

Towards p -adaptive spectral/ hp element methods for modelling industrial flows

D. Moxey, C. D. Cantwell, G. Mengaldo, D. Serson, D. Ekelschot, J. Peiró, S. J. Sherwin and R. M. Kirby

Abstract There is an increasing requirement from both academia and industry for high-fidelity flow simulations that are able to accurately capture complicated and transient flow dynamics in complex geometries. Coupled with the growing availability of high-performance, highly parallel computing resources, there is therefore a demand for scalable numerical methods and corresponding software frameworks which can deliver the next-generation of complex and detailed fluid simulations to scientists and engineers in an efficient way. In this article we discuss recent and upcoming advances in the use of the *spectral/ hp element method* for addressing these modelling challenges. To use these methods efficiently for such applications, is critical that computational resolution is placed in the regions of the flow where it is needed most, which is often not known *a priori*. We propose the use of spatially and temporally varying polynomial order, coupled with appropriate error estimators, as key requirements in permitting these methods to achieve computationally efficient high-fidelity solutions to complex flow problems in the fluid dynamics community.

1 Introduction

Computational modelling is now regularly used in the fluid dynamics community, giving insight into flow problems where experimentation is too difficult, impracti-

David Moxey

College of Engineering, Mathematics and Physical Sciences, University of Exeter, e-mail: d.moxey@exeter.ac.uk

Chris Cantwell, Douglas Serson, Dirk Ekelschot, Joaquim Peiró, Spencer Sherwin

Department of Aeronautics, Imperial College London e-mail: c.cantwell, d.serson14, d.ekelschot, j.peiro, s.sherwin@imperial.ac.uk

Gianmarco Mengaldo

Division of Engineering and Applied Sciences, California Institute of Technology, e-mail: mengaldo@caltech.edu

Robert M. Kirby

Scientific Computing and Imaging Institute, University of Utah, e-mail: kirby@cs.utah.edu

cal or costly to realise. The complex geometries and time constraints involved in modern industrial studies imply that, to date, most numerical simulations are restricted to being steady in time. This limits their capabilities, particularly when the problem of interest involves fundamentally unsteady flow dynamics, such as vortex shedding. However, with the wider availability and reducing cost of large-scale computing power, academic and industrial fluid dynamicists are increasingly looking to perform finely-detailed unsteady simulations. These high-fidelity simulations will allow us to obtain deeper insight into many challenging engineering problems, where steady-state solvers struggle to capture the relevant unsteady flow structures.

One of the main challenges in conducting such simulations is that the complex geometries that are a natural consequence of studying industrial problems will inherently generate flow structures across a large range of time and length scales. From a practical perspective, it becomes difficult or impossible to predict where numerical resolution is required in the computational domain before the simulation is run in order to accurately resolve the flow. Since uniform refinement across very large domains is computationally prohibitive, the community is turning to adaptive methods, where resolution is dynamically adjusted within the domain as a function of time, in order to overcome this issue.

The spectral/ hp element method [11] – in which an unstructured elemental decomposition capable of resolving complex geometries is equipped with high-order polynomial bases are used to give routes to convergence in terms of h and p – has been used in academic applications for several years. However, it is now emerging as one of the enabling technologies for the simulation of high-fidelity industrial simulations. From a numerical perspective, these methods offer attractive properties such as low diffusion and dispersion errors, meaning that for smooth solutions fewer degrees of freedom are required to retain the same accuracy as compared to traditional low-order methods [19]. From a computational perspective, the use of a higher polynomial order leads to compact data structures and enables a balance between the computational and memory intensiveness of the method. This is increasingly becoming a key factor in the efficient use of modern many-core hardware.

On the whole, the development of adaptive methods has been mostly focused around h -adaption, where the elements are refined or coarsened in order to adjust the numerical resolution. The use of p -adaption, on the other hand, has received far less attention. Most of the work in this area has focused on hp -adaption, which has been an area of significant attention with various works investigating these techniques for elliptic problems [24, 4, 8] that are not necessarily immediately applicable for fluid-based problems. However, p -adaption has been shown to be a viable technique in a study by Li and Jameson [14], where adaption in p was shown to provide the highest accuracy with respect to the numerical resolution and computing time.

However high-order methods have presented inherent difficulties that have only started to be overcome in the last few years. These challenges are both mathematical and practical. On the theoretical side, there has been a need to overcome stability issues arising due to aliasing of the solution [17] and timestep size [6]; investigate the generation of curved meshes which conform to the boundary of complex three-dimensional domains [22, 21]; and investigate parallel scaling of these meth-

ods [27]. On the practical side, the mathematical complexity of the methods has necessitated the development of software frameworks [2] to improve accessibility to academia and industry. These developments now mean that these high-order methods are being applied in very high Reynolds number flows that are of significant interest to, for example, the aerodynamics and aeronautics community [15].

In this article, we will discuss some practicalities of implementing spectral/ hp element solvers which use a spatially variable polynomial order across computational domain. We do this both in the context of incompressible and compressible flow. For the former we use a continuous Galerkin approach to solving a semi-implicit form of the incompressible Navier-Stokes equations; for the latter we use a discontinuous Galerkin projection with an explicit time-stepping method. Section 2 discusses the formulation of these methods and how variable polynomial orders are handled in each case. Section 3 illustrates the capabilities of adaptivity in p , before concluding with a brief outlook in Section 4.

