# Numerical Evaluation of P-multigrid Method for the Solution of Discontinuous Galerkin Discretizations of Diffusive Equations

H. L. Atkins[*] and B. T. Helenbrook[†]

This paper describes numerical experiments with P-multigrid to corroborate analysis, validate the present implementation, and to examine issues that arise in the implementations of the various combinations of relaxation schemes, discretizations and P-multigrid methods. The two approaches to implement P-multigrid presented here are equivalent for most high-order discretization methods such as spectral element, SUPG, and discontinuous Galerkin applied to advection; however it is discovered that the approach that mimics the common geometric multigrid implementation is less robust, and frequently unstable when applied to discontinuous Galerkin discretizations of diffusion. Gauss-Seidel relaxation converges ≈ 40% faster than block Jacobi, as predicted by analysis; however, the implementation of Gauss-Seidel is considerably more expensive that one would expect because gradients in most neighboring elements must be updated. A compromise *quasi* Gauss-Seidel relaxation method that evaluates the gradient in each element twice per iteration converges at rates similar to those predicted for true Gauss-Seidel.

## I.   Introduction

The discontinuous Galerkin (DG) method has recently been applied to a range of fluid models from simple linear acoustics[1–4] to complete Navier-Stokes,[5–8] as well as related models in electromagnetics.[9,10] Much of the interest in DG stems from two inherent properties of the method. First, DG's finite-element underpinnings provides a rigorous and robust means of discretizing partial differential equations on unstructured grids; and the use of unstructured grids is widely thought to be the most effective approach to modeling complex geometries. Second, and in contrast to traditional finite-element methods, the use of local discontinuous basis functions results in a local mass matrix and leads to a compact discretization. These inherent properties of DG lead to a highly compact discretization that is well suited to propagation dominated problems that are efficiently resolved by explicit time marching methods.

Most of DG's advantageous features (accurate, robust, compact) persist when applied to partial differential equations having a diffusive or elliptic component, such as the Navier-Stokes equations. However, in such cases, the DG discretization becomes very stiff,[11] and the use of an explicit time marching method is expensive. As a consequence, the DG method with explicit time marching is non-competitive for most problems involving diffusion, and is intractable in cases having a broad range of spatial scales such as high Reynolds number flows. While DG has been successfully applied to the Navier-Stokes using explicit time marching, these instances have been limited to cases with low Reynolds numbers.[5–7,12]

Efforts to develop efficient implicit solvers for DG have included optimized relaxation schemes,[11] analysis of traditional geometric multigrid,[13] GMRES,[14–16] and P-multigrid.[17–20] Analysis of geometric multigrid indicates mesh that independent results are possible; however, the required restriction operators are complex and would be difficult to implement for general unstructured grids. A large effort in GMRES can be found in open literature; however the method incurs a large storage penalty, and none have achieved mesh independent convergence for high Reynolds number flows (or other highly stiff cases). P-multigrid is the most promising of these approaches, as well as the simplest to apply. The method provides mesh independent convergence and can be implemented with little storage or work overhead.

[*]Senior Research Scientist, NASA Langley Research Center, Hampton VA, AIAA Senior member
[†]Professor, Mechanical and Aeronautical Engineering Dept., Clarkson University, Potsdam, NY, member

This paper describes numerical experiments with P-multigrid that corroborate the analysis presented in the companion paper,[20] and validate the present implementation. This paper will also discuss attributes of the various combinations of relaxation schemes, discretization methods, and P-multigrid methods that become apparent when considering the implementation of these methods. The first section of this paper describes the model problem and several variations of the DG discretization that will be considered in this study. The second section describes block Jacobi and Gauss-Seidel relaxation schemes, and discusses issues that arise in the implementation of the Gauss-Seidel method. The third section describes P-multigrid and two approaches for its implementation. The last section presents numerical experiments with the various discretizations and relaxation schemes.

## II.   Model Problem, DG Discretizations

### A.   Model Problem

The Poisson equation is written as a set of first order equations,

$$\nabla^2 u = f(x) \quad \Rightarrow \quad \begin{array}{l} \vec{\sigma} - \nabla u = 0 \\ -\nabla \cdot \vec{\sigma} + f(x) = 0, \end{array} \tag{1}$$

defined on the domain $\Omega$, where $u$ denotes the solution to the Poisson equation, $x$ is a point in $\Omega$, $\vec{\sigma}$ is the gradient vector, and f(x) is a given forcing function.

The domain is partitioned into a collection of non-overlapping elements $\Omega_l$, where the subscript $l$ uniquely identifies an arbitrary element. Let $\Omega_n$ denote an arbitrary element neighboring to $\Omega_l$, let $\partial\Omega_{l,n}$ denote the edge segment between elements $\Omega_l$ and $\Omega_n$. For elements that lie on domain boundaries, let the boundary edge segments be denoted as $\partial\Omega_{l,b}$ when they need to be distinguished. Finally, let $\{e_l\}$ denote the set of all edge segments of $\Omega_l$, including any that may lie on the domain boundary.

