Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 213 (2006) 330–357
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

High-order discontinuous Galerkin methods using an *hp*-multigrid approach

Cristian R. Nastase *, Dimitri J. Mavriplis

Department of Mechanical Engineering, University of Wyoming, 1000 E. University Avenue, Laramie, WY 82071-3295, United States

Received 13 April 2005; received in revised form 12 August 2005; accepted 15 August 2005
Available online 7 October 2005

Abstract

The goal of this paper is to investigate and develop a fast and robust algorithm for the solution of high-order accurate discontinuous Galerkin discretizations of non-linear systems of conservation laws on unstructured grids. Herein we present the development of a spectral *hp*-multigrid method, where the coarse "grid" levels are constructed by reducing the order (p) of approximation of the discretization using hierarchical basis functions (p-multigrid), together with the traditional (h-multigrid) approach of constructing coarser grids with fewer elements. On each level we employ variants of the element-Jacobi scheme, where the Jacobian entries associated with each element are treated implicitly (i.e., inverted directly) and all other entries are treated explicitly. The methodology is developed for the two-dimensional non-linear Euler equations on unstructured grids, using both non-linear (FAS) and linear (CGC) multigrid schemes. Results are presented for the channel flow over a bump and a uniform flow over a four element airfoil. Current results demonstrate convergence rates which are independent of both order of accuracy (p) of the discretization and level of mesh resolution (h).

PACS: 47.11.+j; 83.85.Pt; 47.15.Ki

Keywords: Computational fluid dynamics; Discontinuous Galerkin finite element methods; High order methods; Numerical methods; Multigrid methods; Compressible flow; Gas dynamics

1. Introduction

While most currently employed CFD algorithms are asymptotically second-order accurate in time and in space, the use of higher-order discretizations in both space and time offers a possible avenue for improving the predictive simulation capability for many applications. This is due to the fact that higher-order methods exhibit a faster asymptotic convergence rate in the discretization error than lower (second)-order methods. For example, with a fourth-order accurate spatial discretization, the error is reduced by a factor of $2^4 = 16$

* Corresponding author. Tel.: +1 307 7665424; fax: +1 307 7662695.
E-mail addresses: nastase@uwyo.edu (C.R. Nastase), mavripl@uwyo.edu (D.J. Mavriplis).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2005.08.022

each time the mesh resolution is doubled, while a second-order accurate method only achieves a $2^2 = 4$ reduction in error with each doubling of the mesh resolution. Since a doubling of mesh resolution in three dimensions entails an increase of overall work by a factor of $2^3 = 8$, achieving an arbitrarily prescribed error tolerance with second-order accurate methods in three dimensions can quickly become unfeasible.

Thus, for increasingly high accuracy levels, higher-order methods ultimately become the method of choice. Therefore, the expectation is that an efficient higher-order discretization may provide an alternate path for achieving high accuracy in a flow with a wide disparity of length scales at reduced cost, by avoiding the use of excessive grid resolution.

On the other hand, for levels of accuracy often associated with mean-flow engineering calculations, higher-order methods have proved to be excessively costly compared to simpler second-order accurate methods. Clearly, because of the different asymptotic nature of these methods, the cost comparison between methods is a strong function of the required levels of accuracy. Nevertheless, for many engineering type calculations, higher-order methods have been found to be non-competitive compared to the simpler second-order accurate methods.

While the formulation of discretization strategies for higher-order methods such as discontinuous Galerkin [1–7] and streamwise upwind Petrov–Galerkin [8] methods are now fairly well understood, the development of techniques for efficiently solving the discrete equations arising from these methods has generally been lagging. This is partly due to the complex structure of the discrete equations originating from fairly sophisticated discretization strategies, as well as the current application of higher-order methods to problems where simple explicit time-stepping schemes are thought to be adequate solution mechanisms, due to the close matching of spatial and temporal scales, such as acoustic phenomena.

The development of optimal, or near optimal solution strategies for higher-order discretizations, including steady-state solutions methodologies, and implicit time integration strategies, remains one of the key determining factors in devising higher-order methods which are not just competitive but superior to lower-order methods in overall accuracy and efficiency.

Recent work by the second author has examined the use of spectral multigrid methods, where convergence acceleration is achieved through the use of coarse levels constructed by reducing the order (p) of approximation of the discretization (as opposed to coarsening the mesh) for discontinuous Galerkin discretizations [9]. The idea of spectral multigrid was originally proposed by Ronquist and Patera [10], and has been pursued for the Euler and Navier–Stokes equations by Fidkowski et al. [11–13] with encouraging results. Implicit multi-level solution techniques for high-order discretizations have also been developed by Lottes and Fisher [14].

In this work, we extend the original spectral multigrid approach described in [9] to the two-dimensional steady-state Euler equations, and couple the spectral p-multigrid approach with a more traditional agglomeration h-multigrid method for unstructured meshes. The investigation of efficient smoothers to be used at each level of the multigrid algorithm is also pursued, and comparisons between linear and non-linear solver strategies are made as well. The overall goal is the development of a solution algorithm which delivers convergence rates which are independent of p (the order of accuracy of the discretization) and independent of h (the degree of mesh resolution), while minimizing the cost of each iteration.

The key ingredient in the p-multigrid approach is to employ a hierarchical basis set together with a modal method. This renders the multigrid inter-level operators almost trivial to implement. This approach is rather different than the nodal method presented in [15], where a non-hierarchical basis (i.e., nodal basis based on Lagrange polynomials) is employed and the multilevel process requires rather complicated grid transfer operators. Moreover, in our methodology the coarse-grids are known a priori and the multilevel methodology is obtained by using known subsets of the original matrix. That is, the coarse grids correspond to a modal expansion in a lower space. This is also different than the algebraic multigrid (AMG) method [16] where a "matrix-free" operator is employed without prior knowledge of the coarse-grids and the multilevel process is obtained from an algebraic standpoint.

Note that the "hp-" terminology is commonly used to denote adaptive spatial and polynomial resolutions. This is referred to as "h-" and "p-adaptivity". Although the current multigrid methodology is not applied adaptively, it does make use of p-coarsened and h-coarsened levels leading to our terminology of either "p-multigrid", "h-multigrid" or "hp-multigrid" for the combined algorithm.

2. Governing equations

The conservative form of the compressible Euler equations describing the conservation of mass, momentum and total energy is given in vectorial form

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

subject to appropriate boundary and initial conditions within a two-dimensional domain Ω. Explicitly, the state vector $\mathbf{U}$ of the conservative variables and the Cartesian components of the inviscid flux $\mathbf{F} = (\mathbf{F}^x, \mathbf{F}^y)$ are:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{pmatrix}, \quad \mathbf{F}^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E_t + p)u \end{pmatrix}, \quad \mathbf{F}^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E_t + p)v \end{pmatrix}, \tag{2}$$

where ρ is the fluid density, (u, v) are the fluid velocity Cartesian components, p is the pressure and E_t is the total energy. For an ideal gas, the equation of state relates the pressure to total energy by:

$$p = (\gamma - 1)\left[E_t - \frac{1}{2}\rho(u^2 + v^2)\right], \tag{3}$$

where $\gamma = 1.4$ is the ratio of specific heats.

3. Spatial discretization

The computational domain Ω is partitioned into an ensemble of non-overlapping elements and within each element the solution is approximated by a truncated polynomial expansion

$$\mathbf{U}(\mathbf{x},t) \approx \mathbf{U}_p(\mathbf{x},t) = \sum_{j=1}^{M} \mathbf{u}_j(t)\phi_j(\mathbf{x}), \tag{4}$$

where M is the number of modes defining the truncation level. The semi-discrete formulation (i.e., continuous in time) employs a local discontinuous Galerkin formulation [2,3,5,6] in spatial variables within each element Ω_k. The weak formulation for Eq. (1) is obtained by minimizing the residual with respect to the expansion function in an integral sense:

$$\int_{\Omega_k} \phi_i \left[\frac{\partial \mathbf{U}_p(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}_p)\right]_k \mathrm{d}\Omega_k = 0. \tag{5}$$

After integrating by parts the weak statement of the problem becomes:

$$\int_{\Omega_k} \phi_i \frac{\partial \mathbf{U}_p}{\partial t}\, \mathrm{d}\Omega_k - \int_{\Omega_k} \nabla\phi_i \cdot \mathbf{F}(\mathbf{U}_p)\, \mathrm{d}\Omega_k + \int_{\partial\Omega_k} \phi_i \mathbf{F}^*(\mathbf{U}_p) \cdot \mathbf{n}\, \mathrm{d}(\partial\Omega_k) = 0. \tag{6}$$

The local discontinuous Galerkin approach makes use of element-based basis functions, which results in solution approximations which are local, discontinuous, and doubled valued on each elemental interface. Monotone numerical fluxes are used to resolve the discontinuity, providing the means of communication between adjacent elements and specification of the boundary conditions. The numerical flux, $\mathbf{F}^*(\mathbf{U}_p) \cdot \mathbf{n}$, is obtained as a solution of a local one-dimensional Riemann problem and depends on the internal interface state, $\mathbf{U}_p^-$, the adjacent element interface state, $\mathbf{U}_p^+$, and the orientation as defined by the normal vector, $\mathbf{n}$, of the interface. An approximate Riemann solver is used to compute the flux at inter-element boundaries and provides the means of imposing boundary conditions. Current implementations include the flux difference splitting schemes of Rusanov [17], Roe [18], HLL [19] and HLLC [20–22].

The discrete form of the local discontinuous Galerkin formulation is defined by the particular choice of the set of basis functions, $\{\phi_i, i = 1, \ldots, M\}$. The basis set is defined on a standard triangle $\hat{\Omega}(\xi, \eta)$ spanning between $\{0 < \xi, \eta < 1\}$. We seek a set of hierarchical basis functions in order to simplify our subsequent spectral multigrid implementation. Defining the first order Lagrange polynomials:

$$L_1 = 1 - \xi - \eta, \quad L_2 = \xi, \quad L_3 = \eta \tag{7}$$

the hierarchical basis set, $\{\phi_i\}$, is fully described by *vertex*,

$$\phi_1^{\mathrm{v}} = L_1, \quad \phi_2^{\mathrm{v}} = L_2, \quad \phi_3^{\mathrm{v}} = L_3 \tag{8}$$

edge,

$$\begin{aligned} \phi_n^{\mathrm{e}1} &= L_1 L_2 \psi_{n-2}(L_2 - L_1), \\ \phi_n^{\mathrm{e}2} &= L_2 L_3 \psi_{n-2}(L_3 - L_2), \\ \phi_n^{\mathrm{e}3} &= L_3 L_1 \psi_{n-2}(L_1 - L_3), \end{aligned} \tag{9}$$

and *bubble*,

$$\phi_{n1,n2}^{\mathrm{b}} = L_1 L_2 L_3 \psi_{n1-1}(L_2 - L_1)\psi_{n2-1}(L_1 - L_3) \tag{10}$$

shape functions, where $2 \leqslant n \leqslant p^{\mathrm{e}}$, $n1 + n2 = p^{\mathrm{b}} - 1$ and $n1, n2 \geqslant 1$. The kernel functions $\psi(z)$ are given as:

$$\psi_{n-2}(z) = \frac{-2\sigma}{n-1} P_{n-2}^{1,1}(z), \tag{11}$$

where $P_n^{\alpha,\beta}$ represents the Jacobi polynomial of order, n, with weights α and β. In our discretization the edge order, p^{e}, and the bubble order, p^{b}, are set to $p^{\mathrm{e}} = p^{\mathrm{b}} = p$, where p is the discretization order within the element. For $p \geqslant 2$ the basis functions within the standard triangle, $\{\phi_i, i = 4 \ldots M\}$, are normalized Lobatto (i.e., $\phi_{n \geqslant 2} = \sigma \int_{-1}^{x} P_{n-1}^{0,0}(z)\, \mathrm{d}z$) functions [23], which take zero values at the end of their definition interval (Fig. 1). The normalization factor, σ, can be used to condition the mass or convection matrices.

Although the choice of basis functions does not affect the accuracy of the method, this choice is dictated by the intended application. In our application (Euler equations) which employs spatial derivatives, our choice of basis functions has a significant advantage over other types of bases described in [7,24,25]. Specifically, the basis used in our implementation [23] is rotationally invariant within the standard triangle. The basis used by Karniadakis and Sherwin [7] for triangular elements makes use of collapsed coordinate systems in order to construct the basis as a tensor-product of Jacobi polynomials and does not preserve the rotational invariance. Although this basis has good orthogonality properties and the tensor-product brings advantages in terms of number of operations required to compute the integral terms via quadrature rules, the lack of rotational invariance is somewhat inconvenient. It is known that in the case of a full orthogonal basis set, the transformation operator from the polynomial space to physical space is identity and the mass matrix becomes diagonal. However, in the steady state case considered here there is no need to employ the mass matrix. Furthermore, our choice of basis is not necessarily optimal but, under adequate scaling, the current basis presents good conditioning properties up to $p = 10$ (not shown), which is high enough for aerodynamic applications. Future work pertaining to time-dependent problems will revisit the issue of optimal choice of basis functions.

