AIAA JOURNAL
Vol. 44, No. 3, March 2006

Application of *p*-Multigrid to Discontinuous Galerkin Formulations of the Poisson Equation

Brian T. Helenbrook*
Clarkson University, Potsdam, New York 13699-5725
and
H. L. Atkins†
NASA Langley Research Center, Hampton, Virginia 23681-2199

The *p*-multigrid is investigated as a method for solving discontinuous Galerkin (DG) formulations of the Poisson equation. Different relaxation schemes and basis sets are combined with the DG formulations to find the best performance. The damping factors of the schemes are determined using Fourier analysis for both one- and two-dimensional problems. One important finding is that the standard approach of forming the coarse *p* matrices separately for each level of multigrid is often unstable. To ensure stability, these matrices must be constructed from the fine grid matrices using algebraic multigrid techniques. Of the relaxation schemes, we find that the combination of Jacobi relaxation with the spectral element basis is fairly effective. The results using this combination are *p* sensitive in both one and two dimensions, but reasonable convergence rates can still be achieved for moderate values of *p* and isotropic meshes. A competitive alternative is a block Gauss–Seidel relaxation. This actually outperforms a more expensive line relaxation when the mesh is isotropic. When the mesh becomes highly anisotropic, the implicit line method and the Gauss–Seidel implicit line method are the only effective schemes. Adding the Gauss–Seidel terms to the implicit line method gives a significant improvement over the line relaxation method.

Nomenclature

A	=	discrete matrix equations
e	=	element edge
$\boldsymbol{F}$	=	vector of discrete source terms
f	=	source function
$G_n(\xi)$	=	spectral element polynomials
$I_n(\xi)$	=	integrated Legendre polynomials
I_{p,p_c}	=	prolongation operator
K	=	segment or quadrilateral element
$L^2(\Omega)$	=	space of square-integrable functions on the domain Ω
M	=	mass matrix
$M_n(\xi)$	=	monomials
$\boldsymbol{n}$	=	outward normal to element boundary
n_d	=	number of iterations performed moving to coarser mesh
n_u	=	number of iterations performed moving to finer mesh
$P(K)$	=	space of polynomial functions
$P_n(\xi)$	=	Legendre polynomials
p	=	polynomial order
R	=	relaxation matrix
$\mathcal{T}$	=	set of segments or quadrilateral elements K that triangulate spatial domain
u	=	Poisson solution
$\boldsymbol{u}_j$	=	solution coefficients for segment j
$\boldsymbol{u}_{j,k}$	=	solution coefficients for element j, k
$\hat{\boldsymbol{u}}$	=	boundary flux function
v	=	scalar test function
$\boldsymbol{x}$	=	position vector
α_j	=	flux function constant
α_r	=	flux function lifting operator
β	=	local discontinuous Galerkin scheme constant
∂K	=	element boundary
η	=	$\mathcal{O}(1)$ flux function constant
θ	=	nondimensional wave number
ν	=	iteration counter
ξ	=	element coordinate
$\Sigma(K)$	=	$[P(K)]^2$
$\boldsymbol{\sigma}$	=	flux vector
$\hat{\sigma}$	=	boundary flux function
$\boldsymbol{\tau}$	=	vector test function
$\boldsymbol{\phi}$	=	vector of basis functions
Ω	=	spatial domain
ω	=	underrelaxation factor
$[\![\ ]\!]$	=	jump in quantity along element edge
$\{\ \}$	=	average of quantity along element edge

Subscripts

h	=	element length
K	=	element

I. Introduction

THE *p*-multigrid is an algorithm used to solve equations arising from high-order finite element discretizations, such as spectral/*hp* formulations, by recursively iterating on solution approximations at different polynomial order p. For example, to solve equations derived using a polynomial approximation order of four, we iterate at an approximation order of $p = 4$, 2, and 1. When a low order is reached, that is, $p = 1$ or 0, a conventional grid coarsening algorithm can be applied to solve for the low-order components of the solution. The p component of this algorithm was proposed by Rönquist and Patera[1] and analyzed by Maday and Munoz[2] for a Galerkin spectral element discretization of the Laplace equation. Helenbrook[3] combined *p*-multigrid with geometric multigrid and applied it to an unstructured streamwise upwind Petrov–Galerkin (SUPG) discretization of the incompressible Navier–Stokes equations (see Ref. 4). Recently, there has also been work combining overlapping Schwarz relaxation methods with *p*-multigrid for spectral element discretizations (see Ref. 5).

Received 12 January 2005; presented as Paper 2005-5111 at the AIAA 17th Computational Fluid Dynamics Conference, Toronto, ON, Canada, 6–9 June 2005; revision received 21 July 2005; accepted for publication 26 August 2005. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/06 $10.00 in correspondence with the CCC.

[*] Assistant Professor, Mechanical and Aeronautical Engineering Department; helenbrk@clarkson.edu. Member AIAA.

[†] Senior Research Scientist, Mail Stop 128, Computational Aerosciences Branch; h.l.atkins@larc.nasa.gov. Member AIAA.

All of the aforementioned work has been for continuous formulations. The p-multigrid has also been used for discontinuous formulations.[6–8] To determine the most effective implementation, we have performed some analysis of p-multigrid for hyperbolic systems.[9] There has also been some recent analysis of moderate Reynolds number convection–diffusion problems (see Ref. 8), but there has not been a thorough investigation of the many possible implementations for diffusion. In this paper, we analyze the efficiency of p-multigrid when applied to discontinuous Galerkin formulations of the Poisson equation. Because discontinuous formulations of the Poisson equation are relatively new, there has been little analysis of this combination. There are many factors that can affect the performance of p-multigrid for these formulations. In the following sections we examine various combinations of discontinuous Galerkin (DG) formulation (see Refs. 10 and 11), polynomial basis functions, and relaxation schemes. Fourier analysis of both one- and two-dimensional problems is performed to assess the iterative efficiency. In two dimensions, we also examine the effect of mesh aspect ratio. The first section of the paper summarizes the DG formulations for the Poisson equation. In Secs. III and IV, a description is given of the polynomial bases used and the relaxation schemes investigated. In Sec. V, the multigrid scheme and the analysis techniques used to determine its efficiency are described. The last two sections give results for one-dimensional and two-dimensional problems, respectively.

II. DG Formulations

The model problem is the Poisson equation in one and two dimensions on a unit segment or unit square domain with periodic boundary conditions. To formulate the DG schemes, we write the Poisson equation in the form

$$\boldsymbol{\sigma} = -\nabla u \tag{1}$$

$$\nabla \cdot \boldsymbol{\sigma} = f(\boldsymbol{x}) \tag{2}$$

where u is the solution to the Poisson problem, $\boldsymbol{x}$ is the position vector, $\boldsymbol{\sigma}$ is a flux vector, and $f(\boldsymbol{x})$ is a given source function.