2 Formulation

This section begins with a brief discussion of the formulation of the spectral/ hp element method, the basis being used to represent elemental expansions and how this relates to discontinuous and continuous formulations. We then describe how this formulation can be adapted to allow variable polynomial order across the computational domain, provide implementation details and give an overview of the techniques required to make this approach computationally tractable.

2.1 Domain discretisation

The domain Ω is subdivided into N_{el} non-overlapping elements Ω^e , such that $\Omega = \bigcup_{e=1}^{N_{el}} \Omega^e$. In two dimensions, these elements are a mixture of quadrilaterals and triangles; in three dimensions, a mixture of hexahedra, triangular prisms, square-based pyramids and tetrahedra are considered. We define a standard element Ω_{st} for each shape. For example, a quadrilateral is defined by $\Omega_{st} = \{(\xi_1, \xi_2) | \xi_1, \xi_2 \in [-1, 1]\}$. We equip each standard region with a set of polynomial basis functions ϕ_n with which to approximate functions. A scalar function u on an element Ω^e is represented by an expansion

$$u(\mathbf{x}) = \sum_{n=1}^{M(e,P)} \hat{u}_n^e \phi_n(\xi), \quad (1)$$

where points $\xi \in \Omega_{st}$, $\mathbf{x} \in \Omega^e$, and the two are related through an invertible mapping $\chi^e : \Omega_{st} \rightarrow \Omega^e$ such that $\mathbf{x} = \chi^e(\xi)$. The upper bound of the summation, $M(e,P)$, defines the number of modes that represent the solution in the element Ω^e and is a function of both the polynomial order and the element type. We let $\mathbb{P}_k(\Omega^e)$ denote

the polynomial space spanned by the $M(e, P)$ basis functions, with k the maximum polynomial order, on the e th elemental region.

In order to represent a function across the entire domain Ω , we must select an appropriate function space to represent our approximation. In this work we will consider two classic discretisations: the continuous (CG) and discontinuous Galerkin (DG) methods, which require the spaces

$$D^{\text{CG}}(\Omega) = \{v \in C^0(\Omega) \mid v|_{\Omega^e} \in \mathbb{P}_k(\Omega^e)\}, \quad (2)$$

$$D^{\text{DG}}(\Omega) = \{v \in L^2(\Omega) \mid v|_{\Omega^e} \in \mathbb{P}_k(\Omega^e)\} \quad (3)$$

with C^0 and L^2 being the usual spaces of continuous and square-integrable functions respectively and k initially considered constant. We note that in the context of discontinuous spectral element methods, significant effort has recently been spent in the development of high-order flux reconstruction schemes [10, 25]. While they are in principle different, these schemes can be cast within the same framework as the discontinuous Galerkin method [9, 18]. Therefore, the adaption technique described hereafter can be directly extended to the flux reconstruction method.

2.1.1 Choice of basis

The choice of the basis ϕ is particularly important when variable polynomial order across elements is required. We opt to use a set of functions that augment the usual linear finite element modes with higher-order polynomials, defined as

$$\psi_p(\xi) = \begin{cases} \frac{1-\xi}{2}, & p = 0, \\ \frac{1+\xi}{2}, & p = 1, \\ \frac{1-\xi}{2} \frac{1+\xi}{2} P_{p-2}^{(1,1)}(\xi), & p \geq 2, \end{cases} \quad (4)$$

where $P_p^{(\alpha,\beta)}(\xi)$ is the p -th order Jacobi polynomial with coefficients α and β . In one dimension on the segment $[-1, 1]$, we have that $\phi_n = \psi_n$ in (1). In higher dimensions, quadrilaterals and hexahedral expansion bases are defined using a tensor product of these one-dimensional functions. Other element types use a similar choice of basis that still permits a tensorial expansion (for more details, see [11]).

There are several advantages to this choice of basis in the context of a mesh of variable polynomial order. The first is that it results in a topological decomposition of the basis, so that the modes of an element can be classified into vertex, edge-interior, face-interior and volume-interior modes. Only vertex, edge and face modes have support which extends to the boundary of the element; interior modes are zero on the boundary. This is depicted for an order 4 quadrilateral in Fig. 1, where black circles represent the boundary modes and grey the interior. When we discuss the modification of any contributions along an edge of the element, this only therefore requires the modification of coefficients along that edge, as opposed to across the entire element. Additionally, this set of modes is *hierarchical*; that is, the degree of

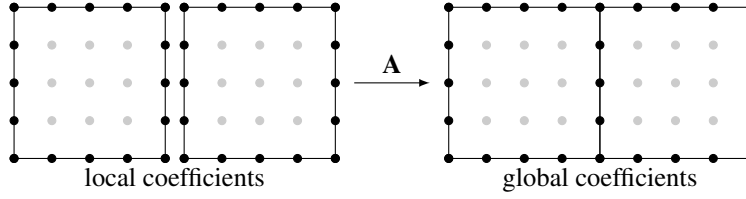


Fig. 1: Diagram describing assembly operation between two \mathbb{P}_4 quadrilaterals.

each basis polynomial $\phi_p(\xi)$ increases as a function of p . This is in contrast to, for example, a classical spectral element method in which Lagrange interpolants define a nodal basis depending on a choice of nodes ξ_j . At order P these are defined as

$$\phi_p(\xi) = \ell_p(\xi) = \prod_{q \neq p}^{q=P} \frac{\xi - \xi_q}{\xi_p - \xi_q}$$

so that every basis function is of the same polynomial order P , whilst still yielding a boundary-interior decomposition.