### B.   DG Discretization

The following describes the discretizations in terms of a single arbitrary element. The solution in each element is approximated in terms of a set of local discontinuous basis functions $\phi = \{b_{l,i}\}$ as $u \approx u_l \equiv \sum_{\{i\}} b_{l,i} u_{l,i}$ and $\vec{\sigma} \approx \vec{\sigma}_l \equiv \sum_{\{i\}} b_{l,i} \vec{\sigma}_{l,i}$ in $\Omega_l$. (Note: $\phi$ is used to denote either a set of basis functions, or a column vector of basis functions depending on the context.) Typically, the local basis set for a given element is taken to be the complete set of polynomials of degree less that or equal to some value $p$ for points inside of the element, and zero for points outside the element. This choice of basis functions allows the discrete equations to be developed for a single element, instead of as a global summation of integrals as is commonly encountered in finite element methods. The governing equations are projected onto each member of the basis set and integrated by parts to obtain a set of equations in an arbitrary element that govern the set of unknown variables $\{u_{l,i}\}$ in that element.

$$\int_{\Omega_l} [b_{l,i}\vec{\sigma}_l + u_l \nabla b_{l,i}] \, d\Omega - \sum_{\{e_l\}} \int_{\partial\Omega_{l,n}} b_{l,i} h_1(u_l, u_n)\vec{n} ds \quad = \quad 0 \qquad \forall \{i\} \tag{2}$$

$$\int_{\Omega_l} [\nabla b_{l,i}\vec{\sigma}_l + f(x)] \, d\Omega - \sum_{\{e_l\}} \int_{\partial\Omega_{l,n}} b_{l,i} h_2(u_l, \vec{\sigma}_l, u_n, \vec{\sigma}_n) \cdot \vec{n} ds \quad = \quad 0 \qquad \forall \{i\}. \tag{3}$$

$\vec{n}$ is the outward unit normal. The integrands of the integrals over element boundaries are replaced by numerical fluxes that account for the discontinuous nature of the solution there. There are several variations of DG discretizations; however, it is shown in reference 21 that most can be written in a unified form, and that all differ only by the way in which the numerical fluxes are defined.

The present work will focus on two forms: the local discontinuous Galerkin method[22] (LDG), and the interior penalty method[23, 24] (IP). They are defined by:

$$\begin{aligned} \text{LDG}: \qquad & h_1(u_l, u_n) = \{\!|u|\!\} - \beta_{l,n}[\![u]\!], \\ & h_2(u_l, \vec{\sigma}_l, u_n, \vec{\sigma}_n) = \{\!|\vec{\sigma}|\!\} + \beta_{l,n}[\![\vec{\sigma}]\!] - \alpha_j \vec{n}[\![u]\!], \end{aligned} \tag{4}$$

American Institute of Aeronautics and Astronautics

$$\text{IP}: \qquad h_1(u_l, u_n) = \{\!|u|\!\},$$
$$h_2(u_l, \vec{\sigma}_l, u_n, \vec{\sigma}_n) = \{\!|\nabla u|\!\} - \alpha_j \vec{n}[\![u]\!], \tag{5}$$

where $\{\!|w|\!\} \equiv (w_l + w_n)/2$ denotes the average of a quantity, and $[\![w]\!] \equiv w_n - w_l$ denotes the jump of a quantity. $\beta_{l,n}$ is a parameter in the range $-\frac{1}{2} \le \beta_{l,n} \le \frac{1}{2}$ that effectively allows the operator to switch from a symmetric centered operator ($\beta_{l,n} = 0$), to a one-sided operator ($\beta_{l,n} = \pm\frac{1}{2}$ with $\beta_{l,n} = -\beta_{n,l}$). The term $\alpha_j$ is a numerical penalty term that is required for stability when $\beta_{l,n} = 0$, and it is defined as $\alpha_j \equiv \eta h^{-1}$, where $\eta \ge 0$ and $h$ is a mesh scale. For the IP method, $\eta$ must be increased when the order $p$ is increased in order to maintain stability. Discretizations similar to LDG($\beta_{l,n} = 0$) and IP, introduced by Brezzi et al.[25] and Bassi et al.[6] respectively, are obtained by replacing $\alpha_j[\![u]\!]$ with an integral function of the solution jump $\alpha_r([\![u]\!])$. This form automatically scales with order to give the required stabilization for any $p$, although a user specified coefficient is usually included in the formulation. LDG is usually applied in one of two distinct forms: either $\beta_{l,n} = 0$ and $\eta > 0$, or $\beta_{l,n} = \pm\frac{1}{2}$ and $\eta = 0$. The former will be denoted as LDG-central and the later as LDG-one-sided.

## III.  Relaxation Schemes

For the purpose of discussing relaxation schemes, equations 2 and 3 are rewritten in the *primal* form obtained by eliminating $\vec{\sigma}_l$ from equation 3.

$$R(U) \equiv AU_l + \sum_{n \in \{n_l\}} B_n U_n = 0. \tag{6}$$

$A$ and $B_n$ are matrices, $U_l$ is a vector containing the solution coefficients $\{u_l\}$, and $\{n_l\}$ is the set of all elements in the stencil except $\Omega_l$. It is worth noting that, for the IP discretization, $\{n_l\}$ contains only the nearest neighboring elements, but for LDG-centered, $\{n_l\}$ additionally contains all the neighbors of the nearest neighbors. For the LDG-one-sided discretization on Cartesian grids, $\{n_l\}$ also contains only the nearest neighboring elements, but for non-Cartesian grids there is a weak coupling to some, but not all, of the neighbors of the nearest neighbors.

### A.  Block Jacobi

The block Jacobi (BJ) relaxation is defined as

$$U_l^{n+1} = U_l^n - \omega A^{-1} R(U^n). \tag{7}$$

An earlier analysis[11] of BJ applied to DG showed that BJ groups all of the eigenvalues to be less than two for all $p$, thus $\omega = 1$ is always stable; however, a slight under relaxation of 0.95 can improve the robustness in some cases. This will be discussed more in the next section.

