Journal of Computational Physics 299 (2015) 56–81

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Dealiasing techniques for high-order spectral element methods on regular and irregular grids

G. Mengaldo [*], D. De Grazia, D. Moxey, P.E. Vincent, S.J. Sherwin

Imperial College London, Department of Aeronautics, South Kensington Campus, London SW7 2AZ, UK

ARTICLE INFO

Article history:
Received 14 October 2014
Received in revised form 10 June 2015
Accepted 28 June 2015
Available online 7 July 2015

Keywords:
Dealiasing
Spectral/*hp* methods
Continuous Galerkin
Discontinuous Galerkin
Flux reconstruction

ABSTRACT

High-order methods are becoming increasingly attractive in both academia and industry, especially in the context of computational fluid dynamics. However, before they can be more widely adopted, issues such as lack of robustness in terms of numerical stability need to be addressed, particularly when treating industrial-type problems where challenging geometries and a wide range of physical scales, typically due to high Reynolds numbers, need to be taken into account. One source of instability is aliasing effects which arise from the nonlinearity of the underlying problem. In this work we detail two dealiasing strategies based on the concept of consistent integration. The first uses a localised approach, which is useful when the nonlinearities only arise in parts of the problem. The second is based on the more traditional approach of using a higher quadrature. The main goal of both dealiasing techniques is to improve the robustness of high order spectral element methods, thereby reducing aliasing-driven instabilities. We demonstrate how these two strategies can be effectively applied to both continuous and discontinuous discretisations, where, in the latter, both volumetric and interface approximations must be considered. We show the key features of each dealiasing technique applied to the scalar conservation law with numerical examples and we highlight the main differences in terms of implementation between continuous and discontinuous spatial discretisations.

1. Introduction

Numerical stability is a fundamental requirement for tools, such as solvers used to model fluid dynamics problems. This is especially true when these tools are used in an engineering context, where reliability and robustness are key factors in its efficacy in the industrial pipeline. While low order finite element methods, which are traditionally popular in such tools, demonstrate a usually satisfying level of numerical stability due to their dissipation properties, high order spectral element methods typically exhibit low dissipation errors, and are therefore affected by a lack of stability which in turn affects their robustness and reliability.

Over the last decade, various attempts have been made to address this problem. Different formulations of the nonlinear terms, for instance, have different aliasing properties. In [1] it is shown that the skew-symmetric form of the convective term results in a reduced amplitude of the aliasing errors in comparison to the conservative and nonconservative forms. Examples of the skew-symmetric methodology for discontinuous Galerkin spectral element methods can be found in [2,3].

[*] Corresponding author.
E-mail addresses: g.mengaldo11@imperial.ac.uk (G. Mengaldo), d.de-Grazia12@imperial.ac.uk (D. De Grazia), d.moxey@imperial.ac.uk (D. Moxey), p.vincent@imperial.ac.uk (P.E. Vincent), s.sherwin@imperial.ac.uk (S.J. Sherwin).

http://dx.doi.org/10.1016/j.jcp.2015.06.032

Approaches such as spectral vanishing viscosity [4] modify the underlying discretised operators in order to add artificial dissipation at high polynomial orders, thereby reducing high-frequency oscillations in the approximate solution, stabilising the method and making it suitable for high Reynolds number large-eddy simulations [5–8]. Polynomial filtering [9–12] adopts a similar strategy, in which an interpolation-based filter is applied locally to each element at each time-step to prevent the buildup of unwanted oscillations in the solution field. An alternative route for stabilisation is represented by the variational multiscale schemes [13] which are based on the model used to approximate the coupling between unresolved and partially resolved scales [14–16]. Finally, in [17] the impact of an exponential based modal filtering and over-integration on underresolved high-order turbulent flow computations is investigated and in [18] a DG method with guaranteed stability is obtained using a non-polynomial modified basis.

In this paper, we focus on the concept of over-integration [19–22], where over-integration refers to the use of more quadrature points than are necessary for a linear operator, but which might equally well be considered as consistent integration of the nonlinear operators. In the formulation of spectral element methods, on any given element one may choose the number of expansion modes used to represent the approximation of a function, and the number of quadrature points used to calculate numerical estimates of integrals which may appear in the formulation.

A popular choice is to couple a set of quadrature points with an equal number of nodal Lagrange polynomials defined at the same points, leading to a collocation method. There are many examples of this throughout the literature, both in terms of the more traditionally utilised continuous Galerkin (CG) and discontinuous Galerkin (DG) formulations, as well as newer extensions such as the flux reconstruction (FR) technique as presented by Huynh [23]. In collocation methods, while most linear operators can be exactly integrated in this setting depending on the choice of quadrature, integrals of nonlinear terms typically incur numerical error. However, the computational efficiencies that can be attained through the use of a collocation formulation, especially given the presence of a diagonal mass matrix, often outweigh the numerical error that is incurred.

To illustrate this point, let us consider errors arising from the integrals of nonlinear products of polynomial expansions of order P. It is well-known that polynomials of order P can be exactly integrated up to machine precision, given some minimum number of quadrature points $Q_{\min}$ [24]. For example Gauss–Lobatto–Legendre quadrature, which is widely adopted in the context of spectral element methods, allows one to obtain exact values of integrals for order $2P-3$ polynomials given P quadrature points [25]. However, when nonlinear quantities are calculated at the quadrature points, a collocation projection leads to an error known as polynomial aliasing being introduced into the obtained integrals, due to an insufficient number of quadrature points. This numerical error may be negligible for well-resolved simulations, such as flow simulations at low Reynolds numbers, but can become a critical issue for under- and marginally-resolved simulations that typically arises in large eddy simulations. In addition, in the presence of non-polynomial functions, such as the compressible Euler and Navier–Stokes equations fluxes written in conservative form, consistent integration cannot be applied with a specific rule as in the case of polynomial functions. In this case, it is still possible to consistently reduce aliasing errors but the number of quadrature points required might become too computationally expensive.

The aim of this paper is to illustrate effective and computationally efficient approaches based on consistent integration to resolve the aliasing issues arising in spectral/*hp* element discretisations. Specifically, we take into consideration three different spectral/*hp* element approximations: the continuous and discontinuous Galerkin methods as well as the flux reconstruction approach. For a scalar partial differential equation (PDE) these approximations can be written as:

$$\begin{aligned}
&\text{CG:} \quad \frac{du}{dt} = \mathbf{M}^{-1}\left(\mathcal{L}+\mathcal{N}\right),\\
&\text{DG:} \quad \frac{du}{dt} = \mathbf{M}^{-1}\left(\mathcal{L}_V+\mathcal{L}_I+\mathcal{N}_V+\mathcal{N}_I\right),\\
&\text{FR:} \quad \frac{du}{dt} = \mathcal{L}_V+\mathcal{L}_I+\mathcal{N}_V+\mathcal{N}_I,
\end{aligned} \tag{1}$$

where u is the approximate solution field, $\mathbf{M}$ is the mass matrix, $\mathcal{L}$ is the linear term and $\mathcal{N}$ is the nonlinear term. In DG and FR both the linear and the nonlinear term are composed by a volumetric term over each spectral element and an interface term on the trace space connecting the elements, which we denote with a subscript V and I, respectively. Henceforth, $\mathcal{N}$ includes aliasing issues arising from both nonlinear and "variable-coefficient" fluxes and to curved geometry, both of which might be described by nonlinear polynomial functions. Note that "variable-coefficient" fluxes and curved geometry, being functions of the spatial coordinates $\boldsymbol{x}$, do not affect the linearity of the underlying PDE which depends on u. However, they introduce aliasing errors. As an example, the advection equation with spatially varying coefficients used in Section 4 is still a linear equation, since the advection velocity does not depend on the solution, but the non-constant velocity coefficients introduce aliasing errors.

To address these aliasing issues we consider two different strategies both based on the concept of consistent integration. With the first approach we *locally* consistently integrate only the nonlinear terms of the PDE being solved without addressing aliasing sources arising from the mapping used in deformed/curved meshes (also referred to as geometrical-aliasing sources). Specifically, with this approach we can address only aliasing arising from the nonlinear or "variable coefficient" fluxes in our problem (also referred to as PDE aliasing). Eq. (1) can therefore be re-written as:

$$
\begin{aligned}
\text{CG:}\quad & \frac{du}{dt} = \mathbf{M}^{-1}\Big|_Q \left(\mathcal{L}\Big|_Q + \mathcal{N}\Big|_{Q_V}\right), \\
\text{DG:}\quad & \frac{du}{dt} = \mathbf{M}^{-1}\Big|_Q \left(\mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_{Q_V} + \mathcal{N}_I\Big|_{Q_I}\right), \\
\text{FR:}\quad & \frac{du}{dt} = \mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_{Q_V} + \mathcal{N}_I\Big|_{Q_I},
\end{aligned}
\tag{2}
$$

where Q is the number of quadrature points used for the linear term and the mass matrix, while Q_V and Q_I indicate the number of quadrature points used for the consistent integration of the volumetric and the interface part of the nonlinear term, respectively. We note the local behaviour of the strategy, which allows us to selectively adjust the quadrature used for the nonlinear term.

In the second approach, we consistently integrate every term of the numerical discretisation which typically implies the over-integration of the linear terms. In this strategy we can address both PDE-aliasing driven instabilities as well as geometrical-aliasing sources arising from deformed/curved elements. Eq. (1) becomes:

$$
\begin{aligned}
\text{CG:}\quad & \frac{du}{dt} = \mathbf{M}^{-1}\Big|_Q \left(\mathcal{L}\Big|_Q + \mathcal{N}\Big|_Q\right), \\
\text{DG:}\quad & \frac{du}{dt} = \mathbf{M}^{-1}\Big|_Q \left(\mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_Q + \mathcal{N}_I\Big|_Q\right), \\
\text{FR:}\quad & \frac{du}{dt} = \mathcal{L}_V\Big|_Q + \mathcal{L}_I\Big|_Q + \mathcal{N}_V\Big|_Q + \mathcal{N}_I\Big|_Q,
\end{aligned}
\tag{3}
$$

where the same number of quadrature points is used *globally* for all the terms. To summarise, the main contributions of this paper are:

- Generalised analysis which encompasses DG, FR and CG discretisations, highlighting the contribution of interface and volumetric terms.
- Local dealiasing strategy which exploits sum-factorisation type approach to improve computational efficiency.
- Comparison of geometrical- and PDE-aliasing.

The paper is structured as follows. In Section 2, we briefly summarise the different CG, DG and FR discretisations, and highlight the sources of aliasing errors which arise in each formulation. Section 3 outlines the *Local* and *Global* dealiasing strategies we adopt to reduce the aliasing errors and improve the stability of the different discretisations considered. In this section we also describe the implementation details which are required to apply such methods. In Section 4, we present practical applications of the dealiasing techniques, highlighting key examples in compressible and incompressible flow simulations. Finally, in Section 5, we outline the results and conclusions of the paper.

2. Aliasing sources in high-order spectral element methods

Aliasing errors in polynomial spectral element methods typically arise through under-integration of the terms appearing in the weak form of the equations to be discretised. When using Gauss–Lobatto–Legendre (GLL) quadrature points, for instance, the minimum number of quadrature points $Q_{\min}$ necessary to exactly integrate any given polynomial of order P, $u(\xi) \in \mathcal{P}_P$, up to machine precision is

$$
Q_{\min} = \frac{P+3}{2}. \tag{4}
$$

In Galerkin methods, we are usually interested in evaluating the L^2 inner product of two polynomials (ϕ_p, ϕ_q) in order to compute, for example, the mass matrix of our discretised problem, where $\phi_p(\xi), \phi_q(\xi) \in \mathcal{P}_P$ are the so-called test or virtual functions, which in our studies are chosen to be the same as the solution expansion. By using Eq. (4) it is possible to calculate the minimum number of GLL quadrature points necessary for the quadrature to be exact as a function of the polynomial order P, as shown in Table 1.

We note that to exactly integrate a linear problem it is necessary to use $Q_{\min} = P + 2$ GLL quadrature points. For nonlinear problems the number of quadrature points needed increases and for quadratic nonlinearities such as the convective term of the incompressible Navier–Stokes equations it becomes $Q_{\min} = \frac{3}{2}(P + 1)$ whereas for cubic nonlinearities $Q_{\min} = 2(P + 1)$ quadrature points are needed for the numerical integration to be exact. The above discussion holds true as long as the nonlinearities belong to the polynomial approximation space used for representing the discretised problem. If instead we have non-polynomial nonlinearities, it is not possible to perform an exact numerical integration using Gauss-type quadrature combined with the rules in Table 1. However, it is still possible to reduce the integration error to machine precision by choosing an integration quadrature rule with a sufficiently high degree. This occurs in, for instance,

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.

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

the compressible Euler and Navier–Stokes equations. When written in conservative form, the flux functions are rational and, therefore, non-polynomial.

In addition to the nonlinearities present in the equations (PDE aliasing sources), additional aliasing may arise from the use of deformed or curved elements. The Jacobian of the isoparametric mapping, which takes a standard element and projects it into the Cartesian space, is by definition a non-constant, non-linear polynomial for curved elements. This introduces a geometrical-aliasing source which may corrupt the solution in a similar manner as the PDE-aliasing and may eventually increase the overall degree of the nonlinearity.

In this section we consider three different high-order methods, namely the continuous Galerkin (CG) method, the discontinuous Galerkin (DG) method and the flux reconstruction (FR) approach. For each of these methods we highlight the sources of PDE- and geometrical-aliasing by considering the multidimensional conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) = 0, \tag{5}$$

where $u = u(\boldsymbol{x}, t)$ is the conserved variable and $\mathbf{F}(u) = [f(u), g(u), h(u)]^{\mathrm{T}}$ is the flux vector that governs the transport of u.

Note that the form of the advection term in Eq. (5) can affect the numerical stability. There is a rich literature, especially for the incompressible Navier–Stokes equations, where the convective term is represented in a skew-symmetric form in order to enhance stability despite losing conservation. In this regard, Malm, Schlatter et al. [26] detailed a possible relationship between consistent integration and the skew-symmetric form of the convection operator, pointing out that for any Galerkin-based method the use of consistent integration recovers imaginary eigenvalues (i.e. skew-symmetry of the convective operator) and therefore stability. In this work we do not consider other possible choices for improving stability other than consistent integration. Nevertheless, the connections between consistent integration and the skew-symmetric form of the convective term remain an interesting aspect which needs to be further explored (see also [2,3]).