2.2 Implementation details

2.2.1 Continuous Galerkin formulation

The key operation of the CG formulation is assembly, wherein local elemental contributions are gathered to impose the C^0 -continuity of the underlying function space, as depicted visually in Fig. 1. The assembly operation associates a vector of concatenated local elemental coefficients $\hat{\mathbf{u}}_l = (\hat{\mathbf{u}}^1, \dots, \hat{\mathbf{u}}^{N_{el}})$ to their global counterparts $\hat{\mathbf{u}}_g$ through an injective map. Here, we note that each $\hat{\mathbf{u}}^e$ corresponds to the vector of local coefficients in Eq. 1. The coefficients in $\hat{\mathbf{u}}_g$ describe the contribution to the solution of the modes which span $D^{CG}(\Omega)$. Mathematically, this operation is expressed through a sparse matrix-vector operation $\hat{\mathbf{u}}_g = \mathbf{A}\hat{\mathbf{u}}_l$. For a uniform polynomial order mesh, the columns of \mathbf{A} are non-zero where local degrees of freedom meet to form global degrees of freedom, and zero otherwise, so that the valency of a global degree of freedom i is defined as the number of non-zero columns in the i -th row of \mathbf{A} . In practice, the high sparsity of \mathbf{A} means that we use array indirection to implement the action of \mathbf{A} without explicitly constructing it, as defined in Algorithm 1.

We note that two arrays are required:

- `map[e][i]` stores the index of the global degree of freedom corresponding to mode i of element Ω^e ;
- `sign[e][i]` stores either 1 or -1 to align modes that are of odd polynomial orders such that the basis remains continuous (see [11] for more details).

Algorithm 1 Continuous C^0 assembly operation

```

for  $e = 1 \rightarrow N_{\text{el}}$  do
  for  $i = 1 \rightarrow M(e)$  do
     $\hat{\mathbf{u}}_g[\text{map}[e][i]] += \text{sign}[e][i] \hat{\mathbf{u}}_i^e[i]$ 
  end for
end for

```

Throughout the rest of this section we will consider a Helmholtz problem

$$\nabla^2 u + \lambda u = f \quad (5)$$

which is later used for incompressible simulations through the use of an operator splitting scheme [13]. This is put into a weak form by defining appropriate finite-dimensional test and trial spaces, multiplying each term by a test function and integrating over the domain. After applying integration by parts we obtain the equation

$$(\mathbf{L} + \lambda \mathbf{M}) \hat{\mathbf{u}}_g = \hat{\mathbf{f}}$$

where \mathbf{L} and \mathbf{M} are the global Laplacian and mass matrices, respectively, and $\hat{\mathbf{f}}$ is the Galerkin projection of f onto $D^{\text{CG}}(\Omega)$. The assembly map is used not only to calculate $\hat{\mathbf{f}}$, but also to construct the matrices \mathbf{L} and \mathbf{M} from their constituent elemental matrices, through the relationship

$$\mathbf{L} = \mathbf{A} \left[\bigoplus_{e=1}^{N_{\text{el}}} \mathbf{L}^e \right] \mathbf{A}^\top. \quad (6)$$

We note that in practice, even at moderately low polynomial orders, $\mathbf{L} + \lambda \mathbf{M}$ is rarely explicitly constructed. The use of the mapping above allows us to apply the action of this operator and leverage the computational optimisations possible due to the rich structure of the elemental matrices.

To modify this procedure for spatially varying polynomial orders, we must address the situation depicted in Fig. 2, where two elements meet that differ in polynomial orders; in this case a \mathbb{P}_3 and \mathbb{P}_7 quadrilateral. In the global space, the edge connecting these elements (depicted in the middle of the figure) should be at most an order 3 polynomial and so some additional logic is required to discard the higher degrees of freedom contributed by the \mathbb{P}_6 quadrilateral in the assembly process. To this end, we note that since we are using a hierarchical basis, Algorithm 1 can remain unchanged by altering the sign and mapping arrays to easily filter out the higher-order contributions. We impose that on the common edge, the coefficients of the sign array on the \mathbb{P}_6 element are set to zero for the highlighted modes corresponding to a polynomial degrees between 4 and 6. This ensures that in the assembly operation, no contribution from these high-frequency modes is included. The corresponding coefficients in the mapping array are set to point to one of the known vertex coefficients to avoid memory overflow errors. We note that if the basis were not to be of a hierarchical construction, then in general, all of the modes along an

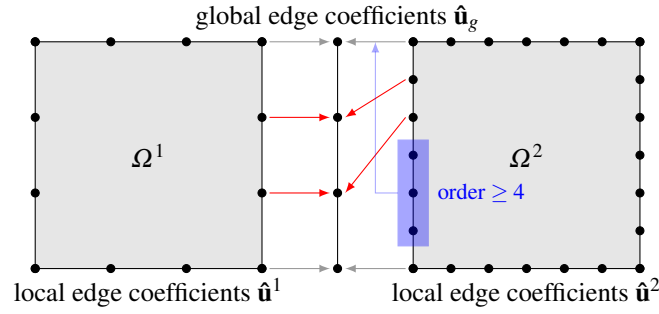


Fig. 2: Diagram describing assembly operation between a \mathbb{P}_3 and \mathbb{P}_6 quadrilateral. The nodes here correspond to vertex and edge modes of the hierarchical basis. Red arrows indicate the usual connectivity; blue arrows indicate modes that are zeroed using the sign array.