Since the basis set is defined in the standard triangle, a coordinate transformation, $\{x = x(\xi, \eta),\ y = y(\xi, \eta)\}$, is required to compute the derivatives and the integrals in physical space $\Omega_k(x, y)$. For iso-parametric elements, the basis functions are expressed as functions of ξ and η, and the coordinate transformation, and its Jacobian within the standard element, $\hat{\Omega}_k(\xi, \eta)$, are given by:

$$\mathbf{x}_p = \sum_{j=1}^{M} \hat{\mathbf{x}}_j \phi_j(\xi, \eta), \quad J_k(\xi, \eta) = \frac{\partial(x, y)}{\partial(\xi, \eta)}. \tag{12}$$

In the simple case of straight-sided elements the mapping is linear and the determinant of the Jacobian, $|J_k|$, and metrics, $\xi_x = \partial\xi/\partial x$, $\xi_y = \partial\xi/\partial y$, $\eta_x = \partial\eta/\partial x$, $\eta_y = \partial\eta/\partial y$, are constant within each element. For the general case, using Eq. (12), the solution expansion and the weak statement within each element, $\hat{\Omega}_k$, becomes:

$$\mathbf{U}_p(\xi, \eta, t) = \sum_{j=1}^{M} \hat{\mathbf{u}}_j(t) \phi_j(\xi, \eta) \tag{13}$$

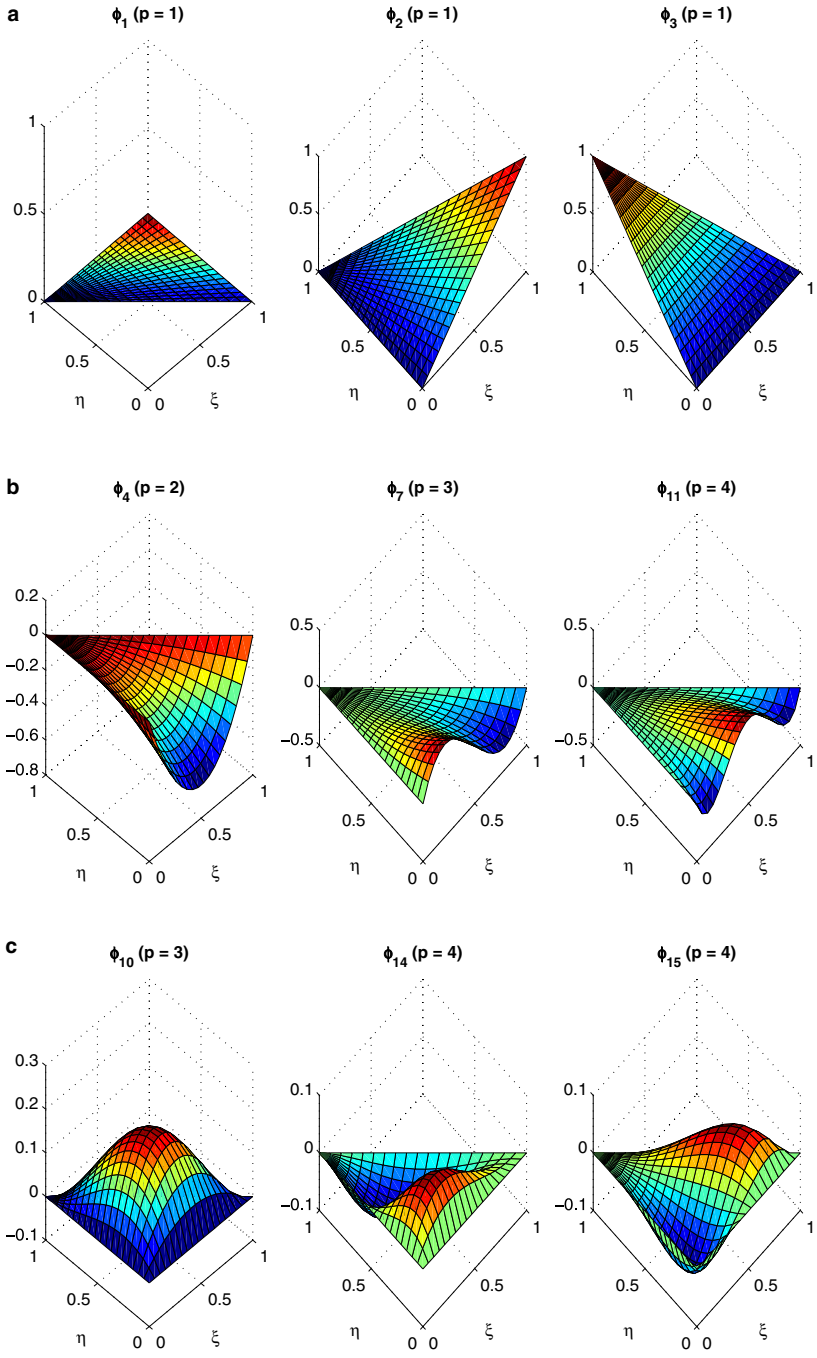

Fig. 1. (a) Vertex, (b) edge and (c) bubble (interior) shape functions on standard triangle.

$$\int_{\hat{\Omega}_k} \phi_i \frac{\partial \mathbf{U}_p}{\partial t} |J_k| \, \mathrm{d}\hat{\Omega}_k - \int_{\hat{\Omega}_k} \nabla \phi_i J_k^{-1} \cdot \mathbf{F}(\mathbf{U}_p) |J_k| \, \mathrm{d}\hat{\Omega}_k + \int_{\partial\hat{\Omega}_k} \phi_i \mathbf{F}^*(\mathbf{U}_p) \cdot \mathbf{n} |J_k| \, \mathrm{d}(\partial\hat{\Omega}_k) = 0. \tag{14}$$

This set of equations is solved in the modal space and the integrals are evaluated by Gaussian quadrature rules, which requires a projection in physical space of the solution values to the quadrature points used in the numerical integration. In order to preserve $p+1$ accuracy order of the numerical approximation, the

element integral uses Gaussian quadrature rules [26,27] which are exact for polynomial degree $2p$ within the standard triangle, while the boundary integral uses Gauss–Legendre–Lobatto quadrature rules which are exact for polynomial degree $2p+1$ [28] (Fig. 2). For boundary elements with curved edges the quadrature rule is no longer exact and a higher order rule must be used in order to preserve the accuracy [29]. In addition the Jacobians must be evaluated at the integration quadrature points, whereas for interior triangles with straight edges, these are constant and need only be evaluated once for each element thus reducing the storage requirements.

Complicated geometries require the use of curved boundaries, especially in the case of high order methods where elements span a bigger portion of boundaries than the case of low order methods. The influence of curvature on the accuracy order is assessed via a projection-based interpolation [30,31] of a smooth non-polynomial function. The projection-based interpolation offers a very close approximation of the flow field solution and, therefore, can be used to assess the convergence rates in the presence of curved boundaries. The function is defined as $f(x,y) = 1/2\sin(4\pi x)\cos(4\pi y)$ spanning between $\{0 < x, y < 1\}$, as illustrated in Fig. 3(a). The projection-based interpolation is performed locally (i.e., element-wise) by transferring the function to the master element. The function is interpolated on the master element and the resulting interpolant is then transferred back to the physical space. Three cases are considered where the computational domain is divided in uniform distributed unstructured elements (triangles). In the Case 1, all the elements are straight-sided (Fig. 3(b)). In the Case 2, all the edges are second order (parabolas) with the maximum deflection 10% of their support, as depicted in Fig. 3(c). Although in real case grids a much smaller curvature is used, it is important to use a large value in oder to assess its influence. Finally, in the Case 3, only the boundary (i.e., exterior) elements are curved and all other (interior) elements are straight-sided (Fig. 3(d)). The discretization error can be assessed by measuring the L_2 or L_∞ norms of the projection error, in a global sense. The L_2 and L_∞ norms of a quantity "w" are defined as:

$$\|w\|_{L_2} = \left(\int_\Omega w^2 \, d\Omega\right)^{1/2}, \quad \|w\|_{L_\infty} = \max|w|, \tag{15}$$

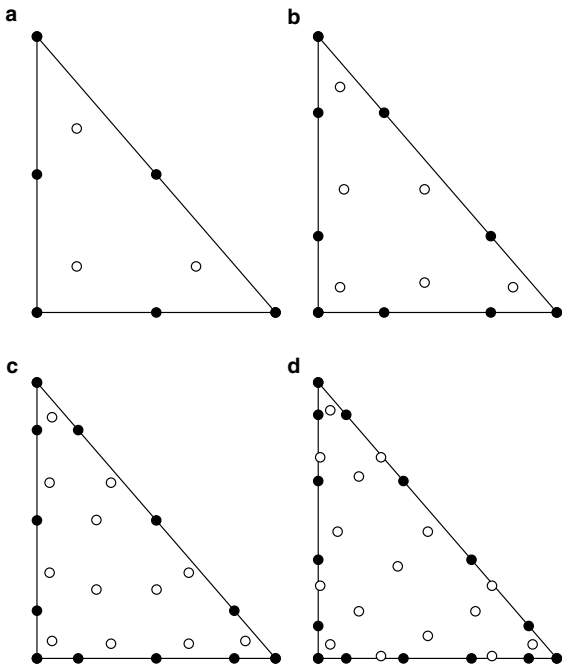

Fig. 2. Interior and edge quadrature points on standard triangle for various approximation orders (○, interior; •, edge). (a) $p = 1$, (b) $p = 2$, (c) $p = 3$, (d) $p = 4$.

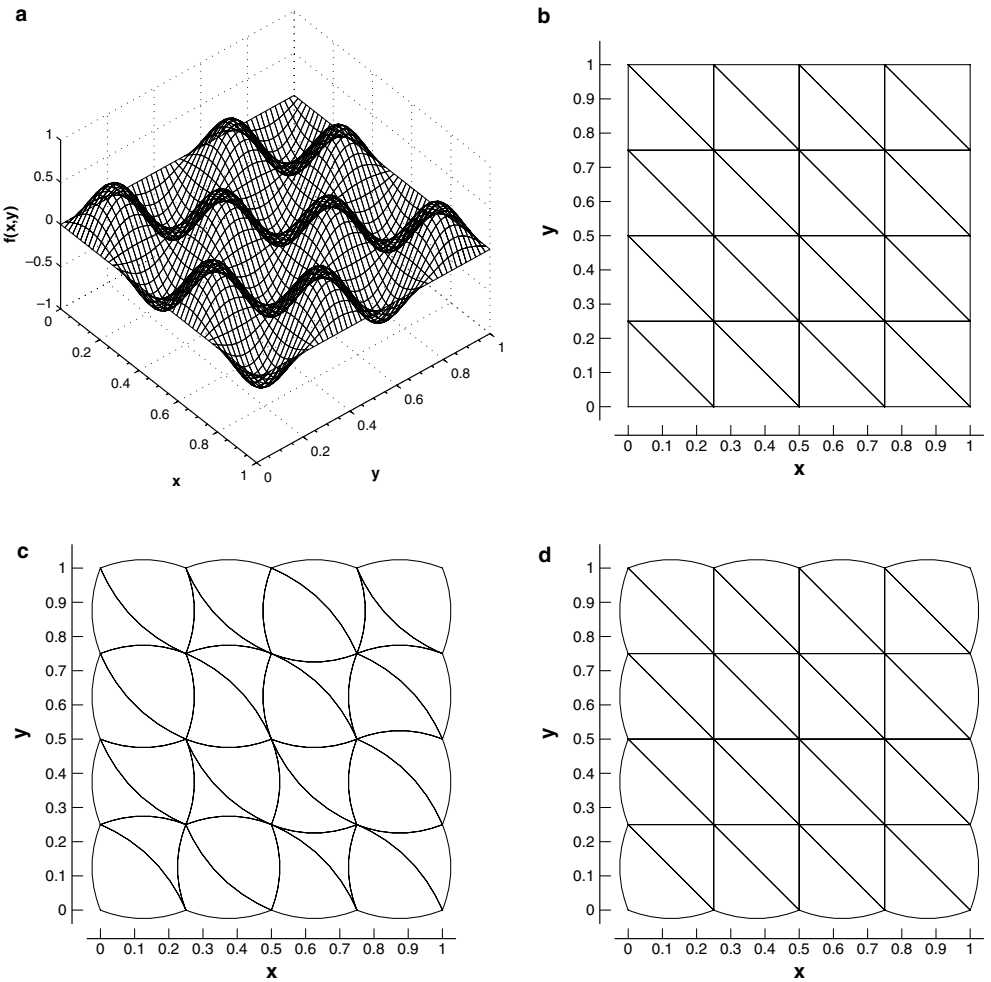

Fig. 3. Projection function and mesh configuration: (a) $f(x,y) = 1/2\sin(4\pi x)\cos(4\pi y)$, (b) Case 1, (c) Case 2, (d) Case 3.

where Ω represents the entire computational domain. Fig. 4 shows the L_2 norm of the projection error for various approximation order (p) and mesh sizes (h). Clearly, the (optimum) $p+1$ accuracy order is obtained in the case of straight-sided elements (Case 1) while a sub-optimal ($\approx p$) accuracy is observed for curved elements (Case 2). Also, the departure form the optimal accuracy increases with the curvature. For example, for a 5% deflection an $\approx p + 1/2$ accuracy order was obtained (not shown). However, the use of curved-sided elements for boundaries only (Case 3) has a minor effect on global accuracy (Fig. 5). The accuracy levels for all cases are summarized in Table 1. Note that the projection error was measured in a global sense and the local accuracy at the boundaries will still suffer form a sub-optimal accuracy level [32]. Therefore, in order to maintain the desired $p+1$ global accuracy order, in all simulations we make use of curved-sided elements for (wall) boundary elements only, and straight-sided elements for all other (interior) elements.

