. On the other hand, the use of the *temporal* ESA framework did not include spatially evolving flow problems (i.e. problem with inflow/outflow boundary conditions) that are commonly found in external aerodynamics. This could trigger spurious reflections of certain waves when they hit regions of mesh expansion (e.g. along turbulent wakes), eventually leading to numerical instabilities and divergence of the simulation [49, 47].

In order to address the first aspect, we focused on the relevant non-dimensional parameter to quantify SVV’s relative intensity is the so-called Péclet number, namely $Pe = vh/\nu_{svv}$, that is based on measures of local velocity v and mesh spacing h . The adopted strategy was to maintain the Péclet number fixed by making $\nu_{svv} \propto vh$. This led to a CG-SVV scaling that is dimensionally equivalent to the upwind-based dissipation of discontinuous spectral/ hp element methods, as explained in [57].

To address the second aspect, we pursued a complementary study [50] focused on the so-called *spatial* ESA, which assumes inflow-/outflow-type boundary conditions and is therefore more relevant to spatially developing problems, as the one encountered in this paper. *Spatial* ESA had initially been conducted for discontinuous spectral/ hp element methods [49, 47] and proved insightful in explaining how spurious reflections occurred in regions of mesh expansion depending on the Riemann flux employed. In this framework, theory prescribes that an incoming wave of frequency ω should be associated with a single (and real-valued) wavenumber $\kappa = \omega/a$, a being the advection velocity. However, in a numerical setting, two complex-valued wavenumbers, κ^p (physical) and κ^u (unphysical), appear in the analysis. Regarding their imaginary part, when numerical diffusion is present, one observes $\Im(\kappa^p) > 0$, which indicates damping of the physical mode during forward propagation, since the solution can be

¹Note that common practice was to directly apply Fourier-type kernel entries \hat{Q}_κ upon the orthogonal hierarchical set of element-wise polynomial modes.

written as

$$(3.14) \quad \mathbf{u} \propto \exp[i(\kappa x - \omega t)] = \exp[-\Im(\kappa) x] \exp\{i[\Re(\kappa) x - \omega t]\},$$

where \Im and \Re correspond to the imaginary and real part, respectively. For the unphysical reflected mode, it is instead necessary that $\Im(\kappa^u)$ be negative, indicating damping during backward propagation ($\Delta x < 0$).

When applied to the CG method in [50], spatial ESA revealed that unless special care is taken during SVV design, unphysical reflected waves can have negligibly small (although negative) values of $\Im(\kappa^u)$. This allows for spurious waves to survive for a long time while interacting with incoming turbulence, eventually causing numerical instability. Hence, a new optimisation procedure was performed for the CG-SVV operator in order to match the diffusion levels of CG to those of standard upwind DG (that uses a Roe-type Riemann solver, in the context of compressible flow problems) at appropriate polynomial orders, this time with a constraint to enforce a sufficiently strong diffusion for the reflected eigenmodes. DG’s numerical diffusion was taken as a reference because DG is regarded as having an adequate balance between dissipation and accuracy. The resulting dispersion and diffusion plots are shown in figure 5 for various orders, with the curves for the spurious modes at the bottom part of the graphs. Once again, the newly designed kernel for the CG-SVV operator, that we named “DG” kernel given its connections to the DG method in terms of diffusion and dispersion characteristics, maintains the Péclet number fixed and guarantees that raising P increases the resolution power per degree of freedom.

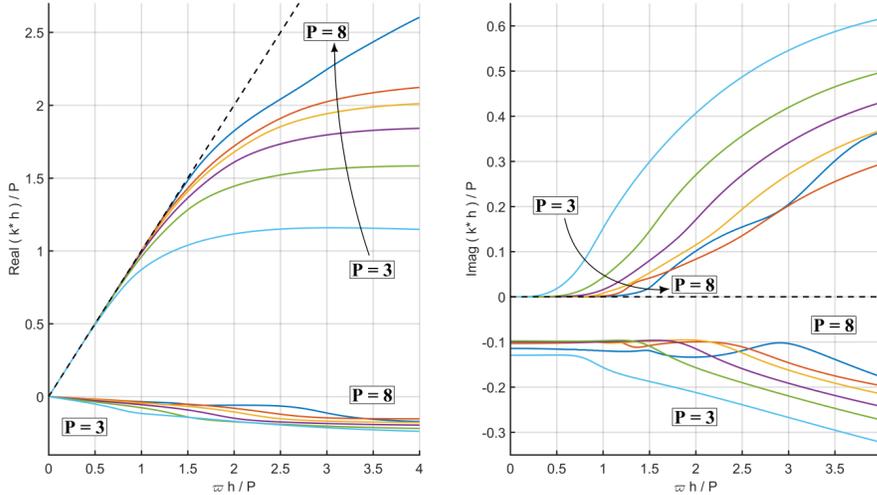


FIG. 5. Numerical dispersion (left) and diffusion (right) plots for CG with the “DG kernel” SVV for various polynomial orders. The unphysical eigenmodes appear on the bottom curves of the graphs. Figure taken from Moura et al. [50].

In order to validate the novel CG-SVV operator based on the “DG” kernel, we used a simple two-dimensional test case that was designed to mimic spatially developing grid turbulence [49, 50]. It consists of a rectangular domain with prescribed unsteady flow at the inlet and symmetry (free-slip) conditions on the sides. This domain was meshed with quadrilaterals in a structured grid pattern featuring two regions. The upstream region had square-shaped elements, whereas the downstream region had a four times larger streamwise mesh spacing. The interface between these

two regions marked a potentially unstable threshold zone across which mesh spacing changed abruptly. Results obtained without SVV for $P = 8$ and $n = 11$ elements in the crossflow direction are shown in figure 6. In particular, the vorticity contour plot obtained without SVV, upon close inspection of the upstream region to the mesh coarsening line, shows the presence of spurious reflections originating at the threshold zone. This case diverged shortly afterwards, along with other test cases where we used different polynomial orders, ranging from $P = 3$, to $P = 7$ (while maintaining the same number of degrees of freedom – i.e. decreasing the number of elements). In particular, at lower polynomial orders, higher Reynolds numbers were required for the simulation to diverge, while for higher polynomial orders, the test cases started to diverge at correspondingly lower Reynolds numbers. When the novel DG-based CG-SVV operator was added, however, up to polynomial order $P = 8$, all cases remained stable no matter how high the Reynolds number tested. Furthermore, no spurious reflections have been noticed. And although solutions at lower orders ($P = 3$ in particular) might have seemed excessively regularised by the “DG” kernel SVV, superior resolution power was effectively recovered at higher orders, such as $P = 4$ and above [53].