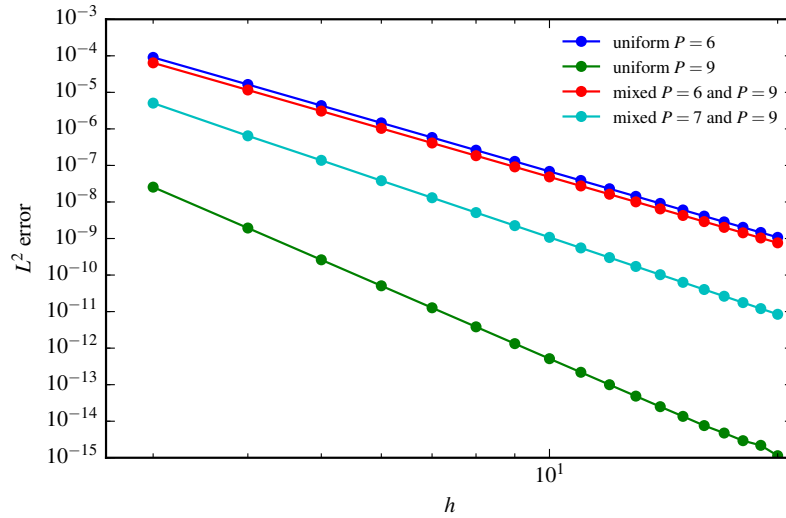


Fig. 3: Convergence of Helmholtz problem for simple square case.

edge can be of equal polynomial order. In this case, the above procedure needs to be modified to perform a polynomial interpolation onto the correct space, rather than simply zeroing elements of the sign array.

As a test of the validity of this approach, we consider the Helmholtz problem in the a square $[-1, 1]^2$, in which \mathbf{f} is defined to obtain a prescribed solution $u(x, y) = \sin(\pi x) \sin(\pi y)$. We consider a series of meshes with h elements in each direction. We then solve Eq. (5) using the continuous Galerkin formulation for four cases.

uniform polynomial orders of $P = 6$ and $P = 9$, and then a mixed order where half of the elements are set to $P = 6$, and half to $P = 9$. Fig. 3 shows the L^2 error of these simulations, where we clearly observe the same convergence rate for all simulations, and the mixed order case has a slightly lower error than the $P = 6$ case as expected. Increasing the mixed order case to $P = 7$ lowers the error so that it lies between the two uniform simulations.

2.2.2 Discontinuous Galerkin formulation

We now briefly discuss the implementation of variable polynomial order in the discontinuous Galerkin (DG) formulation, which is described in greater detail in [5]. The use of DG is widely increasing in modern fluid dynamics codes and is especially popular for discretising hyperbolic or mixed hyperbolic-parabolic systems, such as the compressible Euler and Navier-Stokes equations, which form the cornerstone of modern aerodynamics problems. To illustrate the problem we consider a simple scalar conservation law

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

Using the variational form of the problem together with the function space $D^{DG}(\Omega)$ defined in Eq. (3) leads to the discontinuous Galerkin method, wherein we consider for each element the ODE system

$$\frac{d}{dt} \int_{\Omega^e} u \phi \, dx + \int_{\partial\Omega^e} \phi \mathbf{F}(u) \cdot \mathbf{n} \, ds = \int_{\Omega^e} \mathbf{F}(u) \cdot \nabla \phi \, dx$$

where ϕ is a test function lying in $\mathbb{P}_k(\Omega^e)$ and \mathbf{n} denotes the normal vector to the element boundary $\partial\Omega^e$. We also assume that these ODEs are discretised explicitly in time, so that at each timestep we must calculate the volume term on the right hand side, calculate the flux term on the left hand side, and then incorporate the flux term into the volume term. The remaining first term on the left hand side, in an explicit timestepping setting, corresponds to the action of the elemental mass matrix. The only place in which we need to consider the application of variable polynomial order is therefore the second part of this process.

We again consider the problem of two quadrilaterals of different orders in Fig. 4. We first note that since the discretisation in time considers all elemental degrees of freedom, we must consider the boundary terms at the higher polynomial order to avoid stability issues, otherwise there are degrees of freedom within the higher-order element that become undetermined. Additionally, we wish to preserve the locally conservative nature of the DG method, implying that in the notation of the figure, we require

$$\int_{\Gamma_1} \mathbf{F}(u) \, ds = \int_{\Gamma_2} \mathbf{F}(u) \, ds$$

where Γ_1 and Γ_2 are the edges of the two elements that intersect to make the trace element Γ .

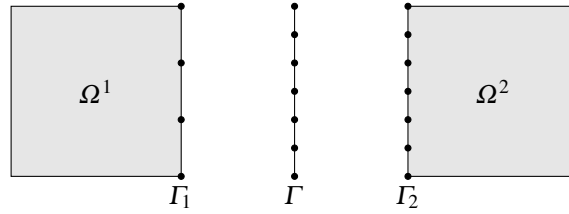


Fig. 4: Diagram describing treatment of variable polynomial order in DG for quadrilateral elements.

To project the trace contributions back into the volume consistently, on the higher side we may simply copy the coefficients directly from Γ to Γ_2 . For Γ_1 , we have a higher-degree polynomial that must be incorporated into a lower degree edge. To do this in a conservative fashion, we perform a change of basis of the elemental coefficients from the hierarchical basis of Equ. (4) onto an orthogonal space of Legendre polynomials. We then apply a low-pass filter, by zeroing the unwanted high-frequency polynomials. This is necessary since the basis functions given in Equ. (4) are not orthogonal and performing a filtering in this space will alter the mean flux, leading to a loss of conservation. Finally, we perform a change of basis back to the lower-order hierarchical basis.