### B.  Block Gauss-Seidel

Gauss-Seidel (GS) is a sweeping algorithm which can be written in a form similar to BJ. The only difference is that the $R(U)$ is evaluated just prior to the update, using most recent data available in all neighboring elements.

$$U_l^{n+1} = U_l^n - \omega A^{-1} R(U^n, U^{n+1}) \tag{8}$$

Although it would appear that GS could be implemented for the same cost as BJ, in fact, this is not the case. It is most efficient to implement DG in the two distinct steps defined by equations 2 and 3, instead of directly implementing the *primal* form. However, the gradient flux $\vec{\sigma}_l$ becomes out of date each time a neighbor is updated. As a result, $\vec{\sigma}_n$ must be reevaluated in almost all neighboring elements, as well as the local element, before $R(U^n, U^{n+1})$ can be properly evaluated and the solution updated. The implementation tested below strikes a compromise in that the gradient flux is evaluated only twice in each element: once just before the update, and again immediately after the update. For this reason, the scheme will be labeled *quasi*-GS. It should be noted that for LDG-one-sided on a Cartesian grid, one particular sweep direction will result in true GS. Alternate sweeping patterns, such as a checkerboard pattern, can also eliminate this difficulty.

American Institute of Aeronautics and Astronautics

It is interesting to note that GS can be applied to the IP discretization without an additional cost penalty. Because the numerical flux depends only on gradients obtained by differentiating the solution, the contribution to the numerical flux from the solution in $\Omega_l$ is not affected by an update occurring in any neighboring element. However, it should also be noted that the IP scheme always computes a gradient twice in each element, although two different methods are used.

One last point concerning the comparisons between analysis and the numerical experiments described in a later section is that the analysis presented in reference 20 is a Fourier analysis, and thus, is purely periodic. However, strictly speaking, GS, is not a periodic operator. In the numerical test, the explicit component of the spatial operator is treated periodically on the computational domain. However, the sweeping pattern of GS corresponds to a lower triangular matrix. Thus, some differences might be expected between the analysis and numerical test for GS.

## IV.   P-multigrid

P-multigrid is an iterative algorithm in which the convergence of a relaxation scheme applied to a set of high-order equations is improved by solving a low-order subset of the equations. The algorithm is applied recursively to the low-order set until the lowest order of $p_c = 0$ or 1 is reached. At this point, P-multigrid can be combined with geometric multi-grid (grid coarsening) to solve the lowest order set of equations.

The approach was first proposed by Rønquist and Patera[26] for spectral element discretizations of elliptic equations, and further analyzed by Maday and Munoz.[27] Helenbrook[17] applied P-multigrid to the Navier-Stokes equations using a SUPG discretization, and coupled P-multigrid with geometric multigrid for the first time.

Typically, the low-order subset is of degree $p_c = p/2$. For hierarchical bases, those for which the low-order basis set $\phi_c$ is a proper subset of the high-order basis set, the low-order set of correction equations is a subset of the high-order equations. The low order equations are solved for the subset of solution coefficients associated with low-order basis while holding the remaining components constant. For any $\phi_c$ in the space spanned by $\phi$, the prolongation operator is given formally by

$$I_{p,p_c} = M^{-1} \left[ \int_{\Omega_l} \phi \phi_c^T d\Omega \right], \tag{9}$$

where

$$M = \left[ \int_{\Omega_l} \phi \phi^T d\Omega \right].$$

The restriction operator is the transpose of the prolongation operator. If $L_p U_p = F_p$ denotes the set of high-order equations of degree $p$, then the low-order correction equation set is given by

$$I_{p,p_c}^T L_p U_p = I_{p,p_c}^T F_p. \tag{10}$$

In spite of its complex appearance, the restriction operator for an ordered hierarchical basis vector is simply

$$I_{p,p_c}^T = [\mathbf{I}, \mathbf{0}], \tag{11}$$

where $\mathbf{I}$ is an $N_{p_c} X N_{p_c}$ identity matrix, $\mathbf{0}$ is an $N_{p_c} X N_{p-p_c}$ null matrix, and $N_x$ denotes the number of members in a basis set of degree $x$. This approach will be referred to as *algebraic* P-multigrid, because the low-order equation set is constructed directly from the high-order set.

An alternate formulation is obtained by employing the methodology commonly used in geometric multi-grid. That is, given a code that implements a relaxation scheme on $L_p U_p = F_p$ for arbitrary $p$, then that code can easily be used to solve a correction equation that is of the form

$$L_{p_c} U_{p_c} = F(U_p, U_{p_c}), \tag{12}$$

where $F(U_p, U_{p_c})$ is a forcing term given by

$$F(U_p, U_{p_c}) = L_{p_c} I_{p,p_c}^T U_p - I_{p,p_c}^T (L_p U_p - F_p). \tag{13}$$

Because of its similarity to traditional geometric multigrid methods, this approach will be referred to as the *rediscretization* P-multigrid method.

American Institute of Aeronautics and Astronautics

Equations 12 and 13 are equivalent to 10 provided $L_{p_c} = I^T_{p,p_c} L_p I_{p,p_c}$. That is, the low-order operator is equal to the upper left block of the high-order operator. This property holds for methods such as SUPG, spectral element, and DG applied to strict first order equations (advection and propagation). However, it is not true for most DG discretizations of second derivative operators. Consider, for example, the $A$ matrix of equation 6. In the case of one-dimensional LDG-one-sided with a monomial basis, the matrices are

$$A_{p=0} = [-2], \qquad A_{p=1} = \begin{bmatrix} -8 & -1 \\ -1 & -2 \end{bmatrix}, A_{p=2} = \begin{bmatrix} -18 & -1 & -7/2 \\ -1 & -9/2 & -3/4 \\ -7/2 & -3/4 & -23/4 \end{bmatrix}. \tag{14}$$

