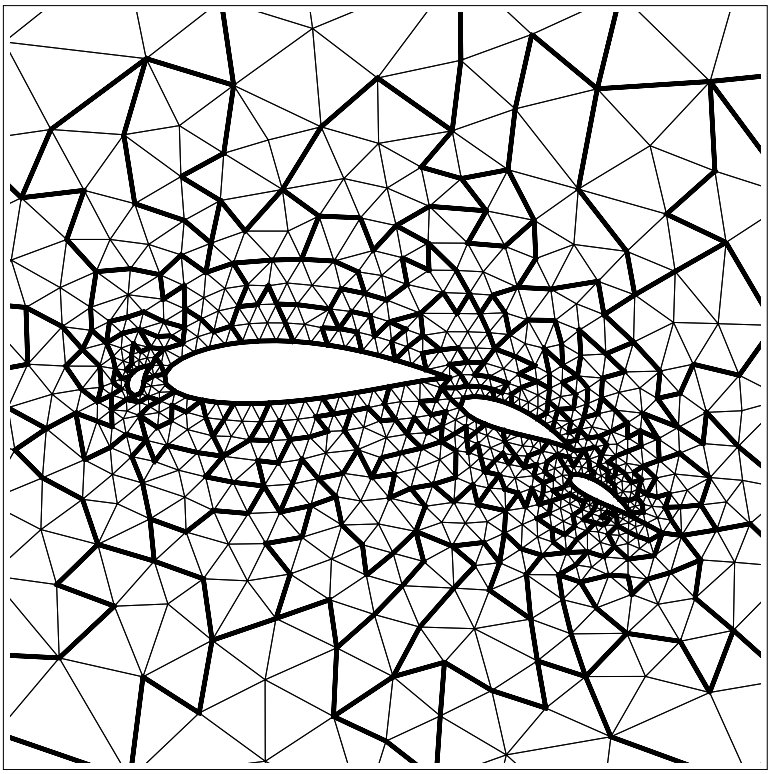
AIAA 2005–1268

High-Order Discontinuous Galerkin Methods using a Spectral Multigrid Approach

Cristian R. Nastase, Dimitri J. Mavriplis
Department of Mechanical Engineering
University of Wyoming, Laramie, Wyoming 82071-3295



43rd AIAA Aerospace Sciences Meeting and Exhibit
January 10–13, 2005/Reno, NV

High-Order Discontinuous Galerkin Methods using a Spectral Multigrid Approach

Cristian R. Nastase, * Dimitri J. Mavriplis †
Department of Mechanical Engineering
University of Wyoming, Laramie, Wyoming 82071-3295

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 inverted directly and all other entries are treated explicitly. The methodology is developed for the non-linear Euler equations, using both non-linear (FAS) and linear (CGC) multigrid schemes, and results are presented for the channel flow over a bump and a four element airfoil. Current results demonstrate convergence rates which are independent of the order of accuracy (*p*) of the discretization, with slight dependence on the level of mesh resolution (*h*).

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 $2^4 = 16$ 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[1,2,3,4,5,6] and Streamwise Upwind Petrov-Galerkin[7] 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.[8] The idea of spectral multigrid was originally proposed by Ronquist and Patera,[9] and has been pursued for the Euler and Navier-Stokes equations by Fidkowski et al.[10,11] with encouraging results. Implicit multi-level solution techniques for high-order discretizations have also been developed by Lottes and Fisher.[12]

In this work, we extend the original spectral multigrid approach described in Ref.[8] to the two-dimensional steady-state Euler equations, and couple the spectral *p*-multigrid approach with a more traditional agglomeration *h*-multigrid method for

*Postdoctoral Research Associate, Member AIAA; e-mail: nastase@uwyo.edu.
†Professor, AIAA Associate Fellow; e-mail: mavripl@uwyo.edu.

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.

Governing Equations

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

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} \tag{2}$$

and the Cartesian components of the inviscid flux $\mathbf{F} = (\mathbf{F}^x, \mathbf{F}^y)$ are:

$$\mathbf{F}^x = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \mathbf{F}^y = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ v(E+p) \end{pmatrix} \tag{3}$$

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

$$p = (\gamma - 1)\left[E - \frac{1}{2}\rho(u^2+v^2)\right] \tag{4}$$

where γ is the ratio of specific heats.

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

$$\mathbf{U}(\mathbf{x},t) \approx \mathbf{U}_p(\mathbf{x},t) = \sum_{j=1}^{M} \mathbf{u}_j(t)\phi_j(\mathbf{x}) \tag{5}$$

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[2,3,4,6] 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:

$$\int_{\Omega_k} \phi_i \left[\frac{\partial \mathbf{U}_p(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}_p)\right]_k d\Omega_k = 0 \tag{6}$$

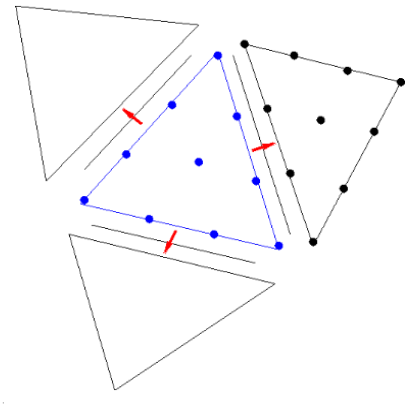


Fig. 1 Discontinuous solution representation illustrating element based modes and inter-element flux evaluations.

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

$$\int_{\Omega_k} \phi_i \frac{\partial \mathbf{U}_p}{\partial t} d\Omega_k - \int_{\Omega_k} \nabla\phi_i \cdot \mathbf{F}(\mathbf{U}_p)\, d\Omega_k + \int_{\partial\Omega_k} \phi_i \mathbf{F}^*(\mathbf{U}_p)\cdot\mathbf{n}\, d(\partial\Omega_k) = 0 \tag{7}$$

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 (Figure 1).