2.3 Efficiency across a range of polynomial orders

As a final note on implementation considerations, a clear observation that can be made when using a variable polynomial order is that the sizes of elemental matrices can vary drastically, particularly when considering three-dimensional elements. This is important since operator evaluations, such as the Laplacian matrix of Eq. (6), form the bulk of the computational cost of the spectral/ hp element method, either in computing quantities such as the inner product or in solving a system of equations in an iterative fashion. Efficient evaluation of these operators across a wide range of polynomial orders is therefore an important component to the efficacy of a variable polynomial order simulation.

The underlying mathematical formulation and tensor-product form of the basis admits a number of different implementation choices for the evaluation of these operators, each of which admits differing performance across polynomial orders and choice of hardware [1, 3, 16, 26]. Furthermore, the results of [20] suggest that for modern hardware, where memory bandwidth is a valuable commodity, elemental operations should be amalgamated wherever possible to minimise data transfer and efficiently utilise the memory hierarchy. In the context of variable polynomial orders, the amalgamation of elements that are of the same type and polynomial order, combined with an appropriate implementation strategy as described in [20], should be performed to maximise the computational performance of the method.

3 Results

This section gives a brief overview of results achieved to date using adaption in the polynomial order p with the compressible and incompressible formulations, focusing on error indicators and how this affects the ability to capture the underlying flow physics, and on the computational cost of these approaches.

3.1 Incompressible flow

In this section, we present an example of a simulation employing adaptive polynomial order for solving the incompressible Navier-Stokes equations, which can be represented as

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla p + \nu \nabla^2 \mathbf{u}, \quad \nabla \cdot \mathbf{u} = 0 \quad (7)$$

where \mathbf{u} is the velocity, p is the pressure, and ν is the kinematic viscosity and, without loss of generality, we set the density to be unity. Given a reference length L and a reference velocity U , the Reynolds number is defined as $Re = \frac{LU}{\nu}$. We solve these equations using a CG-approach and a semi-implicit velocity-correction scheme [13], whereby (7) is separated into an explicit convective term, an implicit Poisson equation for pressure and three further implicit Helmholtz equations for the velocity components.

In the procedure we employed, the polynomial order is adjusted during the solution based on an estimate of the discretisation error in each element. This estimate for the error (sometimes called *sensor*) was based on the one used for shock capture in [23]. In the present work, this is defined as

$$S_e = \frac{\|u_P - u_{P-1}\|_{2,e}^2}{\|u_P\|_{2,e}^2}, \quad (8)$$

where u_P is the solution obtained for the u velocity using the current polynomial order P , u_{P-1} is the projection of this solution to a polynomial of order $P-1$, $\|\cdot\|_2$ is the L_2 norm and the subscript e indicates that this refers to a single element.

Considering this estimate for the discretisation error, the adaptive procedure can be summarized as:

1. Advance the equations for n_{steps} time steps.
2. Calculate S_e for each element.
3. Modify the polynomial order in each element:
 - if $S_e \geq \epsilon_u$ and $P < P_{\text{max}}$, increase P by 1;
 - if $S_e \leq \epsilon_l$ and $P > P_{\text{min}}$, decrease P by 1;
 - maintain same P if none of the above is true.
4. Project the solution to the new polynomial space.
5. Repeat for n_{runs} .

where ε_u is the tolerance above which the polynomial order is increased, $\varepsilon_l \leq \varepsilon_u$ is the tolerance below which the polynomial order is decreased and P_{\min} and P_{\max} are the minimum and maximum polynomial orders imposed on the procedure.

It is important to note that changing the polynomial order during the solution is costly, due to the need to assemble and decompose the linear systems for the implicit part of the method. Therefore, the choice of n_{steps} plays a key role in obtaining an efficient solution. A lower value of n_{steps} will lead to the refinement step being performed more frequently, at the expense of a higher average computational cost per timestep.

To illustrate this method, we consider quasi-3D simulations of the incompressible flow around a NACA0012 profile, shown in Fig. 5b, with Reynolds number $Re = 50,000$ and angle of attack $\alpha = 15^\circ$. A spectral/hp discretisation is applied in the xy plane, with the span direction discretised by a Fourier series, as proposed in [12]. The adaptive procedure was employed only in the spectral/hp plane, with a fixed number of modes used in the Fourier direction.

Fig. 5 shows the distribution of polynomial order obtained using $n_{\text{steps}} = 4,000$, $P_{\min} = 2$ and $P_{\max} = 9$. It is clear that the boundary layers and the regions of turbulent separated flow are represented by high order polynomials, while lower orders are used in regions of laminar flow far from the wing. In this case, the average number of degrees of freedom per element is approximately 49, which is equivalent to the value for a constant $P = 6$ simulation.

Table 1 compares the cost of this simulation using the adaptive procedure with the cost for several different values of constant polynomial order, and with using the same variable polynomial order distribution without performing the adaptive procedure. We note that for this value of n_{steps} , the refinement procedure corresponds to 5% of the computational cost. This is more than offset by the gains obtained from using a more efficient distribution of degrees of freedom, with the adaptive case presenting roughly the same cost as the $P = 7$ case, and being 35% faster than the $P = 9$ case.

Table 1: Comparison of the computational cost of adaptive order case of figure 5 with constant uniform polynomial order and with variable order without adaptive procedure. The computational costs are normalized with respect to the adaptive order case.