4. The implicit steady state solver

Neglecting the temporal derivative term, the system of equations (Eq. (14)) associated with each element becomes:

$$\mathbf{R}(\mathbf{U}_p) = \mathbf{S}_p, \tag{16}$$

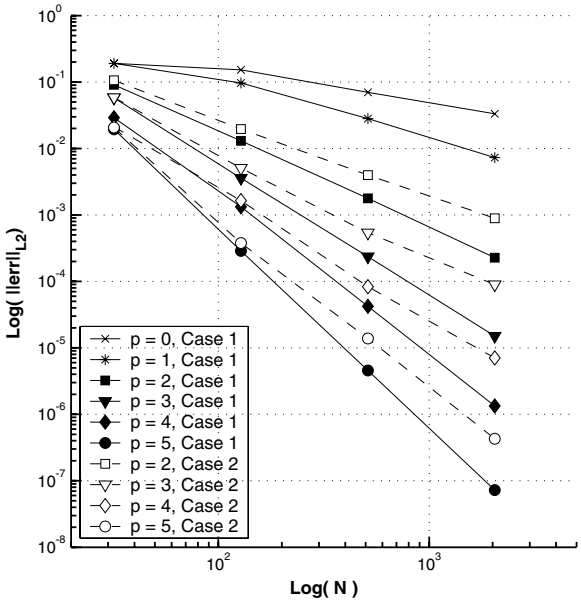

Fig. 4. The L_2 norm of the projection error as a function of h/p-refinement.

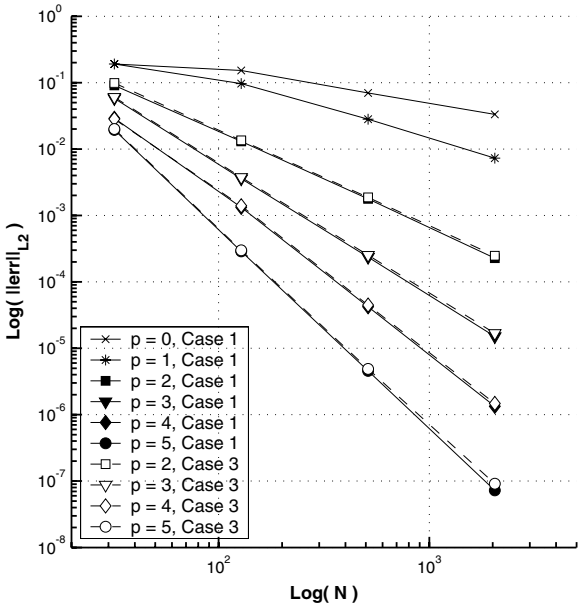

Fig. 5. The L_2 norm of the projection error as a function of h/p-refinement.

where $\mathbf{R}(\mathbf{U}_p)$ is the non-linear residual and $\mathbf{S}_p$ is the source term. Although in the case of the Euler equations $\mathbf{S}_p = 0$, the use of a source term will facilitate the introduction of the multigrid algorithm in the next section. We use variants of the element-Jacobi scheme to solve this system of equations. The Newton iteration associated with Eq. (16) yields at each "$n+1$" step:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}_p}\right]^n \Delta \mathbf{U}_p^{n+1} = \mathbf{S}_p - \mathbf{R}(\mathbf{U}_p^n), \quad \mathbf{U}_p^{n+1} = \mathbf{U}_p^n + \alpha \Delta \mathbf{U}_p^{n+1}, \tag{17}$$

Table 1
The slopes of the L_2 norm of the projection error as a function of h/p-refinement for Case 1 (straight-sided elements), Case 2 (curved-sided elements) and Case 3 (curved-boundaries)

p	Case 1	Case 2	Case 3
0	1.09	–	–
1	1.86	–	–
2	2.92	2.23	1.89
3	3.95	2.91	3.90
4	4.98	3.93	4.94
5	5.98	4.89	5.84

where α is a parameter used for robustness to keep $\|\alpha\Delta\mathbf{U}_p^{n+1}/\mathbf{U}_p^{n+1}\|_{L_\infty} \leqslant 10\%$. The element-Jacobi scheme can be viewed as an approximate Newton scheme where the full Jacobian matrix is replaced by the block diagonal entries representing the coupling between all modes within each element, $[\partial\mathbf{R}/\partial\mathbf{U}_p] = [D]$, thus neglecting the coupling between neighboring element modes, which arises through the inter-element flux evaluations (Fig. 6). The $[D]$ blocks represent small dense matrices associated with each grid element. These element matrices are inverted using Gaussian elimination to produce a lower-upper (LU) factorization of each element matrix. In the case of the two-dimensional Euler equations (Eq. (1)) with cubic triangular elements ($p = 3$), the block diagonal matrix ($[D]$) contains 40×40 entries for each element. The non-linear iteration equation (17) becomes:

$$\Delta\mathbf{U}_p^{n+1} = [D^n]^{-1}(\mathbf{S}_p - \mathbf{R}(\mathbf{U}_p^n)). \tag{18}$$

This solver denoted as the non-linear element Jacobi (NEJ). A second variant of this solver it the *quasi* non-linear element Jacobi (qNJ). This variant employs "k" quasi non-linear iterations, where only the residual, $\mathbf{R}(\mathbf{U}_p^k)$, is updated, and the block diagonal matrices, $[D^n]$, are kept constant from the outer-iteration "n". Therefore, the $(k+1)$ th step is:

$$\Delta\mathbf{U}_p^{k+1} = [D^n]^{-1}(\mathbf{S}_p - \mathbf{R}(\mathbf{U}_p^k)). \tag{19}$$

This approach is expected to yield same converge rates per cycle as in the NEJ variant, with a much improved performance in terms of CPU time. A third variant of this solver is denoted as the linearized element Jacobi

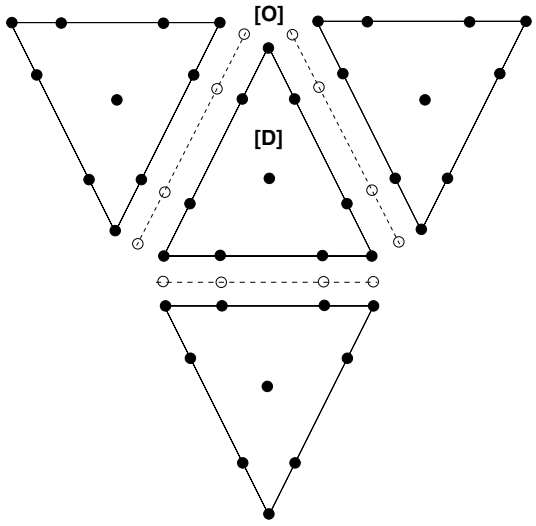

Fig. 6. Discontinuous solution representation illustrating element based modes and inter-element flux evaluations (●, element modes; ○, edge modes).

(LEJ) method. In this approach, the full Jacobian matrix is retained, but is decomposed into block diagonal $[D]$ and off-diagonal $[O]$ components:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}_p}\right]^n = [D^n] + [O^n]. \tag{20}$$

An iterative procedure can now be written by taking the $[O]$ components, which contain terms arising from the inter-element flux evaluations, to the right-hand-side of Eq. (17). In matrix form the $(k+1)$th step of the linearized element Jacobi step is written as:

$$\Delta \mathbf{U}_p^{k+1} = [D^n]^{-1}\left(\mathbf{S}_p - \mathbf{R}(\mathbf{U}_p^n) - [O^n]\Delta \mathbf{U}_p^k\right). \tag{21}$$

Note that the linearized element Jacobi scheme involves a dual iteration strategy, where each nth outer non-linear iteration entails "k" inner linear iterations. The advantage of this formulation is that the non-linear residual $\mathbf{R}(\mathbf{U}_p^n)$ and the Jacobian entries $[D^n]$ and $[O^n]$ are held constant during the linear iterations. This can significantly reduce the required computational time per cycle for expensive non-linear residual constructions. Because this scheme represents an exact linearization of the element-Jacobi scheme (Eq. (18)), both approaches can be expected to converge at the same rates per cycle (asymptotically) [33]. On the other hand, the linearized element Jacobi scheme requires extra storage for the $[O]$ Jacobian blocks, which may not be feasible for large three-dimensional problems.

The convergence of Eq. (21) can be further accelerated by using a Gauss–Seidel strategy where the off-diagonal matrices are split into lower, $[L]$, and upper, $[U]$, contributions (i.e., $[O] = [L] + [U]$). This last solver variant (LGS) becomes:

$$\Delta \mathbf{U}_p^{k+1} = [(D+L)^n]^{-1}\left(\mathbf{S}_p - \mathbf{R}(\mathbf{U}_p^n) - [U^n]\Delta \mathbf{U}_p^k\right) \tag{22}$$

which again involves a dual iteration strategy, but follows an ordered sweep across the elements using latest available neighboring information in the Gauss–Seidel sense. In this work, we employ a frontal sweep along the elements which begins near the inner boundary and proceeds toward the outer boundary, using the numbering assigned to the grid elements from an advancing front mesh generation technique [34].

Note that a non-linear element Gauss–Seidel approach is also possible, based on the element-Jacobi solver, which does not require the storage of the off-diagonal $[O]$ blocks. This approach is not considered in the current work. All the simulation results shown here are performed using the HLLC flux only.

The boundary conditions are imposed via the approximate Riemann solver at every iteration. In the case of the Newton solver, the boundary conditions are naturally included in the diagonal $[D]$ term as a result of the full linearization of the governing equations [22], and no additional treatment is required.

5. Single grid results

The accuracy of the spatial discretizations and the efficiency of the solution schemes described above are evaluated for the Euler equations using a test problem consisting of the compressible channel flow over a bump. A series of four grids on this configuration have been generated, consisting of $N = 505$, 1047, 2015 and 4093 triangular elements, respectively, in order to study the grid convergence of the discontinuous Galerkin discretizations of various orders. For each case the solution was converged to machine zero residual in the discretization error studies. Fig. 7 shows the grid configuration and the Mach contour lines for a freestream Mach number of $M_\infty = 0.25$. The initial condition is set to a solution obtained beforehand with $p = 0$. The full domain extends from $-2 \leqslant x \leqslant 2$ in the stream-wise direction and from $0 \leqslant y \leqslant 2$ in the cross-stream direction, with top and bottom wall boundaries. For this particular case, the grid consists of $N = 1047$ triangular elements, and the discretization order is $p = 4$ (i.e., fifth-order accurate). The discretization error can be assessed by measuring the L_2 norm of the entropy error for this isentropic flow. The entropy error is defined as $\mathrm{d}s = s - s_\infty$, where s_∞ is the free stream entropy.

Fig. 8 shows the accuracy (i.e., the L_2 entropy error norm) of the steady-state solution for 1st, 2nd, 3rd and 4th order accurate discretizations as a function of the number of triangular elements. For two-dimensional configurations the number of elements, N, is proportional to $1/h^2$, where h represents an approximation of

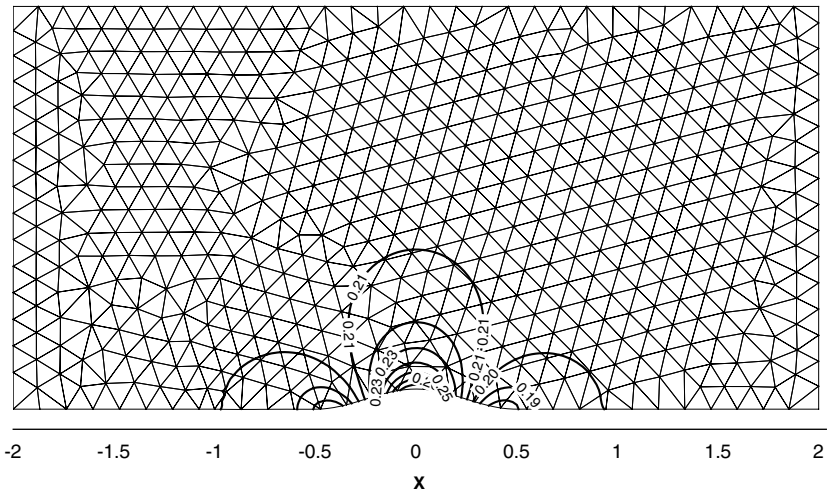

Fig. 7. Solution contours on a domain of $N = 1047$ elements and $p = 4$.