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. Current implementations include the flux difference splitting schemes of Rusanov[13] (Eq. (8)), Roe[14] (Eq. (9)) and HLLC[15,16,17] (Eq. (10)), all of which are summarized below:

$$\mathbf{F}^*_{Rusanov} = \frac{1}{2}[\mathbf{F}(\mathbf{U}_p^-) + \mathbf{F}(\mathbf{U}_p^+) + |\lambda_{max}|(\mathbf{U}_p^- - \mathbf{U}_p^+)] \tag{8}$$

$$\mathbf{F}^*_{Roe} = \frac{1}{2}[\mathbf{F}(\mathbf{U}_p^-) + \mathbf{F}(\mathbf{U}_p^+) + \mathbf{R}\Lambda\mathbf{L}(\mathbf{U}_p^- - \mathbf{U}_p^+)] \tag{9}$$

$$\mathbf{F}^*_{HLLC} = \frac{1}{2}[\mathbf{F}(\mathbf{U}_p^-) + \mathbf{F}(\mathbf{U}_p^+) + (\mathbf{S}^- - |\mathbf{S}_M|)\mathbf{U}^{*-} + (\mathbf{S}^+ + |\mathbf{S}_M|)\mathbf{U}^{*+} - \mathbf{S}^-\mathbf{U}_p^- - \mathbf{S}^+\mathbf{U}_p^+] \tag{10}$$

where $\mathbf{S}$ is the acoustic wave velocity, $\mathbf{U}^*$ is an average intermediate state, $\mathbf{R}$ and $\mathbf{L}$ are the left and right eigenvectors, and λ_{max} is the maximum value of the eigenvalues Λ.

The discrete form of the local discontinuous Galerkin formulation is defined by the particular choice of the set of basis

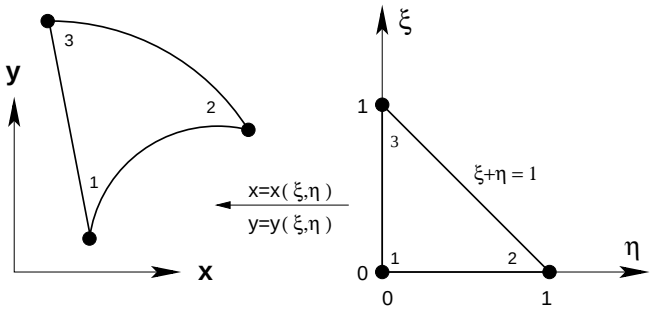


Fig. 2 Mapping between the curvilinear triangle, $\Omega_k(x,y)$, and the standard triangle, $\widehat{\Omega}_k(\xi,\eta)$.

functions, $\{\phi_i, i = 1 \ldots M\}$. The basis set is defined on a standard triangle $\widehat{\Omega}(\xi,\eta)$ spanning between $\{0 < \xi,\eta < 1\}$, as illustrated in Figure 2.

We seek a set of hierarchical basis functions in order to simplify our subsequent spectral multigrid implementation. Defining the first order Lagrange polynomials:

$$L_1 = 1-\xi-\eta, \qquad L_2 = \xi, \qquad L_3 = \eta \tag{11}$$

the hierarchical basis set, $\{\phi_i, i = 1 \ldots 10\}$, is fully described by *vertex*,

$$\phi_1^v = L_1, \quad \phi_2^v = L_2, \quad \phi_3^v = L_3 \tag{12}$$

edge,

$$\begin{aligned} \phi_{n+2}^e &= L_1 L_2 \psi_n(L_2 - L_1) \\ \phi_{n+2}^e &= L_2 L_3 \psi_n(L_3 - L_2) \\ \phi_{n+2}^e &= L_3 L_1 \psi_n(L_1 - L_3) \end{aligned} \tag{13}$$

and *bubble*,

$$\phi_{n1,n2}^b = L_1 L_2 L3 \psi_{n1-1}(L_2 - L_1)\psi_{n2-1}(L_1 - L_3) \tag{14}$$

shape functions, where $n = p-2$, $n \geq 0$, $n1+n2 = p-1$ and $n1, n2 \geq 1$. The full description of the basis function set is given in Ref.[18,19] The hierarchy of vertex, edge and bubble modes is guaranteed by the Lobatto functions $\psi(x)$ which take zero values at the end of their definition interval. 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 is given by:

$$\mathbf{x}_p = \sum_{j=1}^{M} \widehat{\mathbf{x}}_j \phi_j(\xi,\eta) \tag{15}$$

while the solution expansion becomes:

$$\mathbf{U}_p(x,y,t) = \sum_{j=1}^{M} \widehat{\mathbf{u}}_j(t)\phi_j(\xi,\eta) \tag{16}$$

In the case of straight-sided elements the mapping is linear and its Jacobian

$$J_k(\xi,\eta) = \left| \frac{\partial(x,y)}{\partial(\xi,\eta)} \right| \tag{17}$$

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. Here we use curved-sided elements for boundary elements only, and straight-sided elements for all other (interior) elements. For the general case, the weak statement within each element, $\widehat{\Omega}_k$, becomes:

$$\begin{aligned} \int_{\widehat{\Omega}_k} \phi_i \frac{\partial \mathbf{U}_p}{\partial t} J_k d\widehat{\Omega}_k - \int_{\widehat{\Omega}_k} \nabla\phi_i \cdot \mathbf{F}(\mathbf{U}_p) J_k d\widehat{\Omega}_k \\ + \int_{\partial\widehat{\Omega}_k} \phi_i \mathbf{F}^*(\mathbf{U}_p)\cdot\mathbf{n} J_k d(\partial\widehat{\Omega}_k) = 0 \end{aligned} \tag{18}$$

This set of equations is solved in the modal space and the integrals are evaluated by Gaussian quadrature rules, which requires a projection of the solution values to the quadrature points used in the numerical integration. The element integral uses Gauss-Legendre quadrature rules within the standard triangle. The boundary integral uses Gauss-Legendre-Lobatto quadrature rules of corresponding order. For boundary elements with curved edges, 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.

The Implicit Steady State Solver

Neglecting the temporal derivative term, the system of equations (Eq. (18)) associated with each element becomes:

$$\mathbf{R}(\mathbf{U}_p) = 0 \tag{19}$$

where $\mathbf{R}(\mathbf{U}_p)$ is the residual. We use variants of the element-Jacobi scheme to solve this system of equations. The Newton iteration associated with Eq. (19) yields at each step "n":

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}_p}\right]^n \Delta\mathbf{U}_p^{n+1} = -\mathbf{R}(\mathbf{U}_p^n) \tag{20}$$

$$\mathbf{U}_p^{n+1} = \mathbf{U}_p^n + \Delta\mathbf{U}_p^{n+1} \tag{21}$$

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.