Case	$Cost$	$\frac{1}{Cost}$
$P = 5$	0.60	1.66
$P = 6$	0.72	1.39
$P = 7$	1.08	0.93
$P = 8$	1.19	0.84
$P = 9$	1.53	0.65
variable order (fixed)	0.95	1.05
adaptive order	1.00	1.00

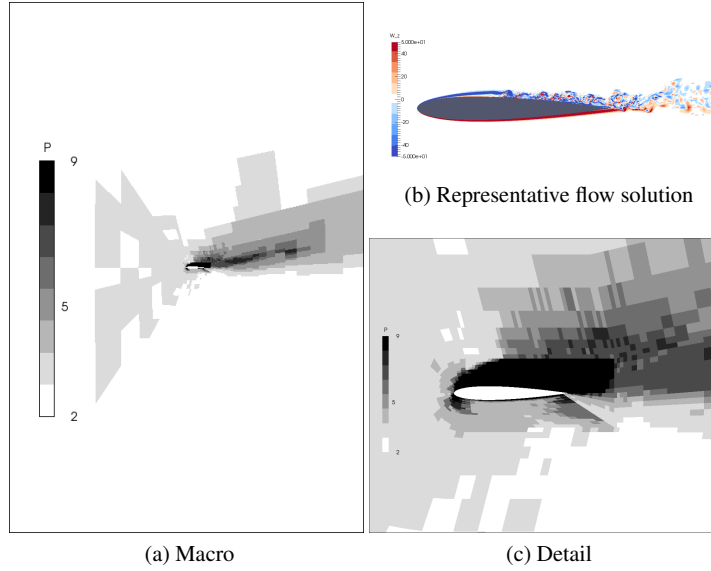


Fig. 5: Polynomial order distribution obtained for incompressible flow around a NACA0012 profile with $Re = 50,000$ and $\alpha = 15^\circ$.

3.2 Compressible flow using explicit timestepping

The accurate solution of compressible flow is an important topic in a number of application areas. For instance, the aeronautical community is concerned with accurately predicting the lift and drag coefficients of different wing configurations whilst keeping the computational cost low. This allows considering a wide range of geometries during the design lifecycle and provides the basis for aerodynamic shape optimization. In these applications, the key to accurately predicting lift and drag lies in determining the regions of the domain which influence these coefficients the most. Adaptive methods, combined with appropriate error estimators, are one route to producing fast, accurate and reliable results. This section describes progress made in [5], where a goal-based error estimator based on an adjoint problem derived from the underlying equations has been applied together with the p -adaptive techniques described in the previous section. The error estimator derives an adjoint problem from a coarsely-resolved base flow, the solution of which determines the areas of the domain which have the greatest sensitivity to the lift and drag coefficients. Although this technique has been explored previously, a review of which can be found in [7], this has mostly focused around h -adaptivity where the element size is refined or coarsened and p -adaptivity at low values of p . The purpose of this work has been to consider a wider range of polynomial orders for this problem.

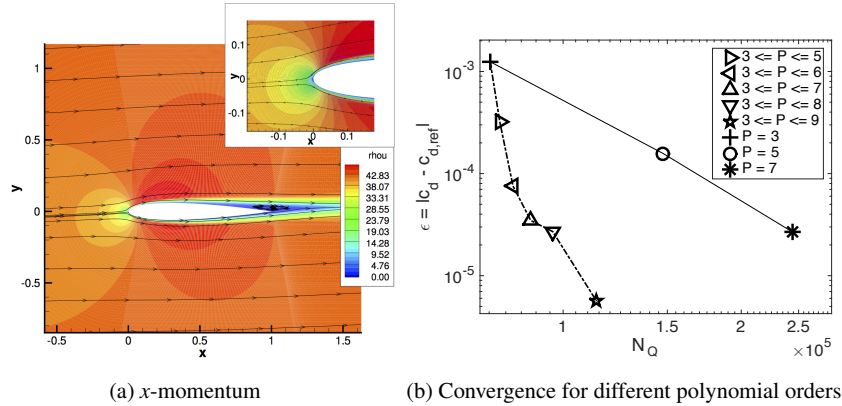


Fig. 6: Variable polynomial order simulations of a compressible laminar NACA0012 wing, taken from [5].

3.2.1 Governing equations

We consider the compressible Navier-Stokes equations written in conservative form

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

where $\mathbf{U} = [\rho, \rho u_1, \rho u_2, \rho u_3, E]^\top$ is the vector of conserved variables, ρ is the density, (u_1, u_2, u_3) the velocity components and E is the specific total energy. $\mathbf{F}(\mathbf{U})$ and $\mathbf{F}_v(\mathbf{U})$ denote the usual inviscid and viscous flux terms respectively, where the ideal gas law is used to close the system. For a more detailed outline, see [5].

3.2.2 Adaptive procedure

Summarising the process at a very high level, the adaptivity procedure runs as follows for this problem:

- Run a low-order simulation to obtain a steady flow field.
- Use this flow field to solve a goal-based adjoint problem by considering an infinitesimal perturbation to the flow field.
- Compute a distribution of the polynomial order according to a goal-based error estimator based on the adjoint solution.
- Using the techniques of Section 2, perform the simulation again to compute a solution with a lower error of the lift or drag.