We next introduce a finite-dimensional space of functions to represent the solution. The domain is subdivided into either uniform length elements (one dimension) or rectangular elements (two dimensions), and, on each element, we use a polynomial basis to describe the solution u and the flux vector σ. Following the notation in Ref. 12, we define the spaces

$$V_h := \left\{ v \in L^2(\Omega) : v|_K \in P(K)\ \forall K \in \mathcal{T}_h \right\} \tag{3}$$

$$\Sigma_h := \left\{ \tau \in [L^2(\Omega)]^2 : \tau|_K \in \Sigma(K)\ \forall K \in \mathcal{T}_h \right\} \tag{4}$$

where $L^2(\Omega)$ is the space of square-integrable functions on the domain Ω, $\mathcal{T}_h$ is the set of segments or quadrilateral elements K that triangulate the domain, and the subscript h refers to the element length associated with a particular mesh. In one dimension, $P(K) = \mathcal{P}_p(K)$ is the space of polynomial functions of at most p degrees on segment K. In two dimensions, $\mathcal{P}_p$ is formed from the tensor product of the one-dimensional set of polynomial functions. $\Sigma(K)$ is equal to $[P(K)]^2$.

All of the DG formulations we analyze are based on the weak form of Eqs. (1) and (2). Multiplying these equations by a scalar test function, $v_h \in P(K)$, and a vector test function $\tau_h \in \Sigma(K)$, respectively, and integrating by parts gives the weak form that is used to find $u_h \in V_h$ and $\sigma_h \in \Sigma_h$:

$$\int_K \sigma_h \cdot \tau \,\mathrm{d}x = -\int_K u_h \nabla \cdot \tau \,\mathrm{d}x + \int_{\partial K} \hat{u}_K n_K \cdot \tau \,\mathrm{d}s \quad \forall \tau \in \Sigma(K) \tag{5}$$

$$-\int_K \sigma_h \cdot \nabla v \,\mathrm{d}x = \int_K f v \,\mathrm{d}x + \int_{\partial K} \hat{\sigma}_K \cdot n_K v \,\mathrm{d}s \quad \forall v \in P(K) \tag{6}$$

Table 1 DG schemes analyzed and their numerical fluxes

Scheme	$\hat{u}_K$	$\hat{\sigma}_K$
Bassi and Rebay[11]	$\{u_h\}$	$\{\sigma_h\}$
Brezzi et al.[13]	$\{u_h\}$	$\{\sigma_h\} - \alpha_r(\llbracket u_h \rrbracket)$
Local DG (LDG)[14]	$\{u_h\} - \beta \cdot \llbracket u_h \rrbracket$	$\{\sigma_h\} + \beta \llbracket \sigma_h \rrbracket - \alpha_j \llbracket u_h \rrbracket$
Interior penalty[15]	$\{u_h\}$	$\{\nabla_h u_h\} - \alpha_j \llbracket u_h \rrbracket$
Bassi et al.[16]	$\{u_h\}$	$\{\nabla_h u_h\} - \alpha_r(\llbracket u_h \rrbracket)$

where $\boldsymbol{n}_K$ is the outward normal to the element boundary ∂K and $\hat{u}_K$ and $\hat{\sigma}_K$ are boundary flux functions. These functions are evaluated along element edges using information from both sides of the element and, thus, provide the interelement coupling in a DG scheme.

The choice of the boundary fluxes distinguishes the various DG schemes.[12] We analyze p-multigrid with the schemes that are listed in Table 1 (Refs. 11 and 13–16). The flux functions for these schemes are also shown. The notation again follows that by Arnold et al.[12]: Braces { } denote the average of a quantity along an edge. Double brackets $\llbracket\ \rrbracket$ denote the jump in a quantity along an edge. For an edge with adjacent elements labeled 1 and 2, the jump in a scalar q is a vector given by $q_1\boldsymbol{n}_1 + q_2\boldsymbol{n}_2$, where q_i, $i \in [1, 2]$, is the value of q evaluated along the edge using the solution from adjacent element i, and $\boldsymbol{n}_i$ is the normal to the edge oriented to point outward from element i. If $\boldsymbol{q}$ is a vector, the jump is given by $\boldsymbol{q}_1 \cdot \boldsymbol{n}_1 + \boldsymbol{q}_2 \cdot \boldsymbol{n}_2$.

In the LDG scheme, the constant β can take values between $-\frac{1}{2}$ and $\frac{1}{2}$. The constant α_j is given by ηh^{-1}, where η is an $\mathcal{O}(1)$ constant and h is a characteristic mesh length normal to the edge. Here, $\alpha_r(\llbracket u_h \rrbracket)$ is defined by a lifting operator[12]; briefly, the following equation defines a vector function, $\boldsymbol{r}_e \in \Sigma_h$, which is nonzero only on the elements on either side of the edge e:

$$\int_\Omega \boldsymbol{r}_e \cdot \tau \,\mathrm{d}x = -\int_e \llbracket u_h \rrbracket \cdot \{\tau\} \,\mathrm{d}s \quad \forall \tau \in \Sigma_h \tag{7}$$

where $\alpha_r(\llbracket u_h \rrbracket)$ is then given by $\eta\{\boldsymbol{r}_e\}$, η is an $\mathcal{O}(1)$ constant, and the braces again denote the average of the discontinuous function $\boldsymbol{r}_e$ along the edge e. In one dimension, $\alpha_r(\llbracket u_h \rrbracket)$ simplifies to the multiplication of $\llbracket u_h \rrbracket$ by a constant that depends on the order of the basis and the length of the elements.

Except for the first scheme in Table 1, the preceding schemes are all consistent and stable. The first scheme is consistent but not stable. It is included so that we can understand how the stability of the scheme affects the multigrid iteration. Note that if β is zero, the LDG scheme and the Brezzi et al.[13] scheme are only different by the magnitude of α_r vs α_j. This is also true of the interior penalty scheme and the Bassi et al.[16] scheme.

III. Basis Functions

Although the specific form of the basis functions used to represent the space $\mathcal{P}_p(K)$ does not affect the final solution for u_h, it can have a strong effect on the efficiency of the relaxation schemes. We investigate several different sets of basis functions. Unlike continuous formulations for which the form of the basis is constrained by continuity requirements, any basis can be easily used in a DG formulation. Table 2 shows the sets of one-dimensional basis functions we investigate and their special properties. The bases are defined on the domain $\xi \in [-1, 1]$. The two-dimensional bases are a tensor product of the one-dimensional bases.