2.1. CG formulation

The CG formulation is based on a continuous discretisation of the computational domain. In particular the approximation space as well as the space of the test functions is defined as

$$\chi^{\delta} = \{v \in C^0(\Omega) : v|_{\Omega_e} \in \mathcal{P}_P(\Omega_e), \forall \Omega_e\}, \tag{6}$$

where $\Omega = \bigcup_{e=1}^{N_{\mathrm{el}}} \Omega_e$ is the mesh representing the computational domain. The weak form of the CG method for Eq. (5) is

$$\left(v, \frac{\partial u}{\partial t}\right)_{\Omega} + \left(v, \nabla \cdot \mathbf{F}(u)\right)_{\Omega} = 0. \tag{7}$$

In order to obtain a discrete form for this equation, we write u, $\mathbf{F}(u)$ and v in terms of an expansion of basis functions of order P, transforming Eq. (7) into its elemental weak form

$$\left(\phi_{pqr}, \frac{\partial u^{\delta}}{\partial t}\right)_{\Omega_e} + \left(\phi_{pqr}, \nabla \cdot \mathbf{F}^{\delta}(u^{\delta})\right)_{\Omega_e} = 0, \quad \forall (p, q, r) \tag{8}$$

where $\phi_{pqr} = \phi_{pqr}(\boldsymbol{x})$ are the basis functions, $(p, q, r) \in \mathbb{N}^3$ denotes a tensor product ordering of the modes, Ω_e is the elemental domain and $(\cdot, \cdot)$ represents the following L^2 inner product

$$(a, b)_{\Omega_e} = \int_{\Omega_e} ab \, \mathrm{d}\boldsymbol{x}. \tag{9}$$

We evaluate the solution u at a set of Q quadrature points in each tensor product direction and denote it with the vector $\boldsymbol{u}_e = \mathbf{B}^e \widehat{\boldsymbol{u}}^e(t)$, where $\mathbf{B}^e = \phi_{pqr}(\boldsymbol{\xi}_m)$ is the basis matrix, the subscript m refers to the m-th quadrature point and we have used the fact that in the standard CG approximation, both the trial and the test functions lie in the same expansion space. The elemental matrix form of Eq. (8) becomes:

$$(\mathbf{B}^e)^{\mathrm{T}} \mathbf{W}^e \, \mathbf{B}^e \, \frac{\mathrm{d}\widehat{\boldsymbol{u}}^e}{\mathrm{d}t} + (\mathbf{B}^e)^{\mathrm{T}} \mathbf{W}^e \left(\mathbf{D}^e_{x_1} \widehat{\boldsymbol{f}}^e(u) + \mathbf{D}^e_{x_2} \widehat{\boldsymbol{g}}^e(u) + \mathbf{D}^e_{x_3} \widehat{\boldsymbol{h}}^e(u)\right) = 0, \tag{10}$$

where $\mathbf{W}^e = \mathbf{W}^e(J)$ is the weight matrix that depends on the elemental Jacobian and $\mathbf{D}^e_{x_i}$ are the elemental differentiation matrices written with respect to the Cartesian coordinates. A more detailed explanation of the geometric definitions and the operations in an arbitrarily-shaped element can be found in Appendix A; the interested reader can also refer to [24]. We can express the partial derivatives in x_i in terms of partial derivatives in standard region coordinates ξ_i and, using the chain rule, we can write:

$$\mathbf{D}^e_{x_j} = \sum_{i=1}^{i=3} \mathbf{\Lambda}^e(\bar{G}_{i/j})\mathbf{D}^e_{\xi_i}, \tag{11}$$

where $\mathbf{\Lambda}(\cdot)$ is a diagonal matrix which contains the factors

$$\bar{G}_{i/j} = \frac{\partial \xi_i}{\partial x_j} \tag{12}$$

on its diagonal. Equation (10) can therefore be recast as

$$(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\,\mathbf{B}^e\,\frac{\mathrm{d}\widehat{\boldsymbol{u}}^e}{\mathrm{d}t} + (\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\left[\left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/1})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\widehat{\boldsymbol{f}}^e(u) + \left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/2})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\widehat{\boldsymbol{g}}^e(u) + \left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/3})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\widehat{\boldsymbol{h}}^e(u)\right] = 0. \tag{13}$$

The final step in constructing the CG discretisation of the problem in Eq. (5) involves assembling the global problem on Ω using the elemental contributions in Eq. (13). By denoting with $\widehat{\boldsymbol{u}}_l$ the vector of all the expansion coefficients from each element $\widehat{\boldsymbol{u}}^e$ and by relating $\widehat{\boldsymbol{u}}_l$ with the global expansion coefficients $\widehat{\boldsymbol{u}}_g$ through the assembly operation $\widehat{\boldsymbol{u}}^e = \mathcal{A}\widehat{\boldsymbol{u}}_g$, we finally obtain the global matrix form for a standard CG approximation

$$\mathcal{A}^{\mathrm{T}}\left[\!\left[(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\right]\!\right]\mathcal{A}\frac{\mathrm{d}\widehat{\boldsymbol{u}}_g}{\mathrm{d}t} + \mathcal{A}^{\mathrm{T}}\left[\!\left[(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/1})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]\!\right]\mathcal{A}\,\widehat{\boldsymbol{f}}_g$$
$$+ \mathcal{A}^{\mathrm{T}}\left[\!\left[(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/2})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]\!\right]\mathcal{A}\,\widehat{\boldsymbol{g}}_g + \mathcal{A}^{\mathrm{T}}\left[\!\left[(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/3})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]\!\right]\mathcal{A}\,\widehat{\boldsymbol{h}}_g = 0, \tag{14}$$

where $[\![\cdot]\!]$ represents the block diagonal direct summation of all local elemental matrices Equation (14) can be discretised in time after the inversion of the global mass matrix $\mathbf{M} = \mathcal{A}^{\mathrm{T}}[\![(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e]\!]\mathcal{A}$:

$$\frac{\mathrm{d}\widetilde{\boldsymbol{u}}_g}{\mathrm{d}t} = -\mathbf{M}^{-1}\Bigg\{\underbrace{\mathcal{A}^{\mathrm{T}}\left[\!\left[(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/1})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]\!\right]\mathcal{A}}_{\mathbf{D_1}}\,\widehat{\boldsymbol{f}}_g + \underbrace{\mathcal{A}^{\mathrm{T}}\left[\!\left[(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/2})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]\!\right]\mathcal{A}}_{\mathbf{D_2}}\,\widehat{\boldsymbol{g}}_g$$
$$+ \underbrace{\mathcal{A}^{\mathrm{T}}\left[\!\left[(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\left(\sum_{i=1}^{i=3}\mathbf{\Lambda}^e(\bar{G}_{i/3})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]\!\right]\mathcal{A}}_{\mathbf{D_3}}\,\widehat{\boldsymbol{h}}_g\Bigg\}, \tag{15}$$

where we indicate with $\mathbf{D}_i$ the global advection matrices. In the CG formulation, if Eq. (5) is nonlinear, the PDE-aliasing arises from the advection terms $\mathbf{D}_1\widehat{\boldsymbol{f}}_g$, $\mathbf{D}_2\widehat{\boldsymbol{g}}_g$ and $\mathbf{D}_3\widehat{\boldsymbol{h}}_g$. On the other hand, the geometrical aliasing arises from the inverse of the mass matrix as well as from the advection matrices. These depend on the elemental Jacobian determinants (hereafter referred to as Jacobians) through the weight matrix $W^e = W^e(J)$. The advection matrices also depend on the geometric factors through W^e as well as $\bar{G}_{i/j}$. In the case of an unstructured domain with deformed/curved elements, the Jacobians and the geometric factors are represented as non-constant polynomials, which may increase the aliasing issues if not properly resolved.

2.2. DG formulation

The DG formulation relies upon a discontinuous discretisation of the computational domain. In particular, the approximation space as well as the space of the test functions is defined as

$$\chi^\delta = \{v \in L^2(\Omega) : v|_{\Omega_e} \in \mathcal{P}_P(\Omega_e), \forall\,\Omega_e\}, \tag{16}$$

where

$$L^2(\Omega) = \left\{ v\ :\ \Omega \to \mathcal{R}\ \Bigg|\ \int_\Omega |v(\boldsymbol{x})|^2 d\Omega \leq \infty \right\}, \tag{17}$$

with $\mathcal{R}$ being the space of real numbers. The weak form of the DG method applied to Eq. (5) is

$$\frac{\partial}{\partial t}(\phi_{pqr}, u^\delta)_{\Omega_e} + \langle \phi_{pqr}, \widetilde{\boldsymbol{f}}(u_+^\delta, u_-^\delta)\rangle_{\Omega_e} - (\nabla\phi_{pqr}, \mathbf{F}(u^\delta))_{\Omega_e} = 0, \tag{18}$$

where the elemental inner products are defined as follows:

$$\begin{aligned}(a,b)_{\Omega_e} &= \textstyle\int_{\Omega_e} ab\,\mathrm{d}\boldsymbol{x}\\ (\boldsymbol{a},\boldsymbol{b})_{\Omega_e} &= \textstyle\int_{\Omega_e} \boldsymbol{a}\cdot\boldsymbol{b}\,\mathrm{d}\boldsymbol{x}\\ \langle a,\boldsymbol{b}\rangle_{\partial\Omega_e} &= \textstyle\int_{\partial\Omega_e} a\boldsymbol{b}\cdot\boldsymbol{n}_e\,\mathrm{d}S\end{aligned} \tag{19}$$

and $\widetilde{\boldsymbol{f}}(u_+^\delta, u_-^\delta)$ is the interface flux which couples the elements of the DG discretisation. This interface flux can be computed in various ways, including simple upwind approaches or more sophisticated Riemann solvers depending on the equation being solved. It is important to note here that the interface flux is a combination of the information coming from two adjacent elements, namely u_+^δ and u_-^δ, which are in general discontinuous.

The matrix form of Eq. (18) is

$$(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\frac{\mathrm{d}\widehat{\boldsymbol{u}}^e}{\mathrm{d}t} - (\mathbf{D}^e_{x_1}\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{f}}^e(u) - (\mathbf{D}^e_{x_2}\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{g}}^e(u) - (\mathbf{D}^e_{x_3}\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{h}}^e(u) + \boldsymbol{b}^e = 0, \tag{20}$$

where $\boldsymbol{b}^e$ is the surface integral introduced in Eq. (19):

$$\boldsymbol{b}^e = \int\limits_{\partial\Omega^e} \phi_{pqr}(\widetilde{\boldsymbol{f}}\cdot\boldsymbol{n}^e)\mathrm{d}S. \tag{21}$$

We can use similar relations to those in Eqs. (11), (12) to rewrite Eq. (20) as follows

$$\begin{aligned}&(\mathbf{B}^e)^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\frac{\mathrm{d}\widehat{\boldsymbol{u}}^e}{\mathrm{d}t} - \left[\left(\sum_{i=1}^{i=3}\boldsymbol{\Lambda}^e(\overline{G}_{i/1})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{f}}^e(u) - \left[\left(\sum_{i=1}^{i=3}\boldsymbol{\Lambda}^e(\overline{G}_{i/2})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{g}}^e(u)\\ &\quad - \left[\left(\sum_{i=1}^{i=3}\boldsymbol{\Lambda}^e(\overline{G}_{i/3})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{h}}^e(u) + \boldsymbol{b}^e = 0,\end{aligned} \tag{22}$$

where the diagonal matrix $\boldsymbol{\Lambda}^e(\overline{G}_i)$ has on its diagonal the geometric factors defined in Eq. (12).

The final matrix form of the DG method can be written as follows

$$\begin{aligned}\frac{\mathrm{d}\widehat{\boldsymbol{u}}^e}{\mathrm{d}t} &= \left[\mathbf{M}^e\right]^{-1}\left\{\left[\left(\sum_{i=1}^{i=3}\boldsymbol{\Lambda}^e(\overline{G}_{i/1})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{f}}^e(u) + \left[\left(\sum_{i=1}^{i=3}\boldsymbol{\Lambda}^e(\overline{G}_{i/2})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{g}}^e(u)\right.\\ &\quad\left. + \left[\left(\sum_{i=1}^{i=3}\boldsymbol{\Lambda}^e(\overline{G}_{i/3})\mathbf{D}^e_{\xi_i}\right)\mathbf{B}^e\right]^{\mathrm{T}}\mathbf{W}^e\mathbf{B}^e\widehat{\boldsymbol{h}}^e(u)\right\} - \left[\mathbf{M}^e\right]^{-1}\boldsymbol{b}^e = 0,\end{aligned} \tag{23}$$

where $\left[\mathbf{M}^e\right]^{-1} = (\mathbf{B}^e)^{-1}(\mathbf{W}^e)^{-1}(\mathbf{B}^e)^{-\mathrm{T}}$ is the inverse of the elemental mass matrix.

In the DG method the source of aliasing errors are similar to those identified for the CG method. Specifically, PDE-aliasing arises from the advection terms whilst geometrical-aliasing emerges from the weight matrices which contain the elemental Jacobians as well as from the $\overline{G}_{i/j}$ terms which include the geometric factors.

A separate discussion is instead needed for the interface fluxes $\widetilde{\boldsymbol{f}}(u_+^\delta, u_-^\delta)$. They are evaluated at the edges of the elements on a given set of points (which may differ from the quadrature points used for discretising the problem within the element domain) and they are in general non-polynomial functions since the contributions to $\widetilde{\boldsymbol{f}}(u_+^\delta, u_-^\delta)$ can come from the left (u_+^δ) and from the right (u_-^δ) element with respect to the edge considered. If Eq. (5) is nonlinear, then this term is another source of PDE-aliasing which introduces additional numerical errors. Note that the boundary term $\boldsymbol{b}^e$ also includes the contribution of the geometric factors and therefore also geometrical-aliasing issues can arise.

2.3. FR formulation

In this formulation the approximation space is equal to that of the DG scheme and it is defined in Eq. (16). The matrix form of the FR approach is similar to that of the DG scheme, where the solution polynomials are piecewise discontinuous. The main difference arises from the use of the differential form of the equations as opposed to the variational form. In addition, the FR approach embeds various high-order spectral element methods including a nodal discontinuous Galerkin method, the spectral difference method and a class of energy-stable schemes [27]. Some well-established connections have been found in the literature. In this paper we show how the connections in [28] hold true also applying both the dealiasing techniques presented.

The FR approach in matrix form can be represented as follows:

$$\frac{d\widehat{\mathbf{u}}^e}{dt} + \left[\sum_{i=1}^{i=3} \mathbf{\Lambda}^e(\overline{G}_{i/1})\mathbf{D}^e_{\xi_i}\right]\widehat{\boldsymbol{f}}^e(u) + \left[\sum_{i=1}^{i=3} \mathbf{\Lambda}^e(\overline{G}_{i/2})\mathbf{D}^e_{\xi_i}\right]\widehat{\mathbf{g}}^e(u) + \left[\sum_{i=1}^{i=3} \mathbf{\Lambda}^e(\overline{G}_{i/3})\mathbf{D}^e_{\xi_i}\right]\widehat{\boldsymbol{h}}^e(u) + \overline{\overline{\boldsymbol{b}^e}} = 0 \tag{24}$$

where $\mathbf{D}_{\xi_i}$ are the elemental differentiation matrices with respect to the standard space, $\mathbf{\Lambda}^e(\overline{G}_{i/j})$ are the same quantities as in Eqs. (15), (23), with $\overline{G}_{i/j}$ being defined in Eq. (12), and the boundary term is

$$\overline{\overline{\boldsymbol{b}^e}} = \left(\overline{\overline{\widetilde{\boldsymbol{f}}^{C,e} \cdot \boldsymbol{n}^e}}\right)\overline{\overline{\mathbf{\Phi}'}} = \left[\overline{\overline{\left(\widetilde{\boldsymbol{f}} \cdot \boldsymbol{n}^e\right) - \left(\widetilde{\boldsymbol{f}}^e \cdot \boldsymbol{n}^e\right)}}\right]\overline{\overline{\mathbf{\Phi}'}}, \tag{25}$$

with $\overline{\overline{\mathbf{\Phi}'}}$ being the derivative of a suitable correction function defined in the standard space, $\overline{\overline{\widetilde{\boldsymbol{f}}^{C,e}}}$ being the standard correction flux defined as the difference between the interface flux $\widetilde{\boldsymbol{f}}$ and the corresponding volumetric flux evaluated at the edge of element e, $\widetilde{\boldsymbol{f}}^e$, where we note that if the interface flux corresponds exactly to the flux coming from element e the difference is zero. This difference needs to be calculated in the standard space since the derivatives of the correction functions belong to the standard space. To obtain the difference between $\widetilde{\boldsymbol{f}}$ and $\widetilde{\boldsymbol{f}}^e$ in the standard space, $\overline{\overline{\Delta\widetilde{\boldsymbol{f}}}}$, we apply a transformation which depends on the Jacobians J as well as on the previously defined geometric terms $\overline{G}_{i/j}$:

$$\overline{\overline{\Delta\widetilde{\boldsymbol{f}}}} = \mathcal{T}(\Delta\widetilde{\boldsymbol{f}}), \tag{26}$$

where $\mathcal{T} = \mathcal{T}(J, \overline{G}_{i/j})$. For a better understanding of the required transformations see [29] whilst for a more detailed and comprehensive discussion concerning the FR approach, the interested reader can refer to Huynh [23] and to Vincent, Castonguay, and Jameson [27,30].

In terms of aliasing considerations, we note that PDE-aliasing sources are exactly the same as the DG approach, and in particular the advection term along with the interface flux $\widetilde{\boldsymbol{f}}$ are directly responsible for introducing PDE-aliasing if Eq. (5) is nonlinear. Regarding geometrical aliasing, whilst the geometric terms seem different if compared to the DG approach, they are in fact identical and they produce the same geometrical aliasing sources.