For an in-depth overview of all of the techniques used in the computation of the adjoint and error estimator, the interested reader should consult [5]. To highlight the

Case	Cost	ϵ
$P = 3$	0.28	1.2×10^{-3}
$P = 5$	0.29	1.57×10^{-4}
$P = 7$	0.64	2.69×10^{-5}
$P = 9$	1.0	–
$3 \leq P \leq 5$	0.31	3.19×10^{-4}
$3 \leq P \leq 6$	0.32	7.44×10^{-5}
$3 \leq P \leq 7$	0.34	3.47×10^{-5}
$3 \leq P \leq 8$	0.36	2.71×10^{-5}
$3 \leq P \leq 9$	0.45	5.63×10^{-6}

Table 2: Summary of normalised CPU cost and error in drag c_d for various constant and spatially variable polynomial orders, compared to a uniform simulation at $P = 9$.

resolution capability of this adaptive method, a series of simulations have been performed to compare the use of variable p with an appropriate error estimator against a uniform refinement in p . We consider the simulation presented in [5], where the laminar subsonic flow over a classical NACA0012 wing geometry is studied at an angle of attack $\alpha = 2^\circ$, Mach number of 0.1 and Reynolds number 5,000. A number of simulations are considered:

- a high resolution case at $P = 9$ is used as a reference solution, the obtained solution for the x -momentum for which can be seen in Fig. 6a;
- uniform polynomial order simulations are performed at $P = 3, 5$ and 7 ;
- variable polynomial orders are performed with $3 \leq P \leq 5 \rightarrow 9$.

To compare these simulations we calculate the error as $\epsilon = \|c_d - c_{d,\text{ref}}\|$ where

$$c_d = \frac{2}{\rho_\infty u_\infty^2 A} \oint_\Gamma \mathbf{u} \cdot [\cos \alpha, \sin \alpha] ds$$

is the drag coefficient, ρ_∞ and u_∞ are the farfield density and velocity, A the frontal area of the wing and $c_{d,\text{ref}}$ denoting the drag coefficient of the reference $P = 9$ case.

The error obtained using these cases can be seen in Fig. 6b, where it is viewed against the number of degrees of freedom N_Q of the resulting mesh, where two distinct trends can be observed. We see that increasing the polynomial order uniformly does reduce the error obtaining in the drag coefficient at a reasonably constant rate. However, the use of the goal-based error estimator, coupled with the use of a variable polynomial order, allows us to greatly reduce the resolution (and therefore the cost) required for these simulations. For example, the simulations at $3 \leq P \leq 8$ and $P = 7$ have very comparable values of ϵ . The main difference is that whereas the uniform case has around 2.5×10^5 degrees of freedom, the variable case needs only 1×10^5 to produce a comparable error, which represents a significant saving in the cost of the simulation. This can be observed in Table 2, where the CPU time for each simulation is reported as a proportion of the reference $P = 9$ case.

4 Conclusions

In this article we have discussed the use and implementation of adaptive polynomial order in the spectral/ hp element method. The canonical flows considered here show the clear benefits of this adaptive process, bringing a reduction in both the computational cost and the number of degrees of freedom required to resolve a given problem. However, there are still a number of challenges that need to be addressed before these methods can be brought to bear on extremely large-scale problems. Numerically, future work should focus around the development of more robust error estimators, particularly in the context of unsteady simulations, perhaps based around an unsteady formulation of the adjoint approach used for compressible simulations in Section 3. We note that this is inherently more expensive than the sub-cell estimator, however it will give a better indication of error throughout the domain. More sophisticated techniques also need to be developed for parallel simulations. In particular, the efficient preconditioning of these systems remains an open problem, and very large-scale simulations require the development of adaptive load-balancing techniques that can be used to re-distribute the workload evenly across processors as the polynomial order changes. Finally, when dealing with complex geometries, techniques need to be developed to couple the change in polynomial order to the treatment of curvilinear surfaces and the elements that connect to them, in order to preserve the accurate representation of the underlying geometry.

Acknowledgements. D.M. acknowledges support from the EU Horizon 2020 project ExaFLOW (grant 671571) and the PRISM project under EPSRC grant EP/L000407/1. D.S. is grateful for the support received from CNPq (grant 231787-/2013-8) and FAPESP (grant 2012/23493-0). D.E. acknowledges support from the EU ITN project ANADE (grant PITN-GA-289428). S.J.S. acknowledges Royal Academy of Engineering support under their research chair scheme. R.M.K. acknowledges support from the US Army Research Office under W911NF1510222 (overseen by Dr. M. Coyle). Computing resources supported by the UK Turbulence Consortium (EPSRC grant EP/L000261/1) and the Imperial College HPC service.

References

1. Cantwell, C., Sherwin, S., Kirby, R., Kelly, P.: From h to p efficiently: Strategy selection for operator evaluation on hexahedral and tetrahedral elements. *Comp. Fluids* **43**(1), 23–28 (2011)
2. Cantwell, C.D., Moxey, D., Comerford, A., Bolis, A., Rocco, G., Mengaldo, G., de Grazia, D., Yakovlev, S., Lombard, J.E., Ekelschot, D., Jordi, B., Xu, H., Mohamied, Y., Eskilsson, C., Nelson, B., Vos, P., Biotto, C., Kirby, R.M., Sherwin, S.J.: Nektar++: An open-source spectral/ hp element framework. *Comp. Phys. Comm.* **192**, 205–219 (2015)
3. Cantwell, C.D., Sherwin, S.J., Kirby, R.M., Kelly, P.H.J.: From h to p efficiently: selecting the optimal spectral/ hp discretisation in three dimensions. *Math. Mod. Nat. Phenom.* **6**, 84–96 (2011)
4. Demkowicz, L., Rachowicz, W., Devloo, P.: A fully automatic hp -adaptivity. *J. Sci. Comp.* **17**(1), 117–142 (2002)