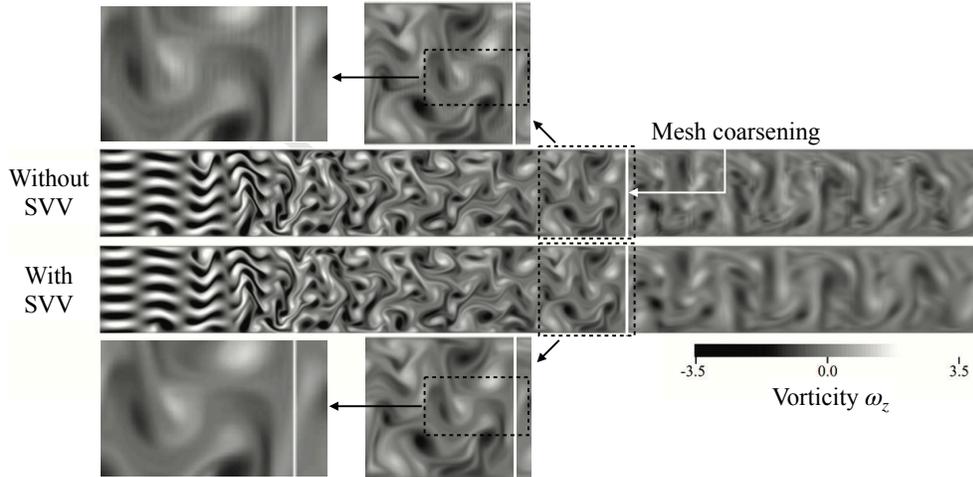


FIG. 6. Vorticity contours of case $P = 8$, $n = 11$, at $Re = 1000$ with (bottom plot) and without (top plot) the novel CG-SVV operator in presence of streamwise mesh coarsening. Close examination shows how the novel CG-SVV kernel is able to suppress spurious reflections originating where the mesh changes resolution. Figure readapted from Moura et al. [50].

The current best practice in *Nektar* has been to use the “DG” kernel in general while employing the “power-law” kernel only along homogeneous directions in case periodic boundary conditions are present. In either case, SVV’s base coefficient takes the final form

$$(3.15) \quad \nu_{svv} = (Pe^*)^{-1} v h / P ,$$

where Pe^* is a reference constant Peclet number (typically of order one), v and h are respectively local measures of advection velocity and mesh size (along the relevant directions of flow propagation), and P is the chosen polynomial order. The entries defining the “power-law” kernel are those given in (3.13), whereas those of the “DG” kernel have no explicit formula (as the output of an optimisation procedure), having

their values given in table form for each order P in [50]. Furthermore, we note that SVV’s kernel operation from (3.11) is accommodated in polynomial space as

$$(3.16) \quad \mathcal{Q} \star \nabla \mathbf{u} \approx \mathbf{R}^{-1} \mathbf{Q} \mathbf{R} \hat{\mathbf{g}} ,$$

in which $\hat{\mathbf{g}}$ denotes the vector of element-wise coefficients in the polynomial expansion of $\nabla \mathbf{u}$, while \mathbf{R} is a “rotation” matrix that projects $\hat{\mathbf{g}}$ onto an orthogonal (Legendre) hierarchical basis, and \mathbf{Q} is the diagonal matrix whose entries are given by the chosen SVV kernel function. Therefore, the proposed CG-SVV operator has the final form

$$(3.17) \quad \nu_{svv} \nabla (\mathcal{Q} \star \nabla \mathbf{u}) \approx (\text{Pe}^*)^{-1} \frac{\nu h}{P} \nabla (\mathbf{R}^{-1} \mathbf{Q} \mathbf{R} \hat{\mathbf{g}}) .$$

This CG-SVV operator has been implemented in `Nektar++` and extensively tested in different test cases of increasing complexity, always displaying strong robustness and preventing numerical instabilities. This approach offers a decent balance between robustness and accuracy for high-order discretizations, being effectively a parameter-free strategy from the practitioners’s perspective and allowing for under-resolved turbulence simulations over complex geometries at high Reynolds numbers. This approach, in conjunction with the dealiasing techniques described in section 3.1, was used to obtain the numerical simulations depicted in section 4. Without the concurrent use of dealiasing and SVV regularization, the numerical simulations presented in this paper would have diverged due to numerical instabilities.

3.3. Computational efficiency. One of the advantages of higher order polynomial expansions is their ability to exploit high arithmetic intensities; that is, a greater number of floating point operations is performed for each byte of data retrieved from memory at higher orders than at lower orders. Although this naturally makes operator evaluations at higher order more expensive per degree of freedom, on modern computing hardware this effect frequently tends to be masked, as the cost of transferring data from memory to the compute unit tends to be far greater than that of performing floating point operations. In essence, this can make floating point operations nearly ‘free’ compared to the transfer of data.

However, the efficient implementation of the building-block finite element operators that make up the discretised version of the incompressible Navier-Stokes equations (3.1) that were used in this work, such as inner products or derivatives, still requires careful consideration. Even small changes in polynomial order can yield a large effect on performance due to the many dependent factors, including both the choice of hardware and the problem under consideration. Our recent work in this area [58] has focused on the development of techniques that can automatically determine the most efficient kernels for these operators at runtime. These kernels are aimed to target both clear performance factors (such as operator type and polynomial order), as well as more subtle aspects, such as whether the choice of basis and whether mathematical techniques such as sum-factorisation can be leveraged to increase performance, and the grouping of elements with equivalent numerical properties to improve performance evaluation. The use of these kernels therefore allows an efficient on-node implementation across a broad range of polynomial orders.

For parallelisation between nodes, `Nektar++` uses a parallelisation model based on distributed MPI. The METIS library [39] is first used to partition the domain based on the dual graph representation of the mesh. Since, for industrial geometries, the mesh generation process yields a hybrid mesh of prismatic and tetrahedral elements, these are weighted appropriately in the partitioning process, so as to approximately

equally distribute the work between processors. Both the pressure Poisson equation and the three scalar velocity correction Helmholtz equations that comprise the operator splitting scheme are solved implicitly, leading to the need for efficient solvers for large, distributed linear algebraic positive definite systems. To achieve this, we apply an iterative preconditioned conjugate gradient (PCG) method for each scalar system. We note, however, that the usual approach taken at lower orders of representing this system as a large, distributed sparse matrix is not generally efficient at higher orders. Instead, when the action of the matrix multiplication is required during PCG iterations, we implement this as many smaller, locally dense elemental matrices, combined with a sparse gather operation to apply C^0 connectivity. The gather operation uses the efficient `Gslib` library in parallel, based on the code `Nek5000` [18]. Our tests have demonstrated that this combination allows both performance and scalability of the PCG solver up to 10^5 cores at high levels of efficiency. Finally, we discuss preconditioning of these methods. All four PCG solvers use an effective and scalable low-energy basis preconditioner [73, 28] and, optionally, projected initial conditions based on storage from solutions at previous timesteps [20]. The solution of the pressure Poisson equation, however, is particularly ill-conditioned, and therefore requires the use of coarse-space preconditioning, which requires the inversion of the linear finite element representation of the mesh. Currently, the coarse space preconditioner uses a direct parallel inverse method developed by Fischer and Tufo [79].

