



A comparison of interpolation techniques for non-conformal high-order discontinuous Galerkin methods

Edward Laughton*, Gavin Tabor, David Moxey

College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK

Received 9 July 2020; received in revised form 21 February 2021; accepted 19 March 2021

Available online xxxx

Abstract

The capability to incorporate moving geometric features within models for complex simulations is a common requirement in many fields. Fluid mechanics within aeronautical applications, for example, routinely feature rotating (e.g. turbines, wheels and fan blades) or sliding components (e.g. in compressor or turbine cascade simulations). With an increasing trend towards the high-fidelity modelling of these cases, in particular combined with the use of high-order discontinuous Galerkin methods, there is therefore a requirement to understand how different numerical treatments of the interfaces between the static mesh and the sliding/rotating part impact on overall solution quality. In this article, we compare two different approaches to handle this non-conformal interface. The first is the so-called mortar approach, where flux integrals along edges are split according to the positioning of the non-conformal grid. The second is a less-documented point-to-point interpolation method, where the interior and exterior quantities for flux evaluations are interpolated from elements lying on the opposing side of the interface. Although the mortar approach has significant advantages in terms of its numerical properties, in that it preserves the local conservation properties of DG methods, in the context of complex 3D meshes it poses notable implementation difficulties which the point-to-point method handles more readily. In this paper we examine the numerical properties of each method, focusing not only on observing convergence orders for smooth solutions, but also how each method performs in under-resolved simulations of linear and nonlinear hyperbolic problems, to inform the use of these methods in implicit large-eddy simulations. © 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Keywords: Spectral element method; Non-conformal mesh; Point-to-point interpolation; Mortar method; Moving geometry

1. Introduction

Problems containing features that move or deform are found in many research areas, but are particularly prevalent in the study of various fluid dynamics phenomena. In particular, aeronautical applications commonly feature rotating or sliding geometries, with typical examples in this area including turbomachinery [1,2], insect and avian flight aerodynamics [3,4], unmanned aerial vehicles [5,6], and HVAC (heating, ventilating, and air conditioning) [7,8]. Being able to accurately model these moving geometries and their subsequent impact on the underlying flow physics is highly important: for example, predicting how the profiles of turbine or compressor blades impact on propulsion efficiency, or how wing profiles affect the performance of wind turbines. Moreover, the cost and difficulty of

* Corresponding author.

E-mail addresses: e1326@exeter.ac.uk (E. Laughton), g.r.tabor@exeter.ac.uk (G. Tabor), d.moxey@exeter.ac.uk (D. Moxey).

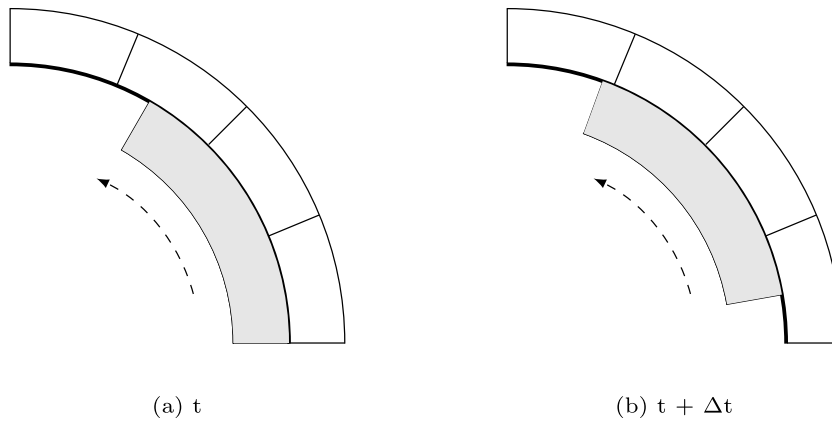


Fig. 1. An example domain showing an inner rotating region and an outer stationary region bounded by a non-conforming interface zone, showing the differing physical positions at two different times.

performing full-scale experimental testing of such geometries can be challenging from the perspective of both instrumentation and expense. For these reasons, computational fluid dynamics (CFD) is now commonplace in the design and modelling process. If a CFD method is to be regarded as universally useful in these application areas, then it must be capable of accurately modelling moving geometry and ideally provide high-fidelity results beyond the scope of physical field tests alone.

Most leading software for CFD is based around lower-order finite volume or finite element methods, typically leveraging the computationally-cheap Reynolds Averaged Navier–Stokes (RANS) equations in combination with a turbulence closure model. However this approach has natural limitations in studying the aforementioned problems at very high levels of fidelity [9]. With the large increases in computational power in recent years, a more recent trend is to instead consider transient simulations that leverage implicit large-eddy simulation (iLES) or under-resolved direct numerical simulation (uDNS) [10,11]. This approach is more computationally expensive than RANS, but also provides greater accuracy and enables high-fidelity simulations of the complex geometries that lie in this regime [12]. The combination of LES with less common high-order methods, either based on continuous or discontinuous Galerkin (CG/DG) methods, has seen significant interest in recent years, particularly in aeronautics applications [13]. From a numerical perspective, high-order methods possess far lower levels of numerical diffusion and dispersion, making them ideally suited to resolving features across long time- and length-scales. This can overcome a significant bottleneck when considering these simulations at lower orders, since very fine grid resolutions are required to overcome the effects of numerical errors [14]. Additionally, from a computational perspective, the larger number of floating-point operations that are required per degree-of-freedom as the polynomial order is increased means that, when equipped with tensor-contraction techniques such as sum factorisation, high-order methods can be used to overcome the memory bandwidth bottlenecks that are common in modern computational hardware [15]. The combination of these effects means that high-order methods can achieve higher accuracy per degree-of-freedom at equivalent or lower computational cost to lower-order methods.

However, these methods are somewhat less well-explored in the simulation of problems involving rotating or sliding geometries, which require the treatment of non-conformal interfaces between elements. In this article, we explore two common approaches to the handling of non-conformal interfaces and compare their numerical performance in a range of linear and nonlinear problems.

1.1. Requirements for moving geometry simulations

One approach to tackling the problem of moving geometry is the sliding mesh method, where the mesh is separated into two or more separate regions, and during the simulation the regions will slide relative to one another. This provides a way to prescribe simple mesh motion via rotation or translation. The most basic problem case is to employ a stationary outer region, with a rotating circular region within it, which is found in many applications, for example modelling flow in a stirred tank [16]. An exaggerated example of this arrangement is shown in Fig. 1,

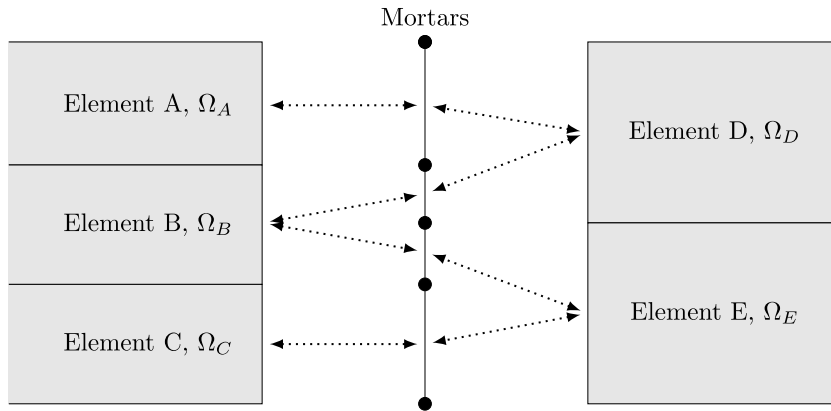


Fig. 2. Mortar construction for a non-conformal interface showing the connection between elements and mortars.

where it is clear that this process results in a non-conformal mesh: i.e. a mesh where elements do not connect to precisely one other element through one of their edges or faces. Most CFD simulations make use of *conformal* meshes, where each edge (in 2D) or face (in 3D) of an element has precisely one neighbouring element. As such, techniques need to be developed in order to accurately preserve solution quality across the non-conformal interface.

In the ‘classical’ spectral element method, where C^0 continuity is imposed between elements in a CG formulation, three main techniques have been evaluated for use in non-conformal meshes. Possibly the most well-known of these arises when performing h -adaptation in an octree-like manner, so that 2-to-1 element subdivisions are obtained in the resulting mesh. In this case, hanging nodes are generated and their values can be constrained through the analytic definition of the basis functions lying along an edge or face, together with the assembly mapping that is used to construct mass and stiffness matrices [17–19]. However, in the sliding mesh case, elements may overlap at arbitrary positions along their edges and faces, making this approach infeasible. Possibly the most widely-adopted approach to implementing generic non-conformal interfaces in the CG setting is the mortar technique [20]. In this setting, one augments the traditional C^0 function spaces for each conformal domain with functions defined on mortar elements at the interface between two domains. The weak form of the problem is then augmented to incorporate a penalty for the jump across the interface in an appropriate manner, so that the convergence order of the scheme is retained. This approach is visualised in Fig. 2, where we note that the mortar elements are constructed at the common intersection points of each element.

An alternative approach to imposing non-conformal conditions, and one which is perhaps less commonly-used, is to instead adopt a point-to-point interpolation across the elemental interface. In this setting, no attempt to construct mortar elements is made and the function space is defined in the usual manner for each conformal domain. However, when values within elements are desired at the left-hand side of the interface, they are obtained by performing a polynomial interpolation from the values on the right-hand side, and vice versa. This approach was first implemented and tested for geophysical problems in [21] in the CG setting, where it was shown to demonstrate convergence-order preserving properties. Within the DG setting this allows for the values of the exterior conserved variables to be obtained from interpolating the interior trace values on the opposing side of the interface. A sample visualisation of this approach is shown in Fig. 3, where dotted arrows denote the evaluation of the high-order polynomial defined by points on the edge of element Ω_A to obtain their values within the boundaries of elements in $\partial\Omega_B$ and $\partial\Omega_C$. This interpolation process can be built into the assembly operation that is used to construct mass and stiffness matrices.

1.2. Non-conformal techniques for the discontinuous Galerkin method

At present, there is a significant interest in the development of high-order fluid dynamics solvers for iLES/uDNS based around the DG method due to its favourable stability properties in these settings [22–24]. In the context of sliding mesh simulations, DG also offers an easier route to the accurate treatment of non-conformal interfaces between elements across the sliding interface, since elements are naturally disconnected as part of the formulation

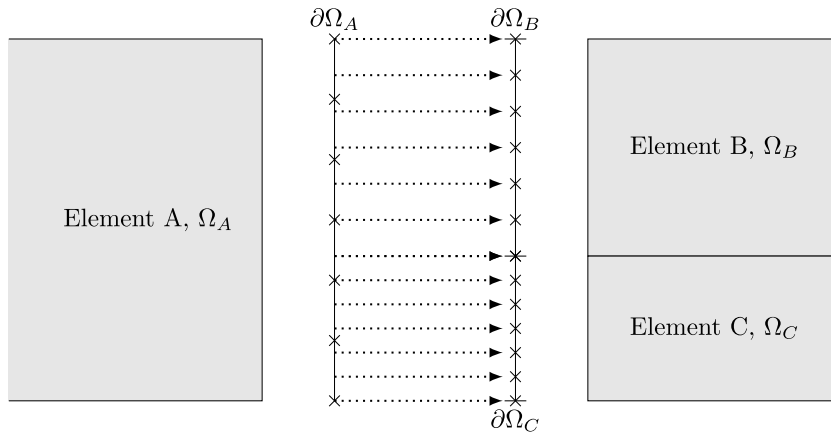


Fig. 3. Schematic representation of point-to-point interpolation across a non-conformal interface. Crosses represent the integration points on the respective trace of each element. Dotted arrows denote a high-order evaluation of the polynomial on the edge of element A at the points required for flux evaluations of elements B and C.

of the method. Additionally, approaches to impose non-conformal interfaces in DG are perhaps less well-explored than in CG.

In the DG formulation, connectivity between elements is imposed through a flux term, which may be either an upwind-type solver for linear problems or a more complex Riemann problem for more general nonlinear hyperbolic systems. These fluxes are computed on integrals across each edge of an element and take the form

$$\int_{\Gamma_e} \tilde{\mathbf{f}}(\mathbf{u}^+, \mathbf{u}^-) \cdot \mathbf{n} \, ds$$

where Γ_e is an edge of element Ω^e , \mathbf{u}^+ and \mathbf{u}^- is a vector of conserved variables on the exterior and interior of the element respectively, $\tilde{\mathbf{f}}$ is the numerical flux function and \mathbf{n} is an outwards-facing normal. The question then is how one computes these integrals, given that the exterior values \mathbf{u}^+ may now lie across more than one element on the other side of the interface.