The Legendre basis is orthogonal, which gives a diagonal mass matrix,

$$M = \int \phi\phi^T \,\mathrm{d}x$$

where ϕ is the vector of basis functions. The integrated Legendre basis is orthogonal with respect to the typical bilinear operator associated with a continuous formulation of the Poisson equation

$$\int \frac{\mathrm{d}\phi}{\mathrm{d}x}\frac{\mathrm{d}\phi^T}{\mathrm{d}x} \,\mathrm{d}x$$

Table 2 Basis sets used

Basis	Symbol	Property
Legendre	$P_n(\xi)$	$\int_{-1}^{1} P_m(\xi) P_n(\xi)\,\mathrm{d}\xi = \delta_{m,n}$
$\int$ Legendre	$I_n(\xi) = \int_{0}^{\xi} P_{n-1}(\xi')\,\mathrm{d}\xi'$	$\int_{-1}^{1} \frac{\mathrm{d}I_m(\xi)}{\mathrm{d}\xi}\frac{\mathrm{d}I_n(\xi)}{\mathrm{d}\xi}\,\mathrm{d}\xi = \delta_{m,n}$
Monomial	$M_n(\xi) = \xi^n$	Quadrature-free integration[17]
Spectral element	$G_n(\xi)$	Nodal at Gauss–Lobatto points (Lagrangian)

The monomial basis is included because it is simple and can be efficient if implemented properly.[17] The spectral element basis is typically used for continuous formulations. Operations at element boundaries are simpler with this basis because only a few basis functions are nonzero at the boundaries. It has also been shown that this basis is well suited for p-multigrid solutions of continuous formulations of the Poisson equation (see Refs. 1, 2, and 9).

IV. Relaxation Schemes

Before describing the relaxation schemes, we introduce some matrix notation for the linear systems of equations generated by the DG formulations. Because DG formulations are dominated by operations on elements, we label the solution coefficients by element, such as $\boldsymbol{u}_j$. This corresponds to the vector of coefficients used to describe the solution on element j. The solution on this element can then be written as $\boldsymbol{\phi}^T \boldsymbol{u}_j$, where $\boldsymbol{\phi}$ is again the vector of basis functions. In one dimension, the vector of coefficients and basis functions for an element is of length $p+1$. In two dimensions, it is of length $(p+1)^2$. In two dimensions, we use a multidimensional indexing for the elements, that is, $\boldsymbol{u}_{j,k}$, where j is the horizontal index and k is the vertical index.

The following matrix is an example of the general form of one-dimensional DG formulations for a mesh with four elements and periodic boundary conditions:

$$\begin{bmatrix} M & 0 & 0 & 0 & D_M & D_R & 0 & D_L \\ 0 & M & 0 & 0 & D_L & D_M & D_R & 0 \\ 0 & 0 & M & 0 & 0 & D_L & D_M & D_R \\ 0 & 0 & 0 & M & D_R & 0 & D_L & D_M \\ S_M & S_R & 0 & S_L & U_M & U_R & 0 & U_L \\ S_L & S_M & S_R & 0 & U_L & U_M & U_R & 0 \\ 0 & S_L & S_M & S_R & 0 & U_L & U_M & U_R \\ S_R & 0 & S_L & S_M & U_R & 0 & U_L & U_M \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \\ \boldsymbol{F}_3 \\ \boldsymbol{F}_4 \end{bmatrix} \tag{8}$$

Each entry in the matrix is a block of dimension $(p+1) \times (p+1)$. The first four rows are the discrete equivalent of Eq. (5). The last four correspond to Eq. (6). The $\boldsymbol{F}_j$ terms are the vectors that result from the variational integration of the source term f on each element. Because all of the DG formulations choose the flux $\hat{u}_K$ to be independent of σ, there is no interelement coupling of σ, and, thus, the upper left quarter of the matrix is block diagonal. This allows σ to be found using local operations and to be eliminated from the problem.

Replacing σ_j by $M^{-1}(D_L u_{j-1} + D_M u_j + D_R u_{j+1})$, we arrive at a circulant pentadiagonal matrix with block entries

$$\begin{aligned} A_{LL} &= 0 - S_L M^{-1} D_L \\ A_L &= U_L - S_L M^{-1} D_M - S_M M^{-1} D_L \\ A_M &= U_M - S_L M^{-1} D_R - S_M M^{-1} D_M - S_R M^{-1} D_L \\ A_R &= U_R - S_M M^{-1} D_R - S_R M^{-1} D_M \\ A_{RR} &= 0 - S_R M^{-1} D_R \end{aligned} \tag{9}$$

All of the iterative schemes are applied to this form of the equations. The two-dimensional system is similar, except with a nine-element stencil. For the interior penalty method and the Bassi et al.[16] scheme, the $\hat{\sigma}$ fluxes do not involve σ. Therefore, S_L and S_R are zero, and the system is tridiagonal. For the LDG method, if β is chosen uniformly as $+\frac{1}{2}$, the $\hat{u}$ flux for the edge between element j and $j+1$ is one sided, involving only u_{j+1}. This implies that D_L is equal to 0. Similarly for $\hat{\sigma}$, S_R becomes zero. We then arrive at a tridiagonal matrix again. For nonperiodic problems, however, β must be chosen to be compatible with the boundary information, and, in general, it is not possible to produce a uniformly compact stencil with the LDG approach.

Given the preceding form for the discrete governing equations, we now describe the iterative schemes. All of the schemes are written in the form

$$R\Delta\boldsymbol{u} + (A\boldsymbol{u} - \boldsymbol{F}) = 0 \tag{10}$$

where R is a relaxation matrix, $\boldsymbol{u}$ is the vector of unknown coefficients for all elements, and A is the stiffness matrix, which is composed of the block entries shown in Eq. (9). $\boldsymbol{F}$ is the entire vector of source terms consisting of the $\boldsymbol{F}_j$ vectors from each element.

The first scheme we investigate is essentially a physical time advancement scheme. R is block diagonal with blocks consisting of the element mass matrix scaled by $1/\omega$. This is exactly what we would arrive at if we analyzed the heat equation instead of the Poisson equation. The relaxation factor ω then corresponds to the physical time step and is taken as the inverse of the maximum eigenvalue of $R^{-1}A$ calculated with $\omega = 1$. This relaxation scheme allows us to verify that the DG implementations are correct because it should give results consistent with the physical behavior of the heat equation. The results should also be independent of the form of the basis functions because the iterative matrix can be obtained from a weighted integral form $(\int \boldsymbol{\phi}\boldsymbol{\phi}^T\,\mathrm{d}x)$.

The next scheme is a Jacobi scheme with R composed of the diagonal elements of A. This is again scaled by a constant $1/\omega$ that is the inverse of maximum eigenvalue of $R^{-1}A$ calculated with $\omega = 1$. This is the least computationally intensive scheme. It is also the only scheme that gives results that are dependent on the form of the polynomial basis. When Legendre polynomials are used, this approach is very similar to the mass matrix approach because the mass matrix is diagonal. The only difference is in the magnitudes of the diagonal entries. This is also true of the spectral-element basis because the mass matrix of the spectral-element matrix is diagonal when integrated using a $p+1$ point Gauss–Lobatto integration rule.

The last scheme is a block Jacobi scheme. In this case, we take R to be the block diagonal matrices of A for each element, that is, A_M. The results for this scheme should also be independent of the basis because the block diagonal term can be obtained from a weighted integral form. The weighted integral form for a particular element is Eqs. (5) and (6), except with the fluxes evaluated assuming the solution for u is nonzero only on the element in question. The block Jacobi scheme is much more expensive than either of the first two schemes because it involves the inversion of a block matrix. Unlike the mass matrix scheme, a simple inversion that can be reused on various element geometries does not exist.