For LDG-central with monomial basis, the matrices are:

$$A_{p=0} = [-1/2 - 2\eta], \qquad A_{p=1} = \begin{bmatrix} -29/8 - 2\eta & 0 \\ 0 & -13/32 - \eta/2 \end{bmatrix}, \tag{15}$$

$$A_{p=2} = \begin{bmatrix} -117/32 + 2\eta & 0 & 11/128 - \eta/2 \\ 0 & -213/128 - \eta/2 & 0 \\ -11/120 - \eta/2 & 0 & -95/1536 - \eta/8 \end{bmatrix}, \tag{16}$$

where $\eta$ is the coefficient of the penalty term. Note that in each case, the lower order matrix does not equal the upper left-block of the high order matrix. Finally, for IP with monomial basis the problem is not so severe. The matrices are:

$$A_{p=0} = [-2\eta], \qquad A_{p=1} = \begin{bmatrix} -2\eta & 0 \\ 0 & 1/2 - \eta/2 \end{bmatrix}, A_{p=2} = \begin{bmatrix} 2\eta & 0 & -1 - \eta/2 \\ 0 & 1/2 - \eta/2 & 0 \\ -\eta/2 & 0 & 1/12 - \eta/8 \end{bmatrix}. \tag{17}$$

For the principle part of the discretization, the low order matrix does equal the upper-left block of the high order matrix. The same is true of the contribution from the penalty terms; however recall that the coefficient of the penalty term should increase with $p$ in order to maintain stability. However, if $\eta$ is frozen at the high-order value while formulating the low order correction, then the low order equations with the appropriate forcing terms are equivalent to the low order component of the high order equations. For methods using the $\alpha_r([\![U]\!])$ penalty term[6, 25] the scaling is built in, and the matrices cannot be made equivalent by adjusting an explicit constant.

Numerical experiments with the LDG-one-sided found the *rediscretization* P-multigrid method to be highly unstable. The analysis in reference 20 confirms this result; and further shows that most discretizations are less robust, and that many are unstable, when *rediscretization* P-multigrid is applied.

The *algebraic* form of P-multigrid is not only more robust, but it can also be more efficient to implement; although it requires some special considerations in the code design. Recall that the formal restriction operator is essentially a trivial operator that simply selects a subset of the high-order equation set. This suggests it is possible to solve the low-order set "in-place"; that is, to use the existing data arrays, and modify the code to operate only on selected subsets of the data. This eliminates the need to allocate any additional storage and to copy data to and from that storage. From this prespective, it is appropriate to view *algebraic* P-multigrid as simply a relaxation in $p$-space in which some equations are solved while holding some variables constant.

The analysis[20] assumes that the lowest order set of equations are solved exactly. In the numerical test, the lowest order set is solved to some finite level of convergence. In most cases, the same relaxation scheme applied to the highest order equations is also applied to all lower order equations, but a sufficiently large number of iterations is applied to the lowest order set such that those equations are effectively solved.

As mentioned earlier, analysis[11] has shown that BJ bounds the eigenvalues of LDG-one-sided to be less than two (and GS to less that one). However, the same is not true of the low-order subset generated by *algebraic* P-multigrid, and some under relaxation can improve robustness of block Jacobi. In the numerical test in the next section, block Jacobi uses a relaxation of 1.0 at the highest level but 0.95 on lower levels.

For the remainder of this paper, the high-order set of equations will be referred to as the high-$p$ equations and the low-order set of correction equations will be referred to as the low-$p$ equations.

# V.  Numerical Test

P-multigrid is evaluated for the Laplace equation defined on a 2D unit domain $0 < x, y < 1$ with periodic boundaries. The domain is partitioned by an $NXN$ Cartesian grid. The initial solution is broadband with significant energy in the highest portion of the spectrum visible on any grid.

$$u(x,y) = F(2x)F(2y) + F(Nx)F(Ny) \qquad (18)$$
$$\text{where} \quad F(s) = \exp(\cos(\pi s) - 1)$$

Figure 1 illustrates a 1D version of the above function for N=16. Such an initial solution was found to be
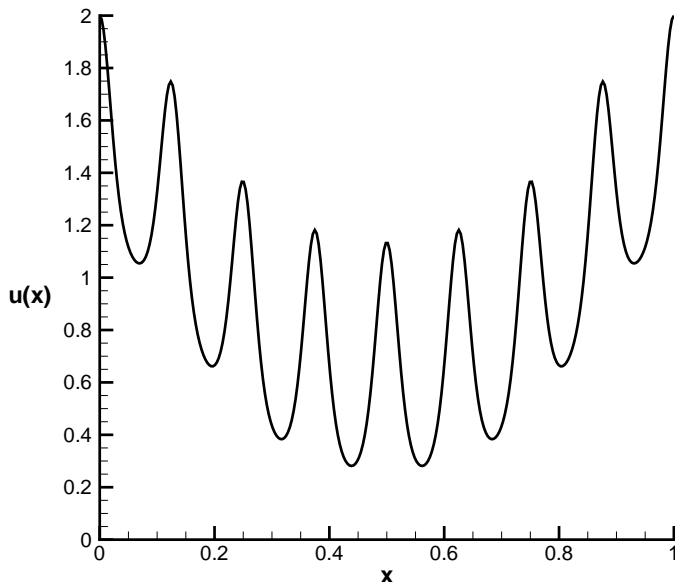


**Figure 1.  Representative form (1D) of initial solution for N=16.**

necessary because of the two level testing approach used in most of the numerical experiments. Although the longest wavelength usually dominates the asymptotic convergence rate, its convergence behavior is governed primarily by the low-order equations which are solved exactly. Thus a two grid test initialized with only long wavelengths will converge artificially fast.

Although accuracy is not a focus of this study, the solution error is examined in each case to verify that the method is converging with the expected order property. The equations are forced with $f(x) = -2 * \omega^2 \cos(\omega x) \cos(\omega y)$ so that the exact solution is non-trivial.