The mortaring approach has been investigated in a number of works from the DG perspective. First, we note that unlike the CG setting which requires modifications to the function space and weak form of the problem, in DG by ‘mortaring’ we only refer to the act of constructing mortar elements on which to compute the flux integral. That is, the integral above is split into multiple integrals, one for each mortar element. This approach was first investigated by Kopriva et al. in the study of both fluid dynamics [25] and electromagnetics problems [26]. The same approach has been used fairly extensively for problems involving sliding meshes; for example by [27] in a hybrid DG-Fourier pseudospectral solver for the incompressible Navier–Stokes equations, in the construction of a spectral difference solver for the compressible Navier–Stokes equations that incorporates sliding grids by Zhang and Liang [28] and more recently in the hyperbolic solver FLEXI [29]. This approach has the significant advantage that it preserves the local conservation property of the DG method, which is important from the perspective of obtaining accurate results that conserve mass (in the case of CFD). However, although mortaring is straightforward in two dimensions, a significant challenge in the use of the mortar approach for general three-dimensional problems is the generation of the mortar elements themselves. In general this could involve the re-meshing of the non-conformal interface between domains at each timestep in order to generate an appropriate mortar space, as adopted by Aguerre et al. [30] for finite volume simulations based on the supermesh construction of Farrell et al. [31]. We note that these works consider only straight-sided elements. In some settings, such as a sliding mesh defined by a translation, this approach could therefore readily be adopted to a high-order setting. However, in problems involving rotation such as in Fig. 1, the interface between regions now becomes curved, thereby significantly increasing the complexity and computational requirements in this approach.

The alternative approach is therefore to consider the point-to-point interpolation method in the DG context, since implementation is relatively straightforward by comparison as it does not require the construction of mortar elements. However, neither the implementation, performance or robustness of this approach for has been thoroughly

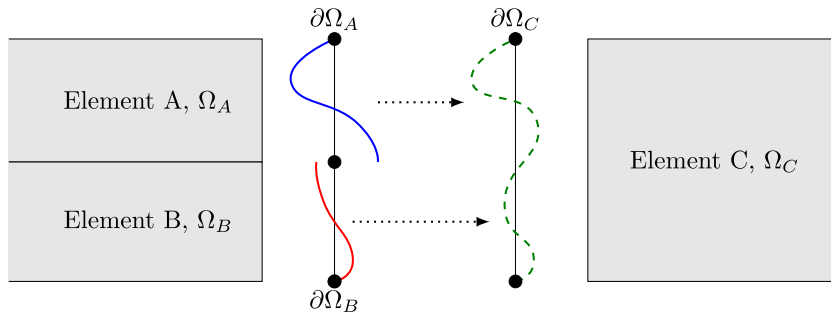


Fig. 4. Interpolation across a non-conformal 2D interface showing the solution of element A (—), element B (—) and the resultant discontinuity in the solution of Element C (---), with element boundaries indicated by the circles (•).

investigated in the literature to date. In particular, potential issues may arise from the discontinuity of fluxes between elements: for smooth solutions and at high polynomial orders, the interpolation between neighbouring non-conformal elements will likely introduce very little error into the resulting solution. However, in the presence of under-resolved simulations, which are more prone to admitting discontinuities in flow solution between elements, the discontinuity may introduce additional numerical error that warrants further study. A prototypical example which demonstrates this in an illustrative manner is shown in Fig. 4. On the left side of the interface, the two discontinuous solutions from elements Ω_A and Ω_B must be sampled at integration points on the skeleton of Ω_C . If the two functions are sufficiently discontinuous, the interpolation procedure could result in spurious noise introduced into the interior of Ω_C .

1.3. Aim of this work

To date, the point-to-point method has not been well-studied in the literature. A study by Kopera and Giraldo [32] is one of the very few references, to the best of the authors' knowledge, that consider the point-to-point interpolation approach in DG, where CG and DG implementations of the interpolation technique are examined and their mass conservation properties are reported. However we note that in this case, only hanging-node type vs. more generic non-conformal interfaces are considered. Additionally, this work was performed in well-resolved cases which may not be the case for more general iLES/uDNS-type problems. In this article, we therefore aim to address this gap in the understanding of the performance of these approaches by performing a comparative study of the mortar and point-to-point techniques. We consider several aspects, including a validation of convergence order for both approaches, the performance of each method in terms of numerical diffusion for a linear transport equation at varying degrees of underresolution, the behaviour of each method when considering the nonlinear problem of the compressible Euler equations across long time periods, and the compressible Navier–Stokes equations in a 3D setting.

The remainder of the paper is structured as follows. In Section 2 we set out the theoretical framework of the two formulations and outline our implementation strategy within the spectral/ hp element framework Nektar++ [33,34]. Section 3 presents the results of our studies for a linear transport equation and the nonlinear compressible Euler equations in two dimensions. In Section 4, we consider more realistic fluid simulations in three dimensions, by examining the transition to turbulence in a canonical Taylor–Green vortex problem. Finally, in Section 5, we draw some brief conclusions and discuss the key performance characteristics of each method.

2. Theory

2.1. The DG formulation of the spectral/ hp element method

In this section, we briefly introduce the discontinuous Galerkin (DG) discretisation of the spectral/ hp element method. A more thorough overview can be found in several other works, e.g. [33–35]. The starting point for the DG formulation is the same as any other typical finite element problem, in that we consider a domain Ω comprised

of non-overlapping elements Ω^e such that $\Omega = \bigcup_e \Omega^e$. Given a general hyperbolic conservation law for conserved variables \mathbf{u} taking the form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = 0, \tag{1}$$

we follow the standard Galerkin approach and, on a single element, construct the weak form via multiplication by a test function v and integrating by parts to obtain

$$\left(v, \frac{\partial \mathbf{u}}{\partial t} \right)_{\Omega^e} + \left\langle v \mathbf{n}, \tilde{\mathbf{f}}(\mathbf{u}^+, \mathbf{u}^-) \right\rangle_{\partial \Omega^e} - (\nabla v, \mathbf{F}(\mathbf{u}))_{\Omega^e} = 0, \tag{2}$$

where $(u, v)_{\Omega^e} = \int_{\Omega^e} uv \, d\mathbf{x}$ and $\langle u, v \rangle_{\partial \Omega^e} = \int_{\partial \Omega^e} uv \, ds$ denote inner products on the volume and surface, respectively. Moreover, $\tilde{\mathbf{f}}$ defines a numerically-calculated flux term which, as explained in the previous section, may take the form of a general Riemann problem, and which depends on the element-exterior and interior velocities \mathbf{u}^+ and \mathbf{u}^- , respectively. Within each element, we represent \mathbf{u} using an expansion of high-order polynomials, so that

$$\mathbf{u}^\delta = \sum_n \hat{\mathbf{u}}_n \phi_n([\chi^e]^{-1}(\mathbf{x})).$$

In this expression, we note that the approximation is defined with the use of a standard (reference) element Ω_{st} , with ϕ_n denoting an appropriate set of basis functions. An isoparametric mapping $\chi^e : \Omega_{st} \rightarrow \Omega^e$ defines a possibly curvilinear element Ω^e , so that $\mathbf{x} = \chi^e(\boldsymbol{\xi})$ for $\boldsymbol{\xi} \in \Omega_{st}$. We additionally equip the standard element with a distribution of quadrature points $\boldsymbol{\xi}_q$ and weight w_q , so that upon selecting test functions $v = \phi_n$ we then evaluate the terms in Eq. (2) as finite summations, i.e.

$$(\nabla \phi_n, \mathbf{F}(\mathbf{u}))_{\Omega^e} \approx \sum_q \nabla \chi^e(\boldsymbol{\xi}_q)^{-T} \nabla \phi_n(\boldsymbol{\xi}_q) \cdot \mathbf{F}(\mathbf{u}(\mathbf{x}_q)) \det(\chi^e(\boldsymbol{\xi}_q)) w_q$$

In this study, we consider only two-dimensional elements and select tensor products of Gauss–Lobatto–Legendre points to evaluate quadrature. As basis functions we adopt the hierarchical modified basis of Karniadakis & Sherwin [35]. Similarly to the classical Lagrange basis, these basis functions have the beneficial property of boundary-interior decomposition, which makes the addition of flux terms into the overall elemental degrees of freedom a straightforward addition operation. In particular we note that the flux integral terms can be considered along each edge i of Ω^e , which we denote by Γ_i^e , as the integral

$$\int_{\Gamma_i^e} \psi_n(\boldsymbol{\xi}) \tilde{\mathbf{f}}(\mathbf{u}^+, \mathbf{u}^-) ds$$

where now ψ_n denotes a basis function with support along edge i . In particular, we note that the solution variables along $\mathbf{y} \in \Gamma_i^e$ can be written as a polynomial expansion

$$\mathbf{u}(\mathbf{y}) = \sum_n \hat{\mathbf{u}}_n \psi_n(\mathbf{y}) \tag{3}$$

As alluded to in the introduction, the central focus of this work is to understand how different evaluations of the flux term in the presence of a non-conformal mesh influence the overall properties and stability of the DG method. In the following sections, we outline the formulation of both the point-to-point interpolation method and the mortar method.

2.2. The point-to-point interpolation method

In the point-to-point interpolation method, the interface is handled using a direct interpolation from one side to the other. That is, when we require the values of exterior conserved variables \mathbf{u}^+ at a spatial position $\mathbf{y}^* \in \Gamma_i^e$, we adopt the following procedure:

- determine a corresponding element Ω^f that contains the point \mathbf{y}^* along an edge Γ_j^f ;
- perform a polynomial interpolation at that position using Eq. (3) in order to determine \mathbf{u}^+ .

This interpolation is performed for every integration point along Γ_i^e , as shown diagrammatically in Fig. 3. Once the trace space (i.e. the collection of all edges in the interface) has been fully populated by interpolation, the DG solver can continue as usual with a Riemann solver to calculate the numerical flux to then be added into elemental coefficient spaces. In order to determine a corresponding element that contains the point \mathbf{y}^* , we require the ability to determine the distance of a desired point from any given edge Γ_j^f . For edges that are straight-sided, this translates into a simple geometric problem which may be solved analytically. However, for curvilinear elements, we must instead utilise the parametric mapping for the two-dimensional element, which gives a coordinate mapping $\mathbf{x} = \boldsymbol{\chi}^e(\boldsymbol{\xi})$. In particular, for each edge in the non-conformal interface, we minimise an objective function $d(\boldsymbol{\xi}; \mathbf{y}^*) = \|\mathbf{x} - \mathbf{y}^*\|_2^2 = \|\boldsymbol{\chi}^e(\boldsymbol{\xi}) - \mathbf{y}^*\|_2^2$, i.e. the square of the Euclidean norm $\|\cdot\|_2$ between a point $\boldsymbol{\xi}$ within the edge and the target point \mathbf{y}^* . This then allows us to determine the corresponding reference space point $\boldsymbol{\xi}^* = \min_{\boldsymbol{\xi}} d(\boldsymbol{\xi}; \mathbf{y}^*)$, so that $\boldsymbol{\chi}(\boldsymbol{\xi}^*)$ has minimum distance to \mathbf{y}^* . An edge that has $\boldsymbol{\chi}(\boldsymbol{\xi}^*) \approx \mathbf{y}^*$ is chosen as the corresponding edge Γ_j^f . In our implementation, this is solved via a gradient-descent method utilising a quasi-Newton search direction and backtracking line search, but other Newton-type methods will provide similar convergence properties. Since this is additionally an expensive operation to be performed for every edge within the interface, we make use of an r -tree structure to reduce the initial search space. The octants that are used to construct the r -tree are defined as the bounding box for each curvilinear edge. In this manner, the r -tree can first be interrogated to determine a subset of possible edges under which to then perform the nonlinear optimisation of distance, which further reduces computational cost.

Finally, in the minimisation process above, we require the evaluation of each polynomial expansion (3) at any arbitrary point $\boldsymbol{\xi}$ in the reference element. Although this can be computed directly from Eq. (3), this would require the evaluation of each basis function at the same arbitrary point. We note that, for numerical integration purposes within the DG scheme, we already naturally represent \mathbf{u} at solution points $\mathbf{y}_q = \boldsymbol{\chi}^e(\boldsymbol{\xi}_q)$ that correspond with Gauss–Lobatto quadrature points in the reference element $\boldsymbol{\xi}_q \in [-1, 1]$. This allows us to rewrite Eq. (3) as a summation in terms of Lagrange interpolants $\ell_q(\boldsymbol{\xi})$ defined using these same points, so that

$$\mathbf{u}(\mathbf{y}) = \sum_q \mathbf{u}(\mathbf{y}_q) \ell_q(\boldsymbol{\xi}).$$

Classically, given this collocation representation, one would then generate a diagonal interpolation matrix $\mathcal{I}(\mathbf{y})$ as outlined in [35], and perform a dot product against a vector of points $\mathbf{u}(\mathbf{y}_q)$ to evaluate $\mathbf{u}(\mathbf{y})$. However, our timings demonstrate that the use of fast summation based on barycentric interpolation techniques described in [36] yield far better performance for this operation. The extension of this technique to higher dimensions is discussed further in Section 4.