The combination of all of these intra- and inter-node performance optimisations, together with a robust meshing pipeline and stabilisation strategies, has yielded an efficient solver at high orders that is capable of tackling complex configurations required for industrially relevant problems at superior computational performance.

4. High-order iLES for the aerodynamic (re)design of the Elemental Rp1 road car. The developments in terms of high-order mesh generation and numerics implemented in `NekMesh` and `Nektar++` that were described in sections 2 and 3 were used to perform large-scale high-fidelity LES simulations of a real road car, namely the Rp1 car [16]. The Rp1 car, produced by the British manufacturer Elemental Cars since June 2016, is a high-performance, light-weight, road-legal vehicle that weighs 580 kg. It is equipped with a 320 bhp engine and accelerates from 0 to 60 mph in just 2.6 seconds. Its current configuration, depicted in Figure 7 along with its digital twin, delivers 400kg of downforce at 150 mph. The main targets of the high-fidelity LES simulations performed were to analyze and improve the external aerodynamic performance of the car. In particular, we analyzed three configurations, namely designs 1, 2, and 3, that we denote as D1, D2, and D3, respectively. D1 is the baseline design, while D2 and D3 are two design upgrades that were implemented to improve aerodynamic performance and pilot comfort. This study was a collaborative effort put forward by Imperial College London, Elemental cars group [15], London Computational Solutions (LCS) [42] and Silicon Graphics International Corp (now Hewlett Packard Enterprise). All the simulations were carried out using the open-source library `Nektar++`. In the following, we describe the configuration of the simulations and we discuss qualitatively the results obtained.

4.1. High-order meshing strategy and simulation configuration. This section outlines the details of the meshes adopted along with the strategies used to generate them (section 4.1.1), and discusses the setup of the simulations performed (section 4.1.2).



FIG. 7. The original design of the Elemental Rp1 road car, a high performance, street legal track car which delivered 400 kg of downforce at 150 mph.

4.1.1. Mesh generation. The hugely complicated nature of the geometry under consideration, which involves hundreds of CAD surfaces and thousands of edges, presented a significant test for the mesh generation process outlined in section 2. The high-order meshes used in this study ranged from approximately 2.2 million to 4 million elements, and they are comprised of a combination of triangular prisms in the near-wall boundary layer region with 6 layers used to resolve separation accurately, and tetrahedra in the rest of the domain.

We adopted three different meshing strategies for the three car designs, D1, D2, and D3. In particular, we initially used the first and second *analytic curving* method described in section 2.3, for the surface mesh generation stage. The first *analytic curving* method was applied to the first design, namely D1, where the creation of a viable mesh took approximately 4 months of iterations between CAD healing and high-order meshing. The final mesh was composed of 2.2 million elements, including 600000 prisms within the boundary-layer mesh. The second *analytic curving* method was instead applied to the second design, namely D2, that was constructed based on the results from D1. The mesh for D2 contained 2.6 million elements, including 700000 boundary layer prisms. While, both *analytic curving* strategies were effective in generating viable meshes for D1 and D2, they were deemed excessively user-driven and time-consuming. The mesh for D3 was obtained with what we named the *projection curving* method, described in section 2.3. Indeed, this method, along with the use of *CADfix* for geometry repair, *Star-CCM+* for linear and boundary-layer mesh generation and *NekMesh* for high-order mesh generation was the least time-consuming and user-driven. On the one hand, it allowed cutting the overall meshing time from 4 months to 1 week. On the other hand, the mesh was created almost without user effort, and the process required one single execution of linear mesh generation via *Star-CCM+* and one single execution of *NekMesh*. No repetitive cycles of CAD healing were required. The final mesh produced for D3 contained approximately 3.5 million elements, including 1.24 million boundary-layer prisms. Although the meshing strategy for D3 is the most robust and efficient, we should point out that allows for inaccuracies in the geometry description, as a small percentage of surface elements are included in the “exclusion set”, thereby remaining straight-sided. However, in the results that we present in section 4.2.2, no obvious unphysical results could be noticed, and the flow physics in those regions of the car that were geometrically identical between D2 and D3 was indistinguishable.

Figure 8 shows the three surface meshes for D1, D2, and D3 obtained via the three different strategies just outlined. It is possible to note that the surface mesh for D2 and D3 has a smoother element distribution than D1, due to the enhanced

triangulation strategies adopted, namely the second *analytic curving*, and *projection curving*. We remark that for all the simulations carried out we did not include the spokes that are instead shown in figure 4.

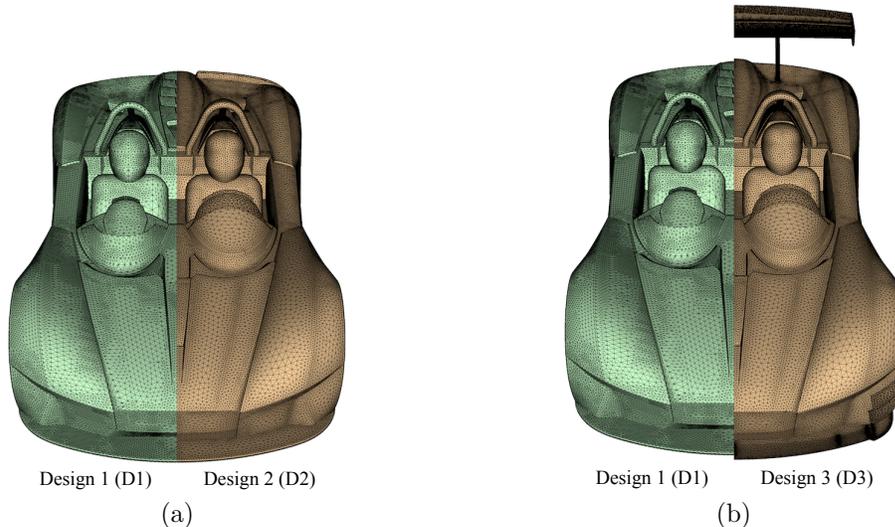


FIG. 8. Surface meshes of the three car designs considered, $D1$, $D2$, and $D3$. Figure (a) shows a comparison between $D1$ and $D2$, while figure (b) shows a comparison between $D1$ and $D3$. It is possible to notice a smoother distribution of surface elements for both $D2$ and $D3$ when compared to $D1$. This is due to the different approaches adopted for the surface mesh generation, where for $D1$ we used the first analytic curving, while for $D2$ and $D3$ we used the second analytic curving and projection curving, respectively.

4.1.2. Simulations setup. The simulation setup required to analyze the aerodynamic of the Rp1 car was challenging due to the high Reynolds number, $Re = 10^6$, based on the length of the car L , and to the complex geometries involved. In order to obtain stable yet accurate simulations, we adopted some simplifications that are commonly adopted when performing the aerodynamic car simulations.