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 diagonal matrix ($[D]$) contains 40×40 entries for each element. The non-linear iteration Eq. (20) becomes:

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

A second variant of this solver is denoted as the linearized element Jacobi method. In this approach, the full Jacobian matrix is retained, but is decomposed into block diagonal $[D]$ and off-diagonal $[O]$ components:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{U}_p}\right]^n = [D^n] + [O^n] \tag{23}$$

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 equation (20). In matrix form the $(k+1)^{th}$ step of the linearized element Jacobi step is written as:

$$\Delta \mathbf{U}_p^{k+1} = [D^n]^{-1}\left(-\mathbf{R}(\mathbf{U}_p^n) - [O^n]\Delta \mathbf{U}_p^k\right) \qquad (24)$$

Note that the linearized element Jacobi scheme involves a dual iteration strategy, where each n^{th} 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. (22)), both approaches can be expected to converge at the same rates per cycle (asymptotically).[20] 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. (24) 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 becomes:

$$\Delta \mathbf{U}_p^{k+1} = ([D+L]^n)^{-1}\left(-\mathbf{R}(\mathbf{U}_p^n) - [U^n]\Delta \mathbf{U}_p^k\right) \qquad (25)$$

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.[21]

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.

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. Figure 3 shows the grid configuration and the Mach contour lines for a freestream Mach number of $M_\infty = 0.2$. The full domain extends from $-2 \le x \le 2$ in the stream-wise direction and from $0 \le y \le 2$ 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

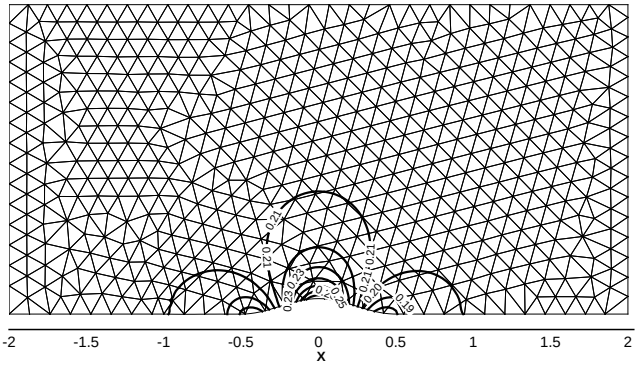


Fig. 3 Solution contours on a domain of $N = 1047$ elements and $p = 4$.

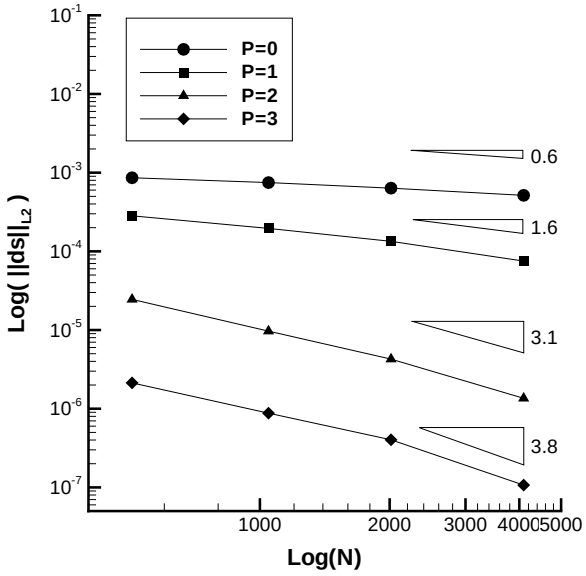


Fig. 4 The L_2 norm of the entropy error as a function of h/p-refinement.

can be assessed by measuring the L_2 norm of the entropy error for this isentropic flow. The entropy error is defined as $ds = s - s_\infty$ where s_∞ is the free stream entropy. The L_2 norm of a quantity w is defined as:

$$||w||_{L_2} = \sqrt{\int_\Omega w^2 d\Omega} \qquad (26)$$

where Ω represents the entire computational domain. Figure 4 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 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

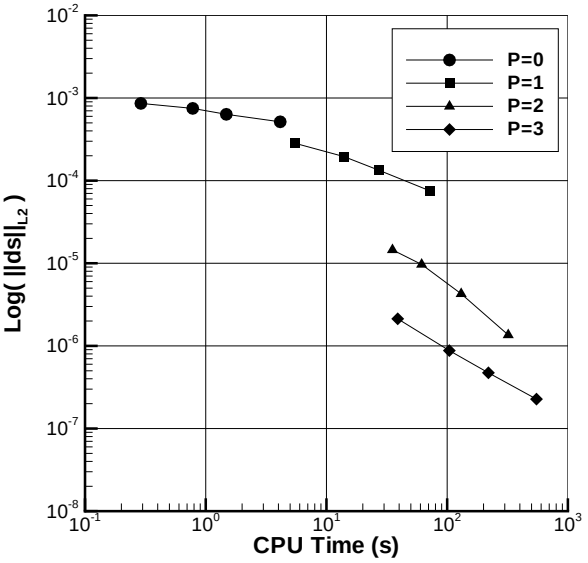


Fig. 5 The L_2 norm of the entropy error as a function of CPU time.

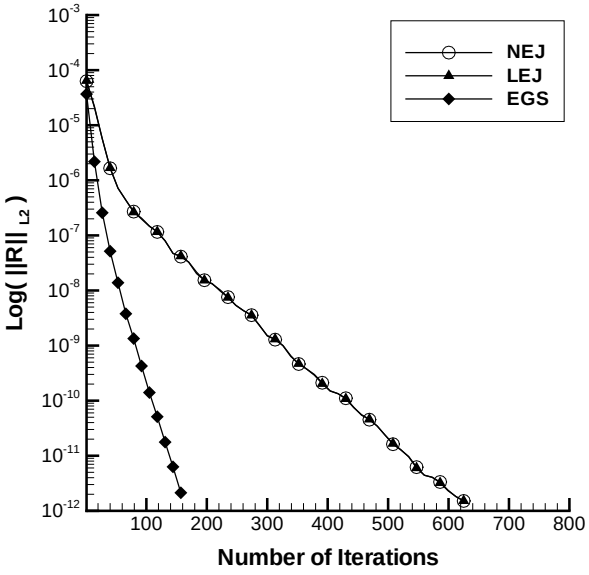


Fig. 6 Comparison of convergence of non-linear element-Jacobi (NEJ), linear element-Jacobi (LEJ), and linear element Gauss-Seidel (EGS) in terms of iterations on a mesh size of $N = 2015$ elements and order $p = 4$.

is given in Figure 5, 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.

Figure 6 depicts the convergence of the 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 element-Jacobi and linearized element-Jacobi schemes converge at similar rates in terms of numbers of cycles, while the element Gauss-Seidel scheme converges substantially faster. When compared in terms of CPU time, the linearized element-Jacobi and element Gauss-Seidel schemes are seen to be substantially more efficient than the 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.