2.3. The mortar method

The second approach we will consider in this paper is the mortar method which maintains the local conservation properties of DG by constructing mortar elements as visualised in Fig. 2. This method applied to the spectral element method was originally developed by Maday et al. [37], and has been used for both incompressible flow [20] and compressible flow problems [25]. We note again that ‘mortar’ in this sense refers to the act of construction of mortar elements so that flux integrals may be expressed as

$$\int_{\Gamma_i^e} \psi_n(\boldsymbol{\xi}) \tilde{\mathbf{f}}(\mathbf{u}^+, \mathbf{u}^-) ds = \sum_{m=1}^M \int_{\Xi_m} \psi_n(\boldsymbol{\xi}) \tilde{\mathbf{f}}(\mathbf{u}^+, \mathbf{u}^-) ds$$

where M is the number of mortars on edge i , and Ξ_m denotes each mortar element. We then construct a polynomial expansion on each mortar element of the same polynomial order. The mortar method is realised by projecting variables from across the interface onto its corresponding mortar element, solving the Riemann problem on the mortars, and then performing an L^2 projection in order to consolidate the contributions from each mortar element. The number of mortars connected to a single interface edge and their relative size is arbitrary, allowing for a wide range of varying mesh circumstances. To give a more concrete definition of the method, we utilise the notation prevalent in Zhang and Liang [28] and Kopriva et al. [26], labelling the two contributing interface segments ‘L’ and ‘R’ as shown in Fig. 5.

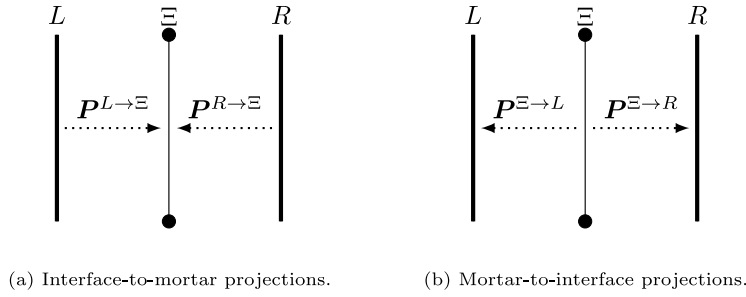


Fig. 5. The left, L , and right, R , interface edges projection relationships from and to the mortar element, Ξ .

First we recall that each edge in the interface Γ_i^e may be represented on a standard segment $-1 \leq \xi \leq 1$ and then mapped using the isoparametric mapping χ^e . Similarly, each mortar element has a similar mapping $-1 \leq z \leq 1$ and, in particular, we may write the relationship between the two as

$$\xi = o + sz$$

where o is the offset of the centre of the mortar relative to the centre of the interface edge, and s is the relative scale factor. The solution on an interface edge can be represented by Eq. (3), so that

$$u^\Omega(\xi) = \sum_p \hat{u}_p^\Omega \phi_p(\xi),$$

where we consider now only a single scalar quantity u for clarity. We can similarly define the solution on the mortar element, Ξ , as

$$u^\Xi(z) = \sum_p \hat{u}_p^\Xi \phi_p(z).$$

To project the solutions from the element onto the mortar we minimise the norm in the L^2 sense, i.e.

$$\int_{-1}^1 (u^\Xi(z) - u^\Omega(\xi)) \phi_j(z) dz = 0, \quad \text{for all } j.$$

When evaluated at all quadrature points, this can be expressed in matrix form as

$$\hat{u}^\Xi = \mathbf{P}^{\Omega \rightarrow \Xi} \hat{u}^\Omega = \mathbf{M}^{-1} \mathbf{S}^{\Omega \rightarrow \Xi} \hat{u}^\Omega,$$

where \mathbf{M} is the standard elemental mass matrix, and $\mathbf{S}^{\Omega \rightarrow \Xi}$ are constructed as

$$S_{i,j} = \int_{-1}^1 \phi_i(o + sz) \phi_j(z) dz, \quad \text{for all } i, j.$$

To apply the mortar method to the DG formulation, we therefore adopt the following approach:

- Construct both the left and right solutions u^+ and u^- onto the mortar using the projection matrices $\mathbf{P}^{L \rightarrow \Xi}$ and $\mathbf{P}^{R \rightarrow \Xi}$ as shown in Fig. 5(a).
- Once the solutions are on the mortar, the Riemann solver can be used to compute the numerical flux \tilde{f} .
- Projecting from the mortars back onto the interface element requires minimising the trace quantities norm in the L^2 sense. For M mortars to the interface element Ω this is as follows

$$\sum_{i=1}^M \left[\int_{\Xi_i} (\hat{f}^\Omega(\xi) - \hat{f}^{\Xi_i}(z)) \phi_j(\xi) d\xi \right] = 0, \quad \text{for all } j.$$

In matrix form the solution to this is

$$\hat{f}^\Omega = \sum_{i=1}^N \left[\mathbf{P}^{\Xi_i \rightarrow \Omega} \hat{f}^{\Xi_i} \right] = \sum_{i=1}^N \left[s_{\Xi_i} \mathbf{M}^{-1} \mathbf{S}^{\Xi_i \rightarrow \Omega} \hat{f}^{\Xi_i} \right],$$

where $\mathbf{S}^{\Xi \rightarrow \Omega}$ is the transpose of $\mathbf{S}^{\Omega \rightarrow \Xi}$ taking care to include the respective scale factors.

Table 1

Convergence rates for the conformal case, the non-conformal mortar method, and the non-conformal point-to-point interpolation method.

Poly. order	Convergence rates for order p			
	3	5	7	9
Conformal	5.19	7.03	8.88	10.75
Mortar	4.69	6.30	8.26	9.98
Interp, $Q = P + 2$	4.09	5.89	7.78	9.56
Interp, $Q = 2P + 2$	4.33	6.00	8.05	9.98

It is also worth noting that for where the geometry of the interface element is identical to the mortar, for example between Ω_A and its corresponding mortar in Fig. 2, the projection matrix is merely the identity matrix so $\hat{\mathbf{u}}^\Xi = \hat{\mathbf{u}}^\Omega$ and $\hat{\mathbf{f}}^\Omega = \hat{\mathbf{f}}^\Xi$. This can be used to reduce computational costs.

3. Results

In this section, we report on the results of a number of two-dimensional tests using both linear and nonlinear problems to evaluate the efficacy of both the mortar and point-to-point interpolation technique. At each stage, we use conformal grids of similar resolutions to provide a benchmark against which to compare. Each method has been implemented within the Nektar++ spectral/ hp element framework [33,34]. In all cases, we consider only explicit timestepping methods with the use of a standard 4th-order Runge–Kutta time integration scheme unless otherwise stated. The timestep used for each case is reported separately.

3.1. Convergence order

In this first case, we test the correctness of our implementation by performing a standard h -convergence study for various polynomial orders p . For this, we select a standard linear transport equation within a domain $\Omega = [-5, 5]^2$, so that in Eq. (1), $\mathbf{F}(u) = \mathbf{v}u$ for a constant velocity $\mathbf{v} = (1, 0)$. We select an initial condition that is non-polynomial, so that $\mathbf{u}(\mathbf{x}, 0) = \sin(2\pi x) \cos(2\pi y)$ together with periodic boundary conditions on all edges, so that the solution propagates indefinitely. Regular grids are constructed using between 81 and 22,500 quadrilateral elements in the conformal case. The non-conformal case incorporates two interfaces to ensure that the periodic boundaries are conformal to one another for ease of implementation. This results in three sub-domains, with the central one shifted vertically by half a cell length to create a non-conformal grid. An example non-conformal grid is shown in Fig. 8(b). For a given value of h , the non-conformal mesh will therefore have a slightly higher number of elements than its conformal counterpart. Polynomial orders of $P = 3$ through $P = 11$ are considered for each grid, and we select $Q = P + 2$ quadrature points in each coordinate direction so as to exactly integrate the mass matrix and remove any spurious aliasing error due to the use of numerical integration. We select a timestep size of $\Delta t = 10^{-3}$ and measure the error after one tenth of a cycle (i.e. $t = 1$) so that error due to timestepping is reduced. In addition, for the point-to-point interpolation method we also investigate an additional setup with $Q = 2P + 2$ number of quadrature points to investigate the effect of dealiasing on the resulting error and to mirror the strategies employed in later sections.

Fig. 6 and Table 1 highlight the convergence properties in the L^2 sense of the two non-conformal methods, together with the conformal interface. To ensure clarity the results in the graph have been trimmed to remove points from each polynomial order after the minimum L^2 error has been reached owing to the finite precision being used. Convergence rates are approximated from the gradients of curves in Fig. 6, besides from at $p = 11$ where this is omitted due to lack of data points. The results of this study show that for smooth solutions and at higher orders the mortar method, dealiased interpolation method and conformal cases all show near identical results. These results ratify both that the solvers are implemented correctly and, moreover, that both non-conformal interface handling methods yield similar convergence rates of around $P + 1$, while in the conformal setting the rates are $P + 2$.

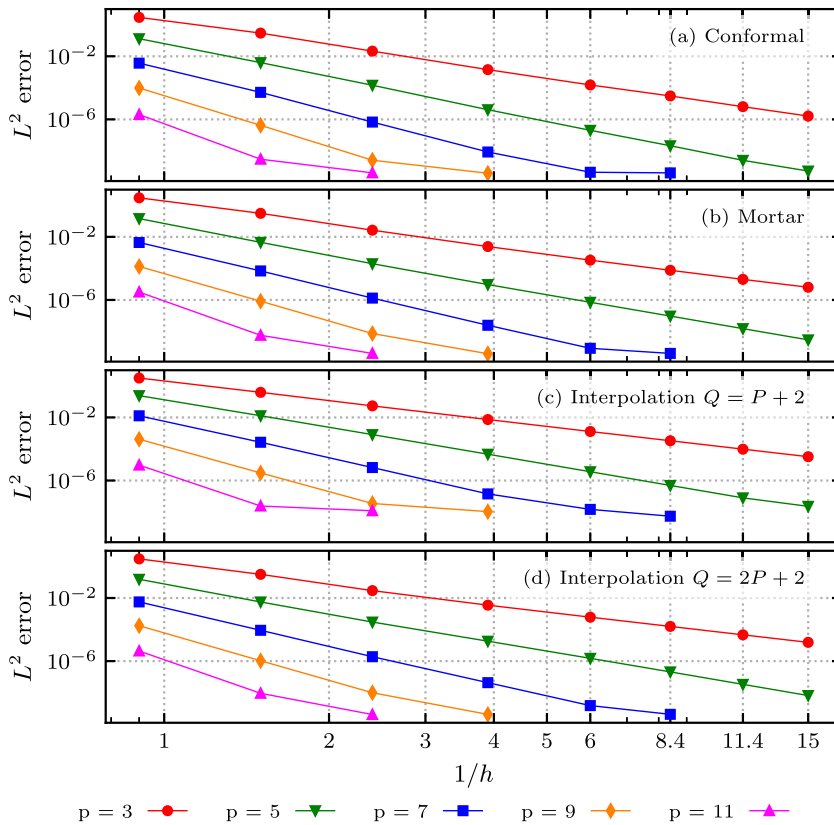


Fig. 6. Convergence properties for the conformal case, the non-conformal mortar method, and the non-conformal point-to-point interpolation method.

3.2. Decay properties

In order to more robustly validate each method, we now consider a more challenging problem at varying degrees of resolution. In order to evaluate the numerical diffusion that is introduced by the presence of an interface, we consider the rotation of a Gaussian in a circular manner using the transport equation. More precisely we utilise the same transport equation as the previous setting but now consider the velocity $\mathbf{v}(x, y) = (-x, y)$, so that the initial scalar Gaussian field $u(\mathbf{x})_{t=0} = \exp(-\|\mathbf{x} - \mathbf{x}_0\|^2 / \sigma^2)$ is rotated around the origin. We consider a domain $\Omega = [-2, 2]$, using an initial starting point $\mathbf{x}_0 = (-0.625, -0.625)$ with $\sigma = 0.1$. The mesh used in this test consists of 16×16 uniformly-sized quadrilateral cells for the conformal case, while in the non-conformal cases the right-hand half of the grid has been displaced by half a cell vertically along the central interface to the left-hand side, as shown in Fig. 7. Constructing the mesh in this way aims to keep a consistent cell density by ensuring the half cell height sections are on the extreme ends of the interface, distant from where the peak crosses the interface. We also note that the selection of \mathbf{x}_0 is designed to place the peak in the centre of a cell to the left of the interface, this is to ensure minimal interaction with the domain boundaries which all have a homogeneous Dirichlet condition imposed on them.

The peak starts at $t = 0$ with a maximum value of 1. Unlike the exact solution, which precisely preserves this peak indefinitely, the non-polynomial nature of the solution field means that we can expect the peak to decrease every rotational cycle due to numerical diffusion introduced by each method. We then measure the L^∞ norm of the solution precisely through a minimisation problem — i.e. we do not solely sample the error at quadrature points, as at lower orders very few quadrature points are used within each element, and this may lead to a significant difference in the observed error. We select a timestep size of $\Delta t = 10^{-3}$, and for each combination of polynomial order and interface handling method, we measure the L^∞ norm after 100 cycles of the Gaussian (i.e. $t = 200\pi$).