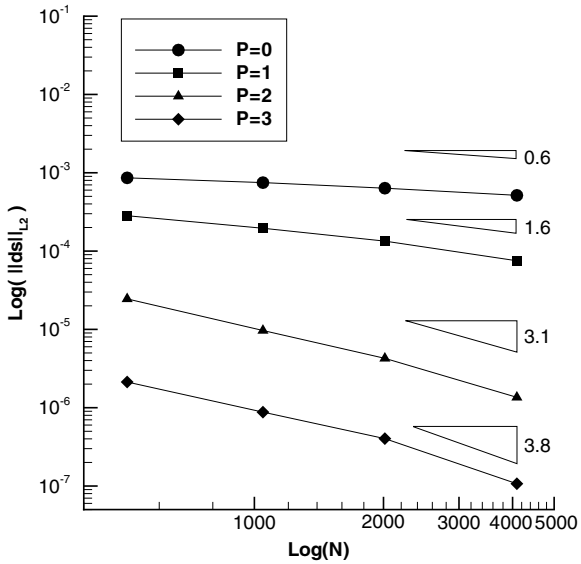

Fig. 8. The L_2 norm of the entropy error as a function of h/p-refinement.

the cell size. The asymptotic slope of these curves indicates that the design accuracy of the various discretization schemes is approximately realized in this study. For example, the final slope of the $p = 3$ curve is 3.8, which is close to the design accuracy value of 4.

A comparison of the computed accuracy versus CPU time is given in Fig. 9, where the various p-discretizations have been converged to machine zero on the various grid configurations using the linearized element-Jacobi driven multigrid scheme described in the next section. In general, for a given level of accuracy, the CPU time decreases when the approximation order is increased, with the benefit increasing for smaller accuracy tolerances.

Fig. 10 depicts the convergence of the non-linear element Jacobi, quasi non-linear element Jacobi, linearized element Jacobi, and linearized element Gauss–Seidel schemes on the mesh of $N = 2015$ elements, for the $p = 4$ discretization. The convergence is measured in terms of overall number of cycles, linear cycles for the linear schemes, and non-linear cycles for the element-Jacobi scheme. As expected, the non-linear element-Jacobi,

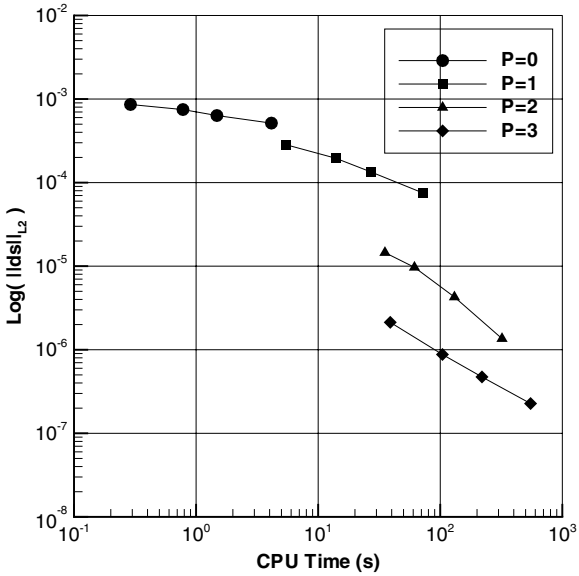

Fig. 9. The L_2 norm of the entropy error as a function of CPU time.

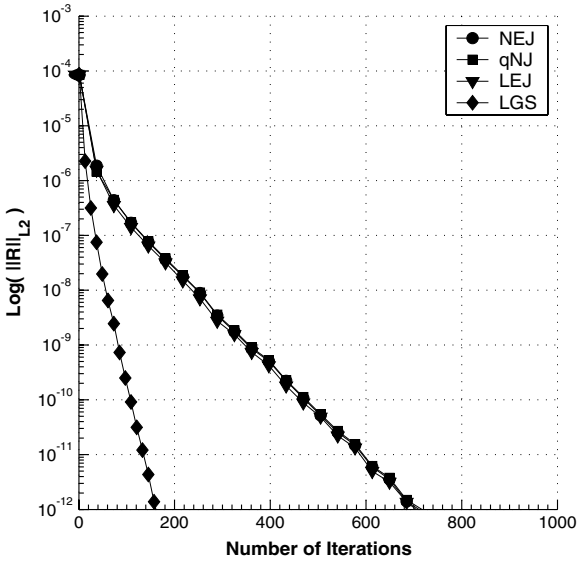

Fig. 10. Comparison of convergence of non-linear element-Jacobi (NEJ), quasi non-linear element-Jacobi (qNJ), linear element-Jacobi (LEJ), and linear element Gauss–Seidel (LGS) in terms of iterations on a mesh size of $N = 2015$ elements and order $p = 4$.

quasi non-linear element Jacobi and linearized element-Jacobi schemes converge at similar rates in terms of numbers of cycles, while the linearized element Gauss–Seidel scheme converges substantially faster. When compared in terms of CPU time, Fig. 11, the linearized element-Jacobi and element Gauss–Seidel schemes are seen to be substantially more efficient than the non-linear element Jacobi scheme. The linearized schemes utilize 10 linear iterations between each non-linear update, and thus result in 10 times fewer non-linear residual and Jacobian evaluations than the element-Jacobi scheme. The savings are substantial due to the fact that these non-linear evaluations include the expensive quadrature integration procedures. The quasi non-linear element Jacobi may prove to be an appropriate compromise for cases where memory limitations are dominant (i.e., only diagonal blocks, $[D]$, storage required).

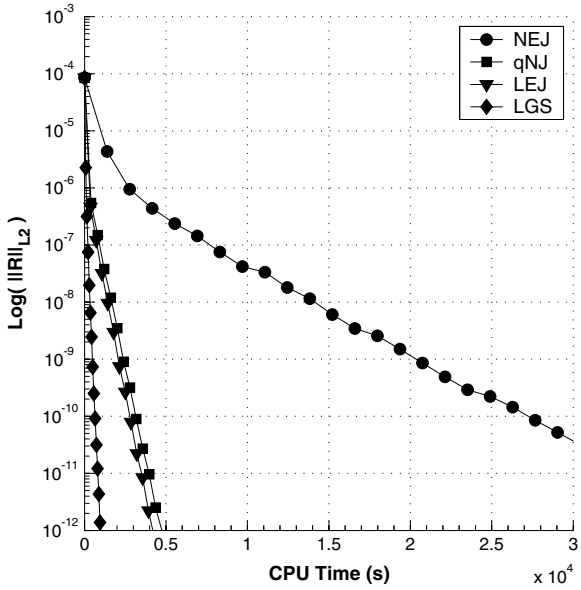

Fig. 11. Comparison of convergence of non-linear element-Jacobi (NEJ), quasi non-linear element-Jacobi (qNJ), linear element-Jacobi (LEJ), and linear element Gauss–Seidel (LGS) in terms of CPU time on a mesh size of $N = 2015$ elements and order $p = 4$.

Due to the superior efficiency of the linearized schemes, the remaining results will make exclusive use of these schemes. Fig. 12 illustrates the convergence of the linearized element-Jacobi solver as measured by the rate of the residual reduction versus the number of iterations, for approximation orders varying from $p = 1$ to $p = 4$, on the mesh of $N = 2015$ elements. Clearly, the method yields a convergence rate which is independent of the order of accuracy of the discretization for a fixed size grid. However, increasing the number of elements, N, has an adverse effect on the convergence rate. In Fig. 13, the convergence rate for $p = 4$ is seen to

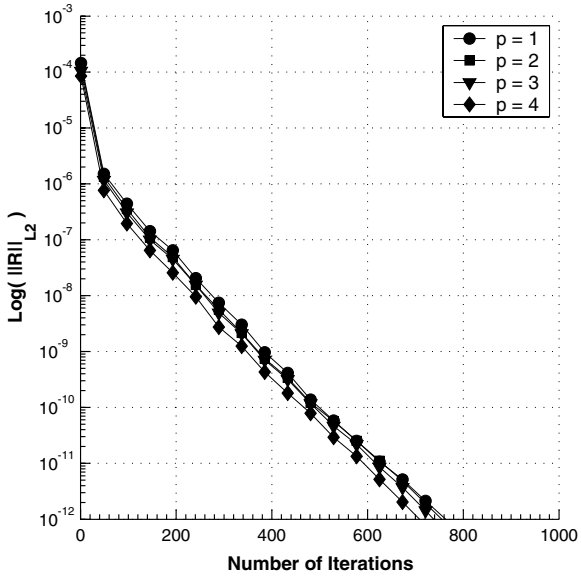

Fig. 12. The L_2 norm of the residual vs. number of linear element-Jacobi (LEJ) cycles on a fixed mesh size of $N = 2015$ elements and various orders (p).

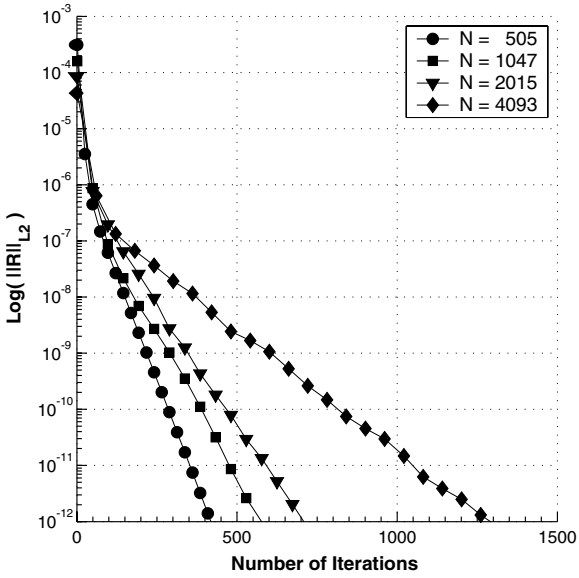

Fig. 13. The L_2 norm of the residual vs. number of linear element-Jacobi (LEJ) cycles for a fixed order $p = 4$.

degrade as the number of mesh elements is increased. This *h*-dependence of the element-Jacobi solver is addressed through the use of an *hp*-multigrid scheme.

6. The *hp*-multigrid approach

Multigrid methods are known as efficient techniques for accelerating convergence to steady state for both linear and non-linear problems [35,33], and can be applied with a suitable existing relaxation technique. The rapid convergence property relies on an efficient reduction of the solution error on a nested sequence of coarse grids.

6.1. General description

The spectral multigrid approach is based on the same concepts as a traditional *h*-multigrid method, but makes use of "coarser" levels which are constructed by reducing the order of accuracy of the discretization, rather than using physically coarser grids with fewer elements. Thus, all grid levels contain the same number of elements, which alleviates the need to perform complex interpolation between grid levels and/or to implement agglomeration-type procedures [33]. Furthermore, the formulation of the interpolation operators, between fine and coarse grid levels, is greatly simplified when a hierarchical basis set is employed for the solution approximation. The main advantage is due to the fact that the lower order basis functions are a subset of the higher order basis (i.e., hierarchical) and the *restriction* and *prolongation* operators become simple projection operators into a lower and higher order space, respectively [11]. Therefore, their formulation is obtained by a simple deletion or augmentation of the basis set. The *restriction* from fine to coarse level is obtained by disregarding the higher order modal coefficients and transferring the values of the low order modal coefficients exactly. Similarly, the *prolongation* from coarse to fine levels is obtained by setting the high order modes to zero and injecting the values of the low order coefficients exactly.