Due to the superior efficiency of the linearized schemes, the remaining results will make exclusive use of these schemes. Figure 8 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

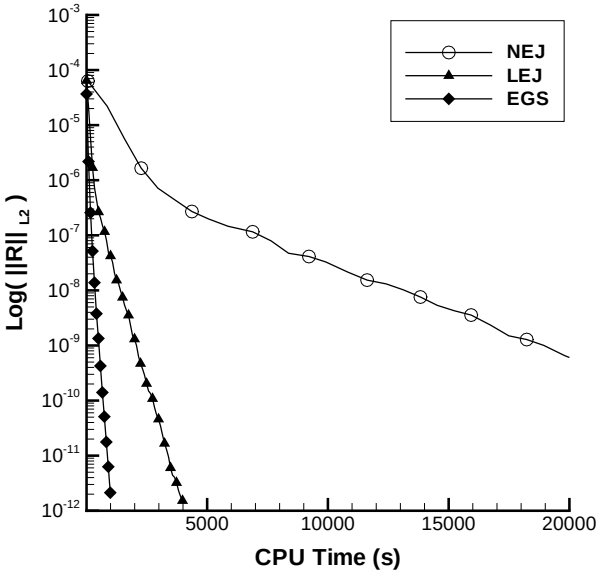


Fig. 7 Comparison of convergence of non-linear element-Jacobi (NEJ), linear element-Jacobi (LEJ), and linear element Gauss-Seidel (EGS) in terms of CPU time on a mesh size of $N = 2015$ elements and order $p = 4$.

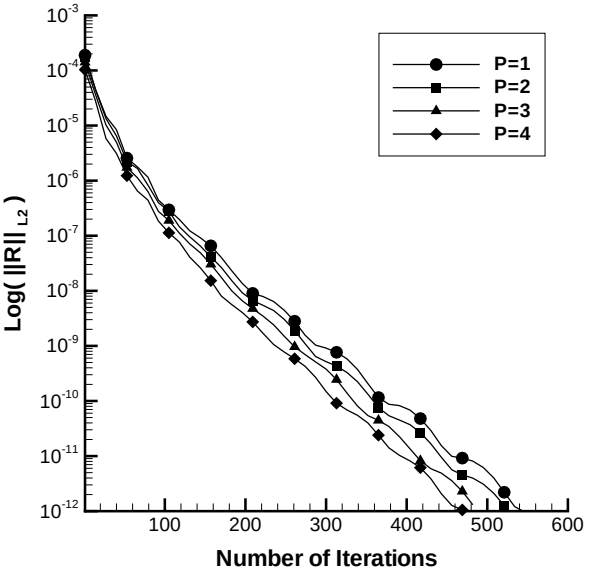


Fig. 8 The L_2 norm of the residual *vs.* number of linear element-Jacobi (LEJ) cycles on a fixed mesh size of $N = 1047$ elements and various orders (p).

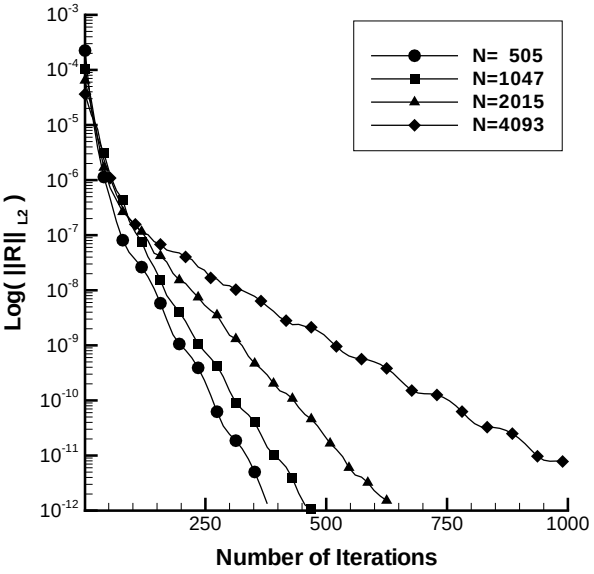


Fig. 9 The L_2 norm of the residual *vs.* number of linear element-Jacobi (LEJ) cycles for a fixed order $p = 4$.

the number of elements, N, has an adverse effect on the convergence rate. In Figure 9, the convergence rate for $p = 4$ is seen to 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.

The hp-Multigrid Approach

Multigrid methods are known as efficient techniques for accelerating convergence to steady state for both linear and non-linear problems,[22,20] 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.

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.[20] 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.[10] 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.[22] 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,

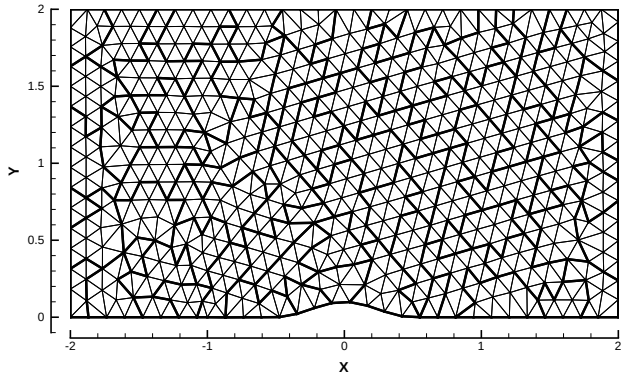


Fig. 10 A typical two level *h*-multigrid mesh configuration.

formed by merging together neighboring fine grid level 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.[23] This procedure has the potential of resulting in a truly *h*- and *p*-independent solution strategy for high-order accurate discontinuous Galerkin discretizations.

Figure 10 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 via the Full Approximation Storage (FAS) scheme. The second is to use the Coarse Grid Correction (CGC) multigrid technique on the linearized problem obtained at each Newton iteration (Eq. (20)). Results are presented for both methods in order to asses their performance.

Non-linear (FAS) hp-multigrid scheme

Figure 11 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*-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 Figure 12, 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

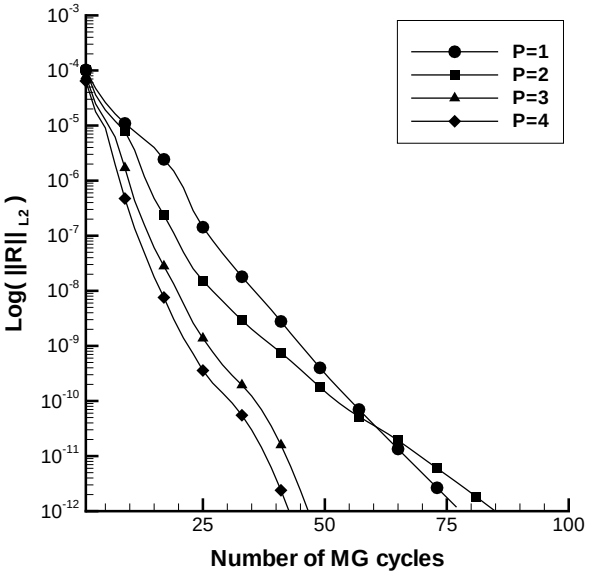


Fig. 11 The *hp*-multigrid convergence *vs.* the number of multigrid (MG) cycles, on a mesh size of $N = 2015$ elements and various orders (*p*), for the channel bump problem.