2.4. Summary

In the previous sections we have highlighted the sources of aliasing in both continuous (CG) and discontinuous (DG and FR) formulations. These are due to PDE nonlinearities as well as to geometrical nonlinearities. In particular, whilst in CG the PDE nonlinearities depend on the volumetric flux only, in DG and FR another source of nonlinearity to take into account depends on the interface flux. In the following sections we perform some numerical experiments with the three methods above, showing the effects of these aliasing issues and explaining different procedures to reduce or eliminate them. Regarding the discontinuous approaches we analyse the role of the interface flux for PDE and geometrical nonlinearities. We also show that the aliasing instabilities of two different versions of DG schemes and two types of FR schemes are equivalent, which is in agreement with previous observations [28].

3. Dealiasing techniques

In this section we describe the two dealiasing approaches considered in this work, namely:

- *Local dealiasing*: PDE-dealiasing through consistent integration of the nonlinear terms[1];
- *Global dealiasing*: PDE- and geometrical-dealiasing through consistent integration of all the terms of the discretisation.

The first takes into account only PDE-aliasing errors and it locally uses a consistent integration of the nonlinear terms of the PDE. The second takes into account both PDE and geometrical aliasing effects.

3.1. Local dealiasing

The first dealiasing approach presented involves the consistent integration of the nonlinear terms in Eq. (5). This approach can be applied to all the three formulations previously introduced, namely the CG, DG and FR discretisations, and addresses the PDE-aliasing issues. In the following, we first consider a one-dimensional case before outlining the extension

[1] As already mentioned, in these nonlinear terms we are including both nonlinear fluxes and variable-coefficient fluxes. The latter still produce a linear PDE but they introduce aliasing errors because of their spatial varying nature.

to two and three dimensions through a tensor product expansion. The implementation presented here has been carried out in the spectral/*hp* element library *Nektar++* [31]. This implementation, and in particular its tensor product extension in both structured and unstructured sub-domains, is computationally efficient for high polynomial orders due to its use of the sum factorisation technique.

Dealiasing in one dimension Consider a 1D approximate polynomial solution $u(\xi)$ of order P represented by an expansion in terms of nodal polynomial functions $\phi_i(\xi)$, such as Lagrange polynomials, on a set of GLL points

$$u^{Q_P}(\xi) = \sum_{i=0}^{Q_P} \hat{u}_i \phi_i(\xi), \qquad Q_P = P. \tag{27}$$

The dealiasing strategy consists of 3 steps.

A) *1D interpolation*
The solution $u(\xi)$ is interpolated onto a larger set of GLL quadrature points

$$u^{Q}(\xi_j) = \sum_{i=0}^{Q_P} \hat{u}_i \phi_i(\xi_j), \quad 0 \le j \le Q, \tag{28}$$

where the number of additional points Q required depends on the degree of the nonlinearity. This step can be achieved through a matrix-vector multiplication

$$\boldsymbol{u}^{Q} = \mathbf{I}_{Q_P \to Q}\, \boldsymbol{u}^{Q_P}, \tag{29}$$

where $\mathbf{I}_{Q_P \to Q}$ is the interpolation matrix

$$\mathbf{I}_{Q_P \to Q}[i, j] = \phi_i^{Q_P}(\xi_j), \tag{30}$$

whose dimensions are $Q \times Q_P$. Note that we could use any set of Q points and not necessarily GLL points. For instance, the larger set of points could be represented by Gauss–Legendre points.

B) *Collocation product*
We then evaluate the nonlinear term on the expanded set of points using the interpolated values of the solution $u^{Q}(\xi_j)$ obtained at the previous step. For simplicity we consider a quadratic nonlinearity so that the product is calculated as

$$f^{Q}(\xi_j) = u(\xi_j) \cdot u(\xi_j) \in \mathcal{P}^{2Q_P} \quad j = 0, \dots, Q. \tag{31}$$

C) *Galerkin projection*
Finally, we perform a projection onto the original set of points. In the following we use a Galerkin projection (GP), which is equivalent to a least squares projection of the nonlinear term evaluated on the larger set of points onto the original set of points, so that

$$\left(f^{Q}(\xi) = u(\xi_j) \cdot u(\xi_j) \in \mathcal{P}^{2Q_P}\right) \xrightarrow{GP_{Q \to Q_P}} \left(f^{Q_P}(\xi) = \widetilde{u}(\xi) \cdot \widetilde{u}(\xi) \in \mathcal{P}^{Q_P}\right) \tag{32}$$

In a similar manner to step A, this can be interpreted as a matrix-vector multiplication

$$\boldsymbol{f}^{Q_P} = \mathbf{GP}_{Q \to Q_P}\, \boldsymbol{f}^{Q}, \tag{33}$$

where $\mathbf{GP}_{Q \to Q_P}$ is the Galerkin projection matrix

$$\mathbf{GP} = \mathbf{M}^{-1} \mathbf{I}_{Q_P \to Q}^{T} \mathbf{W} \tag{34}$$

with $\mathbf{M}[i, j] = \int \phi_i^{Q_P} \phi_j^{Q_P} d\xi$ being the mass matrix and $\mathbf{W}[i, i] = \int \phi_i^{Q} d\xi$ being the diagonal Gauss quadrature weight matrix defined using Q points. Note that the dimensions of $\mathbf{GP}_{Q \to Q_P}$ are $Q_P \times Q$.

To summarise, Fig. 1 depicts a schematic representation of the dealiasing procedure described above for one-dimensional problems.

We note that this procedure is exact when the isoparametric mapping that describes the coordinates of a physical element Ω_e is affine and therefore the Jacobian is constant. In addition, we note that if $\mathbf{M}$ is diagonal then $\mathbf{M}$ has diagonal components of the weights at P set of points. In this case we observe that the projection is an interpolation matrix with the rows scaled by the inverse of the P quadrature point weights whereas the columns are rescaled by the quadrature point integration weights.

Tensor product extension to 2D and 3D For the sake of simplicity, in this section we consider a 2D tensor product element and we remark where necessary the differences in terms of computational costs between the 2D and the 3D tensor-product element cases.

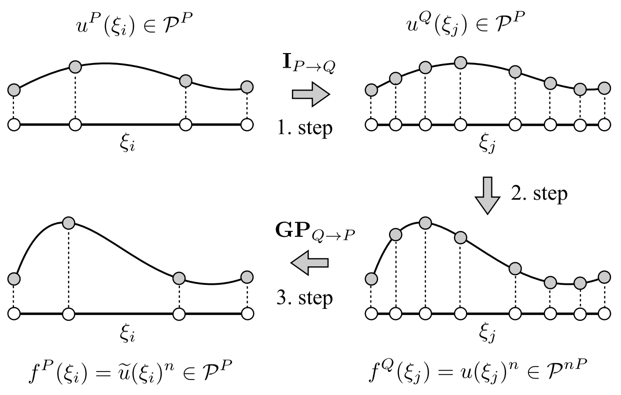

$u^P(\xi_i) \in \mathcal{P}^P$
$u^Q(\xi_j) \in \mathcal{P}^P$
$\mathbf{I}_{P\to Q}$
1. step
ξ_i
ξ_j
2. step
$\mathbf{GP}_{Q\to P}$
3. step
ξ_i
ξ_j
$f^P(\xi_i) = \widetilde{u}(\xi_i)^n \in \mathcal{P}^P$
$f^Q(\xi_j) = u(\xi_j)^n \in \mathcal{P}^{nP}$

Fig. 1. Conceptual flow chart of the *Local* dealiasing approach through consistent integration of the nonlinear terms for 1D problems.

Consider a 2D approximate polynomial solution $u(\xi_1, \xi_2)$ of order P represented through a nodal expansion basis, such as Lagrange polynomials, onto a set of GLL points

$$u^{Q_P}(\xi_1, \xi_2) = \sum_{i,j=0}^{Q_P} \hat{u}_{ij}\phi_i(\xi_1)\phi_j(\xi_2). \tag{35}$$

In the 2D (equivalently 3D) tensor product case, the dealiasing technique consists of the same conceptual steps as the 1D case but the operations to perform are more numerous and consequently computationally more costly. However, the exploitation of the tensor-product properties allows an efficient implementation through the use of the sum-factorisation technique.

A) *2D/(3D) interpolation*
The first step involves interpolating the solution $u(\xi_1, \xi_2)$ onto a larger set of GLL quadrature points as follows:

$$u^Q(\xi_{1_\ell}, \xi_{2_m}) = \sum_{i,j=0}^{Q_P} \hat{u}_{ij}\phi_i(\xi_{1_\ell})\phi_j(\xi_{2_m}), \quad 0 \le \ell, m \le Q, \tag{36}$$

where the number of additional points Q required depends on the degree of the nonlinearity. The interpolation performed is divided into two (or three in 3D) sub-steps as shown in Fig. 2. The first sub-step, in Fig. 2(a), is the interpolation of the solution along ξ_1 whereas the second, in Fig. 2(b) is the interpolation along ξ_2. This implementation allows a reduction of the floating point operations required.
In this case the overall number of floating point operations required is $\mathcal{O}(Q \times Q_P^2 + Q^2 \times Q_P)$ while by doing a 2D interpolation for a non-tensor product basis the floating point operations needed becomes $\mathcal{O}(Q^2 \times Q_P^2)$. Following a similar procedure for the 3D case gives a number of floating point operations that is $\mathcal{O}(Q \times Q_P^2 + Q^2 \times Q_P^2 + Q^3 \times Q_P)$ versus $\mathcal{O}(Q_P^3 \times Q^3)$ operations otherwise required for a full 3D interpolation. Further details of this type of sum factorisation can be found in [24].
Note that also in this case we could have used any larger set of points (such as Gauss–Legendre points) and not necessarily GLL points.

B) *Collocation product*
The second step involves evaluating the nonlinear term on the expanded set of points using the interpolated values of the solution $u^Q(\xi_{1_\ell}, \xi_{2_m})$ obtained at the previous step. As with the 1D case, we evaluate the product of the interpolated values $u^Q(\xi_{1_\ell}, \xi_{2_m})$ using a collocation projection as follows

$$f^Q(\xi_{1_\ell}, \xi_{2_m}) = u(\xi_{1_\ell}, \xi_{2_m}) \cdot u(\xi_{1_\ell}, \xi_{2_m}) \in \mathcal{P}^{2Q_P}, \quad 0 \le \ell, m \le Q, \tag{37}$$

where we again consider a quadratic nonlinearity for the sake of simplicity.

C) *Galerkin projection*
The third step involves performing a projection of the nonlinear term onto the original set of points. To do so we use a Galerkin projection (GP). Similar to the first step, we split this operation into two sequential sub-operations as shown in Fig. 3. First (Fig. 3(a)) we perform a Galerkin projection in direction ξ_2 and successively (Fig. 3(b)) we perform a Galerkin projection in direction ξ_1.

In Fig. 4 we show an overview of the steps above for the 2D/3D tensor-product case.

A similar tensor-product based approach can also be used for triangles in 2D as well as prismatic and tetrahedral elements in 3D as reported in Appendix C.

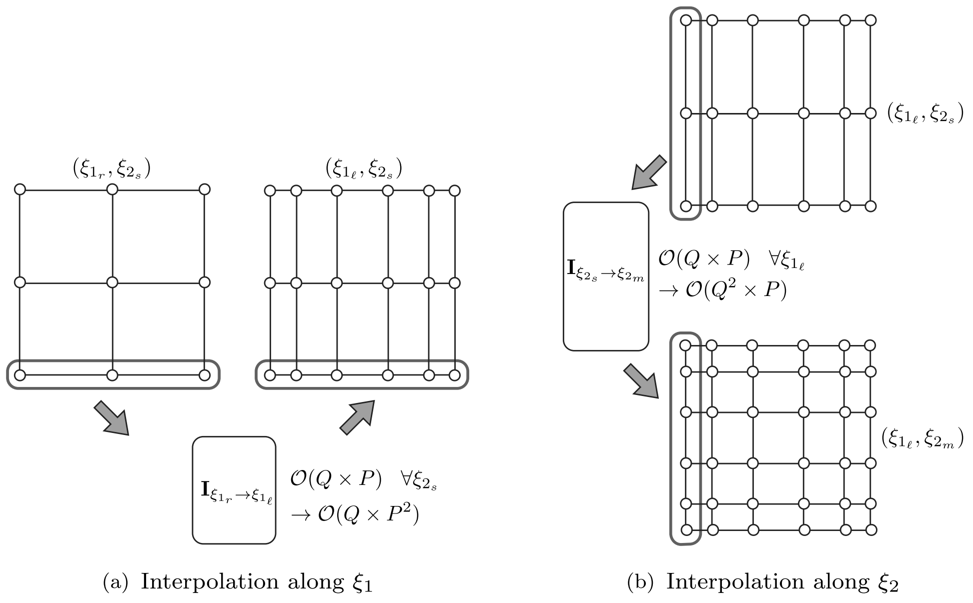

(ξ_{1_r}, ξ_{2_s})
$(\xi_{1_\ell}, \xi_{2_s})$
$\mathbf{I}_{\xi_{1_r}\to\xi_{1_\ell}}$
$\mathcal{O}(Q \times P)\ \ \forall \xi_{2_s}$
$\to \mathcal{O}(Q \times P^2)$
(a) Interpolation along ξ_1
$(\xi_{1_\ell}, \xi_{2_s})$
$\mathbf{I}_{\xi_{2_s}\to\xi_{2_m}}$
$\mathcal{O}(Q \times P)\ \ \forall \xi_{1_\ell}$
$\to \mathcal{O}(Q^2 \times P)$
$(\xi_{1_\ell}, \xi_{2_m})$
(b) Interpolation along ξ_2

Fig. 2. Interpolation for 2D (equivalently 3D) tensor-product elements.

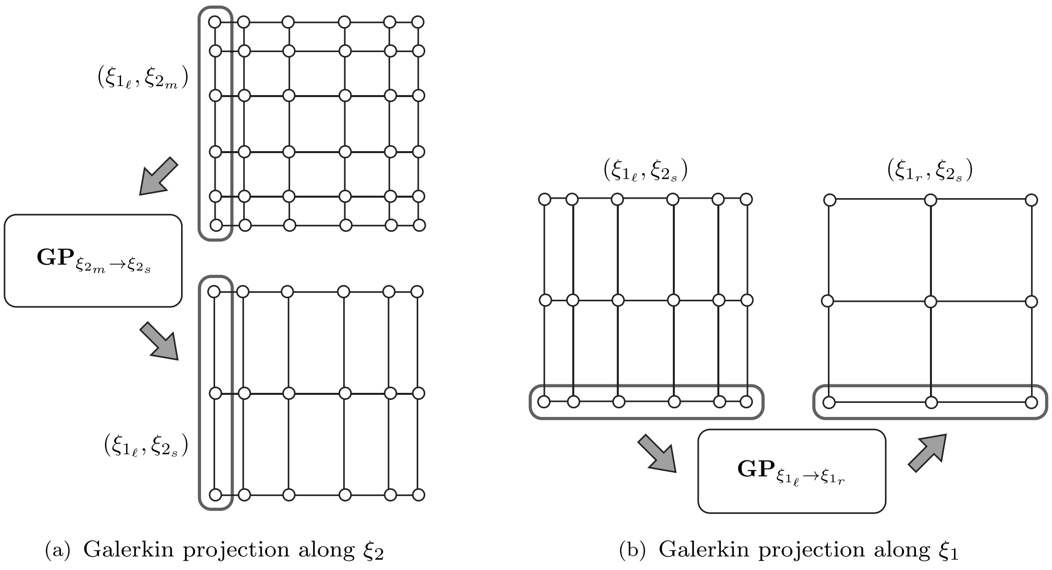
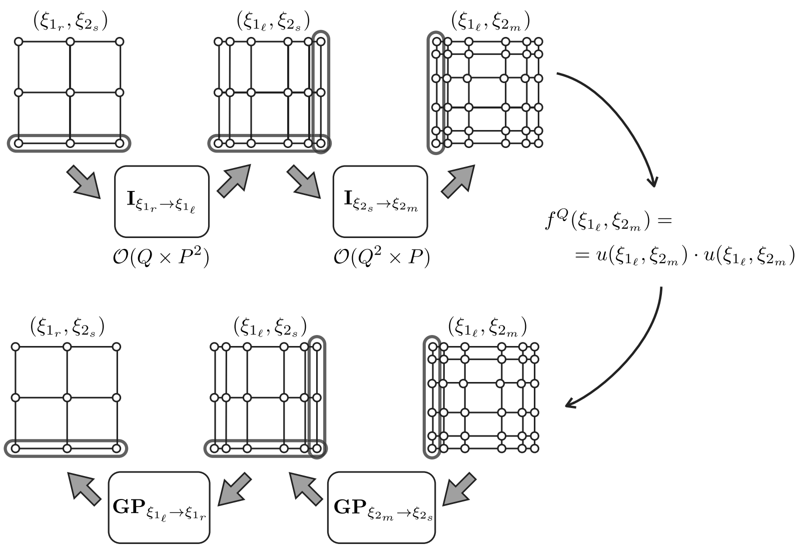

$(\xi_{1_\ell}, \xi_{2_m})$
$\mathbf{GP}_{\xi_{2_m}\to\xi_{2_s}}$
$(\xi_{1_\ell}, \xi_{2_s})$
(a) Galerkin projection along ξ_2
$(\xi_{1_\ell}, \xi_{2_s})$
(ξ_{1_r}, ξ_{2_s})
$\mathbf{GP}_{\xi_{1_\ell}\to\xi_{1_r}}$
(b) Galerkin projection along ξ_1

Fig. 3. Galerkin projection for 2D tensor-product elements.