In practice, any of the preceding schemes can be transformed into a block Gauss–Seidel scheme by calculating the residual, $A\boldsymbol{u} - \boldsymbol{F}$, and updating the solution element by element. If the elements are ordered left to right, then bottom to top, this corresponds to adding to R the block matrices of A that couple the element being updated to the elements to the left and down (essentially the lower triangle of A). To keep R circulant, these block matrices are added even when the position of the corresponding element has wrapped around the domain due to periodicity. This maintains the periodicity of the problem, which makes the analysis easier.

V. The *p*-Multigrid

To implement the p-multigrid algorithm, restriction and prolongation operators are needed in addition to the relaxation scheme. Restriction consists of moving solution residuals from a space of high polynomial order to a lower order. We typically choose the order of the polynomial spaces such that the coarse space has a degree, $p_c = p/2$. In some cases, however, we skip directly from order p to order 1. Prolongation is the reverse operation in which the solution correction from the low-order space is transferred to the higher-order space. For a basis ϕ_c that is contained in the space spanned by a higher-order basis ϕ, the prolongation operator on an element is given by

$$I_{p,p_c} = M_K^{-1} \int_K \phi \phi_c^T \, \mathrm{d}x \tag{11}$$

This is a matrix of dimension $p \times p_c$ that takes a correction represented using the basis ϕ_c and gives an equivalent representation using the basis ϕ. For hierarchical bases such as $P_n(\xi)$, $I_n(\xi)$, and $M_n(\xi)$ (bases in which the lower-order basis functions are a subset of the higher-order functions), the prolongation operator takes a very simple form. If the functions are organized from low to high order then the operator is simply an identity matrix of dimension $p_c + 1$ followed by $p - p_c$ rows of zeros. In all cases, the restriction operator is the transpose of prolongation.

We use a linear multigrid algorithm with a V cycle. This algorithm can be written as a recursive subroutine as follows:

```
cycle($p$) {
    if ($p$ = smallest $p$) {
        $\boldsymbol{u}_{[p]} = A_{[p]}^{-1}(\boldsymbol{F}_{[p]})$
        return
    }
    Relaxation:
        $\boldsymbol{u}_{[p]} = \boldsymbol{u}_{[p]} + R^{-1}(\boldsymbol{F}_{[p]} - A_{[p]}\boldsymbol{u}_{[p]}) \quad \nu = 0, \ldots n_d$
    Restriction:
        $p_c = p/2$
        $\boldsymbol{F}_{[p_c]} = I_{p,p_c}^T(\boldsymbol{F}_{[p]} - A_{[p]}\boldsymbol{u}_{[p]})$
        $\boldsymbol{u}_{[p_c]} = 0$
    Recursion:
    cycle($p_c$)
    Prolongation:
        $\boldsymbol{u}_{[p]} = \boldsymbol{u}_{[p]} + I_{p,p_c}\boldsymbol{u}_{[p_c]}$
    Relaxation:
        $\boldsymbol{u}_{[p]} = \boldsymbol{u}_{[p]} + R^{-1}(\boldsymbol{F}_{[p]} - A_{[p]}\boldsymbol{u}_{[p]}) \quad \nu = 0, \ldots n_u$
    return
}
```

The subscript $[p]$ indicates which polynomial space is being used. For the highest-order space, $\boldsymbol{u}_{[p]}$ is the solution to the discrete Poisson equation. For lower values of p coefficient $\boldsymbol{u}_{[p]}$ is a solution correction that will be prolongated to a higher-order space. For each level, n_d relaxation steps are performed on the first entry to the subroutine and n_u relaxations are performed after the prolongation step. For the results presented here, we use $n_d = 1$ and $n_u = 0$. We also require that the matrix equations at the coarsest level are directly inverted. Unless otherwise noted, the results presented here are for a cycle with only two levels. The function shown uses $p_c = p/2$. When using block Jacobi relaxation, we also investigate the case of a direct jump to $p_c = 1$.

The stiffness matrices $A_{[p]}$ at the coarse levels are determined using two different techniques. In the first, the standard approach is used in which these matrices are evaluated in the same way as outlined earlier for the fine level. In the results, this is referred to as the rediscretization approach. In the second, the coarse level stiffness matrices are evaluated using an algebraic approach. In this case, we use the restriction and prolongation operators

$$A_{[p_c]} = I_{p,p_c}^T A_{[p]} I_{p,p_c} \tag{12}$$

For some of the schemes, these two approaches yield the same results. Specifically, the interior penalty scheme can be formulated totally in terms of u and does not require static inversion of σ (see Eq. 3.11 in Ref. 12). Furthermore, if η is fixed, the constant α_j does not change with p. Thus, compression by Eq. (12) simply reduces the order of the stiffness matrix yielding exactly the same results as if it were derived directly. The constant α_r in the Bassi et al.[16] scheme changes with p resulting in a small difference in the two approaches. The first three schemes in Table 1 all require static inversion of σ. In this case, the two different approaches of deriving $A_{[p_c]}$ give very different results.

VI. Analysis Techniques

For the analysis, we assume that the source function f is zero everywhere because the source term does not affect the convergence rate. To determine the convergence rates, we examine the eigenvalues of the multigrid iteration. For a two-level iteration with no source term, one multigrid cycle can be simplified to the following form:

$$\boldsymbol{u}^{[\nu+1]} = \left(I - R^{-1}A_{[p]}\right)^{n_u} \left(I - I_{p,p_c} A_{[p_c]}^{-1} I_{p,p_c}^T A_{[p]}\right) \times \left(I - R^{-1}A_{[p]}\right)^{n_d} \boldsymbol{u}^{[\nu]} \tag{13}$$

where I is the identity matrix.

Because the matrices are circulant, the discrete Fourier transform can be used to determine the eigenvalues. In one dimension, we assume that the solution on each element has the form

$$\boldsymbol{u}_j = \tilde{\boldsymbol{u}} e^{ij\theta} \tag{14}$$

where $\tilde{\boldsymbol{u}}$ is a vector of dimension $(p + 1)$ and θ can take values of $-\pi$ to π by increments of $2\pi/N$, where N is the number of elements. For all of the results presented, N is chosen large enough such the results are essentially continuous functions of θ. Substitution of the form given by Eq. (14) reduces the dimension of the eigenvalue problem to $p + 1$. This is then solved numerically for each value of θ. The maximum eigenvalue over the range $-\pi < \theta < \pi$ excluding $\theta = 0$ is the damping factor of the iterative scheme. Zero is excluded because Laplace's equation on a periodic domain is only determined up to a constant. This free constant results in a undamped mode at $\theta = 0$ no matter what iterative scheme is used.