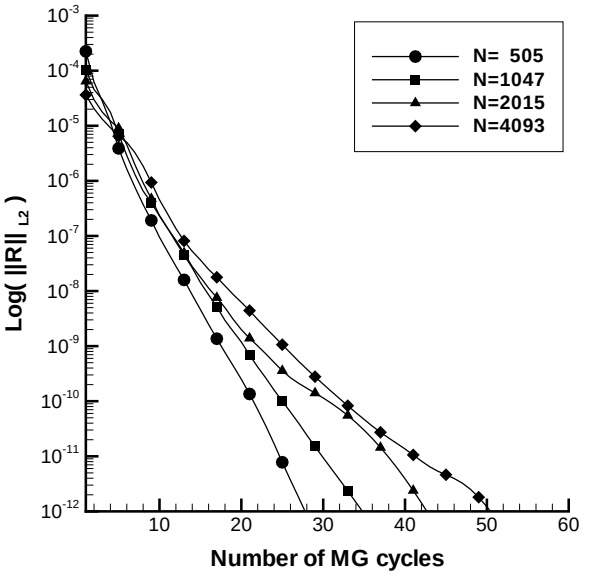


Fig. 12 The *hp*-multigrid convergence *vs.* the number of multigrid (MG) cycles, on various fine grid problem sizes and order $p = 4$, for the channel bump problem.

is achieved in 50 multigrid cycles or less, and only a slight h-dependence is observed (i.e. the $N = 505$ case requires 30 cycles, while the $N = 4093$ case requires 50 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.

Figure 13 illustrates the increases in solution efficiency for the $p = 4$, $N = 4093$ channel bump flow case using lin-

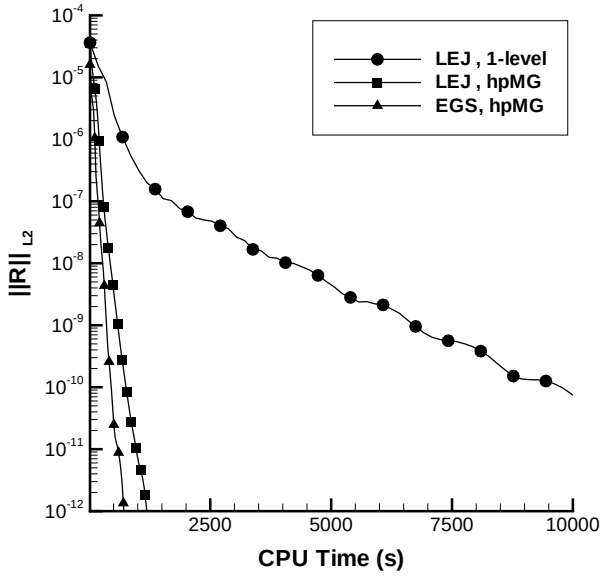


Fig. 13 Comparison of convergence of linear element-Jacobi (LEJ) and linear element Gauss-Seidel (EGS) in terms of CPU time, using one-level and *hp*-multigrid (hpMG) schemes, on a mesh size of $N = 4093$ elements and order $p = 4$, for the channel bump problem.

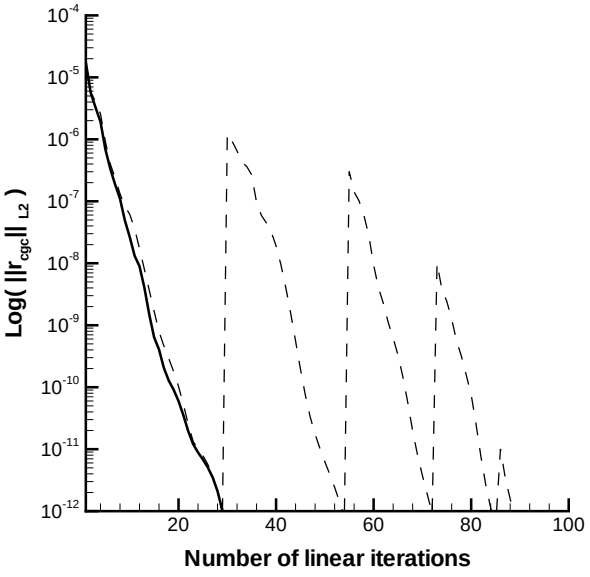


Fig. 14 The linear (CGC) *hp*-multigrid convergence *vs.* the number of linear iterations, on a mesh size of $N = 4093$ elements and order $p = 4$, for the channel bump problem (– non-optimized, - - optimized).

earized element Jacobi as a solver without multigrid, as a smoother within *hp*-multigrid, and using a linearized element Gauss-Seidel *hp*-multigrid approach. While the multigrid approach results in an order of magnitude increase in solution efficiency over the single grid approach, the Gauss-Seidel driven multigrid approach results in an additional efficiency improvement of 40%.

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 equation (20). 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 Figure 14, for the case of the channel bump flow with $p = 4$ on the mesh of $N = 4093$ elements. In the so-called non-optimized linear multigrid (CGC) scheme, the linear problem is driven to machine zero at each non-linear cycle, as seen 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 iterations. The advantage of this approach, is that the computation of the non-linear residuals and Jacobians, which involve expensive quadrature integration procedures, need 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:

$$||r^k_{cgc}||_{L2} \leq \frac{||R^n||_{L2}}{2^n} \tag{27}$$

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. In the case of linear element-Jacobi the linear iteration residual is obtained from Eq. (24) as:

$$r^k_{cgc} = -\mathbf{R}(\mathbf{U}^n_p) - [O^n]\Delta\mathbf{U}^k_p - [D^n]\Delta\mathbf{U}^k_p \tag{28}$$

In Figure 14 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. Figure 15 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 more than a factor of 3, due to the infrequent (*i.e.* 5) evaluations of the non-linear residuals, Jacobians and integration quadratures.