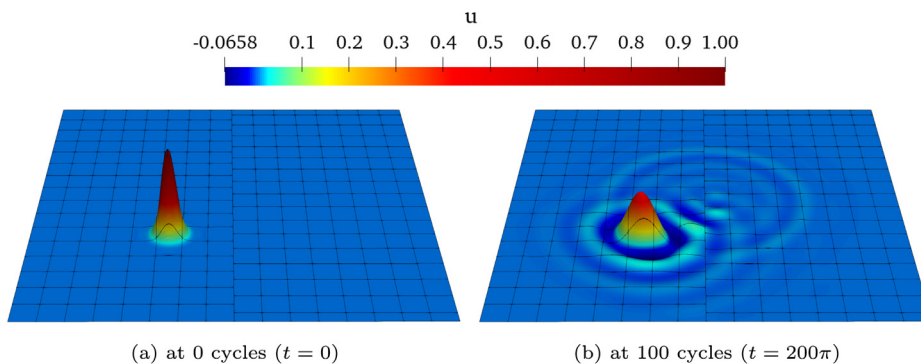


Fig. 7. The decay in the Gaussian peak over the 100 cycles for the 4th order point-to-point interpolation method.

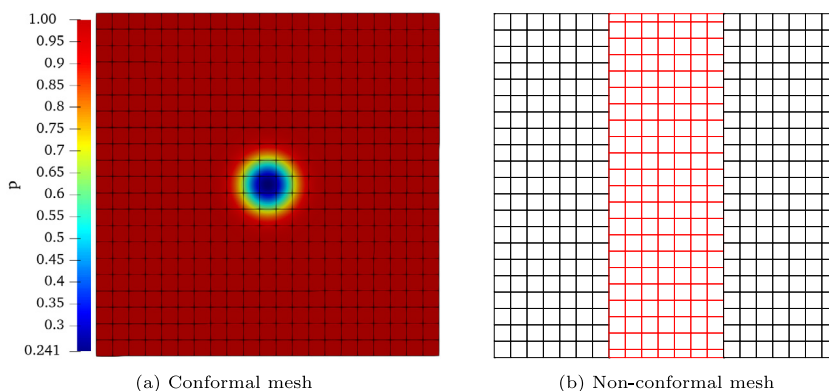


Fig. 8. The two meshes used for the isentropic Euler vortex cases, (a) the conformal case has the initial projection of the pressure field overlaid.

Table 2
Results of Gaussian peak value after 100 cycles for varying basis orders.

Poly. order	$\ \cdot \ _{L^\infty}$ after 100 cycles						
	4	5	6	7	8	9	10
Conformal	0.4403	0.6870	0.8291	0.9302	0.9758	0.9933	0.9984
Mortar	0.4387	0.6819	0.8286	0.9293	0.9756	0.9931	0.9984
Point-to-point	0.4812	0.7056	0.8394	0.9476	0.9720	0.9936	0.9980

We note that at lower polynomial orders, the solution will be underresolved by design — our aim in this series of simulations is to examine how this affects numerical stability across the methods and/or if there are significant differences in performance of the methods. The results of these experiments are shown in Table 2.

Two trends are evident from the presented results. At the lowest polynomial order of $P = 4$, we see a reasonable level of difference in the point-to-point method vs. the mortar and conformal grids. The oscillations of numerical error at this order are clearly evident in Fig. 7(b) which shows the 4th order point-to-point interpolation method after 100 cycles. Curiously, the values observed at $P = 4$ through $P = 7$ are higher for the point-to-point method than both the mortar/conformal cases, indicating that the point-to-point method is somewhat better able to resolve the peak of the Gaussian. It is also clear that as the polynomial order increases this difference in the maximum value decreases, so that at $P \geq 8$ the results are essentially identical. Broadly speaking, however, the performance of the methods is reasonably comparable across the range of polynomial orders.

3.3. Long-time advection of an isentropic vortex

In order to examine the non-conformal methods in more realistic problems, whilst still considering their long-term stability and diffusion properties, we now move on to consider a nonlinear problem. In particular, we consider the compressible Euler equations in two dimensions. In this case, the conserved variables are given as $\mathbf{u} = [\rho, \rho u, \rho v, E]$ with ρ being the density, (u, v) the fluid velocity, E the specific total energy, and

$$\mathbf{F}(\mathbf{u}) = \begin{bmatrix} \rho u & \rho v \\ p + \rho u^2 & \rho uv \\ \rho uv & \rho v^2 + p \\ u(E + p) & v(E + p) \end{bmatrix},$$

where p is the pressure. To close the system we need to specify an equation of state; in this case we use the ideal gas law $p = \rho RT$ where T is the temperature and R is the gas constant.

To consider long-term stability, we opt to study an isentropic vortex that is advected at constant velocity through periodic boundaries. This is a commonly used benchmark when testing numerical discretisation of the compressible Euler equations, particularly for higher-order codes, as it is one of the few problems that admits an exact solution calculable at all times whilst also being relatively simple to implement [38–41].

For our purposes, we consider a domain $\Omega = [-5, 5]^2$. At any given time t , the solution for the isentropic vortex is given by the equations

$$\begin{aligned} \rho &= \left(1 - \frac{\beta^2 (\gamma - 1) e^{2f}}{16\gamma\pi^2}\right)^{\frac{1}{\gamma-1}}, \\ u &= \left(u_0 - \frac{\beta e^f (y - y_0)}{2\pi}\right), \\ v &= \left(v_0 - \frac{\beta e^f (x - x_0)}{2\pi}\right), \\ E &= \frac{\rho^\gamma}{\gamma - 1} + \frac{1}{2}\rho (u^2 + v^2), \end{aligned} \tag{4}$$

where $f = 1 - (x - x_0)^2 + (y - y_0)^2$. We select an initial vortex position $(x_0, y_0) = (0, 0)$ with strength $\beta = 5$ and $\gamma = 1.4$, and advect the velocity in the x -direction with velocity $(u_0, v_0) = (1, 0)$. The initial vortex size and location can be seen in Fig. 8(a).

The concept behind this series of simulations is much the same as the rotating Gaussian peak; i.e. we wish to cycle the vortex through the domain a number of times, and compare the error as a function of time for each interface method. This will be undertaken for polynomial orders, $P = 3$ through $P = 7$ on a grid of fixed size, where the lower orders are expected to be under-resolved and the higher orders somewhat more resolved. To impose this, a single pair of periodic conditions at the constant x boundaries were used so that $\mathbf{u}(-5, y) = \mathbf{u}(5, y)$, while the constant y boundaries were set to free-stream conditions. Although it may seem more natural to impose periodic boundaries on all of the edges of the domain, in a similar fashion to [40] we found that this leads to a gradual accumulation of numerical error, which left unchecked eventually causes simulations to diverge. The solution, proposed in [40] and [41], is to impose farfield conditions at constant y boundaries, which allows recirculated waves of accumulated numerical error to escape the domain and avoid premature divergence. This is particularly important in this case, as in order to further reduce sources of artificial dissipation, we elect to use the exact Riemann solver of Toro [42] to calculate the numerical flux $\tilde{\mathbf{f}}(\mathbf{u}^+, \mathbf{u}^-)$. We note that although this is computationally expensive, cheaper solvers such as the Roe solver may introduce additional numerical diffusion [43].

One additional consideration that needs to be taken in this nonlinear regime is the order of integration used to evaluate integrals in the weak form of Eq. (2). Aliasing errors are a well known phenomenon in this regime, due to the cubic nonlinearity that arises in the definition of the Euler equations, as well as the non-polynomial flux term which calculated between elements [38]. Typically it is necessary to use a higher order of quadrature than is used for linear problems, in order to remove sources of aliasing error due to under-integration of these terms. For this reason, we consider two different numbers of quadrature points with $Q = P + 2$ and $Q = 2P + 2$, respectively.

The conformal case consists of a singular domain made up of a 21×21 regular quadrilateral mesh as shown in Fig. 8(a). The resulting non-conformal mesh consists of the domains 7×21 , 7×22 and 7×21 elements, as

Table 3Computational costs for the isentropic Euler vortex with $Q = P + 2$ cases.

Case	Average cost per timestep (s)				
	P3Q5	P4Q6	P5Q7	P6Q8	P7Q9
Conformal	0.0202	0.0299	0.0430	0.0577	0.0794
Point-to-point	0.0224	0.0325	0.0464	0.0617	0.0679
Mortar	0.0872	0.0983	0.1135	0.1297	0.1708

Table 4Computational costs for the isentropic Euler vortex with $Q = 2P + 2$ cases.

Case	Average cost per timestep (s)				
	P3Q8	P4Q10	P5Q12	P6Q14	P7Q16
Conformal	0.0359	0.0555	0.0848	0.1176	0.1646
Point-to-point	0.0366	0.0558	0.0848	0.1185	0.1638
Mortar	0.1112	0.1307	0.1872	0.2325	0.2696

shown in Fig. 8(b). The periodic boundary also allows us to conveniently express the time in cycles, where one cycle is the length of time taken for the vortex to propagate through the domain and return to its initial position. In our case, with a propagation speed of $u_0 = 1$ and domain length $L = 10$, this leads to the same exact solution every $t = 10$. We can calculate the exact solution at any time, t , by moving the vortex centre by $u_0 t$ in the x -direction and making sure to account for the periodic condition. We select a fixed timestep size of $\Delta t = 10^{-3}$ and use the same explicit 4th-order Runge–Kutta timestepping scheme as in previous results.

In Fig. 9, we visualise the L^2 error of the density field ρ , denoted by L_ρ^2 , for a simulation spanning 100 cycles of the vortex through the domain. This figure yields a number of interesting features that warrant further discussion. Firstly, as validation of our results, we note that the broad characteristics of the conformal error broadly agree with those seen in other work and, in particular, those of [40]. More generally, we observe that the conformal method and mortar method yield extremely close results for all polynomial and quadrature orders under observation, which we would expect given the similar levels of resolution and the local conservation properties of the mortar method.

However, when considering the point-to-point interpolation method, there are indeed clear differences in comparison to the mortar and conformal methods. Perhaps the most obvious peculiarity is the possible relationship between odd numbers of quadrature points and the long term stability of the interpolation method; P3Q5 and P5Q7 in Fig. 9 show significant divergence at low cycle counts. The errors in this case appear to be related to aliasing error: as the integration order is increased to $Q = 8$ and $Q = 12$ respectively, the results remain consistent with those found by the mortar method and the reference conformal case.

To investigate the effect of aliasing and integration order further, additional point-to-point interpolation simulations were run at $P = 5$ with quadrature orders ranging between $Q = 7$ and $Q = 13$. Fig. 10 depicts the L_ρ^2 error for these cases. The pronounced abnormality at $Q = 7$ is clearly visible, and indeed at $Q = 8$, there is a sudden increase in error after ~ 70 cycles which is indicative of further long-time increases in error. However, for $Q \geq 9$, we observe much more consistent trends and better agreement with the mortar and conformal cases. More generally then, we can state that so long as appropriate levels of aliasing are used so that $Q = 2P + 2$, the interpolation method closely follows the same trend as the mortar method and the benchmark conformal case, with the same reduction in error as polynomial order increases. These cases are all visualised in a single Fig. 11 to highlight this more clearly.