Unless otherwise noted, the first several results are for the LDG-one-sided discretization with $p = 4$. Figure 2(a) shows the convergence of BJ for a 2-level P-multigrid scheme in which the residual of the low-$p$ equations are converged to 0.001 below the residual of the high-$p$ equations. The effective spectral radius of convergence $\sigma$ is $\approx 0.71$, and agrees well with the analysis'[20] prediction of 0.73. Furthermore, the asymptotic convergence rate is independent of mesh size.

The analysis[20] assumes the low-$p$ equations are solved exactly. However, in practice, the low-$p$ equations will be only approximately solved. Figure 2(b) shows the convergence of the low-$p$ equations when the convergence criterion is 0.001. To examine the effect of only approximately solving the low-$p$ equations, as well as to establish that the low-$p$ equations are adequately solved in the above result, test are performed in which the convergence criterion on the low-$p$ equations is increased to 0.01, 0.1, and 0.5. For 0.01 and 0.1 (not shown) the convergence rates of the high-$p$ equations are nearly identical to those given above. In the case with 0.5, shown in figure 3(a) and (b), the asymptotic rate is still approximately 0.71, although the extremely rapid convergence that occurred over the first few iterations in the earlier case is not present, and

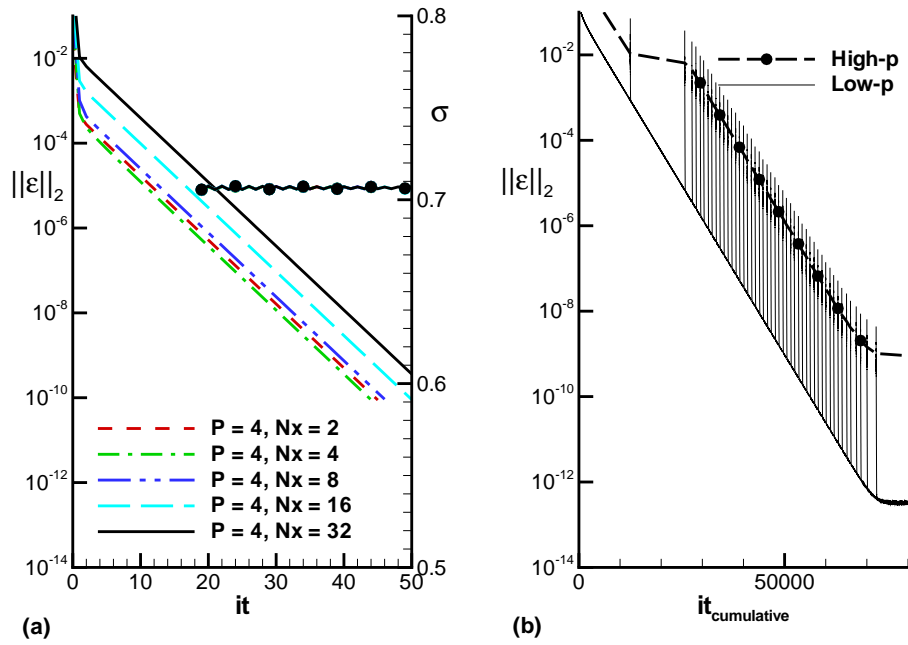American Institute of Aeronautics and Astronautics

**Figure 2. Convergence of 2-level P-multigrid using block Jacobi: in (a) $||\epsilon||_2$, line only; $\sigma$, line with symbol.**
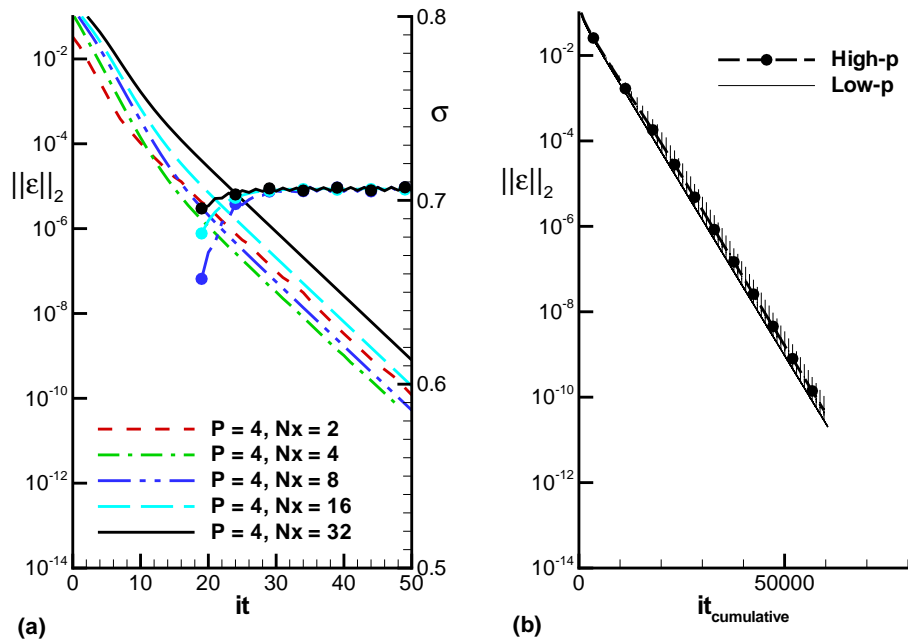


**Figure 3. Convergence of 2-level P-multigrid using block Jacobi with relaxed low-$p$ convergence criterion: in (a) $||\epsilon||_2$, line only; $\sigma$, line with symbol.**

American Institute of Aeronautics and Astronautics

the convergence rates on the coarsest meshes are slightly erratic. Comparing figures 2(b) with 3(b), we note that the total number of iterations on the low-$p$ equations, and hence the cost, is not greatly reduced by increasing the convergence criterion from 0.001 to 0.5, as one might have expected. A convergence criterion of 0.01 is used in the remaining results.