The results for two dimensions are obtained in a similar way. In this case, we assume we have quadrilateral meshes and a tensor product basis formed from one of the one-dimensional bases given in Sec. III. To analyze the problem, Fourier transforms are performed in both directions using $\exp[i(j\theta_x + k\theta_y)]$. This reduces the problem to a $(p + 1)^2$ eigenvalue problem that can be solved for each combination of θ_x and θ_y in the domain $[-\pi, \pi]^2$. This analysis technique can also be extended to non-tensor-product bases on any type of structured mesh, but to limit the scope we will only study the combination of tensor-product bases on quadrilaterals.

VII. One-Dimensional Results

In this section, we present one-dimensional results for the mass matrix iterative scheme, the Jacobi relaxation scheme, the block Jacobi iterative scheme, and, last, the improvement to these schemes when implemented as a Gauss–Seidel relaxation. We begin with the mass matrix scheme because this iteration is physically analogous to the unsteady heat equation. By comparing the two, we can validate the analysis techniques and also examine the accuracy of the various DG formulations in Fourier space. The remainder of the section focuses on the efficiency of p-multigrid.

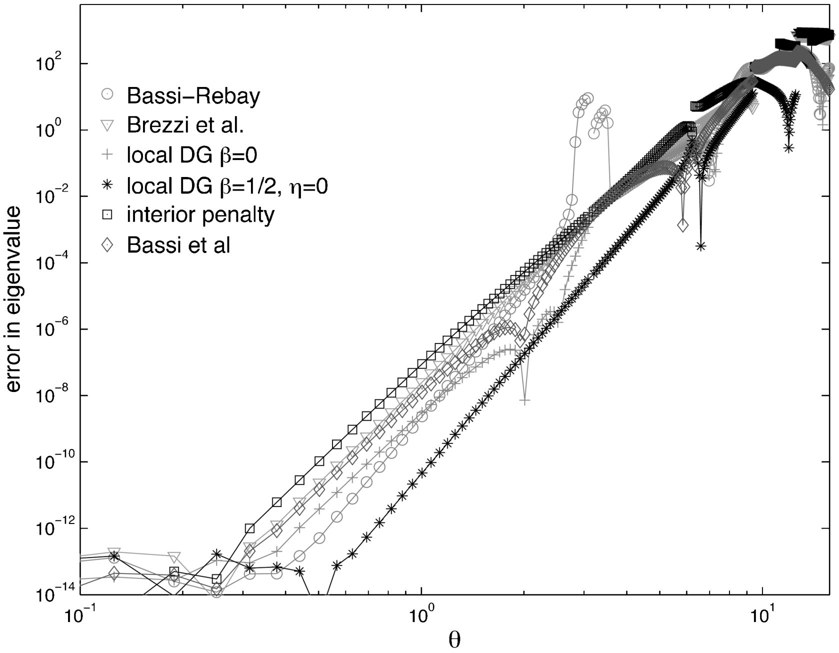

Fig. 1 Accuracy of eigenvalues for several DG formulations, $p = 4$.

A. Mass Matrix Relaxation

The eigenvalues of the matrix $R^{-1}(A\boldsymbol{u})$ calculated using the mass matrix preconditioner with $\omega = 1$ should correspond to the eigenvalues determined from the Fourier transform of the continuous problem

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0 \tag{15}$$

Figure 1 shows the difference between the eigenvalues of the discrete schemes and the continuous problem as a function of θ for $p = 4$. Here θ can be interpreted as the wave number of the eigenmodes nondimensionalized by the element length Δx. The analytic eigenvalues in terms of θ are given by $-\theta^2/\Delta x^2$. In Fig. 1, the error is nondimensionalized by Δx^2. We also have unrolled the $p + 1$ eigenvalues in θ; at any value of θ we have $p + 1$ eigenvalues. The larger eigenvalues correspond physically to eigenfunctions with a wave number that is shifted by an integer multiple of 2π from θ. We order the eigenvalues by magnitude and then assign the eigenvalues to the expanded domain $[0, (p + 1)\pi]$. For these results, the constant η for the penalty terms of the DG schemes is chosen as 1.

The eigenvalues for a continuous formulation are expected to converge at a rate of θ^{2p+1} (Ref. 18). All of the discontinuous schemes converge at this rate, θ^9, except two, the Bassi and Rebay[11] scheme and the one-sided LDG scheme ($\eta = 0$, $\beta = \frac{1}{2}$). These two schemes converge with a rate of θ^{12}. These are also the only schemes that are consistent when p is equal to 0. Unfortunately, both of these schemes have drawbacks. The Bassi and Rebay scheme is not stable, and this manifests itself as the large increase in error around $\theta = \pi$ in Fig. 1. At $\theta = \pi$, one of the eigenvalues is zero, which indicates that there is an element scale odd–even decoupling problem. The uniformly one-sided LDG scheme can only be used for periodic problems or for problems where the flux is specified on one side of the domain and the temperature on the other. For any other configuration, β must change sign to be compatible with the boundary information. If β changes sign and no α_j term is included, this scheme also has a nonphysical zero eigenvalue.

The one other exceptional result is the interior penalty scheme. With η equal to unity, the interior penalty method converges well for wavelengths greater than the element scale ($\theta < \pi$), however, at higher wave numbers the scheme loses stability and the eigenvalues actually change sign. This is impossible to see from Fig. 1 because it shows the absolute value of the error, and all of the errors are large at high wave numbers. By adjusting the penalty constant, we can make the interior penalty scheme identical to the Bassi et al.[16] scheme, which is stable. (The Brezzi et al.[13] scheme and the LDG scheme with $\beta = 0$ can also be made equivalent by adjusting η.) For $p = 4$, the interior penalty scheme with $\eta = 12.5$ is equivalent to the Bassi et al.[16] scheme. To avoid this loss of stability, we, therefore, increase η for the interior penalty scheme. We use a value of $\eta = 20$ and also parametrically investigate the effect of η on our results.

Before beginning our examination of p-multigrid, we establish some baseline properties of the mass matrix relaxation scheme. Figure 2 shows the damping of the mass matrix scheme as a function of θ for $p = 4$ using the Brezzi et al.[13] scheme, whose properties are typical of most of the DG schemes considered. The eigenvalues are unrolled in θ in the same way as for Fig. 1. Figure 2 shows that the mass matrix relaxation scheme has good damping at high wave numbers, but the damping deteriorates at smaller wave numbers. All of the other schemes are similar except for the Bassi and Rebay[11] scheme. Because the Bassi and Rebay scheme has a zero eigenvalue at $\theta = \pi$, the damping factor is exactly 1 at this point.

Figure 3 shows the magnitude of the eigenvalue spectrum for the multigrid iteration when using the mass matrix iteration applied to the Brezzi et al.[13] scheme with $p = 4$. Two sets of curves are shown. The curves marked with a plus are obtained using the rediscretization approach for obtaining the coarse grid stiffness matrix. The curves marked with a square are obtained using the algebraic approach. We again unroll the results in θ, but because of the nonmonotonic behavior of the eigenvalues, it is difficult to be sure that we have shifted the eigenvalues to the correct θ domain. Examining Fig. 3, we see that the multigrid iteration does an excellent job of eliminating the error modes that are not removed by relaxation alone. The algebraic approach totally removes three of the five error components for all values θ. This is because the coarse p operator is perfectly consistent with the fine p operator, and solving the coarse p operator completely eliminates the $p/2 + 1$ low-order error modes of the high-order system. For the rediscretization approach, there is a difference between the coarse and fine space operators, and, for this reason, only two of the three modes of the coarse space are totally eliminated. The overall damping factor of the rediscretization approach is slightly better: 0.84 vs 0.87 for the algebraic case. We show later that this is exceptional; in most cases the algebraic approach gives a smaller damping factor. In either case, the maximum damping factor (slowest convergence rate) is independent of the number of elements in the grid so that the convergence rates are grid independent.