3.2. Global dealiasing

In order to address both PDE- and geometrical-aliasing issues, one can over-sample all the terms present in the numerical formulation. For each of the CG, DG and FR formulations considered, we perform the following steps:

A) Over-sample the solution polynomial onto a larger set of points, $u^P \to u^Q$ with $Q > Q_P$;
B) Evaluate the flux onto the larger set of points $\mathbf{F}(u^{Q_P}) \to \mathbf{F}(u^Q)$;
C) Over-sample the geometric factors $\frac{\partial x_i}{\partial \xi_j}$ from which the Jacobian determinant J and geometric factors $\mathbf{\Lambda}(\overline{G}_i)$ are calculated;
D) Over-sample the differentiation matrices $\mathbf{D}_{\xi_i}$;
E) (*DG/FR only*) Over-sample the boundary terms $\boldsymbol{b}^e$ or $\bar{\bar{\boldsymbol{b}}}^e$;
F) Assemble the right-hand side onto the larger set of points;
G) Project the calculated solution back to the original set of coefficients through a Galerkin projection.



(ξ_{1_r}, ξ_{2_s})
$(\xi_{1_\ell}, \xi_{2_s})$
$(\xi_{1_\ell}, \xi_{2_m})$
$\mathbf{I}_{\xi_{1_r}\to\xi_{1_\ell}}$
$\mathcal{O}(Q \times P^2)$
$\mathbf{I}_{\xi_{2_s}\to\xi_{2_m}}$
$\mathcal{O}(Q^2 \times P)$
$f^Q(\xi_{1_\ell}, \xi_{2_m}) =$
$= u(\xi_{1_\ell}, \xi_{2_m}) \cdot u(\xi_{1_\ell}, \xi_{2_m})$
(ξ_{1_r}, ξ_{2_s})
$(\xi_{1_\ell}, \xi_{2_s})$
$(\xi_{1_\ell}, \xi_{2_m})$
$\mathbf{GP}_{\xi_{1_\ell}\to\xi_{1_r}}$
$\mathbf{GP}_{\xi_{2_m}\to\xi_{2_s}}$

Fig. 4. Conceptual flow chart of the *Local* dealiasing approach for the 2D/3D tensor-product case.

Note that we do not make any distinctions between the 1D or the 2D/3D cases, nor between different element shapes, since here we do not explicitly exploit the tensor product advantages (although some operators may still have a tensor product form) used in the previous sections for the PDE-dealiasing through consistent integration of the nonlinear terms. This approach is similar in nature to the one presented by Kirby and Karniadakis [19]. However, here we additionally perform an over-sampling of the geometric terms, the Jacobian and the geometric factors. Based on the greater number of floating operations to be performed, this dealiasing technique is more computationally expensive than the consistent integration of the nonlinear terms presented in the previous section but can address both PDE- and geometrical-aliasing issues.

4. Numerical results

In this section, we consider the impact of the dealiasing schemes outlined above, when applied to continuous and discontinuous discretisations, from two perspectives. First, we perform a quantitative study of the effects of these dealiasing techniques using a simple transport advection equation for which a periodic solution is known, but where the advection velocity is a nonlinear polynomial in space. We then consider various cases including deformed elements, the contribution of boundary terms in the dealiasing process, and reinforce the inequalities which lead to exact integration of discretised nonlinear functionals. We finally consider qualitative effects, giving examples of how dealiasing can enhance the stability of both compressible flows and incompressible Large Eddy Simulations. Even if, in the examples presented, the dealiasing techniques show a stabilising effect, dealiasing does not guarantee stability and in other flow studies the authors experienced instabilities even applying a consistent integration of the flux functions. In this case, for stabilising the simulation, they added an artificial viscosity of the type described in [32] and [7].

In our experiments, we chose Gauss–Lobatto–Legendre points for consistent integration. The use of a Gauss–Legendre quadrature is more efficient for reducing the aliasing issues; however, Gauss–Lobatto–Legendre quadrature is less expensive because the interpolants do not need to be evaluated at the elemental boundaries to project flux terms into the volume.

4.1. PDE aliasing

To understand the effect of the PDE-dealiasing techniques presented in the previous section, we consider the transport equation

$$\begin{cases} \dfrac{\partial u}{\partial t} + a_x \dfrac{\partial u}{\partial x} + a_y \dfrac{\partial u}{\partial y} = 0 \\ u(x, y, t = 0) = \exp\{-41[(x + 0.3)^2 + y^2]\} \\ u(x_b, y_b, t) = 0, \\ a_x = \pi\, y^{P_{adv}} g(t), \quad a_y = -\pi\, x^{P_{adv}} g(t), \end{cases} \tag{38}$$

where u is the conserved variable, a_x and a_y are the advection velocities along the x and y directions, (x_b, y_b) denote the boundaries of the numerical domain, P_{adv} is the polynomial order of the flux and $g(t)$ is a time dependent periodic function

$$g(t) = \cos(\pi t/T) \tag{39}$$

Table 2
L_2 errors for the different dealiasing strategies applied to the CG, DG and FR formulations: Case A mesh as in Fig. 6(a).

	Case A – LMM		
	CG	DG	FR-G2
$Q = 5$	0.00175597416164	0.00517893324996	0.00517893324999
$Q = 5, Q_V = 6, Q_I = 5$	–	0.00484792022345	0.00484792022348
$Q = 5, Q_V = 7, Q_I = 5$	–	0.00484792022345	0.00484792022348
$Q = 5, Q_V = 8, Q_I = 5$	–	0.00484792022345	0.00484792022348
$Q = 5, Q_V = 6, Q_I = 6$	0.00175619445714	0.00483526988463	0.00483526988467
$Q = 5, Q_V = 7, Q_I = 7$	0.00175619445714	0.00483526988463	0.00483526988467
$Q = 5, Q_V = 8, Q_I = 8$	0.00175619445714	0.00483526988463	0.00483526988467
	Case A – EMM		
	CG	DG	FR-DG
$Q = 6, Q_V = 6, Q_I = 6$	–	0.00238095562954	0.00238095562953
$Q = 6, Q_V = 7, Q_I = 6$	–	0.00238095562954	0.00238095562953
$Q = 6, Q_V = 8, Q_I = 6$	–	0.00238095562954	0.00238095562953
$Q = 6, Q_V = 6, Q_I = 6$	0.00174622356541	0.00238095562954	0.00238095562953
$Q = 6, Q_V = 7, Q_I = 7$	0.00174622356541	0.00238095562954	0.00238095562953
$Q = 6, Q_V = 8, Q_I = 8$	0.00174622356541	0.00238095562954	0.00238095562953
$Q = 6$	0.00174622356541	0.00238095562954	0.00238095562953
$Q = 7$	0.00174622356566	0.00238095562954	0.00238095562953
$Q = 8$	0.00174622356570	0.00238095562954	0.00238095562953

on the time interval $[0, T]$. The solution u evolves in time in such a way that the initial data is recovered at every period T, that is

$$u(x, y, 0) = u(x, y, T). \tag{40}$$

We select a period $T = 1$ and a final time $4T$ so that the final solution is equal to the initial condition in order to calculate the error. This strategy is the same as that adopted in [33] and it is widely used to evaluate properties of numerical methods when the exact solution is not available.

In the following, we first perform a comparative study between the *Local* and *Global* dealiasing approaches on the case where the advection velocities is linear, so that $P_{adv} = 1$. This will highlight the connections between the two dealiasing approaches and ensures that some general properties hold true for all the results presented in the following sections. We then show how increasing P_{adv}, and therefore the order of the advection velocities, can lead to a more important role of the interfaces in the case of the discontinuous discretisations. In all the simulations we use a tensor-product Lagrangian polynomial basis of order $P = 4$.

4.1.1. Comparative study of the different dealiasing techniques

In this section we show the results obtained using *Local* and *Global* dealiasing techniques for the test case in Eq. (38) when $P_{adv} = 1$ on a regular grid of quadrilaterals, so that no geometrical aliasing effects are present.

For each simulation we use a forward Euler time integration scheme and the time step was chosen to be sufficiently small in order to consider the temporal error negligible. We remark that any explicit time integration scheme such as 2nd or 4th order Runge–Kutta schemes can be used. In space we consider each of the CG, DG and FR discretisations in turn, where the polynomial order is set to $P = 4$ and the initial condition is given by a collocation projection (i.e. using $Q = 5$ quadrature points). Throughout the following numerical results, we will make extensive use of relative L_2 errors defined as

$$L_2 = \sqrt{\sum_{i=0}^{N} (u_i - u_{i,exact})^2 w_i}, \tag{41}$$

where u_i is the numerical solution calculated at each quadrature point, $u_{i,exact}$ is the exact solution, N is the total number of quadrature points and w_i are the quadrature weights.

In Table 2 we show the L_2 errors for the different dealiasing strategies applied to the CG, DG and FR formulations for the mesh shown in Fig. 6(a). Here, 'LMM' stands for lumped mass matrix, which is the diagonal matrix obtained when $Q = 5$ quadrature points are used to obtain a collocation scheme, while 'EMM' refers to the exact mass matrix obtained for higher quadrature orders. This table illustrates the saturation of the L_2 error up to machine precision for $Q = 6$ as well as the equivalence between *Local* and *Global* dealiasing when the number of quadrature points used for the nonlinear terms are the same in the two approaches, since the geometric terms for a regular grid are constant. We also note that the equivalence of the DG and FR schemes presented in [28] holds when using a larger number of quadrature points than a collocation method.

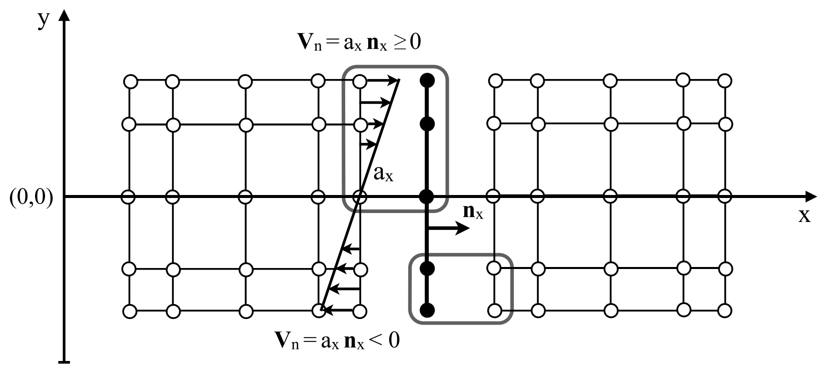


Fig. 5. Upwind flux at the interface between two elements: aliasing arising from mixed information coming from the left and the right element.

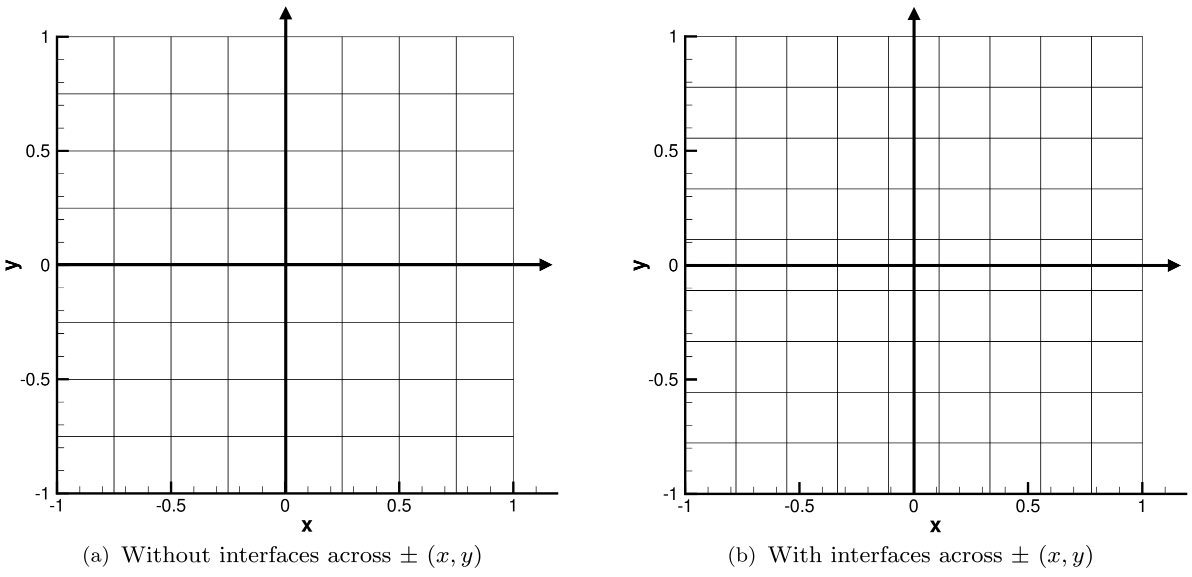

1
0.5
y
0
-0.5
-1
-1
-0.5
0
0.5
1
x
(a) Without interfaces across $\pm (x, y)$
(b) With interfaces across $\pm (x, y)$

Fig. 6. Meshes for the 2D linear advection test cases: Case A in (a) and Case B in (b).

Regarding the dealiasing of the interface fluxes, we note that we obtain a saturation of the error up to machine precision. This is because the interface fluxes for the mesh in Fig. 6(a) are polynomial functions since their values come from just one side with respect to a given boundary.[2] If we instead consider the mesh in Fig. 6(b), we are in a case where the interface fluxes come from both sides of a given interface as shown in Fig. 5; therefore, they do not lie in the polynomial space and error saturation up to machine precision cannot be expected as presented in Table 3. This means that, in general, it is difficult to fully control the PDE-aliasing arising from the interfaces because, in this case the functions representing the interface fluxes lie outside a polynomial space. However, the error does decrease as the number of quadrature points is increased.

4.1.2. Role of dealiasing for higher order advection velocities

In this section, we show the results using advection velocities with non-constant coefficients (i.e. spatially varying polynomials), so that the order P_{adv} is chosen to be greater than one in Eq. (38). The numerical parameters are chosen to be identical to those in the previous section, and we use the grid depicted in Fig. 6(a) to avoid non-saturation of the error when dealiasing the interfaces.

In Fig. 7, each point represents the difference between the L_2 error calculated using (Q_V, Q_I) quadrature points and the error calculated using $(Q_V + 1, Q_I + 1)$ quadrature points in the LMM case.[3] We note that the machine-precision saturation

[2] Note that all the simulations in this and the following section were run using an upwind interface flux.

[3] Note that the L_2 was calculated on an enriched quadrature space.

Table 3
L_2 errors for the different dealiasing strategies applied to the DG and FR formulations: Case B mesh as in Fig. 6(b).