Figure 4(a) shows the results for GS applied to LDG-one-sided for the particular sweep direction that produces true GS. The numerical convergence rate approaches $\approx 0.585$ which is very close to the predicted value of 0.58. Figure 4(b) gives the convergence history of the finest grid for all possible sweep directions. The variation in convergence with sweep direction is minimal, and the quasi-GS scheme presented here performs similar to the true GS scheme.
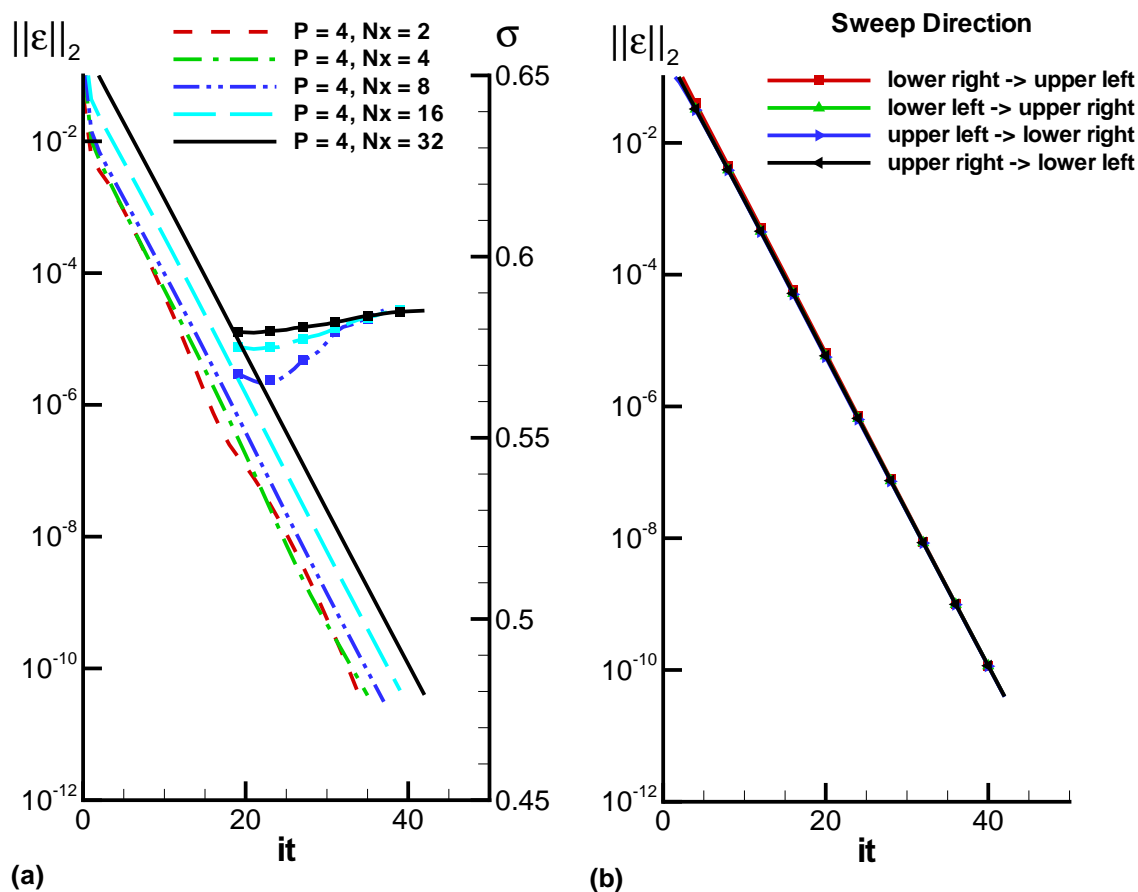


**Figure 4. Convergence of 2-level P-multigrid using Gauss-Seidel: (a) $||\epsilon||_2$, line only; $\sigma$, line with symbol. (b) finest grid for all sweep directions.**

Table 1 gives the convergence rates for BJ and quasi-GS using both LDG-central and LDG-one-sided. Also shown is the convergence rates predicted by the analysis given in 20. The agreement is good in every case but LDG-central with BJ. The reason for this discrepency is not known. Quasi-GS agrees well with the analysis for GS for both LDG-central and LDG-one-sided.

The analysis[20] predicts that the convergence rate of the 2-level scheme is about the same as that of the V-cycle. Figures 5 and 6 are for a V-cycle in which the minimum order is set at $p_c = 1$. For the case shown in figure 5, one iteration is performed at all levels but that of the lowest-order level, which was solved to a convergence criterion of 0.01. In the numerical test, the convergence rate increases to $\approx 0.78$, which is greater than that of the two-level cycle. However, increasing the number of iterations on the intermediate levels to 2, shown in figure 6(a), improves the convergence rate to slightly better than that of the 2-level cycle. Figure 6(b) shows similar results for GS applied to LDG-one-sided.

Although the two-level scheme with $p = 1$, $p_c = 0$ (not shown) converges well; V-cycles with $p_c = 0$ and

American Institute of Aeronautics and Astronautics

| | scheme | | | |
|---|---|---|---|---|
| | BJ, LDG-central | GS, LDG-central | BJ, LDG-one-sided | GS, LDG-one-sided |
| Numerical | 0.6 | 0.47 | 0.71 | 0.58 |
| Analysis | 0.75 | 0.51 | 0.73 | 0.58 |

Table 1. Comparison of convergence rates with analysis of 20.



Figure 5. Convergence of V-cycle P-multigrid using block Jacobi with one iteration at intermediate levels: $||\epsilon||_2$, line only; $\sigma$, line with symbol.