Multigrid strategies are based on a recursive application of a two-level solution mechanism, where the second (coarser) grid is solved exactly, and used to accelerate the solution on the finer grid [35]. Because the exact solution of the coarse grid problem at each multigrid cycle is most often prohibitively expensive, the recursive application of multigrid to solve the coarse grid problem offers the preferred approach for

minimizing the computational cost of the multigrid cycle, thus resulting in a complete sequence of coarser grids. For spectral (p)-multigrid methods, the recursive application of lower order discretizations ends with the $p = 0$ discretization on the same grid as the fine level problem. For relatively fine meshes, the (exact) solution of this $p = 0$ problem at each multigrid cycle can become expensive, and may impede the h-independence property of the multigrid strategy. The $p = 0$ problem can either be solved approximately by employing the same number of smoothing cycles on this level as on the finer p levels, or the $p = 0$ problem can be solved more accurately by performing a larger number of smoothing cycles at each visit to this coarsest level. In either case, the convergence efficiency will be compromised, either due to inadequate coarse level convergence, or to excessive coarse level solution cost. An alternative is to employ an h-multigrid procedure to solve the coarse level problem at each multigrid cycle. In this scenario, the p-multigrid scheme reverts to an agglomeration multigrid scheme once the $p = 0$ level has been reached, making use of a complete sequence of physically coarser agglomerated grids, thus the designation hp-multigrid. Agglomeration multigrid methods make use of an automatically generated sequence of coarser level meshes, formed by merging together neighboring fine grid elements, using a graph algorithm. First-order accurate ($p = 0$) agglomeration multigrid methods for unstructured meshes are well established and deliver near optimal convergence rates [36]. This procedure has the potential of resulting in a truly h- and p-independent solution strategy for high-order accurate discontinuous Galerkin discretizations. Fig. 14 illustrates the second agglomerated level used for the channel bump flow problem using the $N = 1047$ grid as the original mesh configuration, where the bold lines indicate the outlines of the agglomerated coarse level cells. This procedure is performed recursively, producing 4 coarse levels for this mesh. Applying the same procedure to the other triangular meshes for the channel bump configuration resulted in 3, 4, and 5 levels for the meshes containing $N = 505$, 2015 and 4093 elements, respectively.

Based on our experience with the linearized element-Jacobi solver, we also consider two ways of applying multigrid to the non-linear Euler equations. The first is to apply multigrid directly to the non-linear problem (Eq. (16)) via the full approximation storage (FAS) scheme. In a two-level p-multigrid method this scheme is given as:

- Iterate the fine-grid level problem and its residual, $\mathbf{r}_p$, using any of the element-Jacobi variants aforementioned:

$$\mathbf{R}_p(\mathbf{U}_p^n) = \mathbf{S}_p, \quad \mathbf{r}_p^n = \mathbf{S}_p - \mathbf{R}_p(\mathbf{U}_p^n). \tag{23}$$

- Obtain the source term for the coarse (i.e., $p - 1$) level by restricting both the solution and the residual:

$$\mathbf{S}_{p-1} = I_p^{p-1}\mathbf{r}_p^n, \quad \mathbf{U}_{p-1}^n = \tilde{I}_p^{p-1}\mathbf{U}_p^n. \tag{24}$$

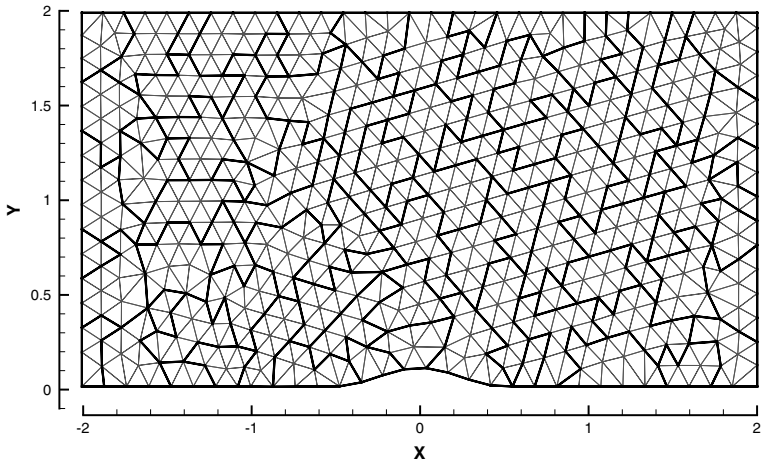

Fig. 14. A typical two level h-multigrid mesh configuration.

- Solve the coarse grid level problem

$$\mathbf{R}_{p-1}(\mathbf{U}^n_{p-1}) = \mathbf{S}_{p-1}. \tag{25}$$

- Calculate the coarse grid error, $\mathbf{e}^n_{p-1}$:

$$\mathbf{e}^n_{p-1} = \mathbf{U}^n_{p-1} - \tilde{I}^{p-1}_p \mathbf{U}^n_p. \tag{26}$$

- Prolongate the coarse grid error and correct the fine-grid level approximation:

$$\mathbf{U}^{n+1}_p = \mathbf{U}^n_p + I^p_{p-1}\mathbf{e}^n_{p-1}. \tag{27}$$

In the case of p-multigrid, $\tilde{I}^{p-1}_p$ and I^{p-1}_p denote the state and residual restriction (i.e., from p to $p-1$) operators, respectively. In the case of a hierarchical basis, I^{p-1}_p is the identity matrix with zero columns appended. Moreover, $I^{p-1}_p = \tilde{I}^{p-1}_p$ for p-multigrid but note that this is not true in the case of h-multigrid. Similarly, the prolongation (i.e., from $p-1$ to p) operator, I^p_{p-1}, is obtained as the transpose of the restriction operator, $I^p_{p-1} = (I^{p-1}_p)^{\mathrm{T}}$.

The second way of applying multigrid to the non-linear set of governing equations is to use the coarse grid correction (CGC) multigrid technique on the linearized problem obtained at each Newton iteration (Eq. (17)). This methodology, sometimes referred as "Newton-multigrid", is given (using the dual iteration strategy) as follows:

- Outer non-linear (nth) iteration. Iterate the discrete linear problem using any of the linearized element-Jacobi variants (LEJ or LGS) aforementioned:

$$\left[\frac{\partial \mathbf{R}_p}{\partial \mathbf{U}_p}\right]^n \Delta\mathbf{U}^{n+1}_p = \mathbf{S}_p - \mathbf{R}_p(\mathbf{U}^n_p). \tag{28}$$

  - Inner linear (kth) iteration. Solve for the fine-grid level correction $\mathbf{w}^k_p = \Delta\mathbf{U}^k_p$ with initial guess $\mathbf{w}^{k=0}_p = 0$:

$$[\mathbf{J}^k_p]\mathbf{w}^k_p = \mathbf{f}^k_p, \tag{29}$$

    where

$$[\mathbf{J}^k_p] = \left[\frac{\partial \mathbf{R}_p}{\partial \mathbf{U}_p}\right]^n, \quad \mathbf{f}^k_p = \mathbf{S}_\mathbf{p} - \mathbf{R}_p(\mathbf{U}^n_p). \tag{30}$$

  - Obtain the source term for the coarse level by restricting the linear residual $\mathbf{r}^k_p$:

$$\mathbf{f}^k_{p-1} = I^{p-1}_p \mathbf{r}^k_p, \quad \mathbf{r}^k_p = \mathbf{f}^k_p - [\mathbf{J}_p]\mathbf{w}^k_p. \tag{31}$$

  - Solve the *coarse grid correction* problem with initial guess $\Delta\mathbf{w}^k_{p-1} = 0$:

$$[\mathbf{J}^k_{p-1}]\Delta\mathbf{w}^k_{p-1} = \mathbf{f}^k_{p-1}, \quad [\mathbf{J}^k_{p-1}] \text{ is a subset of } [\mathbf{J}^k_p]. \tag{32}$$

  - Prolongate the coarse grid correction and update the fine-grid correction:

$$\Delta\mathbf{U}^{k+1}_p = \Delta\mathbf{U}^k_p + I^p_{p-1}\Delta\mathbf{w}^k_{p-1}. \tag{33}$$

- Fine-grid non-linear update:

$$\mathbf{U}^{n+1}_p = \mathbf{U}^n_p + \Delta\mathbf{U}^{k+1}_p. \tag{34}$$

In this implementation the basis set is hierarchical beginning at $p = 1$. Therefore, the Jacobian, $[\mathbf{J}^k_{p-1}] = [\partial\mathbf{R}_p(\mathbf{U}_p)/\partial\mathbf{U}_p]^n$, (and its inverse) represents a subset of $[\mathbf{J}^k_{p\geqslant 2}]$ and requires no additional operator for its construction. Once the $p = 1$ level is reached the state variable, $\mathbf{U}_p$, and the residual, $\mathbf{R}_p(\mathbf{U}_p)$, are restricted to $p = 0$, via two different operators defined as follows:

$$\mathbf{U}_h = \frac{1}{3}\sum_{i=1}^{3}\mathbf{U}^i_{p=1}, \quad \mathbf{R}_h(\mathbf{U}_h) = \sum_{i=1}^{3}\mathbf{R}_p(\mathbf{U}^i_{p=1}), \tag{35}$$

where $\{i = 1,\ldots,3\}$ is the modal index corresponding to $p = 1$. Therefore, in the case of h-multigrid (i.e., $p = 0$), the fine grid problem becomes $\mathbf{R}_h(\mathbf{U}_h) = \mathbf{S}_h$, and both FAS and CGC algorithms are obtained in a similar fashion, with the exception of the $[\mathbf{J}^k_h] = [\mathbf{J}^k_{p=0}]$ term, in the case of CGC algorithm, which needs to be evaluated once at every non-linear nth step for all h-levels as $[\mathbf{J}^k_h] = [\partial\mathbf{R}_h(\mathbf{U}_h)/\partial\mathbf{U}_h]^n$, where the restriction of the state variable and its residual to a coarse level, H, is obtained as:

$$\mathbf{U}_H = \frac{1}{A_H}\sum_{k=1}^{N_h}(\mathbf{U}_h A_h)_k, \quad \mathbf{R}_H(\mathbf{U}_H) = \sum_{k=1}^{N_h}(\mathbf{R}_h(\mathbf{U}_h))_k, \tag{36}$$

where N_h is the number of elements used in the agglomeration, A_h is the fine level elemental area, and $A_H = \sum_{k=1}^{N_h}(A_h)_k$ is the coarse level area. This two-level multigrid can be easily extended to a multi-level scheme.

For robustness it is important to augment the resulting multi-level hp-multigrid with a full multigrid (FMG) technique, in order to provide a good initial guess for the fine level problem. Moreover, the use of FMG is critically important in the case of the CGC scheme for it is known that the Newton iteration will diverge if the initial guess is not close enough to the final solution. In our hp-multigrid approach, the solution process begins at the coarsest grid level ($p = 0$), using all the h-levels available, and ends at the fine level where all the p- and h-levels are used to advance to solution to the desired accuracy, as depicted in Fig. 15. Alternatively, the FMG strategy can be initiated at the coarsest h-level, but no advantage over the latter approach was found, at least for the inviscid grids/problem considered. This will be further investigated in a future work pertaining viscous flows.

Results are presented for both, FAS and CGC, multigrid methods in order to asses their performance. Unless otherwise stated, all the simulated results are obtained via FMG using 5 V-cycles per level, starting at $p = 0$ level with uniform freestream initial conditions.

6.2. Channel flow over a bump

In the context of hp-multigrid methodology, the first case considered is of a compressible channel flow over a bump, with the flow and geometrical parameters as defined in Section 5.

6.2.1. Non-linear (FAS) hp-multigrid scheme

Fig. 16 illustrates the convergence rate of the residual as a function of the non-linear (FAS) hp-multigrid cycles for various p-order discretizations for a fixed mesh resolution ($N = 2015$), using a multigrid V-cycle with 10 linear element Jacobi smoothing passes on each grid level, including the agglomerated levels. While p-

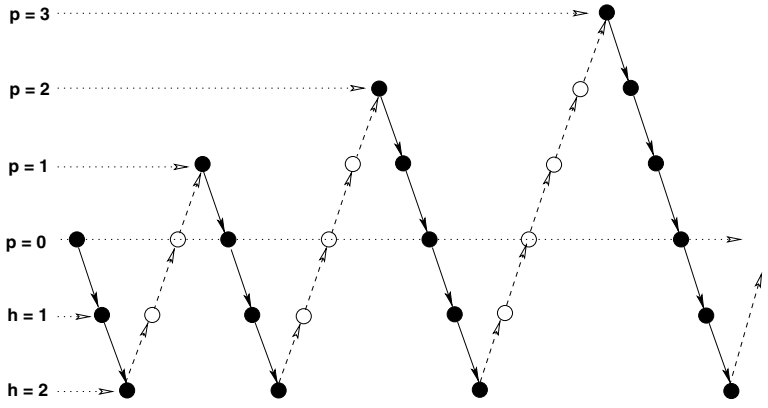

Fig. 15. Full hp-multigrid (FMG) levels for $p = 3$ and $h = 2$ (–, restriction; --, prolongation; ●, smoothing; ○, update).