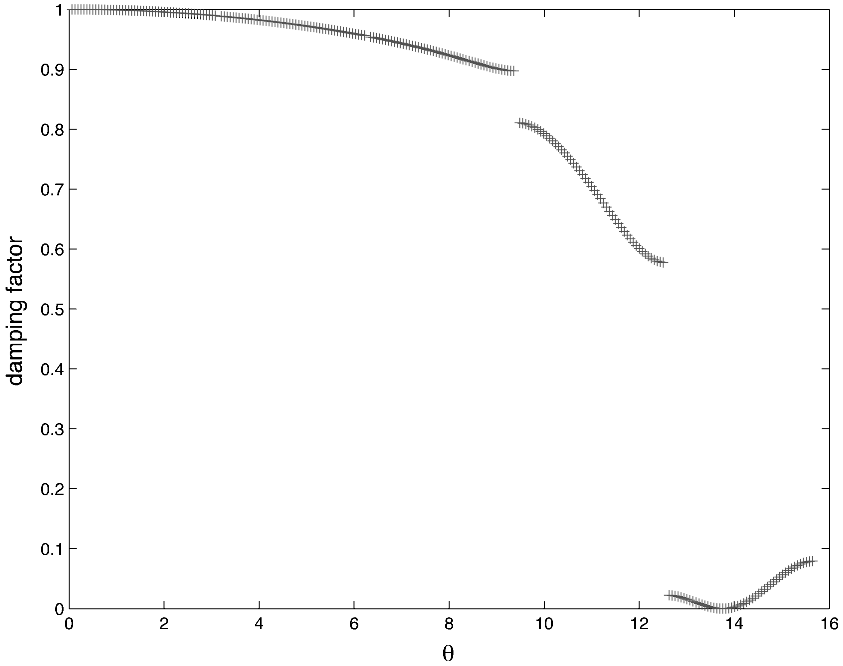

Fig. 2 Damping factor for the mass matrix preconditioner applied to the Brezzi et al.[13] scheme at $p = 4$.

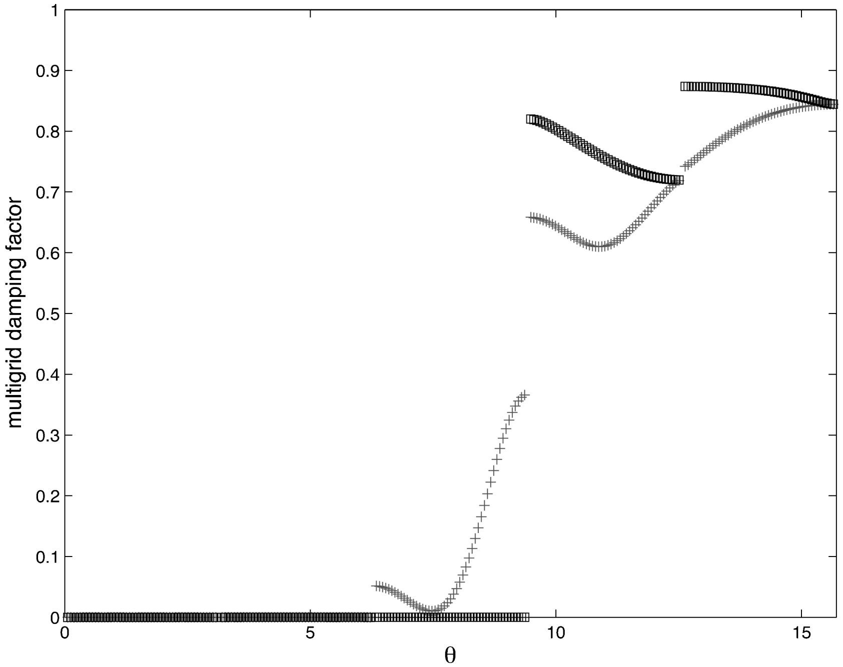

Fig. 3 Damping factor for mass matrix preconditioner applied to Brezzi et al.[13] scheme at $p = 4$ with multigrid: +, using rediscretization coarse grid operator and □, algebraically derived coarse grid operator.

Table 3 gives the damping factors for all of the schemes at polynomial degree of $p = 1$, 2, 4, and 8. In each case, $p_c = p/2$. Most of the schemes are not consistent when $p = 0$, and so we do not expect good results for the $p = 1$ case. However, it would be useful if we could coarsen to $p = 0$ because this system could then be solved using standard mesh-based multigrid techniques. In addition to examining the dependence on p, we also vary the stabilization constant η from one-quarter to four times its baseline value. For the Bassi and Rebay[11] scheme and the LDG one-sided scheme, the dashed entries indicate that the scheme does not depend on η. Unless otherwise noted, the baseline value of η for all of the schemes is one, except for the interior penalty scheme for which we use 20 for the reasons discussed earlier. In Table 3, a u indicates that the iteration is unstable.

A scan of the entries reveals that there are many more unstable iterations when using the rediscretization multigrid approach to determine A_c. The only unstable entries when using the algebraic approach occur for the interior penalty scheme and the Bassi et al.[16] scheme. Both of these schemes lose stability when η is too small. Thus, the problem is with the scheme itself and not the multigrid iteration. For the Bassi et al. scheme, as long as η is kept larger than one, this problem is avoided. For the interior penalty scheme, however, as we change p from four to eight with η/η_0 fixed at one, the scheme changes from stable to unstable. Thus, the penalty parameter for the interior penalty scheme must be adjusted with p. All of the remaining unstable iterations in Table 3 are caused by inconsistency between A_c evaluated with the rediscretization approach and A.