The Rp1's unique double diffuser design and the the road-tyre interaction were carefully modeled with some simplifications that allowed a significantly easier simulation setting while not undermining the accuracy of the results. Rp1's double diffuser design has intakes and outlets on the body of the car. To model these accurately, we extracted the corresponding surfaces and constructed an appropriate velocity boundary field, by enforcing a prescribed average normal velocity on each surface. The boundary condition has to also satisfy the no-slip condition at the edges of the inlet surfaces and this was imposed by solving a Helmholtz problem where the positive Helmholtz coefficient was chosen to provide a layer of desired thickness at the edges. This was possible by using a manifold solver developed by Cantwell et al. [6]. In addition, the road-tyre interaction was modeled using a contact patch along with rotational boundary conditions to a simplified tyre geometry. This allowed efficiently modeling the rotation of the tyre without the need for a moving mesh. Finally, we simulated only half of the car, by using a symmetry plane. These simplifications allowed a faster simulation setup and simulation runtime, without altering the flow features that were of interest in this study.

The simulations were initialized adopting a similar strategy to Lombard et al. [43],

where we began at a lower Reynolds number, $Re = 10^4$. The initial conditions were imposed impulsively so that $u/U = 1$, where U is a reference velocity, and all non-moving surfaces are treated as no-slip boundaries. At the outflow, we imposed the boundary condition developed by Dong et al. [14], which balances the kinetic energy influx through the outflow boundary condition to prevent instability. The Reynolds number is then gradually increased by a factor of ten until the target Reynolds number, $Re = 10^6$, is reached. Between increases, the flow is evolved for two convective time units $t_c = L/U$ to allow the flow to adapt to the new flow physics dictated by the higher Reynolds number. We note that the Reynolds number used in this study, $Re = 10^6$, is still lower than the experimental value of approximately $Re = 2 \times 10^6$. Nevertheless, to the authors' knowledge, this is the highest Reynolds number ever attempted for a real car geometry, using iLES in conjunction with spectral/ hp element methods.

The simulations were run at a polynomial order 4 (fifth order of accuracy) with a second order implicit-explicit time-stepping. To increase the accuracy of the integration, a seventh order of quadrature was used, that consisted of the so-called global consistent integration introduced in Mengaldo et al. [48], and described in section 3.1. This allowed us to mitigate aliasing-driven instabilities. We note that the use of a higher quadrature meant that the meshes were generated at a higher order of $P = 7$ to ensure the elements were valid with this quadrature. In addition, the SVV kernel presented in section 3.2 was used to filter the under-resolved scales and enhance the stability of the simulations. Without the use of both, consistent integration and SVV was not possible to obtain numerically stable calculations.

In terms of computational requirements, the simulation were performed in parallel using 1000 to 4000 cores, depending on node availability on a SGI UV machine, and it took approximately 3 days to evolve the flow for one convective length. The high-fidelity flow simulations performed allowed unique insights into the aerodynamic performance of the Rp1's complex geometry, highlighting a number of important regions for aerodynamic optimisation. These results are discussed next.

4.2. Results. In this section, we discuss the results obtained on the three car designs introduced earlier, D1, D2 and D3. In particular, in section 4.2.1, we present a comparison between D1 and D2, while in section 4.2.2, we present a comparison between D1 and D3. The results focus on iso-contour quantities that allowed us to identify important flow features suitable to optimization.

4.2.1. First design stage. The first simulation performed was for design 1 (D1), that is the first baseline model of the car that was produced, followed by the simulation of design 2 (D2). The latter design, D2, implements an upgrade of the aerodynamic package with respect to D1. Figure 9 shows the meshes and associated results for D1 and D2. In particular, figure 9-(a), shows the meshes for the two designs (that are the same as depicted in figure 8), while figure 9-(b) shows the associated flow results. Figure 9-(c), shows the same flow results as the right image in figure 9-(a), from a sideview perspective. The flow results depicted in figure 9 show iso-contours of the total pressure coefficient $Cp_0 = 1 - (u/u_\infty)^2 = 0$, and they are colored as pressure. From the high-order D1 simulation two key findings were made, neither of which were identified in low-order RANS simulations [77]. Firstly, there appeared to be strong vortical structures hitting the drivers helmet (see figures 9-(b) and 9-(c) for D1). Indeed, aerodynamic disturbances were also reported by the test drivers, confirming the finding of our iLES. These vortical structures were generated by the windscreen, and the issue was fixed in D2 with a redesigned console area in front of

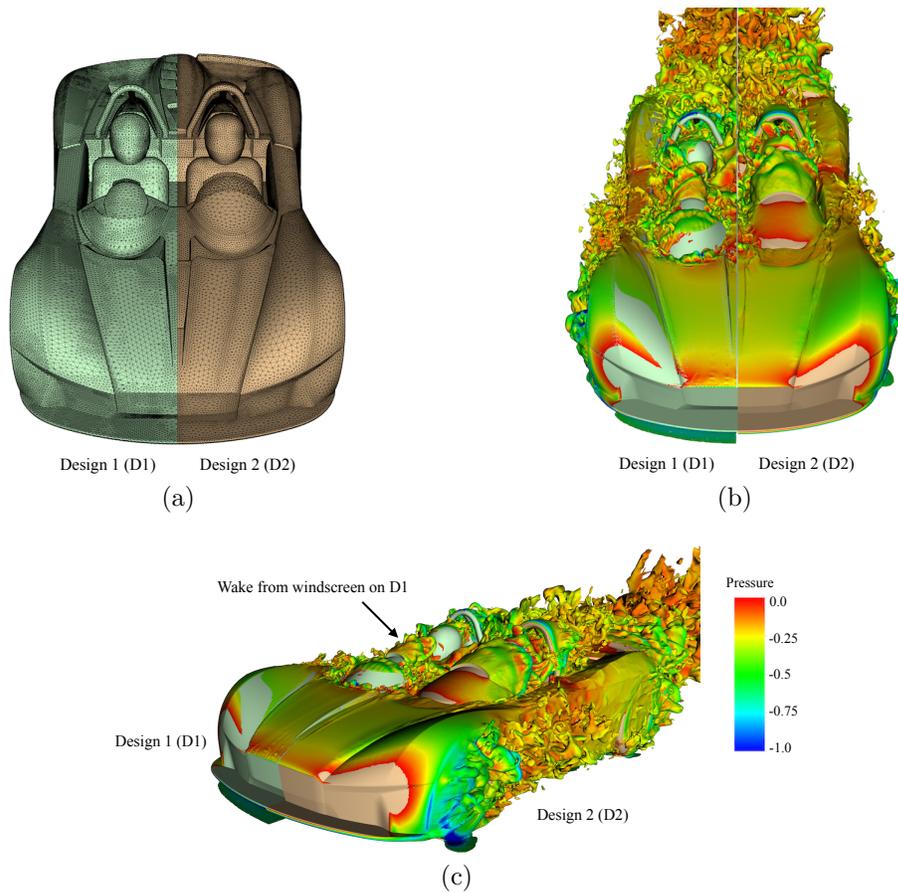


FIG. 9. Comparison between *D1* and *D2*, in terms of mesh [figure (a)], and flow physics [figures-(b) and -(c)]. The images depicting the flow physics shows the total pressure coefficient for $C_{p0} = 0$ and the iso-contours are colored in pressure.

the driver, significantly enhanced the driving experience. The *D2* iso-contour shows clearly that these structures now pass cleanly over the drivers head and are less noisy in terms of small-scale vortices (figures 9-(b) and 9-(c), for *D2*). Secondly, the roll hoop produced significantly more drag and separation than predicted. *D2* had a redesigned roll hoop with an aerofoil profile as opposed to a circular cylinder. This aerofoil profile was slightly angled to help control the flow over the new Gurney flap at the rear of the car. This, combined with redesigned diffusers on the underside, led to increased downforce over *D1*. The trend of changes in downforce and drag between the *D1* and *D2* simulations was well predicted by the high-order simulations and matched the trend in the RANS results [77].