	Case B – LMM	
	DG	FR-G2
$Q = 5$	0.00279358970635	0.00279358970635
$Q = 5, Q_V = 6, Q_I = 5$	0.00265346210971	0.00265346210971
$Q = 5, Q_V = 7, Q_I = 5$	0.00265346210971	0.00265346210971
$Q = 5, Q_V = 8, Q_I = 5$	0.00265346210971	0.00265346210971
$Q = 5, Q_V = 6, Q_I = 6$	0.00266014915428	0.00266014915428
$Q = 5, Q_V = 7, Q_I = 7$	0.00265747641942	0.00265747641942
$Q = 5, Q_V = 8, Q_I = 8$	0.00265532284115	0.00265532284115
	Case B – EMM	
	DG	FR-DG
$Q = 5, Q_V = 6, Q_I = 5$	0.00155395894760	0.00155395894760
$Q = 5, Q_V = 7, Q_I = 5$	0.00155395894760	0.00155395894760
$Q = 5, Q_V = 8, Q_I = 5$	0.00155395894760	0.00155395894760
$Q = 5, Q_V = 6, Q_I = 6$	0.00155395894760	0.00155395894760
$Q = 5, Q_V = 7, Q_I = 7$	0.00155235541132	0.00155235541132
$Q = 5, Q_V = 8, Q_I = 8$	0.00155131919159	0.00155131919159
$Q = 6$	0.00155395894760	0.00155395894760
$Q = 7$	0.00155235541132	0.00155235541132
$Q = 8$	0.00155131919159	0.00155131919159

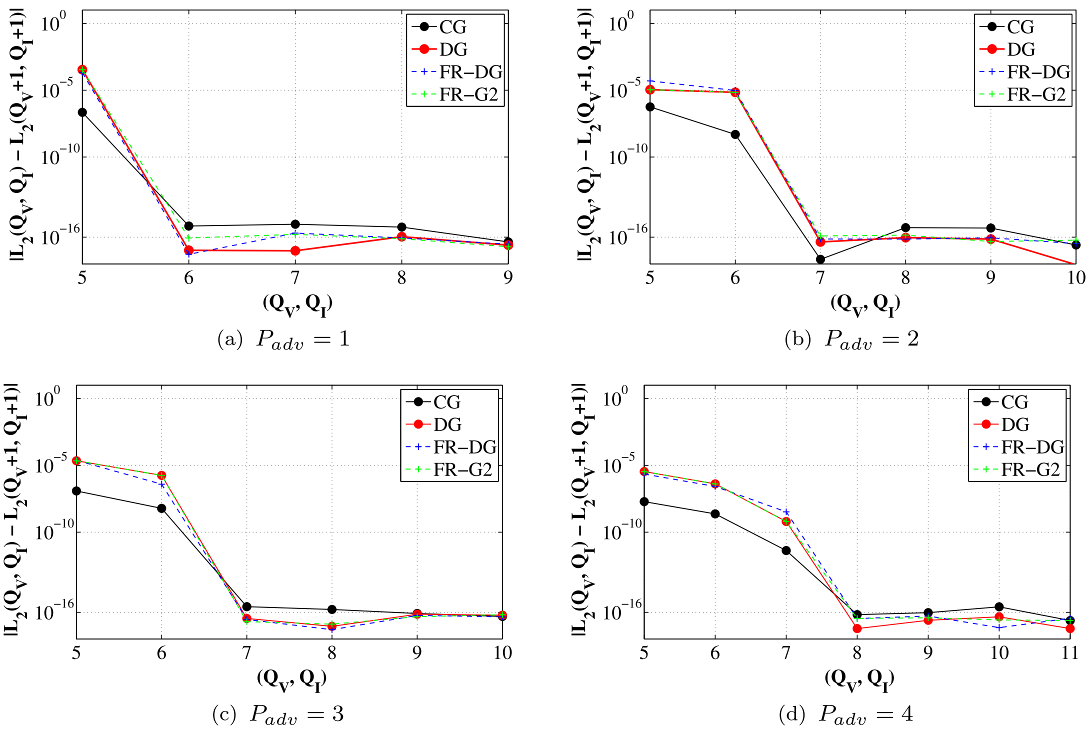


Fig. 7. Differences between the L_2 errors obtained with (Q_I, Q_V) and $(Q_I + 1, Q_V + 1)$ vs. (Q_I, Q_V) using the *Local* dealiasing technique for the case of LMM.

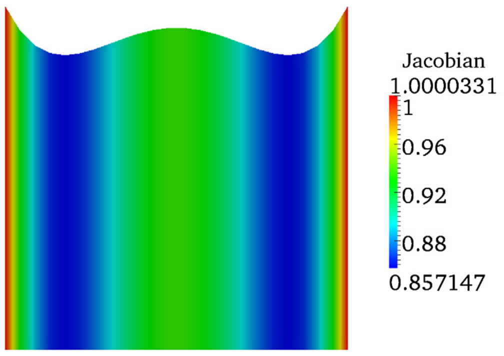
of the error satisfies the inequality:

$$\widetilde{Q} \geq P + \frac{P_{adv}}{2} + \frac{3}{2}, \qquad P \geq P_{adv}, \tag{42}$$

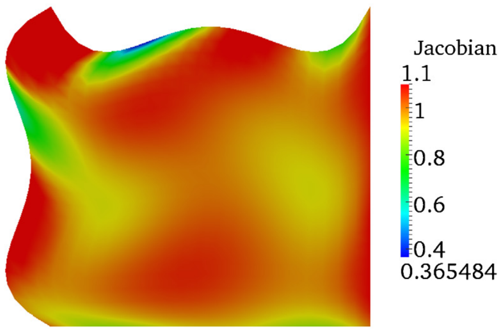
where $\widetilde{Q}$ is the minimum number of quadrature points to exactly integrate the PDE-aliasing for a given polynomial advection velocity possessing a nonlinearity of order P_{adv} and for a given expansion order P.

Table 4
Percentage difference between the L_2 errors for *Local* (Q_V), *Local* (Q_I) and *Local* (Q_V, Q_I) at error saturation.

	LMM			
	$\%diff(L_2(Q_V)_{sat} - L_2(Q_V, Q_I)_{sat})$		$\%diff(L_2(Q_I)_{sat} - L_2(Q_V, Q_I)_{sat})$	
	DG	FR-G2	DG	FR-G2
$P_{adv} = 1$	0.2616	0.2616	7.7022	7.7022
$P_{adv} = 2$	5.5913	5.5913	3.2610	3.2610
$P_{adv} = 3$	10.410	10.410	12.464	12.464
$P_{adv} = 4$	5.9054	5.9054	3.5643	3.5643
	EMM			
	$\%diff(L_2(Q_V)_{sat} - L_2(Q_V, Q_I)_{sat})$		$\%diff(L_2(Q_I)_{sat} - L_2(Q_V, Q_I)_{sat})$	
	DG	FR-DG	DG	FR-DG
$P_{adv} = 1$	0.0	0.0	0.0	0.0
$P_{adv} = 2$	0.1262	0.1262	1.3184	1.3184
$P_{adv} = 3$	0.1770	0.1770	0.0842	0.0842
$P_{adv} = 4$	−0.1014	−0.1014	0.9086	0.9086



(a) Top edge curved



(b) Top and left edges curved

Fig. 8. Examples of 4th order meshes employed to investigate geometrical aliasing.

In Appendix B.1, Fig. 17 shows the actual L_2 errors vs. (Q_V, Q_I) for the LMM case while Tables 6, 7, 8 and 9, show the tabulated values of the L_2 errors for $P_{adv} = 1, 2, 3, 4$, which also include the EMM case. From these results it is possible to see how, for the discontinuous formulations, the use of *Local* dealiasing on both the volumetric and interface fluxes was, for almost all of the cases considered, the best choice in terms of L_2 error among the dealiasing techniques applied, except for EMM $P_{adv} = 4$ where instead *Local* dealiasing of only the volumetric flux provided a better result.

To further quantify the effects of interface dealiasing, in Table 4 we show the difference between the L_2 errors obtained for *Local* dealiasing of only the volumetric flux, and *Local* dealiasing of both volumetric and interface fluxes, taken at the point where the error saturates according to Fig. 7 for each value of P_{adv}. We see from this data, how the difference between performing only volumetric dealiasing and volumetric plus interface dealiasing increases as P_{adv} increases except for the last case ($P_{adv} = 4$). The gap between using a lumped (LMM) and an exact (EMM) mass matrix is notable, and is due to the implicit *Global* dealiasing in the case of EMM which in turn reduces the effects of the interface dealiasing (through the use of a quadrature that can integrate a higher polynomial order).

4.2. Link between geometrical- and PDE-aliasing

Now that the aliasing effects of the discretisation of the PDE have been examined, in this section we show how geometrical-aliasing can be linked to PDE-aliasing, with some differences arising from the fact that geometrical nonlinearities play a slightly different role in the underlying formulation compared to PDE nonlinearities.

We consider a mesh composed of a single standard quadrilateral element $\Omega_{st} = [-1, 1]^2$, where we curve either only the top edge, or both top and left edges. For each configuration, we applied different curvatures to the edges by means of Legendre polynomials of order $P = 1, 2, 3$ and 4. In Figs. 8(a) and 8(b) we show two examples for the mesh of order four.

In each case, the isoparametric mapping, and therefore the Jacobian determinant, is represented by an expansion of basis functions over the standard element, so that for example a $P = 4$ expansion contains $5 \times 5 = 25$ different unique modes. Fig. 9 shows the magnitude of each modal coefficients of the Jacobian determinant for the 4th order case when projected onto an orthonormal basis of order 5 (leading to 36 coefficients), in order to emphasise that the polynomial space is sufficiently large to capture the Jacobian mapping. In particular, we note that the Jacobian for both cases is effectively of fourth order, but in the case of Fig. 9(a) we have nonzero modes only in the x-direction up to the fifth coefficient,

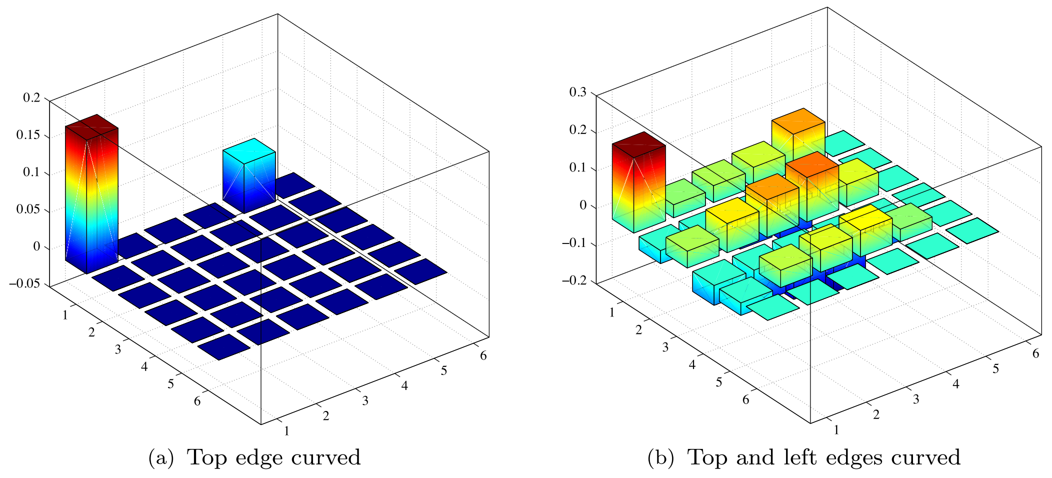


Fig. 9. Coefficients in an orthonormal expansion basis for the 4th order meshes showed in Fig. 8.

whereas in the case of Fig. 9(b) the nonzero modes span both x- and y-directions again up to the fifth coefficient. The sixth coefficient in both directions is zero, showing adequate support for the Jacobian determinant expansion.

The test case we consider is identical to that of Eq. (38), aside from the use of spatially-constant advection velocities

$$a_x = \frac{\pi}{2}\, g(t), \quad a_y = -\frac{\pi}{2}\, g(t), \tag{43}$$

where $g(t)$ is defined in Eq. (39). We choose a period $T = 0.5$ and a final time $40T$. For the time-integration we used a 2nd-order Runge–Kutta scheme while the polynomial order was $P = 14$ in order to have a sufficiently resolved problem. The initial condition was applied using a collocation projection. Throughout the results in this subsection, we used a *Global* dealiasing technique in order to target the geometrical aliasing.

We first present estimates of the quadrature necessary for correctly integrating a deformed or curved mesh. For doing so, we take into account the leading order within the problem under investigation, which is the mass matrix for both CG and DG discretisations. In Fig. 10 each point denotes the difference between two mass matrices, the first obtained for Q and the second for $Q + 1$ quadrature points and it is calculated as follows

$$|\Delta \mathbf{M}|_{L_2} = \sqrt{\sum_{i,j} \left[\mathbf{M}_{i,j}(Q) - \mathbf{M}_{i,j}(Q+1)\right]^2}. \tag{44}$$

Table 10 in Appendix B.2 quantifies the results presented in Fig. 10. We note that the mass matrix does not change (approximately up to machine precision) after we have reached the minimum number of quadrature points to exactly integrate the polynomial order used for describing the geometry, given by the inequality

$$\begin{aligned} &\widetilde{Q} \geq P + \frac{P_{geom}}{2} + \frac{3}{2}, \\ &P \geq P_{geom}, \end{aligned} \tag{45}$$

where $\widetilde{Q}$ is the minimum number of quadrature points to exactly integrate the mass matrix for a given polynomial of order P_{order} describing the geometry and for a given expansion order P. Eq. (45) is identical to Eq. (42) with the only exception being that P_{geom} now refers to the polynomial order of the geometry deformation. The above result is consistent with the Gauss–Lobatto–Legendre quadrature rules.

Fig. 11 shows the differences in terms of L_2 errors as defined in Eq. (44) instead of differences between mass matrices. The main points to note are the following:

- CG reaches a saturation of the error approximately up to eleven or twelve digits for all of the meshes. This is due to the tolerances used for inverting the mass matrix which, in this case, do not achieve machine precision.
- DG and FR show machine-precision saturation of the error for the linear mesh (i.e. $J \in \mathcal{P}_1$) whilst the saturation happens at a higher level as the mesh order increases. This behaviour is due to the interface fluxes which come from both the left (internal domain) and right (boundary condition) sides of a given interface, since the normals are spatially varying across the curved edge(s). Therefore, for complex geometries, it is difficult to fully control the geometrical aliasing arising from the interfaces.
- The non-saturation of the error to machine precision for each of the spatial discretisations is not due to rational functions which do not lie in the polynomial space. In fact, the geometric terms encounter cancellations which lead to just one geometric factor which does belong to the polynomial space considered. Thus, it can be fully represented within the polynomial representation used for these test cases.

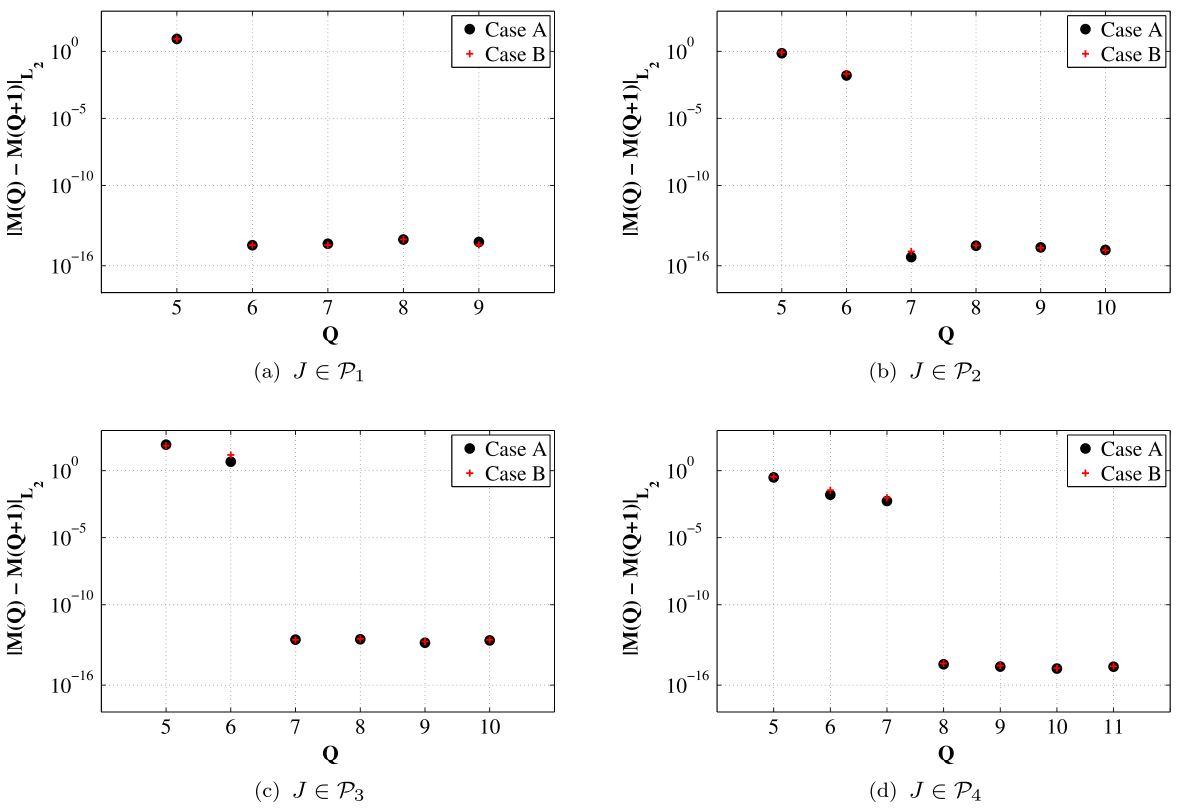


Fig. 10. L_2 norm of the difference between mass matrices calculated with Q and $Q + 1$ GLL points for the four orders considered and using both the mesh configurations shown in Fig. 8. Case A corresponds to Fig. 9(a) and Case B corresponds to Fig. 9(b).