In addition we also used this problem to investigate the computational costs associated with each interface method. These simulations were run on a single core of a dual-socket Intel Xeon Gold 5120 system, equipped with 256 GB of RAM, with the solver pinned to a specific core in order to reduce the influence of kernel core and socket reassignment mid-process. The time taken per timestep for the $Q = P + 2$ cases is shown in Table 3, and the $Q = 2P + 2$ cases in Table 4. This shows that the conformal and point-to-point interpolation simulation timings are very similar with a small percentage cost associated with the interpolation. Of note are the $P = 7$ results, which show the point-to-point interpolation cost as lower than the equivalent conformal case. A further investigation of this

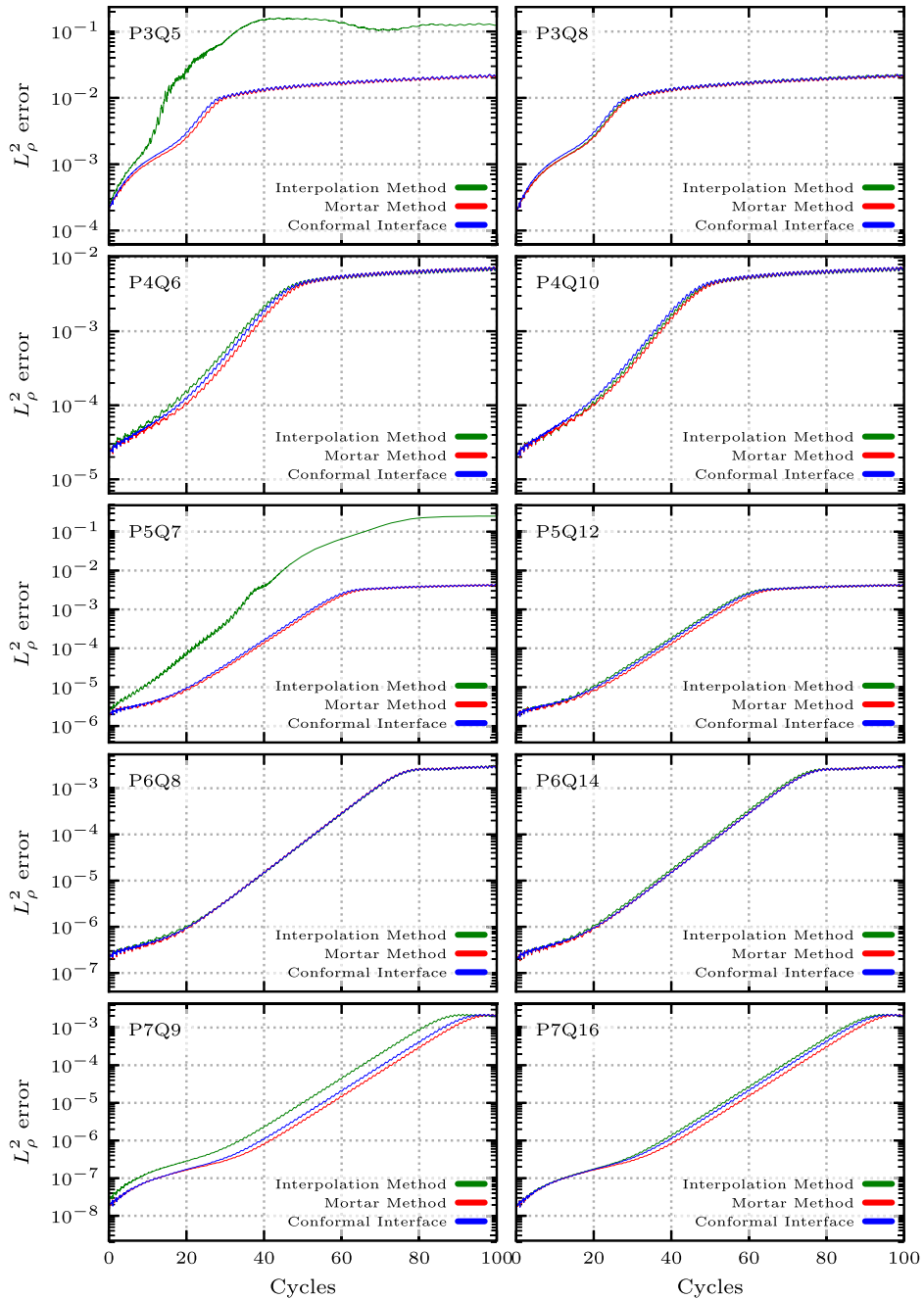


Fig. 9. The evolution of the L_p^2 error as a function of cycles of the vortex through the domain. Rows of the figure denote polynomial order $P = 3$ through $P = 7$, and columns denote integration orders $Q = P + 2$ and $Q = 2P + 2$, respectively.

case showed that the small variation in the number of elements leads to a reduction in walltime of the evaluation of DG volume terms, possibly owing to the strategies used in Nektar++ to evaluate collective operations more effectively [44]. The mortar method shows a larger cost, which decreases in proportion to the other methods as the polynomial order and quadrature point number is increased. For example, at P3Q5 it is over four times as expensive, whilst at P7Q16 it is less than double the conformal cost. This suggests that in the mortar method cases, at least in this setup, the total computational cost is dominated by the interface handling. This is to be expected,

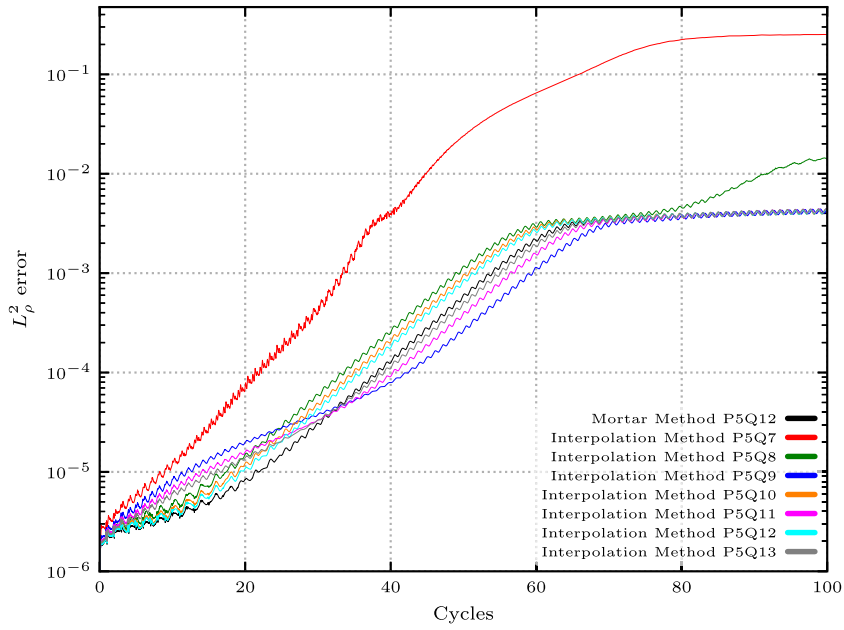


Fig. 10. The evolution of the L_p^2 error over time for $P = 5$ with the interpolation method at various quadrature point numbers. P5Q12 for the mortar method is also shown as a comparison.

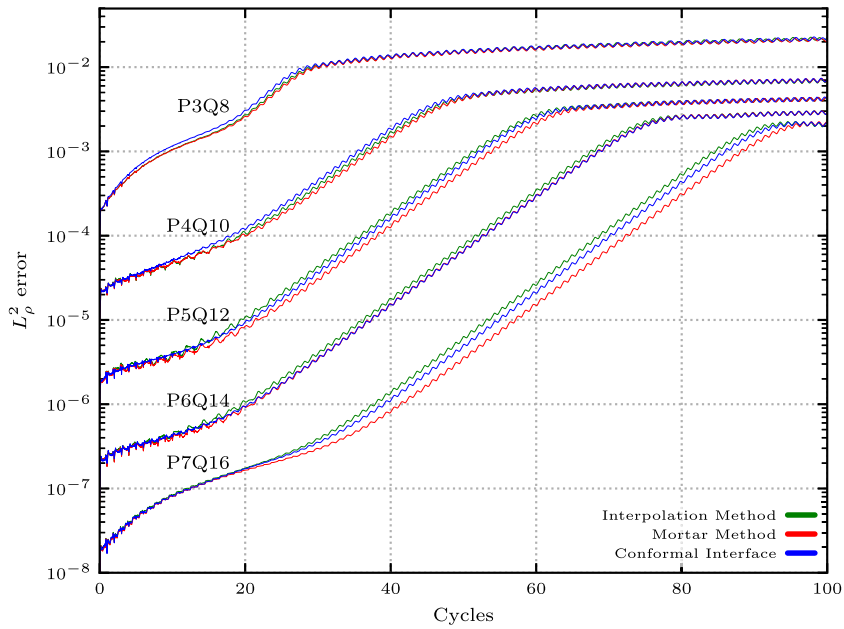


Fig. 11. The evolution of the L_p^2 error over time for P3, P4, P5, P6 and P7 with $Q = 2P + 2$.

since the projection both to and from mortars involves more costly operations than a straightforward interpolation in the point-to-point approach. We note here that although we have taken some steps to optimise the mortar method implementation (e.g. through the caching of \mathbf{S} matrices defined in the previous section), more in-depth techniques such as those found in [45] may help in reducing walltime for parallel execution in particular.

4. Extension to three-dimensional cases

In this section we consider the extension of our two-dimensional simulations to a realistic three-dimensional fluid dynamics case. In particular, we consider the extension to the compressible Navier–Stokes equations, which in conservative form may be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \nabla \cdot \mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u}), \tag{5}$$

where $\mathbf{u} = [\rho, \rho u, \rho v, \rho w, E]$ is the vector of conserved variables in terms of density ρ , velocity $\mathbf{v} = (u_1, u_2, u_3) = (u, v, w)$ and E is the specific total energy. In three dimensions we have that

$$\mathbf{F}(\mathbf{u}) = \begin{bmatrix} \rho u & \rho v & \rho w \\ p + \rho u^2 & \rho uv & \rho uw \\ \rho uv & \rho v^2 + p & \rho vw \\ \rho uw & \rho vw & \rho w^2 + p \\ u(E + p) & v(E + p) & w(E + p) \end{bmatrix},$$

We again use the ideal gas to close the system. The tensor of viscous forces $\mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u})$ is defined as

$$\mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} 0 & 0 & 0 \\ \tau_{xx} & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \tau_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \\ A & B & C \end{bmatrix},$$

with

$$A = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\partial_x T,$$

$$B = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\partial_y T,$$

$$C = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\partial_z T,$$

where in tensor notation the stress tensor $\tau_{x_i x_j} = 2\mu(\partial_{x_i} u_j + \partial_{x_j} u_i - \frac{1}{3}\partial_{x_k} u_k \delta_{ij})$, μ is the dynamic viscosity calculated using Sutherland’s law, k is the thermal conductivity and δ_{ij} is the Kronecker delta.

From a numerical perspective, we adopt the same discontinuous Galerkin formulation to discretise Eq. (5). However we note that the inclusion of the viscous term $\mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u})$ requires additional treatment, in particular a careful selection of flux terms in order to preserve spatial accuracy. In the simulations below, we adopt the local discontinuous Galerkin (LDG) approach, wherein an auxiliary variable $\mathbf{q} = \nabla \mathbf{u}$ is introduced and discretised alongside Eq. (5). With a careful choice of alternating fluxes (so that $\tilde{\mathbf{q}} = \mathbf{q}^+$ and $\tilde{\mathbf{u}} = \mathbf{u}^-$, or vice versa), high-order accuracy can be preserved [46].

4.1. Implementation considerations

In order to extend the formulation in Section 2 to three dimensions for a non-conformal Navier–Stokes simulation, consideration has to be given to a number of implementation changes, which we briefly outline in this section.

Both the mortar and point-to-point interpolation method require the evaluation of the solution at arbitrary points within the skeleton of the mesh. In 2D simulations, this requires evaluation within an interval; however in 3D this could conceivably be evaluated in either quadrilateral or triangular faces, depending on the element type: for example hexahedra possess purely quadrilateral faces, whereas tetrahedra possess triangular faces. Although the barycentric evaluation approach proposed in Section 2 naturally extends to higher-dimensional quadrilaterals and hexahedra through a tensor product of one-dimensional evaluations, for triangular elements and other three-dimensional shapes, most discontinuous Galerkin implementations documented in the literature are based around the selection of a set of cubature points combined with Lagrange interpolants as basis functions. Typical examples of such distributions are the Fekete [47] or electrostatic points [48], which both provide better conditioning of operators when compared to evenly-spaced points [35]. In this case, interpolation may be done via the typical route of generating an interpolation matrix with the aid of a Vandermonde matrix, as outlined in [49].

However, we note that in the spectral/ hp formulation of Karniadakis & Sherwin [35] which forms the numerical basis for Nektar++, higher d -dimensional simplices, as well as other hybrid shape types such as prisms and tetrahedra, are represented instead on a collapsed coordinate space, denoted by $\boldsymbol{\eta} \in [-1, 1]^d$. We give a brief overview of this formulation here, leaving further details to the aforementioned reference. Each collapsed coordinate spaces is mapped to the desired reference elemental shape through the use of Duffy transformations. For example, in a triangular element with reference coordinates $\{(\xi_1, \xi_2) \mid \xi_1, \xi_2 \geq -1, \xi_1 + \xi_2 \leq 0\}$, we have that

$$\eta_1 = 2 \frac{1 + \xi_1}{1 - \xi_2} - 1, \quad \eta_2 = \xi_2$$

Evaluation of quadrature therefore occurs on the collapsed space which, being a quadrilateral or hexahedron in two or three dimensions, may be equipped with a set of tensor-product integration points. Typically, this is chosen to be a set of Gauss–Lobatto points in the η_1 direction, and Gauss–Radau points in the η_2 direction in order to avoid explicit evaluations near the (removable) singularity in the Duffy transformation which occurs when $\xi_2 = 1$. In this manner, barycentric interpolation can still be applied in order to increase the computational speed of the simulation for triangular elements, as well as other higher-dimensional shapes. This is a topic of broader interest and under investigation in a wider range of areas [50].