Table 3 Multigrid damping factors in one dimension with mass matrix relaxation scheme

	A_c, rediscretization			A_c, algebraic		
η/η_0	1/4	1	4	1/4	1	4
		$p = 1$				
Bassi and Rebay[11]	——	u	——	——	0.998	——
Brezzi et al.[13]	u	u	u	0.77	0.72	0.89
LDG, $\beta = 0$	u	0.69	0.80	0.85	0.69	0.80
LDG, $\beta = 1/2$, $\eta = 0$	——	u	——	——	0.75	——
Interior penalty	0.8	0.95	0.988	0.8	0.95	0.988
Bassi et al.[16]	u	u	u	u	0.54	0.88
		$p = 2$				
Bassi and Rebay	——	0.998	——	——	0.998	——
Brezzi et al.	0.78	0.64	0.86	0.78	0.72	0.88
LDG, $\beta = 0$	0.93	0.79	0.71	0.93	0.79	0.71
LDG, $\beta = 1/2$, $\eta = 0$	——	0.68	——	——	0.74	——
Interior penalty	0.6	0.87	0.97	0.6	0.87	0.97
Bassi et al.	u	0.75	0.83	u	0.67	0.94
		$p = 4$				
Bassi and Rebay	——	u	——	——	0.9992	——
Brezzi et al.	u	0.84	0.95	0.85	0.87	0.96
LDG, $\beta = 0$	u	u	0.81	0.98	0.92	0.83
LDG, $\beta = 1/2$, $\eta = 0$	——	u	——	——	0.93	——
Interior penalty	u	0.85	0.97	u	0.85	0.97
Bassi et al.	u	u	0.94	u	0.83	0.86
		$p = 8$				
Bassi and Rebay	——	u	——	——	0.9997	——
Brezzi et al.	u	0.98	0.993	0.96	0.98	0.993
LDG, $\beta = 0$	u	u	u	0.994	0.98	0.95
LDG, $\beta = 1/2$, $\eta = 0$	——	u	——	——	0.98	——
Interior penalty	u	u	0.98	u	u	0.98
Bassi et al.	u	u	0.99	u	0.98	0.991

Although the damping factor is not improved for every case when using the algebraic A_c, we conclude that the algebraic approach for evaluating A_c is correct because it guarantees that the coarse-p problem is consistent with the high-p problem. To reduce the possible combinations to be investigated, in the remaining results only the algebraic A_c is used. We also drop the interior penalty scheme because it is of the same form as the Bassi et al. scheme but requires manual adjustment of η with p. The Bassi and Rebay[11] scheme is dropped as well because the damping factor for this scheme is very close to one. In fact, we can get results arbitrarily close to unity for this scheme by increasing the resolution in θ. This is again because of the lack of stability of the scheme.

Looking at the dependence of the schemes on η for the algebraic cases, we see that the Brezzi et al.[13] scheme performs best at $\eta = 1$ or $\eta = \frac{1}{4}$. The LDG scheme with $\beta = 0$ performs best at $\eta = 4$ for all p except $p = 1$. At any p, the Brezzi et al. scheme and the LDG $\beta = 0$ scheme only differ by the constant α_j and α_r. Because we are no longer going to use the rediscretization method for evaluating A_c, the variation of this constant with p becomes irrelevant for multigrid. Examining the combined results of Brezzi et al. and LDG, noting that for $p = 1$, 2, 4, and 8, the ratio of α_r to α_j is 0.5, 4.5, 12.5, and 40, respectively, we find that the minimum in damping factor occurs when $\eta\alpha_j\Delta x = 2.0$ for all values of p. Thus, we also eliminate the Brezzi et al. scheme and only use the LDG $\beta = 0$ scheme with $\eta = 4$ as the baseline in the remaining results. The Bassi et al.[16] scheme works best with $\eta = 1$, which is used as the baseline for this scheme in the remaining results.

Examining the remainder of the entries in Table 3, we see that there is a strong sensitivity to p in the damping factor. The $p = 2$ and $p = 1$ entries are comparable when using the algebraic coarsening technique, and so we may be able to coarsen all of the way to $p = 0$. At higher p the damping factor degrades. To show why this occurs, Fig. 4 shows the damping factor for the same conditions as shown in Fig. 2, except at $p = 8$. For the multigrid scheme to work well, the wave numbers that are truncated in moving to the coarser space must be damped well by the relaxation scheme. Figure 4 shows that only the highest-order mode is suitably damped when using the mass matrix scheme. This implies that we must either move to the space $p_c = p - 1$ or use a more effective relaxation scheme.

Table 4 Multigrid damping factors in one dimension with Jacobi relaxation scheme for different basis sets

			Basis			
Scheme	η	p	$P_n(\xi)$	$I_n(\xi)$	$M_n(\xi)$	$G_n(\xi)$
LDG, $\beta = 0$	4	1	0.86	0.86	0.86	0.88
LDG, $\beta = 1/2$	0	1	0.82	0.82	0.82	0.80
Bassi et al.[16]	1	1	0.73	0.73	0.73	0.73
LDG, $\beta = 0$	4	2	0.83	0.92	0.92	0.67
LDG, $\beta = 1/2$	0	2	0.86	0.94	0.94	0.65
Bassi et al.	1	2	0.80	0.90	0.90	0.69
LDG, $\beta = 0$	4	4	0.90	0.97	0.995	0.69
LDG, $\beta = 1/2$	0	4	0.96	0.97	0.997	0.77
Bassi et al.	1	4	0.91	0.98	0.996	0.78
LDG, $\beta = 0$	4	8	0.97	0.99	1.0	0.91
LDG, $\beta = 1/2$	0	8	0.99	0.98	1.0	0.86
Bassi et al.	1	8	0.97	0.99	1.0	0.90

B. Jacobi Relaxation

The Jacobi relaxation results depend on the form of the polynomial basis used; therefore in this case we investigate different bases. Table 4 shows the damping factors for various basis sets and DG formulations. These results show that the spectral-element basis is superior to all of the other bases when using Jacobi relaxation. The integrated Legendre basis is very poor. Although this basis is orthogonal with respect to the bilinear product associated with the diffusive operator, this property is not valuable because of the boundary coupling. The monomial form also performs poorly for $p > 2$. This is not unexpected because monomials lack an orthogonality property for $p > 1$. The Legendre basis performs moderately well, but not quite as well as the spectral element basis. This may be because the Legendre basis functions are all nonzero at the element boundaries, which leads to greater interelement coupling.

Compared to the continuous case, the discontinuous p-multigrid iteration does not perform as well. For a continuous spectral-element formulation, the combination of Jacobi preconditioning and p-multigrid gives p independent results in one dimension.[1] Table 4 shows that there is sensitivity to p. However, the damping factors are still reasonably good for moderate values of p.

Table 4 also shows there is only weak sensitivity to the DG formulation chosen even though the schemes are very different. This suggests that the damping factors may not vary significantly with the parameters of the DG schemes. To verify this, we recalculate the results with the spectral element basis with η four times larger. All of the results with η four times smaller give damping factors closer to one, and so these are not shown. The results for the LDG $\beta = 0$ scheme and the Bassi et al.[16] scheme are shown in Table 5. Large improvements in the damping factor can be obtained by increasing η; however, increasing η degrades the accuracy of the scheme. For the LDG scheme, with $\eta = 16$, the error at any θ is nearly a factor of 10 larger than the values shown in Fig. 1, which are calculated with $\eta = 1$.