4.2.2. Second design stage. The final simulation performed in this study is for design 3 (*D3*), which is a full aerodynamic upgrade over the previous two designs, *D1* and *D2*. This car is designed to achieve extremely high levels of downforce specifically for track racing. This design includes a fully redesigned floor, front splitter and the addition of a rear wing. The ride height has also been altered, raising the car at the rear for increased diffuser performance and lowering the front of the car to increase

in-ground effect of the front splitter.

Figure 10, similarly to figure 9, shows again the total pressure coefficient, colored by pressure. The results are consistent with the D2 simulation and are shown alongside

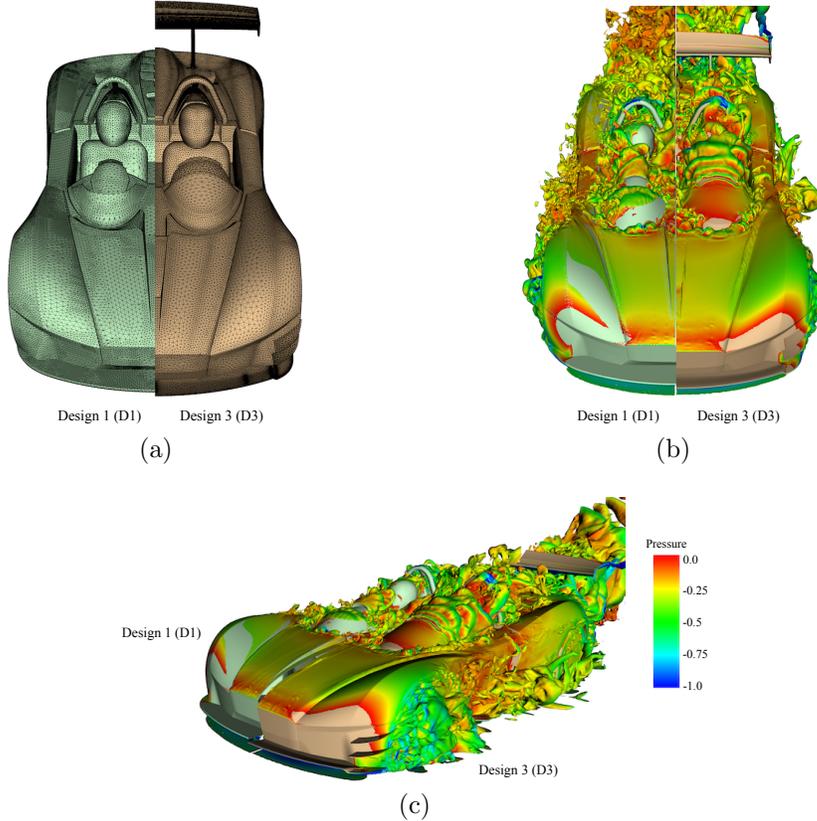


FIG. 10. Comparison between D1 and D3, in terms of mesh [figure (a)], and flow physics [figures (b) and (c)]. The images depicting the flow physics shows the total pressure coefficient for $Cp_0 = 0$ and the iso-contours are colored in pressure.

the D1 results. The trends in increasing downforce were well predicted by the high-order simulation. In figure 10, we note that it is possible to observe a noticeable offset in the geometries of the two cars, due to the alteration in ride height.

Although we used an approach for meshing D3 that can be aggressive in preserving mesh validity versus geometrical accuracy, thereby leaving a non-negligible number of elements straight-sided as opposed to curvilinear (the number of straight-side elements was $< 5\%$), the flow physics is in agreement with D1 and D2, and no unphysical results could be readily noticed. However a more comprehensive study should be undertaken in order to draw stronger conclusions on whether the compromised geometric accuracy of our most robust meshing pipeline is a compromise worth taking. The early results here show that it may well be.

5. Conclusions. This work demonstrates the significant progress in leveraging Nektar++ for the goal of moving high-order CFD from academia to industry, and some of the benefits that high-order methods can provide to the industrial community. Significant developments had to be made in order to overcome obstacles in both mesh

generation and solver technologies.

The novel high-order meshing workflow described in section 2 allowed the robust, efficient and accurate high-order mesh generation of an extremely complex geometry. This was a key to prevent unphysical diffusion being generated at no-slip surfaces that constitute the geometry of the car, and to avoid using an excessive number of linear elements in proximity of curved geometrical features. Indeed, the meshing framework has proven reliable and viable for industrial geometries, and it was a key enabler to successfully accomplish the simulations presented in section 4.

Along with the high-order meshing framework, the novel numerical technologies described in this section 3 were essential to successfully produce the CFD workflow described in this paper. More specifically, the consistent integration for mitigating aliasing-driven instabilities in conjunction with the CG-SVV based on the “DG” kernel for treating under-resolved scales were the keys to obtain stable simulations and accurate simulations. The use of the novel CG-SVV approach, in particular, was instrumental to produce high-fidelity results. In fact, the CG-SVV operator is based on ESA’s estimates for spectral/*hp* element methods, where numerical dispersion and diffusion characteristics are quantified in wavenumber/frequency space, as in classical von Neumann analyses. Although mostly feasible for simple model problems, such as one-dimensional linear advection, ESA’s estimates for spectral/*hp* methods have been shown to hold more generally for nonlinear problems and even turbulence simulations [55, 53, 17]. These estimates quantify how the numerical error (diffusion in particular) increases at large wavenumbers/frequencies, affecting especially the smallest, poorly- and under-resolved scales. When interpreted alongside implicit LES experiments, ESA reveals that spectral/*hp* discretisations of higher polynomial orders perform best because they affect the large and intermediate flow scales as little as possible, while gradually introducing a sufficiently strong diffusion to regularise the smaller scales. This way, the governing equations and thus the turbulence physics are solved more accurately for a wider range of scales, rather than having to rely on (often restrictive) modelling hypothesis. We note that past studies also focussed on how numerical diffusion acts as a turbulence model [45, 56, 9, 10, 52]. However the ESA framework was deemed the most promising route to achieve high-fidelity implicit LES results.

The simulations in section 4 represent the first high-order (having solution polynomials = 4) accurate results of a real automotive geometry by means of implicit LES technologies using spectral/*hp* element methods. They represent one of the first attempts at bridging the gap between applied mathematics academic research and industrial applications within the context of CFD.