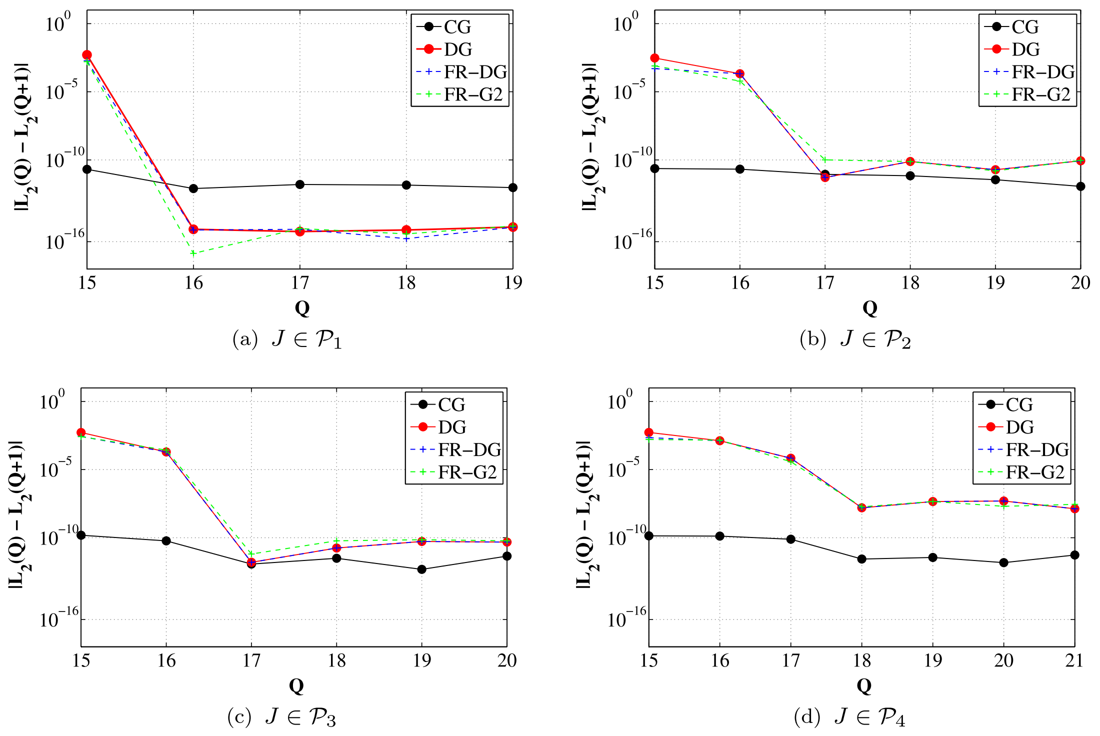


Fig. 11. Differences between the L_2 errors obtained with Q and $Q + 1$ GLL points for the four orders considered using the mesh in Fig. 8(a).

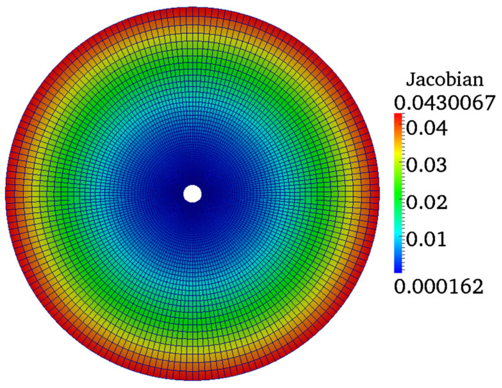


Fig. 12. Jacobian distribution on the mesh used for the compressible inviscid simulation of a cylinder.

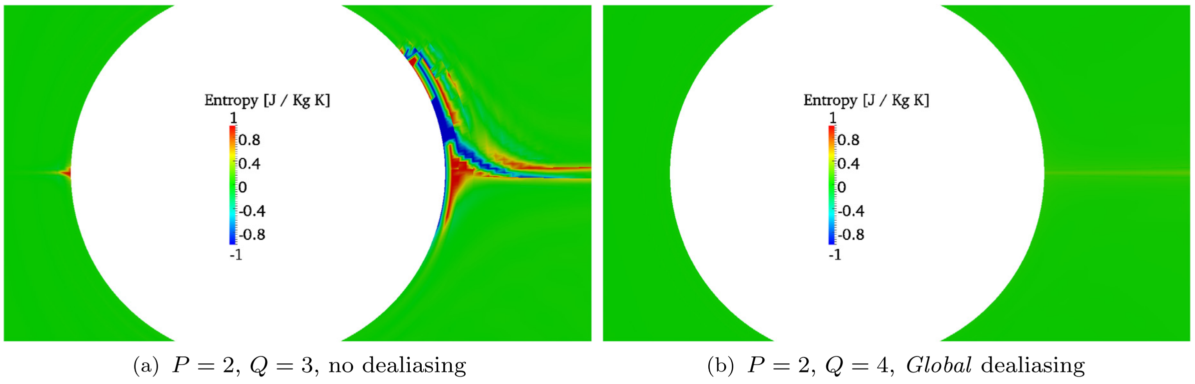


Fig. 13. Numerically-generated entropy for $P = 2$.

As a final comment, we note that the connections between DG and FR established in [28] also hold for a deformed/curved mesh.

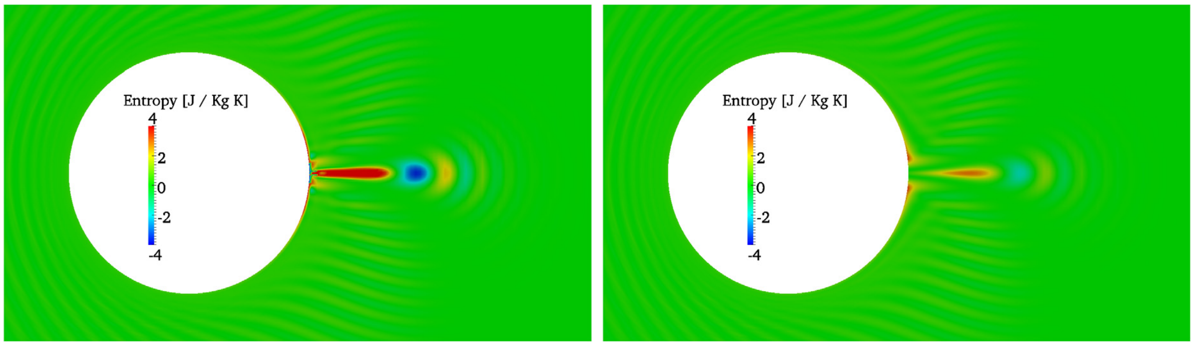
4.3. *Flow applications*

Having considered the quantitative effects of the *Local* and *Global* dealiasing techniques, in this section we consider two examples that highlight how stability can be enhanced by using appropriate dealiasing strategies. We begin by considering a compressible Euler test case, and show that the lack of dealiasing can lead to the buildup of numerical entropy in the solution field. Secondly, we consider the LES of a NACA 0012 wingtip, and show how dealiasing can increase the robustness of the simulation. As previously stated, however, the consistent integration does not imply stability and in some simulations the high frequency modes of the nonlinear flux functions are not eliminated through dealiasing. In this case, other strategies need to be applied (eventually in conjunction with consistent integration) such as the use of artificial viscosity or modal filtering.

4.3.1. *Compressible inviscid subsonic flow past a cylinder*

In this section, we present an inviscid compressible flow past a cylinder governed by the compressible Euler equations. We set the Mach number to 0.2, use an FR–DG scheme for the spatial discretisation and a 4th-order Runge–Kutta scheme for the time-integration. We use a circular mesh composed of 9612 quadrilateral elements and the cylinder is described by third order splines in order to achieve a high order description of the geometry. In Fig. 12 we show the mesh used for simulations with the Jacobian determinant distribution. We use two different polynomial orders: $P = 2$ and $P = 3$ and the *Global* dealiasing technique. Fig. 12 indicates that the elements of the mesh are of good quality and not highly distorted, although a third order curvature is applied to the cylinder wall. We therefore expect PDE aliasing to play a bigger role than geometrical aliasing.

In Fig. 13(a) we show the numerically-generated entropy for the case $P = 2$, $Q = 3$ without dealiasing techniques applied. We can clearly see a buildup of entropy from the posterior stagnation point, which is sensitive to aliasing-driven instabilities due to the strong gradients of the solution in this region. Note that ideally for an inviscid simulation, the entropy should be numerically zero. In this case the simulation diverged after a few timesteps; however, by applying the *Global* dealiasing techniques using $Q = 4$, we were able to reduce the aliasing errors. In Fig. 13(b) we show a snapshot of the solution field taken at the same value of time as Fig. 13(a). We note that the numerically-generated entropy at the



(a) $P = 3$, $Q = 4$, no dealiasing (b) $P = 3$, $Q = 5$, *Global* dealiasing

Fig. 14. Numerically-generated entropy for $P = 3$.

Table 5
Maximum and minimum values across the mesh of numerically-generated entropy for both $P = 2$ and $P = 3$.

	$P = 2$	
	Entropy-Min [J/Kg K]	Entropy-Max [J/Kg K]
NO	−96.36	94.35
Global $Q = 4$	−0.86666943708297	1.59384922648788
Global $Q = 5$	−0.85200075803792	1.54735068797723
Global $Q = 6$	−0.85130979874330	1.54637692206548
Global $Q = 7$	−0.85130979868331	1.54637692206548
	$P = 3$	
	Entropy-Min	Entropy-Max
NO	−1118.61	4347.60
Global $Q = 5$	−1.67319315760229	3.49726118185079
Global $Q = 6$	−1.16291305386484	3.39484597010811
Global $Q = 7$	−1.13721148529237	3.39967822044820
Global $Q = 8$	−1.13547005462042	3.39977179691589
Global $Q = 9$	−1.13517999136988	3.39977189419113

posterior stagnation point is no longer present. The simulation for $P = 3$ and $Q = 4$ depicted in Fig. 14 was also unstable. As in the previous case (i.e. $P = 2$, $Q = 3$), the use of the *Global* dealiasing technique stabilised the simulation.

In Table 5 we report the maximum and minimum values of entropy across the computational domain for all the cases considered. As we can see, the numerically-generated entropy, which in this case can be seen as a measure of the numerical error, roughly saturates at $Q = 6$ for $P = 2$ and at $Q = 8$ for $P = 3$. Clearly the saturation is not up to machine precision because of the high-order description of the geometry and the deformed elements throughout the domain which introduce additional 'nonlinearities' into the problem. Also, the right-hand side of the compressible Euler equations is composed by rational functions. This in turn means that the right-hand side cannot be represented in a polynomial space and, therefore, a machine-precision saturation of the solution cannot be expected. However, a minimisation of the relative error can be seen as Q is increased and this may also indicate that the PDE-aliasing is potentially dominant for this problem.

Note that, for both the polynomial orders considered, we also applied the *Local* dealiasing approach but the results are not shown here, as they are similar to those obtained using the *Global* dealiasing technique. This further reinforces the role of the PDE-aliasing for this problem.

4.3.2. Incompressible viscous flow past a wing tip

In this section we show the results of an incompressible viscous flow past a NACA 0012 wingtip, originally studied experimentally by Chow et al. [34].

The simulation was obtained using the CG approach, with a velocity correction scheme being used to discretise the incompressible Navier–Stokes equations [35]. While the experimental Reynolds number was set at $\mathrm{Re} = 4.6 \times 10^6$, initial simulations were performed at a lower Reynolds number of $\mathrm{Re} = 1.2 \times 10^6$. *Local* dealiasing, when combined with a spectral vanishing viscosity to dampen high-frequency energy buildup, yielded a stable simulation. However, in order to increase the Reynolds number and bring the simulation in line with experimental conditions, we found that without an increase in the global quadrature order (i.e. *Global* dealiasing technique), the simulation quickly became unstable. This highlights the role of geometric as well as PDE aliasing under these conditions for the mesh and polynomial resolution considered.

In Fig. 15 we illustrate the dynamics of the flow by showing the isocontours of helicity for a $P = 3$ approximation, where the domain is represented using prismatic elements around the boundary and tetrahedral elements in the rest of the domain. In Figs. 16(a) and 16(b), for the same simulation, we represent the aliasing errors. The top image shows up to 30%

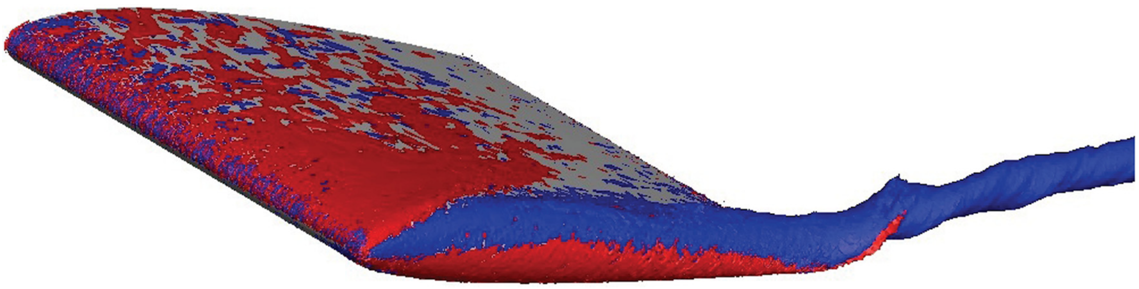


Fig. 15. Helicity for an incompressible viscous flow over a NACA 0012 wing.