(a)        (b)

Figure 6. Convergence of V-cycle P-multigrid using with 2 relaxation steps on intermediate orders: $||\epsilon||_2$, line only; $\sigma$, line with symbol; (a) block Jacobi, (b) Gauss-Seidel

$p \geq 2$ converge poorly ($\sigma = 0.94$ on an N=32 grid and is mesh dependent.) This suggests an approach in which an embedded cycle is applied to converge the $p = 1$ level as illustrated in figure 7(a). Figure 7(b) shows that applying an embedded cycle with 40 iterations gives results identical to that obtained when $p_c = 1$.



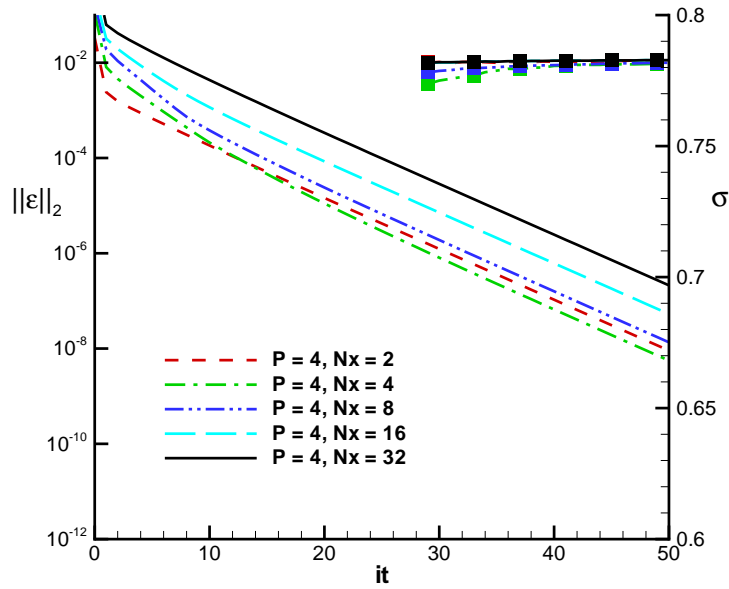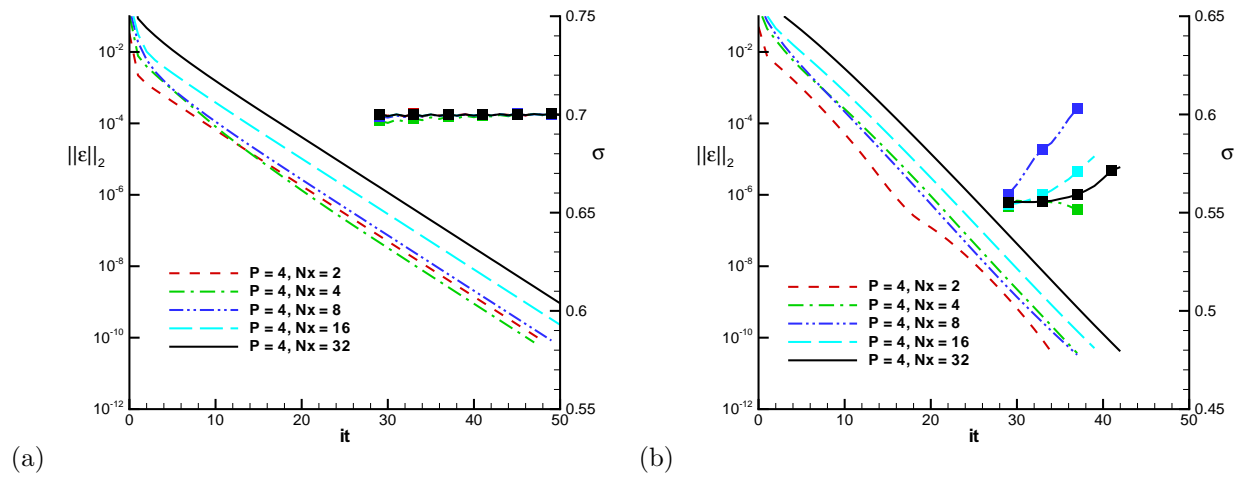(a)                                                                                     (b)

**Figure 7.** (a) Embedded V-cycle to solve $p = 1$; (b) Convergence of V-cycle P-multigrid using Gauss-Seidel with 2 relaxation steps on intermediate orders and an embedded cycle with 40 iterations at $p = 1$, $||\epsilon||_2$, line only; $\sigma$, line with symbol.

## VI.    Conclusions

Numerical tests with P-multigrid confirm that the scheme behaves essentially as the analysis of reference[20] predicts. It is confirmed that GS does converge better than BJ, but the numerical test reveals that GS can be considerably more expensive, and more complicated to implement, that one might expect. Tests with various sweep directions suggest that the quasi-GS scheme presented here converges as well as the true GS method. It is found that the *rediscretization* P-multigrid is unstable for LDG-one-sided, as predicted in the analysis. However, the *algebraic* form of P-multigrid can be easily implemented by solving the low order equation subset "in place", reusing much of the the existing storage and algorithms(code). Tests in which the low-$p$ convergence criterion varied suggested that the low-$p$ equations do not need to be exactly solved; however, increasing the criterion did not significantly reduce the overall work. A V-cycle with $p_c = 1$ converges at a rate similar to the two-level cycle. Reducing $p_c$ to zero results in poor convergence. However solving the $p = 1$ level with an embedded cycle restores the overall convergence rate to that of a V-cycle with $p_c = 1$.

## References

[1]H. L. Atkins and David P. Lockard. A high-order method using unstructured grids for the aeroacoustic analysis of realistic aircraft configurations. AIAA Paper 99-1945, 1999. Fifth Joint CEAS/AIAA Aeroacoustics Conference, May 10-12, Bellevue, WA.