Another significant issue to overcome in the case of mortaring is the construction of the mortar space. As noted in the preceding sections, for generic interfaces between unstructured grids, this can pose a significant challenge, although several techniques have been demonstrated in the literature to handle cases where the geometry is extruded and thus elicits structure that can be exploited. In the remainder of this section, we opt therefore to consider only the point-to-point interpolation approach, since the preceding section clearly demonstrates near-identical behaviour when compared to normal conformal simulations, and the aforementioned references demonstrate the viability of this approach in three-dimensional simulations. The more pertinent question is therefore how the point-to-point interpolation approach performs in this setting, which has yet to be examined in these cases to the best of the authors' knowledge.

4.2. Simulation of a Taylor–Green vortex

In order to examine the performance of the point-to-point interpolation method, we consider the simulation of a Taylor–Green vortex at a Reynolds number $\text{Re} = 1600$, which has become a benchmark case for the evaluation of higher-order CFD codes. In this case, starting vortices are defined in a periodic box $\Omega = [-L\pi, L\pi]^3$, given a reference length L , which break down into turbulent eddies before decaying due to viscous effects. The initial conditions are given in primitive variables (\mathbf{v}, p) as

$$\begin{aligned} u &= V_\infty \sin(x/L) \cos(y/L) \cos(z/L), \\ v &= -V_\infty \cos(x/L) \sin(y/L) \cos(z/L), \\ w &= 0, \\ p &= \rho_\infty V_\infty^2 \left[\frac{1}{\gamma \text{Ma}_\infty^2} + \frac{1}{16} (\cos(2x/L) + \cos(2y/L)) \cdot (\cos(2z/L) + 2) \right], \end{aligned}$$

with the Reynolds number $\text{Re} = \rho_\infty U_\infty L / \mu$ and the Prandtl number $\text{Pr} = 0.71$. Although the Taylor–Green vortex problem is traditionally examined in the setting of an incompressible flow, we approach this limit by considering flows with low compressibility effects so that the Mach number $\text{Ma}_\infty = 0.1$. A simulation is then conducted across the time interval $t_c \in [0, 20]$, where the convective timescale $t_c = t V_\infty / L$. We select an explicit second-order Runge–Kutta time integration scheme, with the timestep adjusted to maintain a Courant–Friedrichs–Lewy (CFL) condition of 0.2.

4.2.1. Kinetic energy dissipation rate

A key quantity of interest in this simulation is the evolution of the kinetic energy dissipation rate ϵ , where

$$\epsilon = -\frac{dE_k}{dt}, \quad E_k = \frac{1}{\rho_\infty |\Omega|} \int_\Omega \frac{1}{2} \rho \|\mathbf{v}\|^2 d\mathbf{x},$$

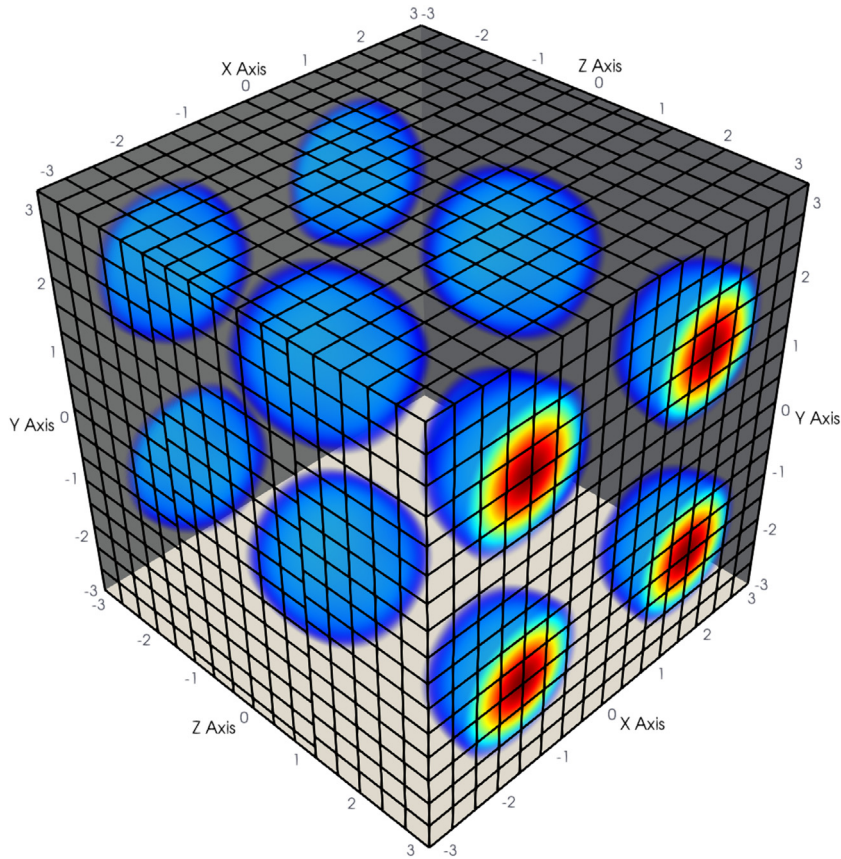


Fig. 12. Non-conformal 3D mesh for the 64^3 DOF Taylor–Green vortex case showing $\|\omega\|$ at $t_c = 0$.

since the peak dissipation is a difficult quantity to resolve closely for under-resolved simulations. Examination of ϵ therefore gives an indication as to the numerical performance of the scheme and inherent numerical diffusion. For an incompressible fluid, $\epsilon = 2\mu\mathcal{E}/\rho_\infty$ where the enstrophy \mathcal{E} is computed as

$$\mathcal{E} = \frac{1}{\rho_\infty|\Omega|} \int_{\Omega} \frac{1}{2}\rho\|\omega\|^2 d\mathbf{x}.$$

This equality does not strictly hold for a compressible fluid. However, as the additional contributions that appear in the exact expression depend on the divergence of the velocity, for this close-to-incompressible case their contribution is very small and can therefore be omitted.

In Fig. 13 we visualise the evolution of ϵ from a number of simulations:

- reference DNS data from a spectral simulation using 512^3 grid points in each spatial direction;
- simulations on a conformal mesh with 16^3 and 32^3 equally-sized hexahedra at polynomial order $P = 4$ and quadrature order $Q = 6$, for a total of 64^3 and 128^3 degrees of freedom (DOF) equivalent resolution;
- simulations on a non-conformal mesh at equivalent levels of resolution and polynomial order to the conformal case, with various levels of dealiasing. The mesh used for the non-conformal simulations as well as the initial vorticity condition is visualised in Fig. 12.

As is seen from the reference spectral data, the maximum kinetic energy dissipation appears at $t_c \approx 9$. The simulations of the conformal DG case at 128^3 clearly demonstrate the capability of the discontinuous Galerkin method to broadly resolve all but the very peak of the simulation even at a factor of 4 reduction in resolution, with greater numerical diffusion occurring when the resolution is reduced further to a 64^3 resolution. These results are broadly in line with a number of other simulations that appear in the literature, for example in [51–54].

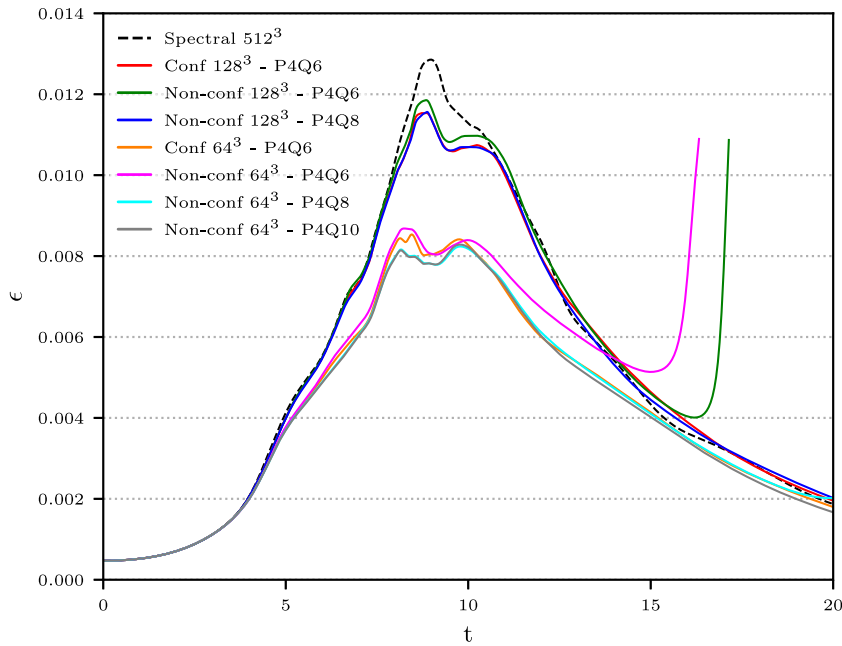


Fig. 13. Evolution of the dimensionless energy dissipation rate in the Taylor–Green vortex, showing the reference spectral data and both conformal and non-conformal at 64^3 and 128^3 DOF.

Of course the central interest of this work is to examine the effect of the point-to-point interpolation in the non-conformal case. Two trends are immediately apparent. Plainly the clearest aspect of Fig. 13 is that when run without any polynomial dealiasing, the simulation becomes unstable at $t_c \approx 16$. From a fluid dynamics perspective, this is during the vortex saturation phase, where breakdown of the vortices is approaching the viscous limit and thus the smallest features are starting to appear in the flow. At under-resolution, this highlights the increased fragility of the point-to-point interpolation approach, likely owing to the appearance of oscillatory effects as structures break down and thus regularity of the solution across the interface decreases. However, as we observe in the previous section and figures, the use of moderate levels of dealiasing, commensurate with what is typically leveraged for underresolved simulations, can stabilise the simulation. Moreover, aside from this instability, it is clear that the non-conformal simulations very closely track the evolution of ϵ when compared to the conformal cases, with only small deviations observed from the conformal cases that can perhaps be attributed to a small difference in number of elements between simulations. Nevertheless, these simulations further emphasise that careful consideration of dealiasing is critical in this setting.

4.2.2. Examination of fluid structures

Although the evolution of ϵ is an important global quantity of interest, we now consider snapshots of the vorticity norm $\|\omega\|$ in order to view structures as they appear in the solution. Fig. 14 for the 64^3 DOF cases, and Fig. 15 for the 128^3 DOF cases present volume renderings near the peak dissipation phase at $t_c = 8$ and at the final time $t_c = 20$ when the smallest scale structures are present. The 64^3 and 128^3 DOF conformal cases are compared to the equivalent P4Q8 point-to-point cases, additionally a non-conformal 64^3 P4Q10 is also shown, in order to examine the effects of higher dealiasing on the resulting solution. The interface locations have been depicted on the domain faces for reference in the non-conformal cases. It is apparent that at $t_c = 8$, very little difference can be seen between all cases and there is a close match between both conformal and non-conformal results. However, at $t_c = 20$ where the smallest vortex features appear, the 64^3 DOF non-conformal P4Q8 case shown in Fig. 14(d) shows a clear buildup of vorticity that aligns with the non-conformal interface. By increasing the dealiasing to $Q = 10$, this buildup is eliminated, as shown in Fig. 14(f), and the resulting vorticity field closely resembles the conformal case in Fig. 14(b). At a higher resolution of 128^3 DOF $\|\omega\|$ snapshots for the same time points are shown in Fig. 15. In this simulation it is clear that there is no accumulation of vorticity around the interfaces for