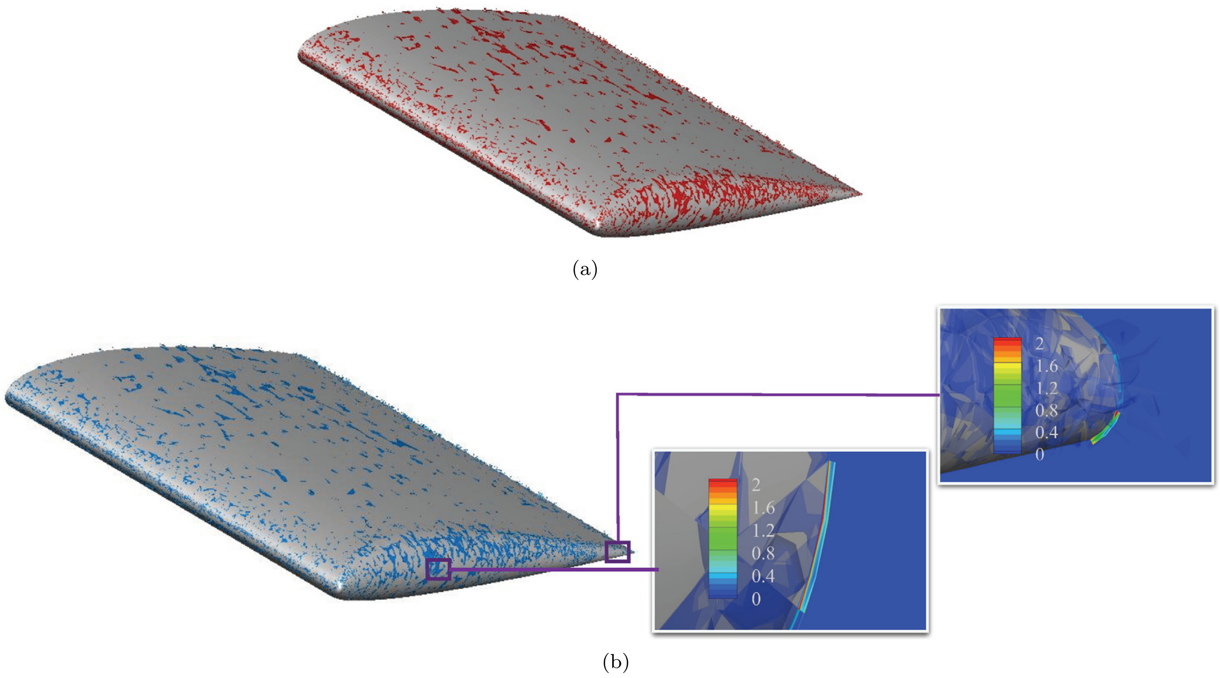


Fig. 16. Aliasing errors at 30% of the magnitude of the nonlinear terms (a) and close up view of the regions near the wing surface (b).

aliasing error with respect to the magnitude of the nonlinear terms while the bottom image shows regions of aliasing near the wing surface where we observe up to 600% error, highlighting the crucial nature of dealiasing in this problem.

5. Summary and conclusions

We have presented two dealiasing approaches based on the concept of consistent integration of the nonlinear terms or over-integration if the integration order is based on exact integration of linear operators. The first technique presented, namely the *Local* approach, targets the PDE-aliasing sources which arise from the nonlinearities contained in the PDE itself. This approach exploits the tensor product structure of the elements to reduce the number of floating point operations, making it computationally efficient at higher polynomial orders. The application of *Local* dealiasing is particularly useful when we have localised nonlinearities such as in the incompressible Navier–Stokes equations if geometric aliasing is not significant. The second technique, namely the *Global* approach, involves the use of higher quadrature order than is typically necessary for linear PDEs. This technique targets both PDE- and geometrical-aliasing sources which arise when deformed or curved elements are used, as often occurs in industrially-relevant problems where complex geometries are present. In contrast to the first approach, this technique is computationally more intensive, but it can address all the aliasing sources arising in any spectral element method when coupled with a sufficiently large number of quadrature points. We applied both the dealiasing techniques to different high-order spectral element methods, showing the main implementation differences between continuous (CG) and discontinuous (DG and FR) discretisations.

Based on the above analysis and supported by numerical examples we can state that:

1. The consistent integration rules presented in [19] are generally true and, for GLL points, can be complemented by the following expression
$$\widetilde{Q} \geq P_{exp} + \frac{P_{order}}{2} + \frac{3}{2},$$
$$P_{exp} \geq P_{order},$$
where $\widetilde{Q}$ is the minimum number of quadrature points to exactly integrate the highest-degree of nonlinearity P_{order} within the problem considered and for a given expansion order P_{exp}. Note that in the above expression P_{order} can be a combination of geometrical and PDE nonlinearities and therefore the overall degree of the nonlinearity can be higher than that dictated by the PDE itself.
2. In a discontinuous discretisation, it is generally not possible to fully control (i.e. up to machine precision) all aliasing effects, since the boundary terms which dictate interface contributions introduce non-polynomial functions into the problem being solved. In addition, geometrical aliasing affects a discontinuous discretisation more than a continuous, due to the geometric factors appearing in the boundary term.
3. Geometrical dealiasing, in contrast to PDE aliasing, is responsible for modifying the mass matrix and it may, therefore, change the numerical dissipation and dispersion properties of the discretisation employed as shown in [36] and [37].
4. The connections between DG and FR presented in [28] hold also for the dealiasing strategies described.
5. The two dealiasing strategies can be used in a complementary manner and when applied to challenging applications they have proven effective and have increased the numerical stability of the simulations although they do not *guarantee* stability. This result is in agreement with the findings in the recent work of Malm, Schlatter et al. [26] where a possible connection between numerical stability and consistent integration has been shown for the incompressible Navier–Stokes equation (i.e. polynomial nonlinearities).

We additionally note that interface dealiasing may play a bigger role as the order of the interface flux function increases, thus becoming comparable or more important than the volumetric dealiasing contribution.

Acknowledgements

This work was supported by the Laminar Flow Control Centre funded by Airbus/EADS and EPSRC under grant EP/I037946. We thank Dr. Colin Cotter for helpful discussions and Jean-Eloi Lombard for his assistance in the generation of results and figures for the NACA 0012 simulation. PV acknowledges the Engineering and Physical Sciences Research Council for their support via an Early Career Fellowship (EP/K027379/1). SJS additionally acknowledges Royal Academy of Engineering support under their research chair scheme. Data supporting this publication can be obtained on request from nektar-users@imperial.ac.uk.

Appendix A. Elemental mapping

In this appendix we describe how the geometric terms are treated and how integration and differentiation are achieved. For the sake of clarity, we consider the 2D case. The extension to 3D is straightforward and can be found in [24].

Each local element of the computational domain can be mapped onto a standard element $\Omega_S = [-1, 1] \times [-1, 1]$. For an arbitrary-shaped straight-sided quadrilateral with vertices A, B, C and D, the mapping onto the standard region is:

$$\mathbf{x} = (x_1, x_2)^T = \chi(\xi_1, \xi_2) = \mathbf{x}^A \frac{1-\xi_1}{2}\frac{1-\xi_2}{2} + \mathbf{x}^B \frac{1+\xi_1}{2}\frac{1-\xi_2}{2} + \mathbf{x}^C \frac{1+\xi_1}{2}\frac{1+\xi_2}{2} + \mathbf{x}^D \frac{1-\xi_1}{2}\frac{1+\xi_2}{2}. \tag{A.1}$$

For an arbitrary-shaped straight-sided triangle with vertices A, B and C, where C is the collapsed vertex, the map is:

$$\mathbf{x} = (x_1, x_2)^T = \chi(\xi_1, \xi_2) = \mathbf{x}^A \frac{-\xi_2 - \xi_1}{2} + \mathbf{x}^B \frac{1+\xi_1}{2} + \mathbf{x}^C \frac{1+\xi_2}{2}. \tag{A.2}$$

An analogous map can be constructed also for three-dimensional straight-sided elements. In the case of curvilinear regions we can use the standard isoparametric mapping which for an arbitrary-shaped curved-sided two-dimensional element is:

$$\mathbf{x} = (x_1, x_2)^T = \chi(\xi, \eta) = \sum_{p,q=0}^{Q_P} \hat{\mathbf{x}}_{p,q}\phi_p(\xi_1)\phi_q(\xi_2), \tag{A.3}$$

where ϕ_p and ϕ_q are the same basis functions used for representing the solution. We note that, if $\mathbf{x}$ is a polynomial of order $G < Q_P$, the mapping is subparametric. An analogous mapping can be adopted for a three-dimensional curvilinear element.

Integration within a general-shaped region To integrate over an element Ω_e of a two-dimensional domain, we transform this region into the standard region Ω_S and we have:

$$\int_{\Omega_e} u(x_1, x_2) dx_1 dx_2 = \int_{\Omega_e} u(\xi_1, \xi_2) |J_{2D}| d\xi_1 d\xi_2 , \tag{A.4}$$

where J_{2D} is the two-dimensional Jacobian defined as

$$J_{2D} = \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2}\frac{\partial x_2}{\partial \xi_1} . \tag{A.5}$$

The partial derivative can be evaluated using the maps defined in Eqs. (A.1), (A.2), (A.3). If the element is straight-sided and has parallel sides, then the partial derivatives and therefore the Jacobian are constant. For deformed elements the Jacobian can be evaluated and stored at the quadrature points. This represents the Jacobian as a polynomial function and can, therefore, increase the polynomial order of the integrand. The integral is then evaluated by an appropriate quadrature rule.

An analogous procedure is used for a three-dimensional element where the three-dimensional Jacobian J_{3D} can be evaluated from the partial derivatives given by the mappings.

Differentiation within a general-shaped region To differentiate over an element Ω_e we use the chain rule:

$$\frac{\partial u}{\partial x_i} = \sum_{j=1}^{d} \frac{\partial u}{\partial \xi_j}\frac{\partial \xi_j}{\partial x_i} , \tag{A.6}$$

where d is the dimension of the domain. For $d = 2$:

$$\begin{pmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{pmatrix}^{-1} = \frac{1}{J_{2D}} \begin{pmatrix} \frac{\partial x_2}{\partial \xi_2} & -\frac{\partial x_1}{\partial \xi_2} \\ -\frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_1} \end{pmatrix} \tag{A.7}$$

The terms on the right-hand side are known from the mapping. We can now evaluate the derivatives as all the partial derivatives can be expressed in terms of differentials with respect to ξ_1 and ξ_2.

Appendix B. Tabulated L_2 errors

B.1. Role of dealiasing for higher order advection velocities

Table 6
L_2 errors for *Local* strategy applied to the DG and FR formulations. Comparative evaluation of the influence of interface dealiasing for $P_{adv} = 1$.

	$P_{adv} = 1$ – LMM		
	CG	DG	FR-G2
$Q = 5$	0.00175597416164	0.00517893324998	0.00517893325001
$Q = 5, Q_V = 6, Q_I = 5$	0.00175619445714	0.00484792022346	0.00484792022350
$Q = 5, Q_V = 7, Q_I = 5$	0.00175619445714	0.00484792022346	0.00484792022350
$Q = 5, Q_V = 5, Q_I = 6$	–	0.00520769016821	0.00520769016825
$Q = 5, Q_V = 5, Q_I = 7$	–	0.00520769016821	0.00520769016825
$Q = 5, Q_V = 6, Q_I = 6$	–	0.00483526988465	0.00483526988468
$Q = 5, Q_V = 7, Q_I = 7$	–	0.00483526988465	0.00483526988468
	$P_{adv} = 1$ – EMM		
	CG	DG	FR-DG
$Q = 6, Q_V = 6, Q_I = 6$	0.00174622356541	0.00238095562953	0.00238095562953
$Q = 6, Q_V = 7, Q_I = 6$	0.00174622356541	0.00238095562953	0.00238095562953
$Q = 6, Q_V = 6, Q_I = 6$	–	0.00238095562953	0.00238095562953
$Q = 6, Q_V = 6, Q_I = 7$	–	0.00238095562953	0.00238095562953
$Q = 6, Q_V = 6, Q_I = 6$	–	0.00238095562953	0.00238095562953
$Q = 6, Q_V = 7, Q_I = 7$	–	0.00238095562953	0.00238095562953

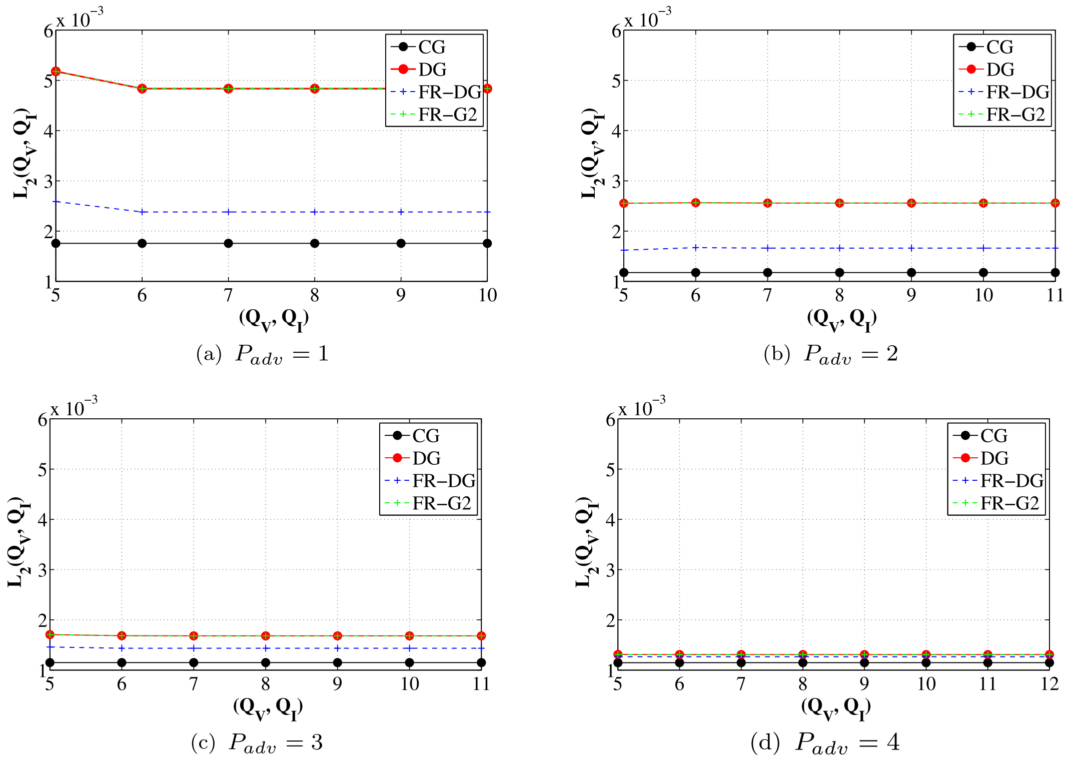


Fig. 17. L_2 errors vs. (Q_I, Q_V) using the *Local* dealiasing technique for the case of LMM.

Table 7
L_2 errors for *Local* strategy applied to the DG and FR formulations. Comparative evaluation of the influence of interface dealiasing for $P_{adv} = 2$.

	$P_{adv} = 2$ – LMM		
	CG	DG	FR-G2
$Q = 5$	0.00117531800986	0.00255493783002	0.00255493783002
$Q = 5, Q_V = 6, Q_I = 5$	0.00117588457750	0.00273091929262	0.00273091929261
$Q = 5, Q_V = 7, Q_I = 5$	0.00117588945474	0.00270213303084	0.00270213303084
$Q = 5, Q_V = 8, Q_I = 5$	0.00117588945474	0.00270213303084	0.00270213303084
$Q = 5, Q_V = 5, Q_I = 6$	–	0.00264115695244	0.00264115695244
$Q = 5, Q_V = 5, Q_I = 7$	–	0.00264249943410	0.00264249943410
$Q = 5, Q_V = 5, Q_I = 8$	–	0.00264249943410	0.00264249943410
$Q = 5, Q_V = 6, Q_I = 6$	–	0.00256594327336	0.00256594327336
$Q = 5, Q_V = 7, Q_I = 7$	–	0.00255904806344	0.00255904806344
$Q = 5, Q_V = 8, Q_I = 8$	–	0.00255904806344	0.00255904806344
	$P_{adv} = 2$ – EMM		
	CG	DG	FR-DG
$Q = 6, Q_V = 6, Q_I = 6$	0.00117437588291	0.00167138551874	0.00167138551874
$Q = 6, Q_V = 7, Q_I = 6$	0.00117437917575	0.00166401725514	0.00166401725514
$Q = 6, Q_V = 8, Q_I = 6$	0.00117437917575	0.00166401725514	0.00166401725514
$Q = 6, Q_V = 6, Q_I = 6$	–	0.00167138551874	0.00167138551874
$Q = 6, Q_V = 6, Q_I = 7$	–	0.00168383066655	0.00168383066655
$Q = 6, Q_V = 6, Q_I = 8$	–	0.00168383066655	0.00168383066655
$Q = 6, Q_V = 6, Q_I = 6$	–	0.00167138551874	0.00167138551874
$Q = 6, Q_V = 7, Q_I = 7$	–	0.00166192034989	0.00166192034989
$Q = 6, Q_V = 8, Q_I = 8$	–	0.00166192034989	0.00166192034989

Table 8
L_2 errors for *Local* strategy applied to the DG and FR formulations. Comparative evaluation of the influence of interface dealiasing for $P_{adv} = 3$.