Acknowledgments. This work has benefited from the technical support and design insight from Elemental Cars and from access to large HPC resources provided by Hewlett Packard Enterprise. We also would like to thank Mark Gammon and his group at ITI Global for providing us with a license to the program CADfix and its CFI interface, and for their technical help and support. Partial financial support was provided by EPSRC under the Platform Grant PRISM: Platform for Research In Simulation Methods (EP/R029423/1) and by the EU Horizon 2020 project ExaFLOW (grant 671571). M. Turner acknowledges the support of EPSRC and Airbus under an Industrial CASE studentship. We also acknowledge support from Imperial College High Performance Computing Service.

- [1] J. L. BENTLEY, *Multidimensional binary search trees used for associative searching*, Communications of the ACM, 18 (1975), pp. 509–517.
- [2] G. A. BLAISDELL, E. T. SPYROPOULOS, AND J. H. QIN, *The effect of the formulation of non-linear terms on aliasing errors in spectral methods*, Applied Numerical Mathematics, 21 (1996), pp. 207–219.
- [3] J. H. BUCKLOW, R. FAIREY, AND M. R. GAMMON, *An automated workflow for high quality cfd meshing using the 3d medial object*, in 23rd AIAA Computational Fluid Dynamics Conference, 2017, p. 3454.
- [4] R. H. BUSH, T. S. CHYCZEWSKI, K. DURAISAMY, B. EISFELD, C. L. RUMSEY, AND B. R. SMITH, *Recommendations for future efforts in RANS modeling and simulation*, in AIAA Scitech 2019 Forum, 2019, p. 0317.
- [5] C. D. CANTWELL, D. MOXEY, A. COMERFORD, A. BOLIS, G. ROCCO, G. MENGALDO, D. DE GRAZIA, S. YAKOVLEV, J. LOMBARD, D. EKELSCHOT, B. JORDI, H. XU, Y. MOHAMIED, C. ESKILSSON, B. NELSON, P. VOS, C. BIOTTO, R. M. KIRBY, AND S. J. SHERWIN, *Nektar++: An open-source spectral/hp element framework*, Computer Physics Communications, 192 (2015), pp. 205–219.
- [6] C. D. CANTWELL, S. YAKOVLEV, R. M. KIRBY, N. S. PETERS, AND S. J. SHERWIN, *High-order spectral/hp element discretisation for reaction–diffusion problems on surfaces: Application to cardiac electrophysiology*, Journal of computational physics, 257 (2014), pp. 813–829.
- [7] G. CASTIGLIONI AND J. A. DOMARADZKI, *A numerical dissipation rate and viscosity in flow simulations with realistic geometry using low-order compressible Navier–Stokes solvers*, Computers & Fluids, 119 (2015), pp. 37–46.
- [8] *Centaur*. <https://www.centaursoft.com/>, 2017.
- [9] J. B. CHAPELIER, G. LODATO, AND A. JAMESON, *A study on the numerical dissipation of the spectral difference method for freely decaying and wall-bounded turbulence*, Computers & Fluids, 139 (2016), pp. 261–280.
- [10] T. DAIRAY, E. LAMBALLAIS, S. LAIZET, AND J. C. VASSILICOS, *Numerical dissipation vs subgrid-scale modelling for large eddy simulation*, Journal of Computational Physics, 337 (2017), pp. 252–274.
- [11] F. DASSI, A. MOLA, AND H. SI, *Curvature-adapted remeshing of CAD surfaces*, Procedia Engineering, 82 (2014), pp. 253–265.
- [12] C. C. DE WIART, K. HILLEWAERT, L. BRICTEUX, AND G. WINCKELMANS, *Implicit LES of free and wall-bounded turbulent flows based on the discontinuous Galerkin/symmetric interior penalty method*, International Journal for Numerical Methods in Fluids, 78 (2015), pp. 335–354.
- [13] S. DEY, R. M. O’BARA, AND M. S. SHEPARD, *Towards curvilinear meshing in 3D: the case of quadratic simplices*, Computer-Aided Design, 33 (2001), pp. 199–209.
- [14] S. DONG, G. E. KARNIADAKIS, AND C. CHRYSOSTOMIDIS, *A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains*, Journal of Computational Physics, 261 (2014), pp. 83–105, <https://doi.org/10.1016/j.jcp.2013.12.042>.
- [15] ELEMENTAL CARS, *Elemental cars*. <http://elementalcars.co.uk/>, 2018.
- [16] ELEMENTAL CARS, *The Elemental RPI*. <http://elementalcars.co.uk/the-rp1/>, 2018.
- [17] P. FERNANDEZ, R. C. MOURA, G. MENGALDO, AND J. PERAIRE, *Non-modal analysis of spectral element methods: towards accurate and robust large-eddy simulations*, Computer Methods in Applied Mechanics and Engineering, 346 (2019), pp. 43–62.
- [18] P. FISCHER, J. KRUSE, J. MULLEN, H. TUFO, J. LOTTES, AND S. KERKEMEIER, *Nek5000: Open source spectral element CFD solver*. <https://nek5000.mcs.anl.gov/index.php/MainPage>, 2008. Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL.
- [19] P. FISCHER, J. MULLEN, ET AL., *Filter-based stabilization of spectral element methods*, Comptes Rendus de l’Academie des Sciences Series I Mathematics, 332 (2001), pp. 265–270.
- [20] P. F. FISCHER, *Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides*, Computer Methods in Applied Mechanics and Engineering, 163 (1998), pp. 193–204.
- [21] P. F. FISCHER, G. W. KRUSE, AND F. LOTH, *Spectral element methods for transitional flows in complex geometries*, Journal of Scientific Computing, 17 (2002), pp. 81–98.
- [22] M. FORTUNATO AND P.-O. PERSSON, *High-order unstructured curved mesh generation using the Winslow equations*, Journal of Computational Physics, 307 (2016), pp. 1–14.
- [23] C. FU, M. UDDIN, AND A. C. ROBINSON, *Turbulence modeling effects on the CFD predictions of flow over a NASCAR Gen 6 racecar*, Journal of Wind Engineering and Industrial Aerodynamics, 176 (2018), pp. 98–111.
- [24] A. GARGALLO-PEIRÓ, X. ROCA, J. PERAIRE, AND J. SARRATE, *Optimization of a regularized*