5. Ekelschot, D., Moxey, D., Sherwin, S.J., Peiró, J.: A p -adaptation method for compressible flow problems using a goal-based error estimator. *Comp. Struct.* **181**, 55–69 (2017)
6. Ferrer, E., Moxey, D., Sherwin, S.J., Willden, R.H.J.: Stability of projection methods for incompressible flows using high order pressure-velocity pairs of same degree: Continuous and discontinuous Galerkin formulations. *Comm. Comp. Phys.* **16**(3), 817–840 (2014)
7. Fidkowski, K.J., Darmofal, D.L.: Review of output-based error estimation and mesh adaptation in computational fluid dynamics. *AIAA J.* **49**(4), 673–694 (2011)
8. Giorgiani, G., Fernández-Méndez, S., Huerta, A.: Goal-oriented hp -adaptivity for elliptic problems. *Int. J. Num. Meth. Fluid.* **72**(1), 1244–1262 (2013)
9. de Grazia, D., Mengaldo, G., Moxey, D., Vincent, P., Sherwin, S.J.: Connections between the discontinuous Galerkin method and high-order flux reconstruction schemes. *Int. J. Num. Meth. Fluid.* **75**(12), 860–877 (2014)
10. Huynh, H.T.: A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In: 18th AIAA Computational Fluid Dynamics Conference, p. 4079 (2007)
11. Karniadakis, G., Sherwin, S.: *Spectral/hp Element Methods for Computational Fluid Dynamics*, second edn. Oxford University Press (2005)
12. Karniadakis, G.E.: Spectral element-Fourier methods for incompressible turbulent flows. *Comp. Meth. Appl. Mech. Eng.* **80**(1–3), 367–380 (1990)
13. Karniadakis, G.E., Israeli, M., Orszag, S.A.: High-order splitting methods for the incompressible Navier-Stokes equations. *J. Comp. Phys.* **97**(2), 414–443 (1991)
14. Li, L.Y., Allaneau, Y., Jameson, A.: Comparison of h - and p -adaptations for spectral difference methods. In: 40th AIAA Fluid Dynamics Conference and Exhibit (2010)
15. Lombard, J.E.W., Moxey, D., Sherwin, S.J., Hoessler, J.F.A., Dhandapani, S., Taylor, M.J.: Implicit large-eddy simulation of a wingtip vortex. *AIAA J.* **54**(2), 506–518 (2016)
16. Markall, G., Slemmer, A., Ham, D., Kelly, P., Cantwell, C., Sherwin, S.: Finite element assembly strategies on multi-core and many-core architectures. *Int. J. Num. Meth. Fluid.* **71**(1), 80–97 (2013)
17. Mengaldo, G., de Grazia, D., Moxey, D., Vincent, P.E., Sherwin, S.J.: Dealiasing techniques for high-order spectral element methods on regular and irregular grids. *J. Comp. Phys.* **299**, 56–81 (2015)
18. Mengaldo, G., de Grazia, D., Vincent, P.E., Sherwin, S.J.: On the connections between discontinuous Galerkin and flux reconstruction schemes: extension to curvilinear meshes. *J. Sci. Comp.* **67**(3), 1272–1292 (2016)
19. Moura, R.C., Mengaldo, G., Peiró, J., Sherwin, S.J.: On the eddy-resolving capability of high-order discontinuous Galerkin approaches to implicit LES/under-resolved DNS of Euler turbulence. *J. Comp. Phys.* **330**, 615–623 (2017)
20. Moxey, D., Cantwell, C.D., Kirby, R.M., Sherwin, S.J.: Optimizing the performance of the spectral/ hp element method with collective linear algebra operations. *Comp. Meth. Appl. Mech. Eng.* **310**, 628–645 (2016)
21. Moxey, D., Ekelschot, D., Keskin, Ü., Sherwin, S.J., Peiró, J.: High-order curvilinear meshing using a thermo-elastic analogy. *Comp. Aided Des.* **72**, 130–139 (2016)
22. Moxey, D., Green, M.D., Sherwin, S.J., Peiró, J.: An isoparametric approach to high-order curvilinear boundary-layer meshing. *Comp. Meth. Appl. Mech. Eng.* **283**, 636–650 (2015)
23. Persson, P.O., Peraire, J.: Sub-cell shock capturing for discontinuous Galerkin methods. AIAA paper **112** (2006)
24. Solín, P., Demkowicz, L.: Goal-oriented hp -adaptivity for elliptic problems. *Comp. Meth. Appl. Mech. Eng.* **193**(1), 449–468 (2004)
25. Vincent, P.E., Castonguay, P., Jameson, A.: A new class of high-order energy stable flux reconstruction schemes. *J. Sci. Comp.* **47**(1), 50–72 (2011)
26. Vos, P.E., Sherwin, S.J., Kirby, R.M.: From h to p efficiently: Implementing finite and spectral/ hp element methods to achieve optimal performance for low- and high-order discretisations. *J. Comp. Phys.* **229**(13), 5161–5181 (2010)
27. Yakovlev, S., Moxey, D., Sherwin, S.J., Kirby, R.M.: To CG or to HDG: a comparative study in 3D. *J. Sci. Comp.* **67**(1), 192–220 (2016)