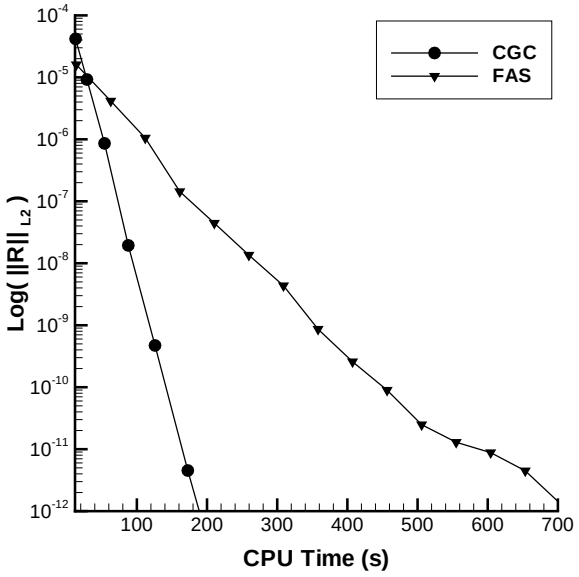


Fig. 15 The CPU time of linear (CGC) *vs.* non-linear (FAS) *hp*-multigrid, on a mesh size of $N = 4093$ elements and order $p = 4$, for the channel bump problem.

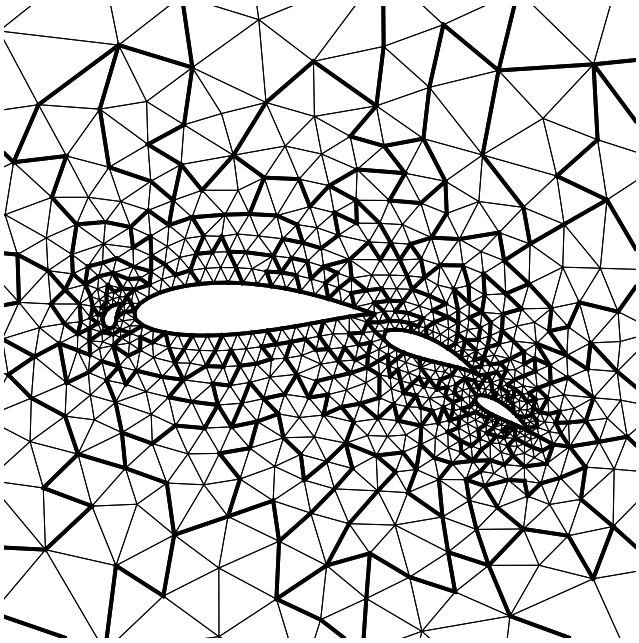


Fig. 16 A typical two level *h*-multigrid mesh configuration.

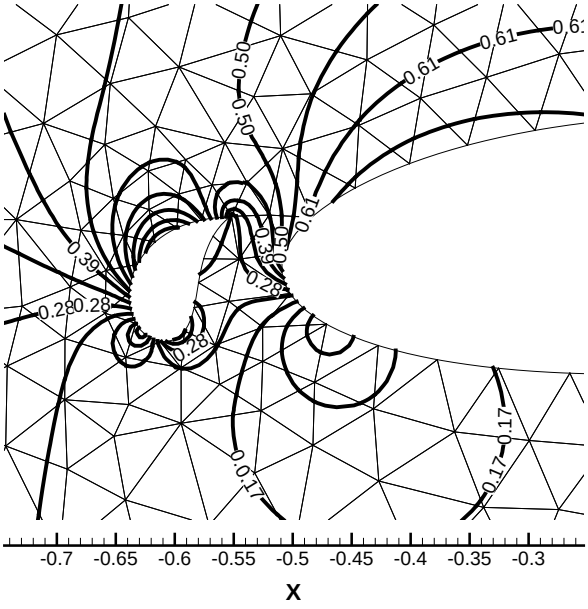


Fig. 17 Close-up of the Mach contour lines near the slat of the four element airfoil.

Flow over a four element airfoil

Non-linear (FAS) hp-multigrid scheme

The next flow configuration is a flow over a four element airfoil of Suddhoo and Hall[24] at zero angle of attack with a freestream Mach number of $M_\infty = 0.3$. This constitutes a more complex configuration, which 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 Gauss-Seidel method.

Three meshes of differing resolution were constructed to study the *h*-dependence of the multigrid solution technique on this configuration. The three meshes contain $N = 1539$, $N = 3055$, and $N = 5918$, elements respectively. The agglomeration procedure was used to construct 3 coarse levels for the $N = 1539$ mesh, 4 coarse levels for the $N = 3055$ mesh, and 5 coarse levels for the $N = 5918$ mesh. An illustration of the second agglomerated level for the $N = 3055$ mesh is shown in Figure 16. Similarly with the previous case, a $p = 0$ solution is first obtained via a *h*-multigrid scheme using uniform free stream values as initial condition. Then, this solution is used to initialize the flow for higher approximation orders. Figure 17 illustrates the computed Mach contours in the region of the leading edge of the main airfoil for the $N = 1539$ mesh using the $p = 4$ discretization.

In Figure 18, the convergence rate of the non-linear (FAS) *hp*-multigrid (using 5 Gauss-Seidel smoothing passes on each grid level of a multigrid V-cycle) scheme is shown for a fixed mesh size of $N = 3055$, for various *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*). Figure 19 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 9 orders of magnitude in under 75 multigrid cycles. The multigrid rates degrade slightly with increasing mesh resolution (*h*-dependence), since the finest mesh requires 65 cycles to achieve the same residual level as that achieved in 50 cycles with the coarsest mesh.

Figures 20 and 21 examine the effectiveness of the h-agglomeration multigrid strategy for the $N = 5918$ finest

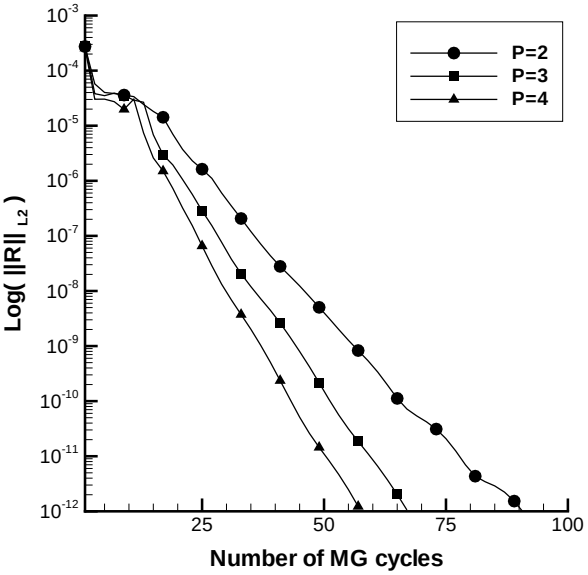


Fig. 18 The *hp*-multigrid convergence *vs.* the number of multigrid (MG) cycles, on a mesh size of $N = 3055$ elements and various orders (p), for the four element airfoil problem