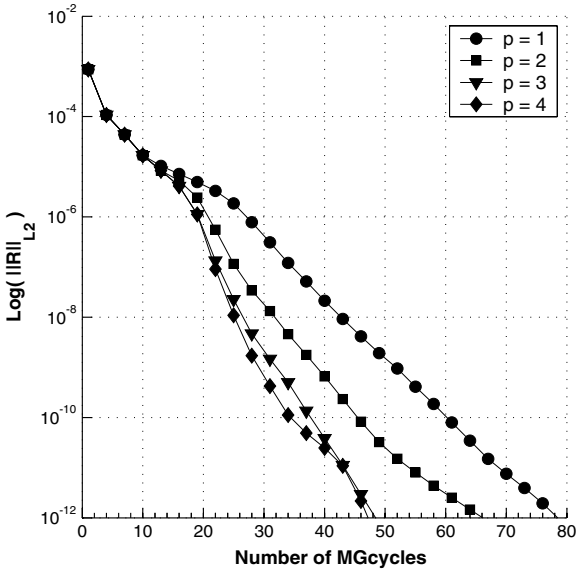

Fig. 16. The full *hp*-multigrid convergence vs. the number of multigrid (MG) cycles, on a mesh size of $N = 4093$ elements and various orders (p), for the channel bump problem.

independent convergence rates are expected, since the Jacobi smoother was shown to be p-independent, convergence actually accelerates slightly with increasing p. Note that although the convergence rate increases, the cost of the higher p discretizations is substantially higher per cycle, due to the higher number of degrees of freedom and larger block matrices involved.

In Fig. 17, the convergence rates for a fixed discretization ($p = 4$) are compared on the various grids for the bump configuration. In all cases, convergence to machine accuracy is achieved in 50 multigrid cycles or less, and only a slight h-dependence is observed (i.e., the $N = 505$ case requires 39 cycles, while the $N = 4093$ case

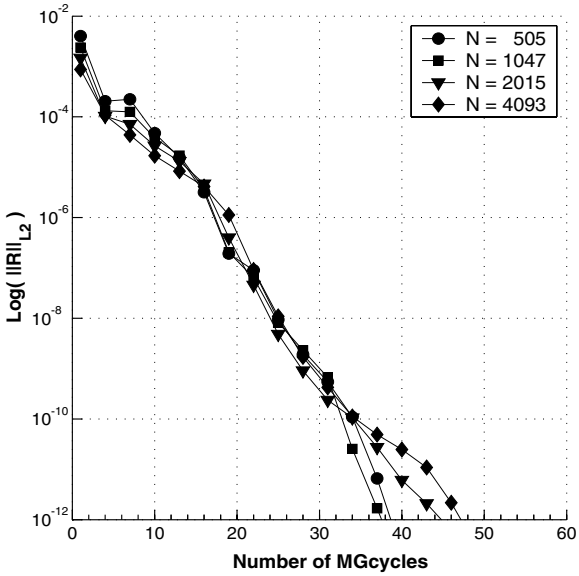

Fig. 17. The full *hp*-multigrid convergence vs. the number of multigrid (MG) cycles, on various fine grid problem sizes and order $p = 4$, for the channel bump problem.

requires 47 cycles). Note that the largest case $N = 4093$ involves a total of 10 multigrid levels, 5 levels from $p = 4$ to $p = 0$, and 5 h-agglomerated levels. The average convergence rates values are given in Table 2.

Fig. 18 illustrates the increases in solution efficiency for the $p = 4$, $N = 4093$ channel bump flow case using linearized element Jacobi as a solver without multigrid (LEJ, 1-level), as a smoother within p-multigrid (LEJ, pMG) and hp-multigrid (LEJ, hpMG), and using a linearized element Gauss–Seidel hp-multigrid approach (LGS, hpMG). While the p-multigrid approach shows a twofold speed-up, the hp-multigrid results in an order of magnitude increase in solution efficiency over the single grid approach. Furthermore, the Gauss–Seidel driven multigrid approach results in an additional efficiency improvement of 40%.

6.2.2. Linear (CGC) hp-multigrid scheme

The linear multigrid approach is based on the use of a Newton scheme to solve the non-linear Euler equations, as given by Eq. (17). Each cycle of this Newton scheme produces a large coupled linear problem, which is solved with the linear multigrid approach. Newton's method provides quadratic convergence of the non-linear problem, provided a sufficiently accurate linear problem solution is computed at each non-linear cycle. This is demonstrated in Fig. 19, for the case of the channel bump flow with $p = 4$ on the mesh of $N = 4093$ elements. A $p = 0$ solution is first obtained via an h-multigrid scheme using uniform free stream values as initial conditions. Then, this solution is used to initiate the full multigrid strategy starting at $p = 1$ level, with three non-linear cycles per FMG-level. In the so-called non-optimized linear multigrid (CGC) scheme, the linear problem is driven to machine zero at each non-linear cycle, as shown in the figure, and quadratic convergence of the non-linear system is obtained, as evidenced by the decreasing jumps in the linear residuals at each new non-linear cycle. In this case, the non-linear problem is solved to machine accuracy in five Newton

Table 2
Convergence rates on various fine grid sizes and order $p = 4$, for the channel bump problem

N	Average rate
505	0.50
1047	0.52
2015	0.52
4093	0.54

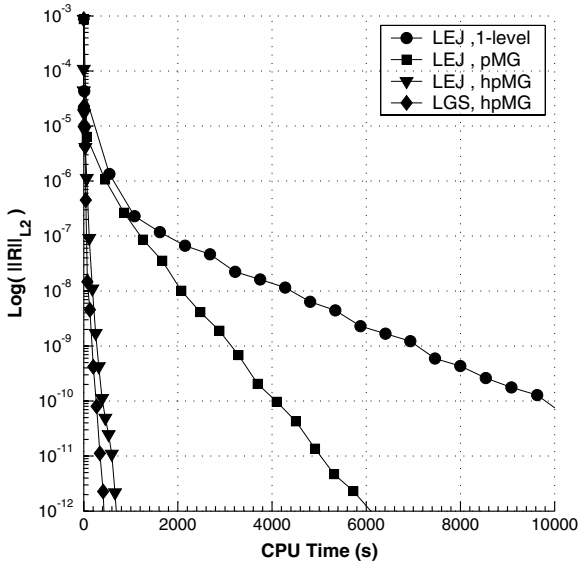

Fig. 18. Comparison of convergence of linear element-Jacobi (LEJ) and linear element Gauss–Seidel (LGS) in terms of CPU time, using one-level, p-multigrid (pMG) and hp-multigrid (hpMG) schemes, on a mesh size of $N = 4093$ elements and order $p = 4$, for the channel bump problem.

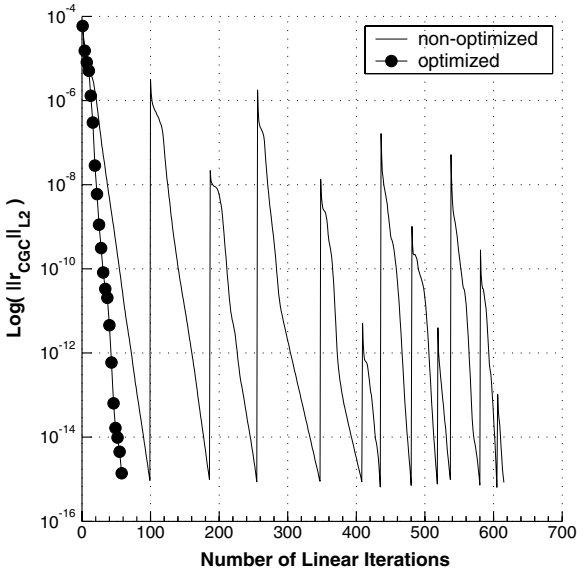

Fig. 19. The linear (CGC) *hp*-multigrid convergence vs. the number of linear iterations, on a mesh size of $N = 4093$ elements and order $p = 4$, for the channel bump problem. The full multigrid (FMG) has three cycles per level.

iterations per FMG level (only three shown). The advantage of this approach is that the computation of the non-linear residuals and Jacobians, which involve expensive quadrature integration procedures, needs only be performed five times through the entire solution procedure in this case. On the other hand, the total number of multigrid cycles in this approach is much larger than in the non-linear FAS multigrid approach, since driving the linear problem to machine accuracy in the initial Newton steps produces little gain in overall non-linear convergence. Therefore, this is referred to as the non-optimized CGC scheme. A simple strategy for optimizing the number of linear multigrid cycles within the Newton solution process is devised by terminating the linear system solution according to the criterion:

$$\|r^k_{\mathrm{cgc}}\|_{L2} \leqslant \frac{\|R^n\|_{L2}}{2^n}, \tag{37}$$

where r^k_{cgc} is the linear (iteration) residual, R^n is the non-linear residual, and k and n denote the current linear and non-linear iteration index, respectively. The linear iteration residual is obtained from Eq. (21) as

$$r^k_{\mathrm{cgc}} = \mathbf{S}_p - \mathbf{R}(\mathbf{U}^n_p) - [O^n]\Delta\mathbf{U}^k_p - [D^n]\Delta\mathbf{U}^k_p \tag{38}$$

in the case of linearized element-Jacobi, and from Eq. (22) as

$$r^k_{\mathrm{cgc}} = \mathbf{S}_p - \mathbf{R}(\mathbf{U}^n_p) - [U^n]\Delta\mathbf{U}^k_p - [(D+L)^n]\Delta\mathbf{U}^k_p \tag{39}$$

in the case of linearized element Gauss–Seidel. In Fig. 19 a dramatic reduction in the overall number of linear system cycles is observed when the optimized exit strategy is used, although the linear multigrid convergence rate is relatively unchanged. Fig. 20 provides a comparison of the CPU-time required by the optimized linear (CGC) multigrid scheme versus the non-linear (FAS) multigrid scheme, both using the same linearized element-Jacobi smoother, for the $N = 4093$, $p = 4$ channel bump case. The optimized linear (CGC) multigrid scheme is seen to reduce the overall CPU time by a factor of 5, due to the infrequent evaluations of the non-linear residuals, Jacobians and integration quadratures.

6.3. Flow over a four element airfoil

The next flow configuration is a flow over a four element airfoil of Suddhoo and Hall [37] at zero angle of attack with a freestream Mach number of $M_\infty = 0.25$. This constitutes a more complex configuration, which

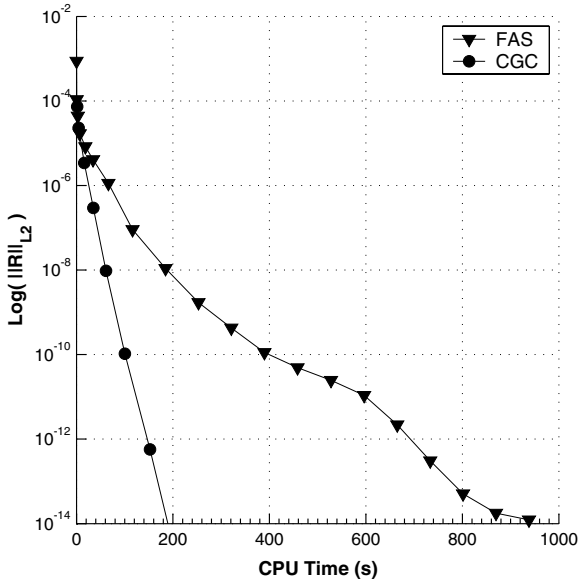

Fig. 20. The CPU time of linear (CGC) vs. non-linear (FAS) *hp*-multigrid, on a mesh size of $N = 4093$ elements and order $p = 4$, for the channel bump problem.

justifies the use of unstructured meshes. The full computational domain extends out to a radius of 10 chords lengths, where a chord length is defined as the span of the four element airfoil ensemble. The results are presented for the solution obtained via the linearized element Gauss–Seidel method.

Three meshes of differing resolution were constructed to study the *h*-dependence of the multigrid solution technique on this configuration. These meshes contain $N = 2142$, $N = 3856$, and $N = 5916$ elements, respectively. The agglomeration procedure was used to construct 4 coarse levels for the $N = 2142$ mesh, 5 coarse levels for the $N = 3856$ mesh, and 6 coarse levels for the $N = 5916$ mesh. An illustration of the second agglomerated level for the $N = 3856$ mesh is shown in Fig. 21.

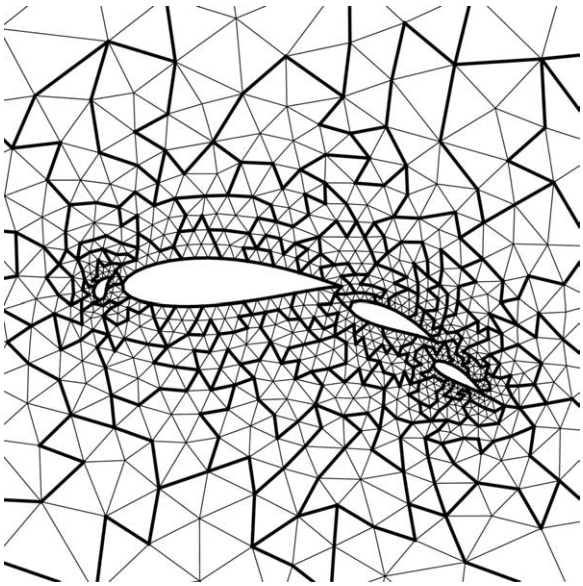

Fig. 21. A typical two level *h*-multigrid mesh configuration.