- distortion measure to generate curved high-order unstructured tetrahedral meshes*, International Journal for Numerical Methods in Engineering, 103 (2015), pp. 342–363.
- [25] G. J. GASSNER, *A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1233–A1253.
- [26] G. J. GASSNER AND A. D. BECK, *On the accuracy of high-order discretizations for underresolved turbulence simulations*, Theoretical and Computational Fluid Dynamics, 27 (2013), pp. 221–237.
- [27] D. GOTTLIEB AND J. S. HESTHAVEN, *Spectral methods for hyperbolic problems*, Journal of Computational and Applied Mathematics, 128 (2001), pp. 83–131.
- [28] L. GRINBERG, D. PEKUROVSKY, S. J. SHERWIN, AND G. E. KARNIADAKIS, *Parallel performance of the coarse space linear vertex solver and low energy basis preconditioner for spectral/hp elements*, Parallel Computing, 35 (2009), pp. 284–304.
- [29] J.-L. GUERMOND AND J. SHEN, *Velocity-correction projection methods for incompressible flows*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 112–134.
- [30] R. HARTMANN AND T. LEICHT, *Generation of unstructured curvilinear grids and high-order discontinuous Galerkin discretization applied to a 3D high-lift configuration*, International Journal for Numerical Methods in Fluids, 82 (2016), pp. 316–333.
- [31] J. HESTHAVEN AND R. KIRBY, *Filtering in Legendre spectral methods*, Mathematics of Computation, 77 (2008), pp. 1425–1452.
- [32] F. HINDENLANG, G. GASSNER, T. BOLEMANN, AND C. D. MUNZ, *Unstructured high order grids and their application in discontinuous Galerkin methods*, in Conference Proceedings, V European Conference on Computational Fluid Dynamics ECCOMAS CFD, 2010, pp. 1–8.
- [33] T. J. R. HUGHES, G. R. FELIÓO, L. MAZZEI, AND J.-B. QUINCY, *The variational multiscale method—a paradigm for computational mechanics*, Computer methods in applied mechanics and engineering, 166 (1998), pp. 3–24.
- [34] ITI-GLOBAL, *CADfix: CAD translation, healing, repair, and transformation*. www.iti-global.com/cadfix, 2017.
- [35] G. S. KARAMANOS AND G. E. KARNIADAKIS, *A spectral vanishing viscosity method for large-eddy simulations*, Journal of Computational Physics, 163 (2000), pp. 22–50.
- [36] G. E. KARNIADAKIS, M. ISRAELI, AND S. A. ORSZAG, *High-order splitting methods for the incompressible Navier-Stokes equations*, Journal of Computational Physics, 97 (1991), pp. 414–443.
- [37] G. E. KARNIADAKIS AND S. A. ORSZAG, *Nodes, modes and flow codes*, Physics Today, 46 (1993), pp. 34–42.
- [38] G. E. KARNIADAKIS AND S. J. SHERWIN, *Spectral/hp element methods for computational fluid dynamics*, Oxford University Press, 2nd ed., 2005.
- [39] G. KARYPIS AND V. KUMAR, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM Journal on Scientific Computing, 2 (1998), pp. 359–392.
- [40] R. M. KIRBY AND S. J. SHERWIN, *Stabilisation of spectral/hp element methods through spectral vanishing viscosity: application to fluid mechanics modelling*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 3128–3144.
- [41] K. KOAL, J. STILLER, AND H. M. BLACKBURN, *Adapting the spectral vanishing viscosity method for large-eddy simulations in cylindrical configurations*, Journal of Computational Physics, 231 (2012), pp. 3389–3405.
- [42] LCS FAST, *London computational solutions*. http://www.lcs-fast.com/aero_main/, 2018.
- [43] J.-E. W. LOMBARD, D. MOXEY, S. J. SHERWIN, J. F. A. HOESSLER, S. DHANDAPANI, AND M. J. TAYLOR, *Implicit large-eddy simulation of a wingtip vortex*, AIAA Journal, 54 (2016), pp. 506–518.
- [44] X.-J. LUO, M. S. SHEPHARD, R. M. O’BARA, R. NASTASIA, AND M. W. BEALL, *Automatic p-version mesh generation for curved domains*, Engineering with Computers, 20 (2004), pp. 273–285.
- [45] L. G. MARGOLIN AND W. J. RIDER, *A rationale for implicit turbulence modelling*, International Journal for Numerical Methods in Fluids, 39 (2002), pp. 821–841.
- [46] S. MARRAS, J. F. KELLY, F. X. GIRALDO, AND M. VÁZQUEZ, *Variational multiscale stabilization of high-order spectral elements for the advection–diffusion equation*, Journal of Computational Physics, 231 (2012), pp. 7187–7213.
- [47] G. MENGALDO, D. DE GRAZIA, R. C. MOURA, AND S. J. SHERWIN, *Spatial eigensolution analysis of energy-stable flux reconstruction schemes and influence of the numerical flux on accuracy and robustness*, Journal of Computational Physics, 358 (2018), pp. 1–20.
- [48] G. MENGALDO, D. DE GRAZIA, D. MOXEY, P. E. VINCENT, AND S. J. SHERWIN, *Dealiasing techniques for high-order spectral element methods on regular and irregular grids*, Journal