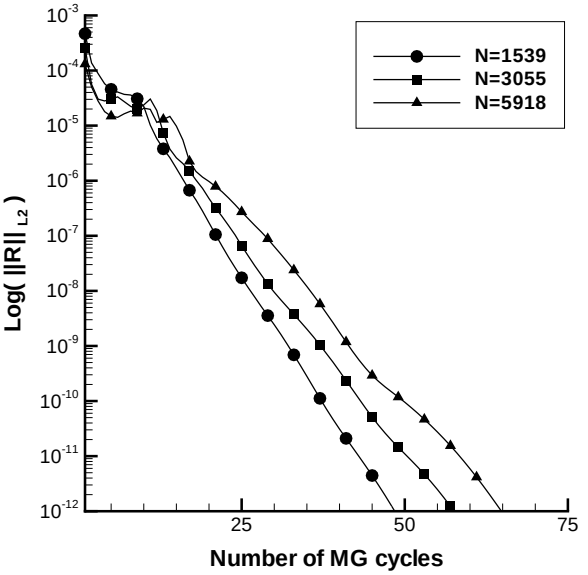


Fig. 19 The *hp*-multigrid convergence *vs.* the number of multigrid (MG) cycles, on various fine grid problem sizes and order $p = 4$, for the four element airfoil problem.

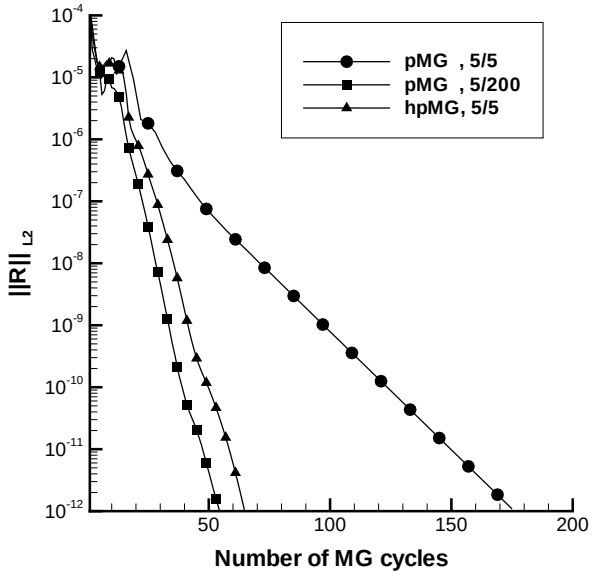


Fig. 20 Comparison of convergence of p-multigrid (pMG) and hp-multigrid (hpMG) in terms of multigrid (MG) cycles, on a mesh size of $N = 5918$ elements and order $p = 4$, for the four element airfoil problem.

mesh problem. In Figure 20, the steady-state solution for $p = 4$ on this mesh is computed using the p-multigrid procedure alone, using 5 linear Gauss-Seidel smoothing cycles on all levels, including the $p = 0$ level. This is compared with a calculation employing 200 smoothing cycles on the $p = 0$ level at each multigrid cycle for better convergence, and with the hp multigrid procedure, employing 5 smoothing cycles on all levels, including the h-agglomerated levels. 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 Figure 21, 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 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.

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 $N = 3055$ mesh using the $p = 4$ discretization, in Figure 22. For the non-optimized linear iteration strategy, where the linear residual is converged to machine accuracy

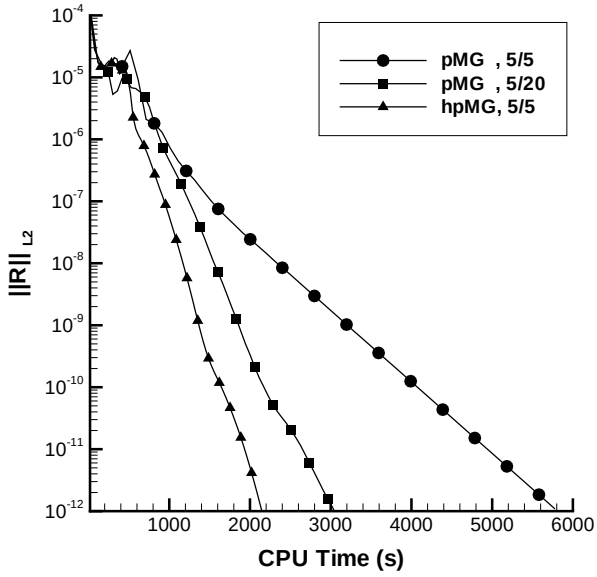


Fig. 21 Comparison of convergence of p-multigrid (pMG) and hp-multigrid (hpMG) in terms of CPU time, on a mesh size of $N = 5918$ elements and order $p = 4$, for the four element airfoil problem.

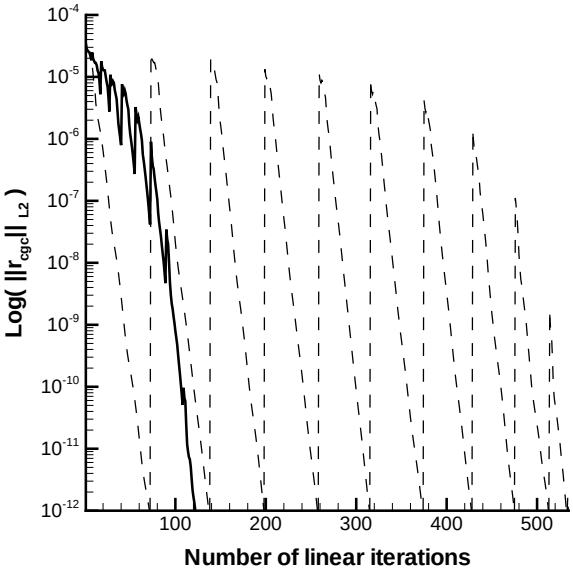


Fig. 22 The linear (CGC) hp-multigrid convergence *vs.* the number of linear iterations, on a mesh size of $N = 3055$ elements and order $p = 4$, for the four element airfoil problem (– non-optimized, - - optimized).

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 to reach machine accuracy. Appropriate non-linear continuation techniques could be used to reduce the total number of non-linear cycles, such as p-continuation.[10] The convergence of the optimized linear (CGC) hp-multigrid scheme, using the linear iteration exit criterion of Eq. (27) is also displayed in Figure 22, showing convergence to machine accuracy of the full non-linear problem in slightly more than 100 linear multigrid cycles. Figure 23 provides a 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 4 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 Figure 23 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 10 times in this approach (at the 10 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.