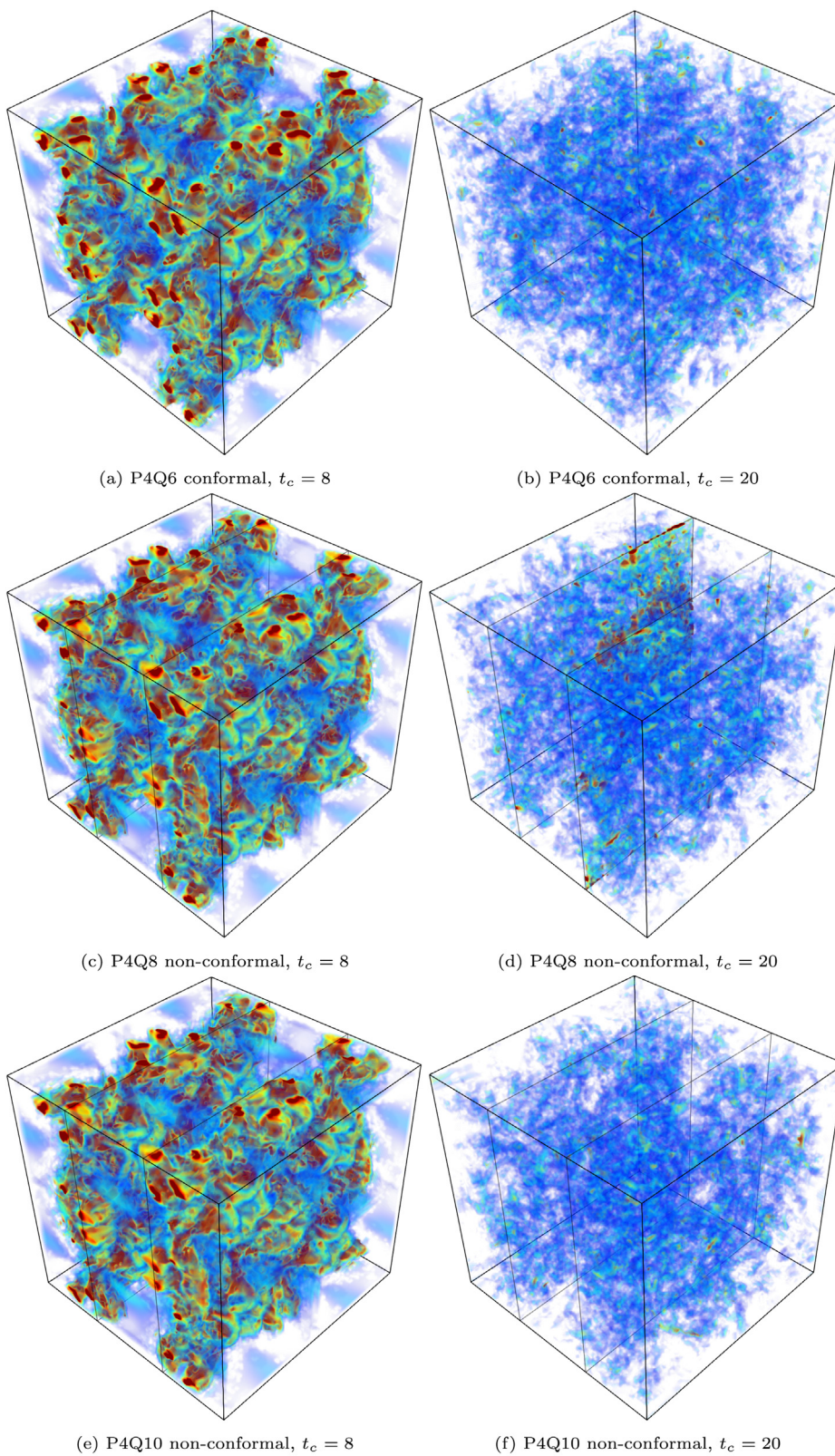


Fig. 14. 64^3 DOF cases showing $\|\omega\|$ at $t_c = 8$ and $t_c = 20$.

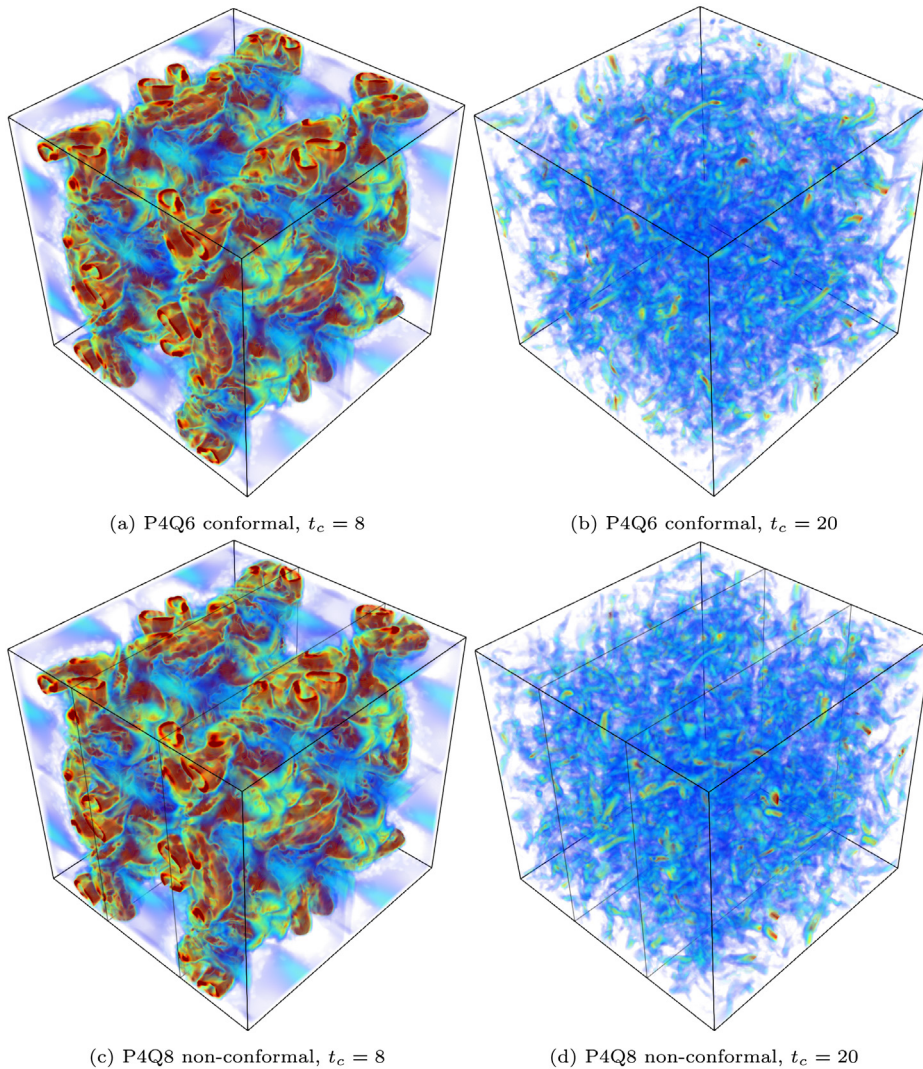


Fig. 15. 128^3 DOF cases showing $\|\omega\|$ at $t_c = 8$ and $t_c = 20$.

the non-conformal P4Q8 case (Fig. 15(d)) and the same small vortex features are present as can be seen in the conformal P4Q6 case (Fig. 15(b)).

4.2.3. Computational cost

The computational costs for the Taylor–Green vortex cases are shown in Table 5. These simulations were run on $8 \times$ AMD Epyc 7742 “Rome” 64 core CPUs, hosted by the Isambard Tier 2 HPC facility, for a total of 512 cores. This shows the fairly large cost incurred by the dealiasing and the handling of the non-conformal interface, as well as the additional communication costs that are imposed in this setting. For example, going from the conformal 64^3 P4Q6 case to the non-conformal 64^3 P4Q10 case results in an approximately six times increase in average computational cost per timestep.

5. Conclusions

In this paper, we have compared the numerical performance of the point-to-point interpolation and mortar techniques, together with equivalent conformal cases, for a number of linear and non-linear hyperbolic conservation

Table 5
Computational costs for the Taylor–Green vortex cases.

Case	Avg. cost per timestep (s)		
	P4Q6	P4Q8	P4Q10
Conformal – 64^3	0.0247	0.0385	0.0607
Point-to-point – 64^3	0.0737	0.1086	0.1510
Conformal – 128^3	0.1918	0.3353	0.6925
Point-to-point – 128^3	0.3037	0.5296	1.0958

law problems. For problems that admit smooth solutions (i.e. which are adequately resolved in space), it is clear that either method is capable of performing equally well, both in terms of preserving the high-order convergence properties of the DG method, and also when considering the advection of structures across very long time periods. Likewise, when considering problems that are marginally- or under-resolved, it is equally clear that the mortar technique yields the most consistently accurate results when compared to the point-to-point interpolation approach.

Although there were relatively minor differences between the point-to-point and mortar methods for the linear Gaussian hump case in the presence of under-resolution, the isentropic vortex and Taylor–Green vortex cases clearly highlight the care that must be taken when using the point-to-point method in such a regime. From the results we observe here, aliasing and oscillatory effects, owing to the discontinuity in polynomial interpolation across elemental interfaces, can have a significant impact on the ability of this method to accurately resolve flow features across long time periods or at a small scale. However, at the same time we note that it would be relatively unusual for higher-order fluid dynamics simulations to be performed in an implicit LES or under-resolved DNS regime without a significant level of dealiasing. As demonstrated in [55], running either compressible Euler or Navier–Stokes simulations without a comparable level of dealiasing to that we present here can yield inaccurate results and potentially lead to instability. Additionally, it is worth considering that in realistic fluid dynamics simulations of e.g. external aeronautics cases, most problems consist of inflow–outflow setups in which structures would be naturally removed from the domain within a far shorter time period than in the cases we consider here, which have been designed to deliberately test the numerical properties of each scheme.

We believe that there are three main factors to consider when choosing an interface handling technique for sliding or moving meshes: desired simulation accuracy, the capability to handle complex geometric interfaces and the resulting computational cost. In terms of accuracy, we have shown that the mortar method yields the results that are essentially identical to that of a conformal grid, and so for accuracy-critical simulations, this would certainly appear to be the most suitable strategy to adopt. Indeed in two dimensions, where the implementation is relatively straightforward, mortaring should be the first choice method to handle non-conformal grids. However the implementation challenge of constructing mortar elements across an arbitrary interface at high-order presents a significant obstacle in three dimensions. This makes the point-to-point method an attractive alternative, particularly in the context of highly parallel simulations. Results here show that the point-to-point interpolation method seems capable of handling a non-conformal interface in all cases, as long as appropriate precautions are taken by dealiasing to a sufficiently high level. The flexibility of supporting arbitrary interfaces in 3D, lower computational cost of interpolation on the interface and ease of implementation are advantages in this setting; however, they must be weighed against the major disadvantages of this method, which is the lack of a formal mass conservation and the requirement for dealiasing (which then further increases computational cost).

Further investigation is warranted to investigate the minimum amount of dealiasing required for the point-to-point interpolation method, and whether it is suitable to overintegrate only on the interface skeleton elements, which would undoubtedly significantly reduce the overall computational cost. Another aspect that we do not consider in terms of computational cost is the relative effort required to set up a moving grid, where the creation of mortars and evaluation of interpolation points needs to be performed at every timestep. It would also be beneficial to compare an efficient 3D mortar method implementation and the point-to-point interpolation method with conformal solutions in a 3D setting for more complex flow problems. Additionally, investigating problems involving shocks would be another area of research to pursue, to validate the point-to-point interpolation and mortar method under more demanding transonic or supersonic conditions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

DM acknowledges support from the EPSRC Platform Grant PRISM under grant EP/R029423/1 and the ELEMENT project under grant EP/V001345/1.

This work used the Isambard UK National Tier-2 HPC Service (<http://gw4.ac.uk/isambard/>) operated by GW4 and the UK Met Office, and funded by EPSRC, UK under grant EP/P020224/1.

Nektar++ is open-source software, released under the MIT license, and is freely available from the project website (<http://www.nektar.info>).