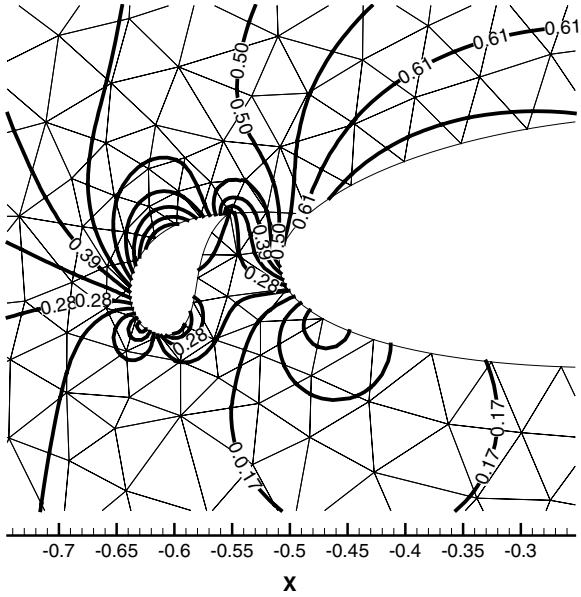

Fig. 22. Close-up of the Mach contour lines near the slat of the four element airfoil.

Similarly with the previous case, uniform free stream values are used as initial conditions for the full-multigrid strategy. For robustness, the full *hp*-multigrid, with five cycles per level (i.e., $p = 0 \ldots 3$) is used to drive the solution to the $p = 4$ level. Fig. 22 illustrates the computed Mach contours in the region of the leading edge of the main airfoil for the $N = 2142$ mesh using the $p = 4$ discretization.

6.3.1. Non-linear (FAS) hp-multigrid scheme

In Fig. 23, the convergence rate of the non-linear (FAS) *hp*-multigrid (using 10 Gauss–Seidel smoothing passes on each grid level of a multigrid V-cycle) scheme is shown for a fixed mesh size of $N = 3856$, for various

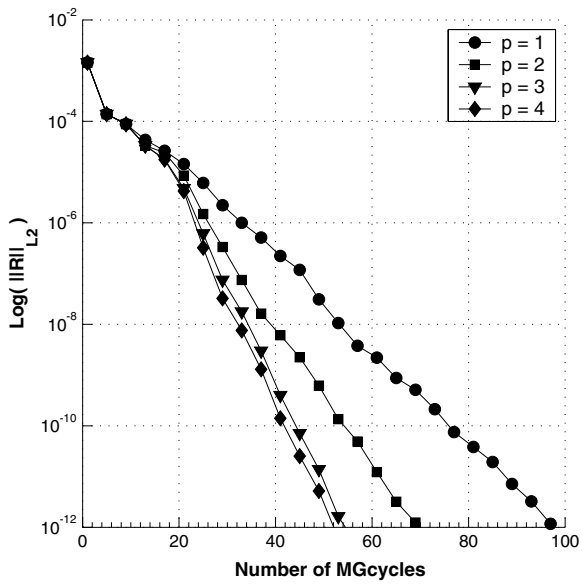

Fig. 23. The full *hp*-multigrid convergence vs. the number of multigrid (MG) cycles, on a mesh size of $N = 5916$ elements and various orders (p), for the four-element airfoil problem.

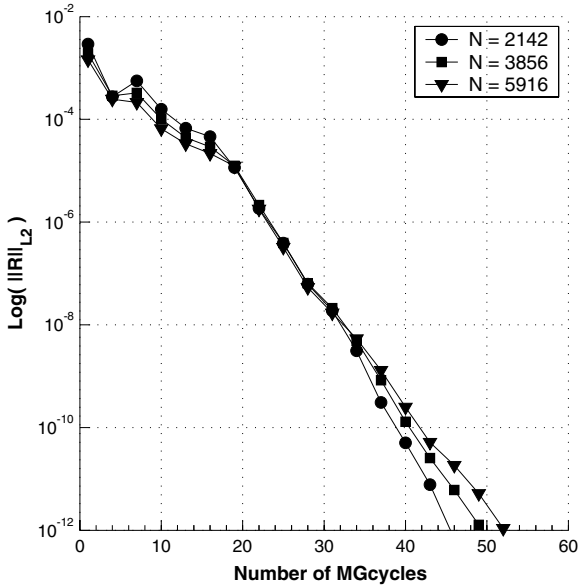

Fig. 24. The full *hp*-multigrid convergence vs. the number of multigrid (MG) cycles, on various fine grid problem sizes and order $p = 4$, for the four-element airfoil problem.

p discretizations. As in the previous case, the convergence rate increases slightly with higher order-accurate discretizations (although the cost of a multigrid cycle increases substantially with p). Fig. 24 shows the convergence rate of the $p = 4$ discretization on the various grids for the four-element airfoil configuration. For all cases, the residuals are reduced more than nine orders of magnitude in approximate 50 multigrid cycles, where the first 20 iteration are actually the FMG. The multigrid rates degrade slightly with increasing mesh resolution (h-dependence) toward the machine zero residual norm values, since the finest mesh requires 52 cycles to achieve the same residual level as that achieved in 46 cycles with the coarsest mesh. The scheme is very nearly h-independent, with average convergence rates given in Table 3.

Figs. 25 and 26 examine the effectiveness of the h-agglomeration multigrid strategy for the $N = 5916$ finest mesh problem. In Fig. 25, the steady-state solution for $p = 4$ on this mesh is computed using the p-multigrid procedure alone, using 10 linear Gauss–Seidel smoothing cycles on all levels, including the $p = 0$ level (pMG, 10/10). This is compared with a calculation employing 200 smoothing cycles on the $p = 0$ level at each multigrid cycle for better convergence (pMG, 10/200), and with the hp multigrid procedure, employing 10 smoothing cycles on all levels, including the h-agglomerated levels (hpMG, 10/10). The convergence of the original p-multigrid scheme is seen to degrade with respect to the hp-multigrid scheme, due to inadequate convergence of the $p = 0$ problems at each cycle. This is remedied by the scheme using more $p = 0$ smoothing cycles, which delivers slightly faster convergence on a multigrid cycle basis than the hp-multigrid scheme. However, as shown in Fig. 26, the additional $p = 0$ smoothing passes increase the cost of the multigrid cycle over the more efficient hp-multigrid scheme, resulting in a loss of efficiency on a CPU-time basis. In this case, the efficiency gain of the hp-multigrid method is moderate, and the number of coarse level $p = 0$ smoothing passes in the p-multigrid scheme has not been optimized. However, for finer meshes, the advantage of the hp-multigrid

Table 3
Convergence rates on various fine grid sizes and order $p = 4$, for the four-element airfoil problem

N	Average rate
2142	0.61
3856	0.62
5916	0.63

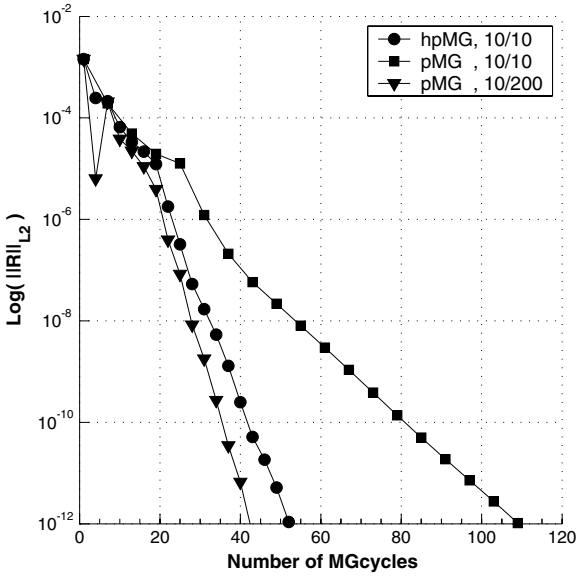

Fig. 25. Comparison of convergence of *p*-multigrid (pMG) and *hp*-multigrid (hpMG) in terms of multigrid (MG) cycles, on a mesh size of $N = 5916$ elements and order $p = 4$, for the four element airfoil problem.

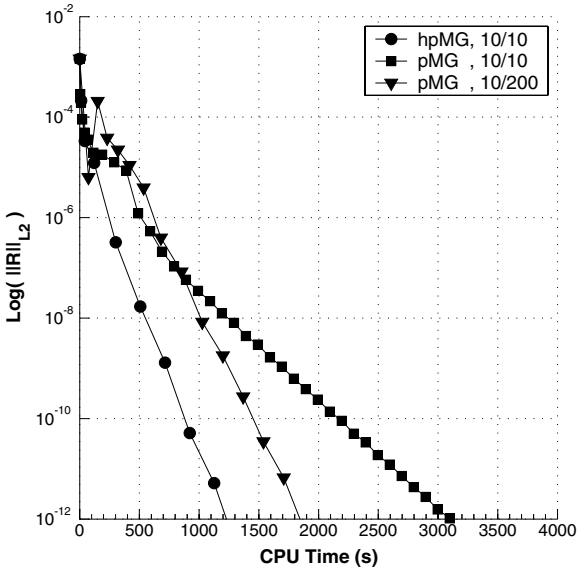

Fig. 26. Comparison of convergence of *p*-multigrid (pMG) and *hp*-multigrid (hpMG) in terms of CPU time, on a mesh size of $N = 5916$ elements and order $p = 4$, for the four-element airfoil problem.

scheme can be expected to increase, as the *p*-multigrid alone scheme will not scale appropriately with h, since more and more coarse $p = 0$ iterations will be required to maintain adequate convergence on the coarsest level.

The *p*-multigrid can be also performed by skipping levels. Fig. 27 shows the convergence vs. the number of cycles obtained using all *p*-levels (i.e., $p = 4, 3, 2, 1, 0$), by skipping one-level (i.e., $p = 4, 2,, 0$) and jumping directly to the coarsest *p*-level (i.e., $p = 4, 0$) for the four-element airfoil problem on the mesh size of $N = 5916$ elements. While the number of iterations increases when skipping levels, the work per MG-cycle will decrease and one might expect an improvement in terms of overall CPU-time. However, Fig. 28 clearly shows that we

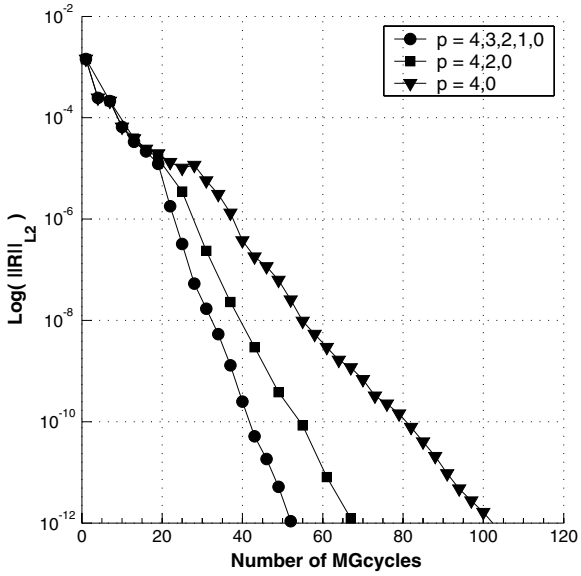

Fig. 27. Comparison of convergence of p-multigrid (pMG) by skipping p-levels in terms of multigrid (MG) cycles, on a mesh size of $N = 5916$ elements and order $p = 4$, for the four-element airfoil problem.

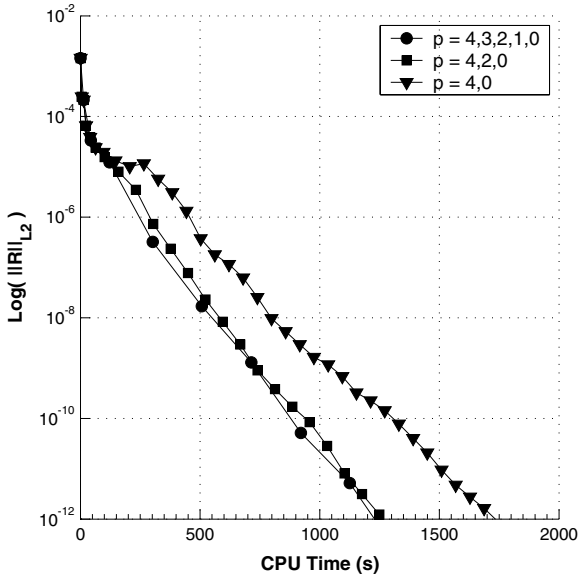

Fig. 28. Comparison of convergence of p-multigrid (pMG) by skipping p-levels in terms of CPU time, on a mesh size of $N = 5916$ elements and order $p = 4$, for the four-element airfoil problem.

obtained the same CPU-time when skipping one-level. Moreover, jumping directly to $p = 0$ level is the worst scenario in terms of both CPU-time and number of MG-cycles.