- of Computational Physics, 299 (2015), pp. 56–81.
- [49] G. MENGALDO, R. C. MOURA, B. GIRALDA, J. PEIRÓ, AND S. J. SHERWIN, *Spatial eigensolution analysis of discontinuous Galerkin schemes with practical insights for under-resolved computations and implicit LES*, Computers & Fluids, 169 (2018), pp. 349–364.
- [50] R. MOURA, M. AMAN, J. PEIRÓ, AND S. J. SHERWIN, *Spatial eigenanalysis of spectral/hp continuous Galerkin schemes and their stabilisation via DG-mimicking spectral vanishing viscosity for high Reynolds number flows*, Journal of Computational Physics, 406 (2020), p. 109112.
- [51] R. C. MOURA, *On the use of spectral element methods for under-resolved simulations of transitional and turbulent flows*, PhD thesis, Imperial College London, 2017.
- [52] R. C. MOURA, G. MENGALDO, J. PEIRÓ, AND S. J. SHERWIN, *An LES setting for DG-based implicit LES with insights on dissipation and robustness*, in Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM 2016, M. Bittencourt, N. Dumont, and J. S. Hesthaven, eds., Springer, 2017, pp. 161–173.
- [53] R. C. MOURA, G. MENGALDO, J. PEIRÓ, AND S. J. SHERWIN, *On the eddy-resolving capability of high-order discontinuous Galerkin approaches to implicit LES / under-resolved DNS of Euler turbulence*, Journal of Computational Physics, 330 (2017), pp. 615–623.
- [54] R. C. MOURA, J. PEIRÓ, AND S. J. SHERWIN, *Implicit LES approaches via discontinuous Galerkin methods at very large Reynolds*, in Direct and Large-Eddy Simulation XI, M. Salvetti, V. Armenio, J. Frohlich, B. Geurts, and H. Kuerten, eds., Springer, 2019, pp. 53–59.
- [55] R. C. MOURA, S. J. SHERWIN, AND J. PEIRÓ, *Linear dispersion-diffusion analysis and its application to under-resolved turbulence simulations using discontinuous Galerkin spectral/hp methods*, Journal of Computational Physics, 298 (2015), pp. 695–710.
- [56] R. C. MOURA, S. J. SHERWIN, AND J. PEIRÓ, *Modified equation analysis for the discontinuous Galerkin formulation*, in Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM 2014, R. M. Kirby, M. Berzins, and J. S. Hesthaven, eds., Springer, 2015, pp. 375–383.
- [57] R. C. MOURA, S. J. SHERWIN, AND J. PEIRÓ, *Eigensolution analysis of spectral/hp continuous Galerkin approximations to advection-diffusion problems: insights into spectral vanishing viscosity*, Journal of Computational Physics, 307 (2016), pp. 401–422.
- [58] D. MOXEY, C. CANTWELL, R. KIRBY, AND S. SHERWIN, *Optimising the performance of the spectral/hp element method with collective linear algebra operations*, Computer Methods in Applied Mechanics and Engineering, 310 (2016), pp. 628–645.
- [59] D. MOXEY, C. D. CANTWELL, Y. BAO, A. CASSINELLI, G. CASTIGLIONI, S. CHUN, E. JUDA, E. KAZEMI, K. LACKHOVE, J. MARCON, G. MENGALDO, D. SERSON, M. TURNER, H. XU, J. PEIRÓ, R. M. KIRBY, AND S. J. SHERWIN, *Nektar++: enhancing the capability and application of high-fidelity spectral/hp element methods*, Computer Physics Communications, 249 (2020), p. 107110.
- [60] D. MOXEY, D. EKELSCHOT, Ü. KESKIN, S. J. SHERWIN, AND J. PEIRÓ, *A thermo-elastic analogy for high-order curvilinear meshing with control of mesh validity and quality*, Procedia engineering, 82 (2014), pp. 127–135.
- [61] D. MOXEY, M. D. GREEN, S. J. SHERWIN, AND J. PEIRÓ, *An isoparametric approach to high-order curvilinear boundary-layer meshing*, Computer Methods in Applied Mechanics and Engineering, 283 (2015), pp. 636–650.
- [62] J. NASH, *C1 isometric imbeddings*, Annals of mathematics, (1954), pp. 383–396.
- [63] *Nektar++*. <http://www.nektar.info/>, 2017.
- [64] *OpenCascade*. <http://www.opencascade.org/>, 2017.
- [65] P.-O. PERSSON AND J. PERAIRE, *Curved mesh generation and mesh refinement using Lagrangian solid mechanics*, in 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, 2009, p. 949.
- [66] *Pointwise*. <http://www.pointwise.com/>, 2017.
- [67] S. B. POPE, *Ten questions concerning the large-eddy simulation of turbulent flows*, New Journal of Physics, 6 (2004), p. 35.
- [68] R. POYA, R. SEVILLA, AND A. J. GIL, *A unified approach for a posteriori high-order curved mesh generation using solid mechanics*, Computational Mechanics, 58 (2016), pp. 457–490.
- [69] M. PRATT, *Introduction to ISO 10303 – the STEP standard for product data exchange*, Journal of Computing and Information Science in Engineering, 1 (2001), pp. 102–103.
- [70] O. SAHNI, X. J. LUO, K. E. JANSEN, AND M. S. SHEPHARD, *Curved boundary layer meshing for adaptive viscous flow simulations*, Finite Elements in Analysis and Design, 46 (2010), pp. 132–139.
- [71] D. SERSON, J. R. MENEGHINI, AND S. J. SHERWIN, *Direct numerical simulations of the flow around wings with spanwise waviness*, Journal of Fluid Mechanics, 826 (2017), pp. 714–731.

- [72] M. S. SHEPHARD, J. E. FLAHERTY, K. E. JANSEN, X. LI, X. LUO, N. CHEVAUGEON, J.-F. REMACLE, M. W. BEALL, AND R. M. O'BARA, *Adaptive mesh generation for curved domains*, Applied Numerical Mathematics, 52 (2005), pp. 251–271.
- [73] S. J. SHERWIN AND M. CASARIN, *Low-energy basis preconditioning for elliptic substructured solvers based on unstructured spectral/hp element discretization*, Journal of Computational Physics, 171 (2001), pp. 394–417.
- [74] S. J. SHERWIN AND J. PEIRÓ, *Mesh generation in curvilinear domains using high-order elements*, International Journal for Numerical Methods in Engineering, 53 (2002), pp. 207–223.
- [75] J. SLOTNICK, A. KHODADOUST, J. ALONSO, D. DARMOFAL, W. GROPP, E. LURIE, AND D. MAVRIPLIS, *CFD vision 2030 study: a path to revolutionary computational aerosciences*, Technical Report CR 2014-218178, NASA, 2014.
- [76] E. TADMOR, *Convergence of spectral methods for nonlinear conservation laws*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 30–44.
- [77] M. TAYLOR. personal communication, 2017.
- [78] T. TOULORGE, C. GEUZAIN, J.-F. REMACLE, AND J. LAMBRECHTS, *Robust untangling of curvilinear meshes*, Journal of Computational Physics, 254 (2013), pp. 8–26.
- [79] H. M. TUFO AND P. F. FISCHER, *Fast parallel direct solvers for coarse grid problems*, Journal of Parallel and Distributed Computing, 61 (2001), pp. 151–177.
- [80] M. TURNER, *High-order mesh generation for CFD solvers*, PhD thesis, Imperial College London, 2017.
- [81] M. TURNER, J. PEIRÓ, AND D. MOXEY, *Curvilinear mesh generation using a variational framework*, Computer-Aided Design, 103 (2018), pp. 73–91.
- [82] B. C. VERMEIRE, S. NADARAJAH, AND P. G. TUCKER, *Implicit large eddy simulation using the high-order correction procedure via reconstruction scheme*, International Journal for Numerical Methods in Fluids, 82 (2016), pp. 231–260.
- [83] P. E. J. VOS, C. ESKILSSON, A. BOLIS, S. CHUN, R. M. KIRBY, AND S. J. SHERWIN, *A generic framework for time-stepping partial differential equations (PDEs): general linear methods, object-oriented implementation and application to fluid problems*, International Journal of Computational Fluid Dynamics, 25 (2011), pp. 107–125.
- [84] A. R. WINTERS, R. C. MOURA, G. MENGALDO, G. J. GASSNER, S. WALCH, J. PEIRÓ, AND S. J. SHERWIN, *A comparative study on polynomial dealiasing and split form discontinuous Galerkin schemes for under-resolved turbulence computations*, Journal of Computational Physics, 372 (2018), pp. 1–21.
- [85] Z. Q. XIE, R. SEVILLA, O. HASSAN, AND K. MORGAN, *The generation of arbitrary order curved meshes for 3D finite element analysis*, Computational Mechanics, 51 (2013), pp. 361–374.
- [86] C. ZHANG, C. P. BOUNDS, L. FOSTER, AND M. UDDIN, *Turbulence modeling effects on the CFD predictions of flow over a detailed full-scale sedan vehicle*, Fluids, 4 (2019), p. 148.
- [87] Z. ZHONG, X. GUO, W. WANG, B. LÉVY, F. SUN, Y. LIU, AND W. MAO, *Particle-based anisotropic surface meshing*, ACM Transactions on Graphics (TOG), 32 (2013), p. 99.