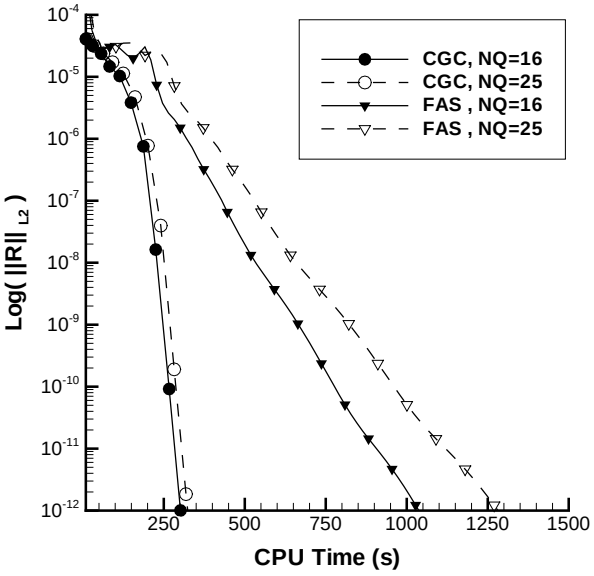


Fig. 23 The CPU time of linear (CGC) *vs.* non-linear (FAS) hp-multigrid, using $NQ = 16$ and $NQ = 25$ number of quadrature points, on a mesh size of $N = 3055$ elements and order $p = 4$, for the four element airfoil problem.

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 *p*-independent and nearly *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[25] or collocation methods[26] 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 different element types.

Acknowledgments

This work was supported by a grant from the Office of Naval Research ONR grant number N00014-04-1-0602.

References

[1]Lesaint, P. and Raviart, P. A., *On a Finite Element Method to Solve the Neutron Transport Equation*, Mathematical Aspects of Finite Elements in Partial Differential Equations, C. de Boor, Academic Press, New York, NY, 1974.

[2]Bassi, F. and Rebay, S., "High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations," *J. Comput. Phys.*, Vol. 138, 1997, pp. 251–285.

[3]Cockburn, B. and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM J. Numer. Anal.*, Vol. 35, No. 6, 1998, pp. 2440–2463.

[4]Warburton, T. C., Lomtev, I., Du, Y., Sherwin, S. J., and Karniadakis, G. E., "Galerkin and Discontinuous Galerkin Spectral/*hp* Methods," *Comput. Methods Appl. Mech. Engrg.*, Vol. 175, 1999, pp. 343–359.

[5]Cockburn, B., Karniadakis, G., and Shu, C.-W., "Discontinuous Galerkin Methods: Theory, Computation and Applications," *Lecture Notes in Computational Science A*, Springer-Verlag, New York, NY, 1999, pp. 69–224.

[6]Cockburn, B. and Shu, C.-W., "Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *SIAM J. Sci. Comput.*, Vol. 16, No. 3, 2001, pp. 173–261.

[7]Hughes, T. J. R. and Brooks, A., "Streamline upwind-Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Comput. Meth. Appl. Mech. Engrg.*, Vol. 32, 1982, pp. 199–259.

[8]Helenbrook, B., Mavriplis, D. J., and Atkins, H., "Analysis of "p"-Multigrid for Continuous and Discontinuous Finite Element Discretizations," *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, 2003, AIAA Paper 2003–3989.

[9]Ronquist, E. M. and Patera, A. T., "Spectral Element Multigrid I: Formulation and Numerical Results," *Journal of Scientific Computing*, Vol. 2, No. 4, 1987, pp. 389–406.

[10]Fidkowski, K. J. and Darmofal, D. L., "Development of a Higher-Order Solver for Aerodynamic Applications," AIAA Paper 2004–0436.

[11]Fidkowski, K. J., Oliver, T. A., J.Lu, and Darmofal, D. L., "p-Multigrid Solution of High-Order Discontinuous Galerkin Discretizations of the Compressible Navier-Stokes Equations," submitted to Journal of Computational Physics.

[12]Lottes, J. and Fischer, P., "Hybrid Multigrid/Schwarz Algorithms for the Spectral Element Method," *Journal of Scientific Computing*, 2004.

[13]Davis, S. F., "Simplified Second-Order Godunov-Type Methods," *SIAM J. Sci. Statist. Comput.*, Vol. 9, No. 3, 1988, pp. 445–473.

[14]Roe, P. L., "Approximate Riemann Solvers, Parameter vectors, and Difference Schemes," *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.

[15]Toro, F. E., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Applied Mechanics, Springer-Verlag, New York, NY, 1999.

[16]Batten, P., Clarke, N., Lambert, C., and Causon, D. M., "On the Choice of Wavespeeds for the HLLC Riemann Solver," *SIAM J. Sci. Comput.*, Vol. 18, No. 2, 1997, pp. 1553–1570.

[17]Batten, P., Leschiner, M. A., and Goldberg, U. C., "Average-State Jacobians and Implicit Methods for Compressible Viscous and Turbulent Flows," *J. Comput. Phys.*, Vol. 137, 1997, pp. 38–78.

[18]Solin, P., Segeth, P., and Zel, I., *High-Order Finite Element Methods*, Studies in Advanced Mathematics, Chapman and Hall, 2003.

[19]Szabo, B. and Babuska, I., *Finite Element Analysis*, John Wiley & Sons, Inc., New York, NY, 1991.

[20]Mavriplis, D. J., "An Assessment of Linear versus Non-Linear Multigrid Methods for Unstructured Mesh Solvers," *J. Comput. Phys.*, Vol. 175, 2002, pp. 302–325.

[21]Mavriplis, D. J., "Unstructured Mesh Generation and Adaptivity," *VKI Lecture Series VKI-LS 1995-02*, mar 1995.

[22]Trottenberg, U., Schuller, A., and Oosterlee, C., *Multigrid*, Academic Press, London, UK, 2000.

[23]Mavriplis, D. J. and Venkatakrishnan, V., "Agglomeration Multigrid for Two Dimensional Viscous Flows," *Computers and Fluids*, Vol. 24, No. 5, may 1995, pp. 553–570.

[24]Suddhoo, A. and Hall, I., "Test cases for the plane potential flow past multi-element airfoils," *Aeronaut. J.*, Vol. 89, 1985, pp. 403–414.

[25]Atkins, H. and Shu, C. W., "Quadrature-Free Implementation of Discontinuous Galerkin Method for Hyperbolic Equations," *AIAA Journal*, Vol. 36, No. 5, 1998, pp. 775–782.

[26]Hesthaven, J. S. and Warburton, T., "Nodal High-orderMethods on Unstructured Grids I. Time-Domain Solution of Maxwell's Equations," *J. of Comp. Physics*, Vol. 181, No. 1, sep 2002, pp. 186–221, ICASE Report No.01-6.