6.3.2. Linear (CGC) hp-multigrid scheme

The linear (CGC) *hp*-multigrid scheme is used to drive the Newton scheme for solving the four-element airfoil flow problem on the $N = 5916$ mesh using the $p = 4$ discretization, in Fig. 29. Similarly to the previous case, a $p = 0$ solution is first obtained via the h-multigrid scheme using uniform free stream values as initial conditions. Then, this solution is used to initiate the full multigrid strategy starting at the $p = 1$ level, with five

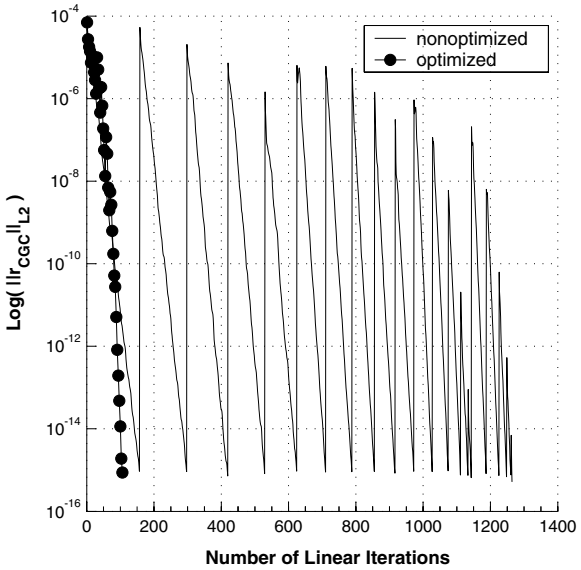

Fig. 29. The full linear (CGC) *hp*-multigrid convergence vs. the number of linear iterations, on a mesh size of $N = 5916$ elements and order $p = 4$, for the four-element airfoil problem.

cycles per FMG-level. For the non-optimized linear iteration strategy, where the linear residual (Eq. (39)) is converged to machine accuracy at each non-linear cycle, quadratic convergence is observed for the non-linear problem, although the increased complexity of this case requires a total of 10 Newton iterations per FMG level to reach machine accuracy (only five shown). Appropriate non-linear continuation techniques could be used to reduce the total number of non-linear cycles, such as *p*-continuation [11]. The convergence of the optimized linear (CGC) *hp*-multigrid scheme, using the linear iteration exit criterion of Eq. (37) together with the FMG strategy, is also displayed in Fig. 29, showing convergence to machine accuracy of the full non-linear problem in slightly more than 100 linear multigrid iterations (or 11 Newton updates). Fig. 30 provides a

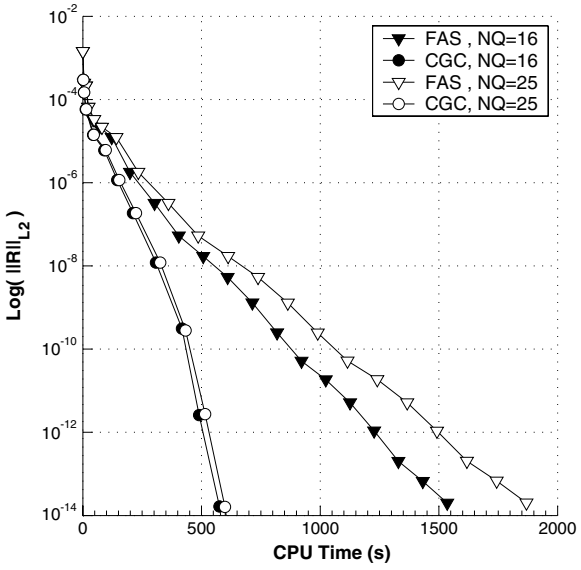

Fig. 30. The CPU time of linear (CGC) vs. non-linear (FAS) full *hp*-multigrid, using $NQ = 16$ and $NQ = 25$ number of quadrature points, on a mesh size of $N = 5916$ elements and order $p = 4$, for the four-element airfoil problem.

comparison of the total CPU time required to converge this problem using the optimized linear (CGC) *hp*-multigrid scheme, and the non-linear (FAS) *hp*-multigrid scheme, demonstrating an efficiency increase of nearly a factor of 3 for the linear multigrid approach over the non-linear approach. In order to demonstrate the advantage of the linear multigrid approach, the same comparison is reproduced in Fig. 30 but using a higher accuracy quadrature integration in the spatial discretization operator for both multigrid schemes. A total of $NQ = 25$ quadrature points are used on each triangle, which corresponds to over-integration for this $p = 4$ discretization (where previously $NQ = 16$ quadrature points where used). While the non-linear (FAS) multigrid solution cost increases by about 25%, the cost of the linear (CGC) multigrid solution is relatively unchanged, since the non-linear residual and thus quadrature evaluations are only performed 11 times in this approach (i.e., at the 11 Newton updates). While this level of quadrature accuracy has little effect on the final solution accuracy, it is instructive to demonstrate the advantages of the linear multigrid approach.

7. Concluding remarks

A high-order discontinuous Galerkin discretization using hierarchical basis functions on triangles has been developed and implemented using a *hp*-multigrid approach. Non-linear element-Jacobi, as well as linearized element-Jacobi and Gauss–Seidel schemes are used as smoothers on each level of the multigrid sequence. The linearized smoothers require additional storage, but are generally more efficient than their non-linear counterparts. The *hp*-multigrid scheme demonstrates both *p*-independent and *h*-independent convergence rates. The coupling of *p*- and *h*-multigrid procedures, through the use of agglomerated coarse levels for unstructured meshes, increases the overall solution efficiency compared to a *p*-alone multigrid procedure, and the benefits of the *hp*-multigrid approach can be expected to increase for finer meshes. The multigrid procedure can itself be applied as a non-linear solver, or as a linear solver for a Newton scheme applied to the non-linear problem. The linear multigrid approach demonstrates superior overall efficiency, provided a suitable linear iteration termination strategy is employed. The linear approach results in a solution strategy which is relatively insensitive to the cost of the non-linear residual construction, including the cost of the quadrature integration procedure used in the spatial discretization. This is significant, since considerable effort has been spent devising quadrature-free discretization constructions [38] or collocation methods [39] in order to reduce the cost of the spatial residual operator. On the other-hand, the linear approach is most appropriate for steady-state or implicit time-integration problems, where relatively few non-linear residual evaluations are required. Additionally, for memory constrained problems (particularly in three dimensions), the additional storage of the linear schemes may prove to be impractical. Future work will concentrate on extending these techniques to the Navier–Stokes equations and into the three-dimensional setting using hybrid element topologies.

Acknowledgment

This work was supported by a grant from the Office of Naval Research ONR Grant No. N00014-04-1-0602.

References

[1] P. Lesaint, P.A. Raviart, in: C. de Boor (Ed.), On a Finite Element Method to Solve the Neutron Transport Equation, Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, New York, NY, 1974.
[2] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (6) (1998) 2440–2463.
[3] T.C. Warburton, I. Lomtev, Y. Du, S.J. Sherwin, G.E. Karniadakis, Galerkin and discontinuous Galerkin spectral/*hp* methods, Comput. Methods Appl. Mech. Eng. 175 (1999) 343–359.
[4] B. Cockburn, G. Karniadakis, C.-W. Shu, Discontinuous Galerkin methods: theory, computation and applications, in: Lecture Notes in Computational Science A, Springer, New York, NY, 1999, pp. 69–224.
[5] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, SIAM J. Sci. Comput. 16 (3) (2001) 173–261.
[6] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138 (1997) 251–285.
[7] G.E. Karniadakis, S.J. Sherwin, Spectral/hp Element Methods for CFD, Oxford University Press, New York, NY, 1999.

[8] T.J.R. Hughes, A. Brooks, Streamline upwind-Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Meth. Appl. Mech. Eng. 32 (1982) 199–259.
[9] B. Helenbrook, D.J. Mavriplis, H. Atkins, Analysis of "p"-multigrid for continuous and discontinuous finite element discretizations, in: Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, 2003, AIAA Paper 2003-3989.
[10] E.M. Ronquist, A.T. Patera, Spectral element multigrid I: formulation and numerical results, SIAM J. Sci. Comput. 2 (4) (1987) 389–406.
[11] K.J. Fidkowski, D.L. Darmofal, Development of a higher-order solver for aerodynamic applications, in: Proceedings of the 42nd Aerospace Sciences Meeting and Exihibit, Reno, NV, 2004, AIAA Paper 2004-0436.
[12] F. Bassi, S. Rebay, Numerical solution of Euler equations with a multiorder discontinuous finite element method, in: K.S.S. Armfield, P. Morgan (Eds.), Proceedings of the Second International Conference on Computational Fluid Dynamics, Sydney, Australia, Springer, Berlin, 2002, pp. 199–204.
[13] K.J. Fidkowski, T.A. Oliver, J. Lu, D.L. Darmofal, p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 207 (2005) 92–113.
[14] J.W. Lottes, P.F. Fischer, Hybrid multigrid/Schwarz algorithms for the spectral element method, SIAM J. Sci. Comput. 24 (2005).
[15] P. Rasentarinera, M.Y. Hussaini, An efficient implicit discontinuous spectral Galerkin method, J. Comput. Phys. 172 (2001) 718–738.
[16] J. Heys, T.A. Manteufell, S.F. McCormick, L.N. Olson, Algebraic multigrid for higher-order finite elements, J. Comput. Phys. 204 (2005) 520–532.
[17] S.F. Davis, Simplified second-order Godunov-type methods, SIAM J. Sci. Statist. Comput. 9 (3) (1988) 445–473.
[18] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[19] A. Harten, P.D. Lax, B. Van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1) (1983) 35–61.
[20] F.E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Applied Mechanics, Springer, New York, NY, 1999.
[21] P. Batten, N. Clarke, C. Lambert, D.M. Causon, On the choice of wavespeeds for the HLLC Riemann solver, SIAM J. Sci. Comput. 18 (2) (1997) 1553–1570.
[22] P. Batten, M.A. Leschiner, U.C. Goldberg, Average-state jacobians and implicit methods for compressible viscous and turbulent flows, J. Comput. Phys. 137 (1997) 38–78.
[23] P. Solin, P. Segeth, I. Zel, High-Order Finite Element Methods, Studies in Advanced Mathematics, Chapman & Hall, London, 2003.
[24] B. Szabo, I. Babuska, Finite Element Analysis, Wiley, New York, NY, 1991.
[25] M. Dubiner, Spectral methods on triangles and other domains, SIAM J. Sci. Comput. 6 (1991) 345–390.
[26] D.A. Dunavant, High degree efficient symmetrical gaussian guadrature rules for the triangle, Int. J. Numer. Meth. Eng. 21 (1985) 1129–1148.
[27] D.A. Dunavant, Economical symmetrical quadrature rules for complete polynomials over a square domain, Int. J. Numer. Meth. Eng. 21 (1985) 1777–1784.
[28] B. Cockburn, S. Hou, C.-W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (545) (1990) 545–581.
[29] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[30] L. Demkowicz, Projection-based interpolation, ICES Report 04-03, The University of Texas at Austin, 2003.
[31] L. Demkowicz, I. Babuska, Optimal p interpolation error estimates for edge finite elements of variable order in 2D, TICAM Report 01-11, The University of Texas at Austin, 2001.
[32] J.J.W. van der Vegt, H. van der Ven, Slip flow boundary conditions in discontinuous Galerkin discretizations of the Euler equations of gas dynamics, Technical Report, National Aerospace Laboratory NLR, 2002.
[33] D.J. Mavriplis, An assessment of linear versus non-linear multigrid methods for unstructured mesh solvers, J. Comput. Phys. 175 (2002) 302–325.
[34] D.J. Mavriplis, Unstructured mesh generation and adaptivity, in: VKI Lecture Series VKI-LS 1995-02, 1995.
[35] U. Trottenberg, A. Schuller, C. Oosterlee, Multigrid, Academic Press, London, UK, 2000.
[36] D.J. Mavriplis, V. Venkatakrishnan, Agglomeration multigrid for two dimensional viscous flows, Comput. Fluids 24 (5) (1995) 553–570.
[37] A. Suddhoo, I. Hall, Test cases for the plane potential flow past multi-element airfoils, Aeronaut. J. 89 (1985) 403–414.
[38] H. Atkins, C.W. Shu, Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations, AIAA J. 36 (5) (1998) 775–782.
[39] J.S. Hesthaven, T. Warburton, Nodal high-order methods on unstructured grids I. Time-domain solution of Maxwell's equations, J. Comput. Phys. 181 (1) (2002) 186–221, ICASE Report No. 01-6.