[2]C. P. A. Blom, R. Hagmeijer, and A. Biesheuvel. Three-dimensional quadrature-free discontinuous galerkin method for computational aeroacoustics. AIAA Paper 2001-2198, 2001. Presented at the 7th AIAA/CEAS Aeroacoustics Conference, Maastrich, May 2001.

[3]M. Choudhari, M. Khorrami, D. Lockard, G. Lilly, and H. Atkins. Slat cove noise modeling: A posteriori analysis of unsteady rans simulations. AIAA Paper-2002-2468, 2002.

[4]P. Rao Preetham and Philip J. Morris. Application of a generalized quadrature free discontinuous galerkin method in aeroacoustics. AIAA Paper 2003-3120, 2003. Presented at the 9th AIAA/CEAS Aeroacoustics Conference, Hilton Head, SC, May 12-14.

[5]F. Bassi and S. Rebay. Discontinuous finite-element high order accurate numerical solution of compressible navier-stokes equations. In *Numerical Methods for Fluid Dynamics V*, pages 295–302. Clarendon, Oxford, England, UK, 1995.

[6]F. Bassi, S. Rebay, G. Mriotti, S. Pedinotti, and M. Savini. A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In R. Decuypere and G. Diebelius, editors, *proceedings of the 2nd European Confer-*

American Institute of Aeronautics and Astronautics

*ence on Turbomachinery, Fluid Dynamics, and Thermodynamics*, Lecture Notes in Mathematics, pages 99–108. Technologisch Instituut, Antwerpen, Belgium, October 1997.

[7] R. M. Kirby, T. Warburton, and G. E. Karniadakis. A discontinuous galerkin spectral/hp method on hybrid grids. *Applied Numerical Mathematics*, 33:393–405, 2000.

[8] S. Scott Collis. The dg/vms method for unified turbulence simulation. AIAA Paper-2002-3124, 2002.

[9] David A. Kopriva, Stephen L. Wordruff, and Y. Y. Hussaini. Discontinuous galerkin spectral element approximation of maxwell's equation. In G. Karniadakis, B. Cockburn, and C.-W. Shu, editors, *Proceedings of the Symposium on Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 355–362. Springer-Verlag, May 1999.

[10] J. S Hesthaven and T. Warburton. High-order/spectral methods on unstructured grids. i. time-domain solution of maxwell's equations. *Journal of Computational Physics*, 181:1–34, 2002.

[11] H. L. Atkins and Shu C.-W. Analysis of preconditioning and relaxation operators for the discontinuous galerkin method applied to diffusion. AIAA Paper 2001-2554, 2001.

[12] S. Scott Collis. Discontinuous galerkin methods for turbulence simulation, 2002. Center for Turbulence Research, Proceedings of the Summer Program.

[13] P. W. Hemker, W. Hoffmann, and M. H. van Raatle. Two-level fourier analysis of a multigrid approach for discontinuous galerkin discretization. *SIAM J. Sci. Comput.*, 25(3):1018–1041, 2003.

[14] F. Bassi and S. Rebay. Gmres discontinuous galerkin solution of the compressible navier-stokes equations. In G. Karniadakis, B. Cockburn, and C.-W. Shu, editors, *Proceedings of the Symposium on Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 197–208. Springer-Verlag, May 1999.

[15] James S. Warsa, Michele Benzi, Todd A. Wareing, and Jim E. Morel. Preconditioning a mixed discontinuous finite element method for radition diffusion. *Numerical Linear Algebra with Applications*, 99:1–18, 2001.

[16] Caroline Lasser and Andrea Toselli. An overlapping domain decomposition preconditioner for a class of discontinuous galerkin approximations of advection-diffusion problems. *Mathematics of Computation*, 72(243):1215–1238, 2003.

[17] B. T. Helenbrook. A two-fluid spectral eleement method. *Comp. Meth. Appl. Mech. Eng.*, 191:273–294, 2001.

[18] B. T. Helenbrook, D. Mavriplis, and H. L. Atkins. Analysis of "p"-multigrid for continuous and discontinuous finite element discretizations. AIAA Paper 2003-3989, 2003. 16nd AIAA Computational Fluid Dynamics Conference, Orlando, Flordia, June.

[19] K. J. Fidkowski, T. A. Oliver, and D. L. Darmofal. "p"-multigrid to discontinuous galerkin formulations of the poisson equation. PhD thesis, Massachusetts Institute of Technology, June, 2004.

[20] B. T. Helenbrook and H. L. Atkins. Application of "p"-multigrid to discontinuous galerkin formulations of the poisson equation. AIAA paper 2005-5111, 2005. Submitted to the 17nd AIAA Computational Fluid Dynamics Conference, Toronto, Ontario, June 6-9, 2005.

[21] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.

[22] Bernardo Cockburn and Chi-Wang Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal of Numerical Analysis*, 35(6):2440–2463, 1998.

[23] Jr. Douglas, J and T. Dupont. Interior penalty procedures for elliptic and parabolic galerkin methods. In R. Glowinski and J. L. Lions, editors, *Proceedings of the Second International Symposium on Computing Methods in Applied Sciences and Engineering, 2d. Versailles, 1975*, volume 58 of *Lecture Notes in Physics*. Springer-Verlag, 1976.

[24] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

[25] F. Brezzi, G. Manzini, D. Marini, Pietra P., and A Russo. Discontinuous finite elements for diffusion problems. In *in Att Covegno in onore di F. Brioschi Istituto Lombardo*, pages 197–217. Accademia di Sccienze e Lettere, Milan, 1999.

[26] Rønquist Einar M. and Anthony T. Patera. Spectral element multigrid. i. formualation and numerical results. *Journal of Scientific Computing*, 2:389–406, 1987.

[27] Yvon Maday and Rafael Munoz. Spectral element multigrid part 2: Theoretical justification. NASA-CR-181761 also ICASE Report No. 88-73, 1988.