	$P_{adv} = 3 - \text{LMM}$		
	CG	DG	FR-G2
$Q = 5$	0.00114955111915	0.00170667104239	0.00170667104239
$Q = 5$, $Q_V = 6$, $Q_I = 5$	0.00114967291833	0.00195005809316	0.00195005809316
$Q = 5$, $Q_V = 7$, $Q_I = 5$	0.00114967894471	0.00185789594691	0.00185789594691
$Q = 5$, $Q_V = 8$, $Q_I = 5$	0.00114967894471	0.00185789594691	0.00185789594691
$Q = 5$, $Q_V = 5$, $Q_I = 6$	–	0.00188971296917	0.00188971296917
$Q = 5$, $Q_V = 5$, $Q_I = 7$	–	0.00189247015965	0.00189247015965
$Q = 5$, $Q_V = 5$, $Q_I = 8$	–	0.00189247015965	0.00189247015965
$Q = 5$, $Q_V = 6$, $Q_I = 6$	–	0.00167138551874	0.00167138551874
$Q = 5$, $Q_V = 7$, $Q_I = 7$	–	0.00168273243243	0.00168273243243
$Q = 5$, $Q_V = 8$, $Q_I = 8$	–	0.00168273243243	0.00168273243243
	$P_{adv} = 3 - \text{EMM}$		
	CG	DG	FR-DG
$Q = 6$, $Q_V = 6$, $Q_I = 6$	0.00114946897363	0.00143565622567	0.00143565622567
$Q = 6$, $Q_V = 7$, $Q_I = 6$	0.00114947378466	0.00143858460677	0.00143858460677
$Q = 6$, $Q_V = 8$, $Q_I = 6$	0.00114947378466	0.00143858460677	0.00143858460677
$Q = 6$, $Q_V = 6$, $Q_I = 6$	–	0.00143565622567	0.00143565622567
$Q = 6$, $Q_V = 6$, $Q_I = 7$	–	0.00143725250381	0.00143725250381
$Q = 6$, $Q_V = 6$, $Q_I = 8$	–	0.00143725250381	0.00143725250381
$Q = 6$, $Q_V = 6$, $Q_I = 6$	–	0.00143565622567	0.00143565622567
$Q = 6$, $Q_V = 7$, $Q_I = 7$	–	0.00143604324582	0.00143604324582
$Q = 6$, $Q_V = 8$, $Q_I = 8$	–	0.00143604324582	0.00143604324582

Table 9
L_2 errors for *Local* strategy applied to the DG and FR formulations. Comparative evaluation of the influence of interface dealiasing for $P_{adv} = 4$.

	$P_{adv} = 4 - \text{LMM}$		
	CG	DG	FR-G2
$Q = 5$	0.00114765457261	0.00131289321292	0.00131289321292
$Q = 5$, $Q_V = 6$, $Q_I = 5$	0.00114767373867	0.00141416909853	0.00141416909853
$Q = 5$, $Q_V = 7$, $Q_I = 5$	0.00114767605890	0.00138729451635	0.00138729451635
$Q = 5$, $Q_V = 8$, $Q_I = 5$	0.00114767606311	0.00138724135413	0.00138724135413
$Q = 5$, $Q_V = 9$, $Q_I = 5$	0.00114767606311	0.00138724135413	0.00138724135413
$Q = 5$, $Q_V = 5$, $Q_I = 6$	–	0.00134093637296	0.00134093637296
$Q = 5$, $Q_V = 5$, $Q_I = 7$	–	0.00135650237293	0.00135650237293
$Q = 5$, $Q_V = 5$, $Q_I = 8$	–	0.00135657503799	0.00135657503799
$Q = 5$, $Q_V = 5$, $Q_I = 9$	–	0.00135657503799	0.00135657503799
$Q = 5$, $Q_V = 6$, $Q_I = 6$	–	0.00130947841396	0.00130947841396
$Q = 5$, $Q_V = 7$, $Q_I = 7$	–	0.00130988788959	0.00130988788959
$Q = 5$, $Q_V = 8$, $Q_I = 8$	–	0.00130988726422	0.00130988726422
$Q = 5$, $Q_V = 9$, $Q_I = 9$	–	0.00130988726422	0.00130988726422
	$P_{adv} - \text{EMM}$		
	CG	DG	FR-DG
$Q = 6$, $Q_V = 6$, $Q_I = 6$	0.00114764124297	0.00126591291016	0.00126591291016
$Q = 6$, $Q_V = 7$, $Q_I = 6$	0.00114764336446	0.00126488867950	0.00126488867950
$Q = 6$, $Q_V = 8$, $Q_I = 6$	0.00114764336711	0.00126488766695	0.00126488766695
$Q = 6$, $Q_V = 9$, $Q_I = 6$	0.00114764336711	0.00126488766695	0.00126488766695
$Q = 6$, $Q_V = 6$, $Q_I = 6$	–	0.00126591291016	0.00126591291016
$Q = 6$, $Q_V = 6$, $Q_I = 7$	–	0.00127762517235	0.00127762517235
$Q = 6$, $Q_V = 6$, $Q_I = 8$	–	0.00127767598996	0.00127767598996
$Q = 6$, $Q_V = 6$, $Q_I = 9$	–	0.00127767598996	0.00127767598996
$Q = 6$, $Q_V = 6$, $Q_I = 6$	–	0.00126591291016	0.00126591291016
$Q = 6$, $Q_V = 7$, $Q_I = 7$	–	0.00126617520356	0.00126617520356
$Q = 6$, $Q_V = 8$, $Q_I = 8$	–	0.00126617189683	0.00126617189683
$Q = 6$, $Q_V = 9$, $Q_I = 9$	–	0.00126617189683	0.00126617189683

B.2. Link between geometrical- and PDE-aliasing

Table 10
L_2 norm of the difference between mass matrices calculated with Q and $Q+1$ GLL points for the four orders considered and using both the mesh in Fig. 8.

	Case A: $[\mathbf{M}(J)_{Q+1} - \mathbf{M}(J)_Q]_{L_2}$			
	$J \in \mathcal{P}^1$	$J \in \mathcal{P}^2$	$J \in \mathcal{P}^3$	$J \in \mathcal{P}^4$
$Q_{\min} - Q_{\min}+1$,	0.7244	0.3144	0.1773	0.1176
Global = 4	0.0160			
***Global* = 5**	**4.4222e–16**			
Global = 6		0.0162		
Global = 7		0.0055		
***Global* = 8**		**3.6038e–15**	0.0135	
Global = 9			0.0079	
Global = 10			0.0027	0.0105
***Global* = 11**			**1.2322e–12**	0.0080
Global = 12				0.0047
Global = 13				0.0016
***Global* = 14**				**3.4482e–15**

	Case B: $[\mathbf{M}(J)_{Q+1} - \mathbf{M}(J)_Q]_{L_2}$			
	$J \in \mathcal{P}^2$	$J \in \mathcal{P}^4$	$J \in \mathcal{P}^6$	$J \in \mathcal{P}^8$
$Q_{\min} - Q_{\min}+1$,	0.8368	0.3472	–	–
Global = 4	0.0210			
***Global* = 5**	**1.1943e–15**			
Global = 6		0.0351		
Global = 7		0.0096		
***Global* = 8**		**4.2100e–15**		

Appendix C. Dealiasing within triangular and tetrahedral regions

Throughout this paper, most of the applications demonstrate how the local and global dealiasing techniques can be applied in the setting of quadrilateral and hexahedral elements. In this appendix, we show that the same arguments are also applicable in the case of tensor-product triangular elements. An extension of this argument to prismatic and tetrahedral elements follows easily.

We consider a tensor product of two one-dimensional hierarchical *hp* expansion basis functions ψ^a and ψ^b, so that in the standard triangle $\Omega_{\mathrm{st}} = \{(\xi_1, \xi_2) \mid \xi_1 \in [-1,1], \xi_1 + \xi_2 \leq 0\}$, an approximate solution u^δ may be written as an expansion

$$u^\delta(\xi_1, \xi_2) = \sum_{p=0}^{P} \sum_{q=0}^{P-p} \hat{u}_{pq} \psi_p^a(\eta_1) \psi_{pq}^b(\eta_2),$$

where the collapsed coordinates $(\eta_1, \eta_2) \in [-1,1]^2$ are given by the Duffy transformation

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

The use of this collapsed coordinate system within the standard quadrilateral region $[-1,1]^2$ leads to a tensor product of quadrature points within the standard triangular region, as shown in Fig. 18.

Although this transformation is singular at the top vertex of the triangle, the use of Gauss–Radau quadrature in the η_2 direction, in which the top vertex is excluded, leads to a formulation without geometric singularities. Additionally, Gauss–Radau points with a choice of $\alpha = 1$ and $\beta = 0$ naturally incorporate the Jacobian term which appears in integrals over the standard triangular region when weighted by a constant factor of $\frac{1}{2}$ [24].

In this setting, the sum-factorised dealiasing methods described in Section 3 can be utilised for triangular elements by applying the stated tensor product logic to the grid of (η_1, η_2) quadrature points, without applying the Duffy transformation

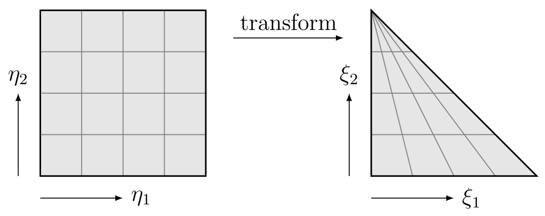


Fig. 18. Illustration of the Duffy transformation between collapsed coordinates $(\eta_1, \eta_2) \in [-1,1]^2$ and Cartesian coordinates $(\xi_1, \xi_2) \in \Omega_{\mathrm{st}}$.

to obtain the desired dealiasing effect in the simplex region. Mathematically, this approach projects the nonlinear terms down to the tensor product space spanned by the (η_1, η_2) space which is typically richer than the space spanned by the triangular expansion in the (ξ_1, ξ_2) space. However, since the inner product using the collapsed coordinates spans the (η_1, η_2) space, this is sufficient to ensure that the nonlinear product is correctly integrated. This logic extends to three dimensions for both prismatic elements and tetrahedra when coupled with suitable extensions of the Duffy transformation to a hexahedral region.

References

[1] G.A. Blaisdell, E.T. Spyropoulos, J.H. Qin, The effect of the formulation of nonlinear terms on aliasing errors in spectral methods, Appl. Numer. Math. 21 (3) (1996) 207–219.

[2] G.J. Gassner, A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods, J. Sci. Comput. 35 (2013) 1233–1253.

[3] G.J. Gassner, A kinetic energy preserving nodal discontinuous Galerkin spectral element method, Int. J. Numer. Methods Fluids 76 (1) (2014) 28–50.

[4] G.-S. Karamanos, G.E. Karniadakis, A spectral vanishing viscosity method for large-eddy simulations, J. Comput. Phys. 163 (1) (2000) 22–50.

[5] R.M. Kirby, G.E. Karniadakis, Coarse resolution turbulence simulations with spectral vanishing viscosity—large-eddy simulations (svv-les), J. Fluids Eng. 124 (4) (2002) 886–891.

[6] R. Pasquetti, Spectral vanishing viscosity method for large-eddy simulation of turbulent flows, J. Sci. Comput. 27 (1–3) (2006) 365–375.

[7] R.M. Kirby, S.J. Sherwin, Stabilisation of spectral h/p element methods through spectral vanishing viscosity: application to fluid mechanics modelling, Comput. Methods Appl. Mech. Eng. 195 (2006) 1667–1691.

[8] C. Xu, Stabilization methods for spectral element computations of incompressible flows, J. Sci. Comput. 27 (1–3) (2006) 495–505.

[9] D. Gottlieb, J.S. Hesthaven, Spectral methods for hyperbolic problems, J. Comput. Appl. Math. 128 (1–2) (2001) 83–131.

[10] P. Fischer, J. Mullen, Filter-based stabilization of spectral element methods, C. R. Acad. Sci., Ser. 1 Math. 332 (3) (2001) 265–270.

[11] P.F. Fischer, F. Loth, G.W. Kruse, Spectral element methods for transitional flows, J. Sci. Comput. 17 (2002) 81–98.

[12] J.S. Hesthaven, R.M. Kirby, Filtering in Legendre spectral methods, Math. Comput. 77 (2008) 1425–1452.

[13] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—a paradigm for computational mechanics, Comput. Methods Appl. Mech. Eng. 166 (1) (1998) 3–24.

[14] C.E. Wasberg, T. Gjesdal, P.B.A. Reif, O. Andreassen, Variational multiscale turbulence modelling in a high order spectral element method, J. Comput. Phys. 228 (19) (2009) 7333–7356.

[15] C. Farhat, A. Rajasekharan, B. Koobus, A dynamic variational multiscale method for large eddy simulations on unstructured meshes, Comput. Methods Appl. Mech. Eng. 195 (2006) 1667–1691.

[16] S. Marras, J.F. Kelly, F.X. Giraldo, M. Vazquez, Variational multiscale stabilization of high-order spectral elements forthe advection–diffusion equation, J. Comput. Phys. 231 (21) (2012) 7187–7213.

[17] G.J. Gassner, A.D. Beck, On the accuracy of high-order discretizations for underresolved turbulence simulations, Theor. Comput. Fluid Dyn. 76:28–50 (2013).

[18] T. Warburton, A low storage curvilinear discontinuous Galerkin method for wave problems, SIAM J. Sci. Comput. 35 (4) (2013) 1987–2012.

[19] R.M. Kirby, G.E. Karniadakis, De-aliasing on non-uniform grids: algorithms and applications, J. Comput. Phys. 191 (2003) 249–263.

[20] M.O. Deville, E.H. Mund, P.F. Fischer, High Order Methods for Incompressible Fluid Flow, Cambridge University Press, 2002.

[21] Y. Maday, E.M. Rønquist, Optimal error analysis of spectral methods with emphasis on non-constant coefficients and deformed geometries, Comput. Methods Appl. Mech. Eng. 80 (1–3) (1990) 80–115.

[22] D. Kopriva, Metric identities and the discontinuous Galerkin spectral element method on curvilinear meshes, J. Sci. Comput. 26 (3) (2006) 301–327.

[23] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, in: 18th AIAA CFD Conference, Miami, FL, June 2007, pp. 25–28.

[24] G.E. Karniadakis, S.J. Sherwin, Spectral/*hp* Element Methods for Computational Fluid Dynamics, Oxford University Press, 2005.

[25] S.A. Orszag, Comparison of pseudospectral and spectral approximations, Stud. Appl. Math. 51 (1972) 253–259.

[26] J. Malm, P. Schlatter, P.F. Fischer, D.S. Henningson, Stabilization of the spectral element method in convection dominated flows by recovery of skew-symmetry, J. Sci. Comput. 77 (2013) 254–277.

[27] P.E. Vincent, P. Castonguay, A. Jameson, A new class of high-order energy stable flux reconstruction schemes, J. Sci. Comput. 47 (1) (2011) 50–72.

[28] 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, Int. J. Numer. Methods Fluids 75 (2014) 860–877.

[29] P.O. Persson, A sparse and high-order accurate line-based discontinuous Galerkin method for unstructured meshes, J. Comput. Phys. 233 (2013) 414–429.

[30] P. Castonguay, P.E. Vincent, A. Jameson, A new class of high-order energy stable flux reconstruction schemes for triangular elements, J. Sci. Comput. 51 (1) (2011) 224–256.

[31] C.D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, et al., Nektar++: an open-source spectral/*hp* element framework, Comput. Phys. Commun. 192 (2015) 205–219.

[32] Y. Maday, E. Tadmor, Analysis of the spectral vanishing viscosity method for periodic conservation laws, SIAM J. Numer. Anal. 26 (4) (1989) 854–870.

[33] R.J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (2) (1996) 627–665.

[34] J.S. Chow, G.C. Zilliac, P.B. Bradshaw, Mean and turbulence measurements in the near field of a wingtip vortex, AIAA J. 35 (1997) 1561–1567.

[35] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 97 (1991) 414–443.

[36] D.A. Kopriva, Implementing Spectral Methods for Partial Differential Equations – Algorithms for Scientists and Engineers, Springer, 2009.

[37] G.J. Gassner, D. Kopriva, A comparison of the dispersion and dissipation errors of Gauss and Gauss–Lobatto discontinuous Galerkin spectral element methods, SIAM J. Sci. Comput. 33 (5) (2011) 2560–2579.