References

- [1] J. Tyacke, N.R. Vadlamani, W. Trojak, R. Watson, Y. Ma, P.G. Tucker, Turbomachinery simulation challenges and the future, *Prog. Aerosp. Sci.* 110 (2019) 100554, <http://dx.doi.org/10.1016/j.paerosci.2019.100554>.
- [2] J.P. Johnston, Effects of system rotation on turbulence structure: A review relevant to turbomachinery flows, *Int. J. Rotating Mach.* 4 (1997) <http://dx.doi.org/10.1155/s1023621x98000098>.
- [3] M. Sun, Insect flight dynamics: Stability and control, *Rev. Modern Phys.* 86 (2014) 615–646, <http://dx.doi.org/10.1103/RevModPhys.86.615>.
- [4] S.P. Sane, The aerodynamics of insect flight, *J. Exp. Biol.* 206 (23) (2003) 4191–4208, <http://dx.doi.org/10.1242/jeb.00663>.
- [5] B.-g. Mi, H. Zhan, Numerical simulation of the static and dynamic aerodynamics of a UAV under wake flows, *J. Adv. Transp.* 2019 (2019) 6326794, <http://dx.doi.org/10.1155/2019/6326794>.
- [6] Q. Guo, Y. Zhu, Y. Tang, C. Hou, Y. He, J. Zhuang, Y. Zheng, S. Luo, CFD simulation and experimental verification of the spatial and temporal distributions of the downwash airflow of a quad-rotor agricultural UAV in hover, *Comput. Electron. Agric.* 172 (2020) 105343, <http://dx.doi.org/10.1016/j.compag.2020.105343>.
- [7] A.S. Farahani, N.M. Adam, K.A. Mohd Ariffin, Simulation of airflow and aerodynamic forces acting on a rotating turbine ventilator, *Am. J. Eng. Appl. Sci.* 3 (2010) <http://dx.doi.org/10.3844/ajeassp.2010.159.170>.
- [8] S.-T. Lien, N. Ahmed, Numerical simulation of rooftop ventilator flow, *Build. Environ.* 45 (2010) 1808–1815, <http://dx.doi.org/10.1016/j.buildenv.2010.02.009>.
- [9] P. Spalart, Reflections on RANS modelling, in: S.-H. Peng, P. Doerffer, W. Haase (Eds.), *Progress in Hybrid RANS-LES Modelling*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 7–24.
- [10] A. Aspdn, N. Nikiforakis, S. Dalziel, J. Bell, Analysis of implicit LES methods, *Commun. Appl. Math. Comput. Sci.* 3 (1) (2008) 103–126, <http://dx.doi.org/10.2140/camcos.2008.3.103>.
- [11] C. Bosshard, M. Deville, A. Dehbi, E. Leriche, UDNS or LES, that is the question, *Open J. Fluid Dyn.* 5 (2015) 339–352, <http://dx.doi.org/10.4236/ojfd.2015.54034>.
- [12] L.W. Chew, L.R. Glicksman, L.K. Norford, Buoyant flows in street canyons: Comparison of RANS and LES at reduced and full scales, *Build. Environ.* 146 (September) (2018) 77–87, <http://dx.doi.org/10.1016/j.buildenv.2018.09.026>.
- [13] J.-E.W. Lombard, D. Moxey, S.J. Sherwin, J.F.A. Hoessler, S. Dhandapani, M.J. Taylor, Implicit large-Eddy simulation of a wingtip vortex, *AIAA J.* 54 (2) (2016) 506–518, <http://dx.doi.org/10.2514/1.j054181>.
- [14] S. Ghosal, An analysis of numerical errors in large-Eddy simulations of turbulence, *J. Comput. Phys.* 125 (1) (1996) 187–206, <http://dx.doi.org/10.1006/jcph.1996.0088>.
- [15] D. Moxey, R. Amici, R.M. Kirby, Efficient matrix-free high-order finite element evaluation for simplicial elements, *SIAM J. Sci. Comput.* 42 (2020) C97–c123, <http://dx.doi.org/10.1137/19m1246523>.
- [16] A. Bakker, R. Laroche, M. Wang, R. Calabrese, Sliding Mesh simulation of laminar flow in stirred reactors, *Chem. Eng. Res. Des.* 75 (1) (1997) 42–44, <http://dx.doi.org/10.1205/026387697523372>.
- [17] L. Demkowicz, J. Oden, W. Rachowicz, O. Hardy, Toward a universal h-p adaptive finite element strategy, part I. Constrained approximation and data structure, *Comput. Methods Appl. Mech. Engrg.* 77 (1) (1989) 79–112, [http://dx.doi.org/10.1016/0045-7825\(89\)90129-1](http://dx.doi.org/10.1016/0045-7825(89)90129-1).
- [18] N. Offermans, A. Peplinski, O. Marin, P.F. Fischer, P. Schlatter, Towards adaptive Mesh refinement for the spectral element solver Nek5000, in: M.V. Salvetti, V. Armenio, J. Fröhlich, B.J. Geurts, H. Kuerten (Eds.), *Direct and Large-Eddy Simulation XI*, Springer International Publishing, Cham, 2019, pp. 9–15.
- [19] W. Bangerth, R. Hartmann, G. Kanschat, Deal.ii—a general-purpose object-oriented finite element library, *ACM Trans. Math. Software* 33 (4) (2007) <http://dx.doi.org/10.1145/1268776.1268779>, 24–es.
- [20] C. Mavriplis, A posteriori error estimators for adaptive spectral element techniques, in: P. Wesseling (Ed.), *Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics*, Vieweg+Teubner Verlag, Wiesbaden, 1990, pp. 333–342.
- [21] D. Rosenberg, A. Fournier, P. Fischer, A. Pouquet, Geophysical–astrophysical spectral-element adaptive refinement (GASPAR): Object-oriented h-adaptive fluid dynamics simulation, *J. Comput. Phys.* 215 (1) (2006) 59–80, <http://dx.doi.org/10.1016/j.jcp.2005.10.031>.

- [22] R. Moura, J. Peiro, S. Sherwin, Direct and Large-Eddy Simulation, in: ERCOFTAC Series, vol. XI, Springer International Publishing, 2019, pp. 53–59, http://dx.doi.org/10.1007/978-3-030-04915-7_8 (Ch. Implicit LES Approaches via Discontinuous Galerkin Methods at Very Large Reynolds).
- [23] R. Moura, G. Mengaldo, J. Peiró, S. Sherwin, On the Eddy-resolving capability of high-order discontinuous Galerkin approaches to implicit LES / under-resolved DNS of Euler turbulence, *J. Comput. Phys.* 330 (2017) 615–623, <http://dx.doi.org/10.1016/j.jcp.2016.10.056>.
- [24] D. Flad, G. Gassner, On the use of kinetic energy preserving DG-schemes for large Eddy simulation, *J. Comput. Phys.* 350 (2017) 782–795, <http://dx.doi.org/10.1016/j.jcp.2017.09.004>.
- [25] D.A. Kopriva, A staggered-grid multidomain spectral method for the compressible Navier–Stokes equations, *J. Comput. Phys.* 143 (1) (1998) 125–158, <http://dx.doi.org/10.1006/jcph.1998.5956>.
- [26] D.A. Kopriva, S.L. Woodruff, M.Y. Hussaini, Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method, *Internat. J. Numer. Methods Engrg.* 53 (1) (2002) 105–122, <http://dx.doi.org/10.1002/nme.394>.
- [27] E. Ferrer, R.H. Willden, A high order discontinuous Galerkin–Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes, *J. Comput. Phys.* 231 (21) (2012) 7037–7056, <http://dx.doi.org/10.1016/j.jcp.2012.04.039>.
- [28] B. Zhang, C. Liang, A simple, efficient, and high-order accurate curved sliding-Mesh interface approach to spectral difference method on coupled rotating and stationary domains, *J. Comput. Phys.* 295 (January) (2015) 147–160, <http://dx.doi.org/10.1016/j.jcp.2015.04.006>.
- [29] N. Kraiss, A. Beck, T. Bolemann, H. Frank, D. Flad, G. Gassner, F. Hindenlang, M. Hoffmann, T. Kuhn, M. Sonntag, et al., FLEXI: A high order discontinuous Galerkin framework for hyperbolic–parabolic conservation laws, *Comput. Math. Appl.* (2020) <http://dx.doi.org/10.1016/j.camwa.2020.05.004>.
- [30] H.J. Aguerre, S.M. Damián, J.M. Gimenez, N.M. Nigro, Conservative handling of arbitrary non-conformal interfaces using an efficient supermesh, *J. Comput. Phys.* 335 (2017) 21–49, <http://dx.doi.org/10.1016/j.jcp.2017.01.018>.
- [31] P. Farrell, M. Piggott, C. Pain, G. Gorman, C. Wilson, Conservative interpolation between unstructured meshes via supermesh construction, *Comput. Methods Appl. Mech. Engrg.* 198 (33–36) (2009) 2632–2642, <http://dx.doi.org/10.1016/j.cma.2009.03.004>.
- [32] M.A. Kopera, F.X. Giraldo, Mass conservation of the unified continuous and discontinuous element-based Galerkin methods on dynamically adaptive grids with application to atmospheric simulations, *J. Comput. Phys.* 297 (2015) 90–103, <http://dx.doi.org/10.1016/j.jcp.2015.05.010>.
- [33] C.D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R.M. Kirby, S.J. Sherwin, Nektar++: An open-source spectral/hp element framework, *Comput. Phys. Comm.* 192 (2015) 205–219, <http://dx.doi.org/10.1016/j.cpc.2015.02.008>.
- [34] 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, S.J. Sherwin, Nektar++: enhancing the capability and application of high-fidelity spectral/hp element methods, *Comput. Phys. Comm.* 249 (2020) 107110, <http://dx.doi.org/10.1016/j.cpc.2019.107110>.
- [35] G. Karniadakis, S. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, second ed., Oxford University Press, Oxford, 2005, <http://dx.doi.org/10.1093/acprof:oso/9780198528692.001.0001>.
- [36] J.-P. Berut, L.N. Trefethen, Barycentric Lagrange interpolation, *SIAM Rev.* 46 (3) (2004) 501–517, <http://dx.doi.org/10.1137/s0036144502417715>.
- [37] Y. Maday, C. Mavriplis, A.T. Patera, Nonconforming mortar element methods - Application to spectral discretizations, in: *Domain Decomposition Methods*, 1989, pp. 392–418.
- [38] D. De Grazia, G. Mengaldo, D. Moxey, P.E. Vincent, S.J. Sherwin, Connections between the discontinuous Galerkin method and high-order flux reconstruction schemes, *Internat. J. Numer. Methods Fluids* 75 (12) (2014) 860–877, <http://dx.doi.org/10.1002/flid.3915>.
- [39] Z.J. Wang, Y. Liu, G. May, A. Jameson, Spectral difference method for unstructured grids II: Extension to the Euler equations, *J. Sci. Comput.* 32 (1) (2007) 45–71, <http://dx.doi.org/10.1007/s10915-006-9113-9>.
- [40] S.C. Spiegel, H. Huynh, J.R. DeBonis, A survey of the isentropic Euler vortex problem using high-order methods, in: *22nd AIAA Computational Fluid Dynamics Conference*, 2015, <http://dx.doi.org/10.2514/6.2015-2444>.
- [41] S.C. Spiegel, H. Huynh, J.R. DeBonis, De-aliasing through over-integration applied to the flux reconstruction and discontinuous Galerkin methods, in: *22nd AIAA Computational Fluid Dynamics Conference*, 2015, <http://dx.doi.org/10.2514/6.2015-2744>.
- [42] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, third ed., Springer, Berlin, New York, 2009.
- [43] X.-s. Li, X.-d. Ren, C.-w. Gu, An improved Roe scheme for all Mach-number flows simultaneously curing known problems, 2017.
- [44] D. Moxey, C. Cantwell, R. Kirby, S. Sherwin, Optimising the performance of the spectral/hp element method with collective linear algebra operations, *Comput. Methods Appl. Mech. Engrg.* 310 (2016) 628–645, <http://dx.doi.org/10.1016/j.cma.2016.07.001>.
- [45] J. Dürrwächter, M. Kurz, P. Kopper, D. Kempf, C.-D. Munz, A. Beck, An efficient sliding mesh interface method for high-order discontinuous Galerkin schemes, *Comput. & Fluids* 217 (2021) 104825, <http://dx.doi.org/10.1016/j.compfluid.2020.104825>.
- [46] B. Cockburn, C.-W. Shu, *SIAM J. Numer. Anal.* 35 (6) (1998) 2440–2463, <http://dx.doi.org/10.1137/S0036142997316712>,
- [47] M.A. Taylor, B.A. Wingate, R.E. Vincent, An algorithm for computing Fekete points in the triangle, *SIAM J. Numer. Anal.* 38 (5) (2000) 1707–1720, <http://dx.doi.org/10.1137/S0036142998337247>.
- [48] J.S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, *SIAM J. Numer. Anal.* 35 (2) (1998) 655–676, <http://dx.doi.org/10.1137/S003614299630587X>.
- [49] J.S. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods*, first ed., Springer-Verlag New York, New York, 2008, <http://dx.doi.org/10.1007/978-0-387-72067-8>.
- [50] E. Laughton, V. Zala, R.M. Kirby, D. Moxey, A. Narayan, Fast barycentric-based evaluation over spectral/hp elements, 2021, in review.

- [51] Z.-G. Yan, Y. Pan, G. Castiglioni, K. Hillewaert, J. Peiró, D. Moxey, S.J. Sherwin, Nektar++: Design and implementation of an implicit, spectral/hp element, compressible flow solver using a Jacobian-free Newton Krylov approach, *Comput. Math. Appl.* 81 (2021) 351–372, <http://dx.doi.org/10.1016/j.camwa.2020.03.009>, Development and Application of Open-source Software for Problems with Numerical PDEs.
- [52] N. Fehn, W. Wall, M. Kronbichler, A matrix-free high-order discontinuous Galerkin compressible Navier–Stokes solver: A performance comparison of compressible and incompressible formulations for turbulent incompressible flows, *Internat. J. Numer. Methods Fluids* 89 (2018) <http://dx.doi.org/10.1002/flid.4683>.
- [53] D. Drikakis, C. Fureby, F. Grinstein, D. Youngs, Simulation of transition and turbulence decay in the Taylor–Green vortex, *J. Turbul.* 8 (2007) <http://dx.doi.org/10.1080/14685240701250289>.
- [54] J.-B. Chapelier, M. de la Llave Plata, F. Renac, Inviscid and viscous simulations of the Taylor–Green vortex flow using a modal discontinuous Galerkin approach, in: 42nd AIAA Fluid Dynamics Conference and Exhibit 2012, 2012, <http://dx.doi.org/10.2514/6.2012-3073>.
- [55] G. Mengaldo, D. De Grazia, P.E. Vincent, S.J. Sherwin, On the connections between discontinuous Galerkin and flux reconstruction schemes: Extension to curvilinear meshes, *J. Sci. Comput.* 67 (3) (2016) 1272–1292, <http://dx.doi.org/10.1007/s10915-015-0119-z>.