We also check that the Jacobi relaxation scheme works well in a multilevel multigrid cycle. Because we are using the algebraic approach to determine the coarse space matrices, the form of the coarse matrices depends on the fine space they are derived from. This means the behavior of the $p = 4$–2 transition depends on the fine-level polynomial degree we start from. To ensure that this does not have any adverse effects, we investigate a V-cycle with four levels starting at $p = 8$ and ending at $p = 1$. The damping factor for LDG, $\beta = 0$ using the Jacobi scheme and $\eta = 4$, is 0.91. This is exactly the same as the two-level case. The result for the Bassi et al.[16] case with $\eta = 1$ is also identical. This is because the highest-order modes are the most difficult to damp and, thus, determine the damping factor. However, if we stop the V-cycle at $p = 0$, the damping factors increase. For both the LDG scheme and the Bassi et al. schemes, the damping factor is 0.98. A similar behavior is seen if we start from $p = 4$ and go to $p = 0$. In the following results, we only investigate a two-level iteration because this provides an accurate prediction of the damping factor of a full V-cycle, and we do not further investigate the $p = 1$–0 transition.

C. Block Jacobi Relaxation

The next relaxation scheme we examine is block Jacobi relaxation. The block Jacobi preconditioner gives results that are independent of the basis; however, there is a strong sensitivity to the DG parameters and so we again investigate various values of η/η_0. The results are shown in Table 6. In Table 6, the results for one-sided LDG are identically zero for all p. In this case, block Jacobi corresponds to a static condensation[18] of the higher-order modes and then a direct solution of the low-order equations. This results in a direct inversion of the equations. For the other two schemes, we get very good damping factors, especially when using the larger values of η.

Although the combination of block Jacobi with multigrid gives good results, we note that in some cases the block Jacobi relaxation scheme by itself is actually unstable. It was shown in Ref. 19 that for LDG with $\beta = \frac{1}{2}$ and $\eta = 0$, block Jacobi is stable but that it converges very slowly. For the LDG, $\beta = 0$ scheme and the Bassi et al.[16] scheme, block Jacobi is actually unstable and neutrally stable, respectively. The stability of the relaxation scheme by itself can easily be improved by using an underrelaxation factor; however, this decreases the multigrid convergence rate. For this reason, we do not introduce a relaxation factor for block Jacobi, but in practical applications a relaxation factor may give more robust results. We also

Table 5 Multigrid damping factors in one dimension with Jacobi relaxation scheme for different values of η

		η/η_0	
Scheme	p	1	4
LDG, $\beta = 0$	1	0.88	0.95
Bassi et al.[16]	1	0.73	0.88
LDG, $\beta = 0$	2	0.67	0.55
Bassi et al.	2	0.69	0.51
LDG, $\beta = 0$	4	0.69	0.59
Bassi et al.	4	0.78	0.60
LDG, $\beta = 0$	8	0.91	0.76
Bassi et al.	8	0.90	0.68

Table 6 Multigrid damping factors in one dimension with block Jacobi relaxation scheme

		η/η_0		
Scheme	p	1/4	1	4
LDG, $\beta = 0$	2	0.62	0.27	0.09
LDG, $\beta = 1/2$, $\eta = 0$	2	——	0.00	——
Bassi et al.[16]	2	u	0.53	0.10
LDG, $\beta = 0$	4	0.81	0.52	0.20
LDG, $\beta = 1/2$, $\eta = 0$	4	——	0.00	——
Bassi et al.	4	u	0.61	0.08
LDG, $\beta = 0$	8	0.94	0.8	0.49
LDG, $\beta = 1/2$, $\eta = 0$	8	——	0.00	——
Bassi et al.	8	u	0.76	0.10

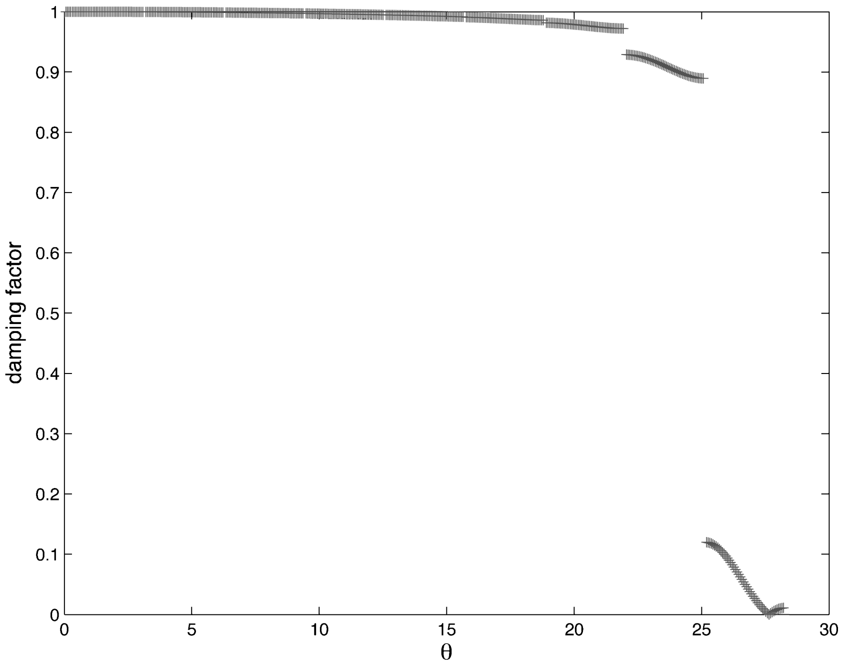

Fig. 4 Damping factor for mass matrix preconditioner applied to Brezzi et al.[13] scheme at $p = 8$.

Table 7 Multigrid damping factors in one dimension with block Jacobi relaxation scheme and $p_c = 1$

		η/η_0	
Scheme	p	1	4
LDG, $\beta = 0$	4	0.58	0.25
LDG, $\beta = 1/2$, $\eta = 0$	4	0.00	——
Bassi et al.[16]	4	0.68	0.11
LDG, $\beta = 0$	8	0.83	0.55
LDG, $\beta = 1/2$, $\eta = 0$	8	0.00	——
Bassi et al.	8	0.80	0.13

Table 8 Multigrid damping factors in one dimension with block Gauss–Seidel relaxation scheme

Scheme	η	p	Factor
LDG, $\beta = 0$	4	2	0.15
Bassi et al.[16]	1	2	0.32
LDG, $\beta = 0$	4	4	0.34
Bassi et al.	1	4	0.41
LDG, $\beta = 0$	4	8	0.68
Bassi et al.	1	8	0.63

Table 9 Multigrid damping factors in two dimensions with Jacobi relaxation scheme

			η/η_0	
Scheme	p	1/4	1	4
LDG, $\beta = 0$	2	0.94	0.90	0.95
LDG, $\beta = 1/2$, $\eta = 0$	2	——	0.93	——
Bassi et al.[16]	2	u	0.86	0.94
LDG, $\beta = 0$	4	0.96	0.90	0.90
LDG, $\beta = 1/2$, $\eta = 0$	4	——	0.93	——
Bassi et al.	4	u	0.92	0.94
LDG, $\beta = 0$	8	0.987	0.96	0.92
LDG, $\beta = 1/2$, $\eta = 0$	8	——	0.96	——
Bassi et al.	8	u	0.96	0.93

Table 10 Multigrid damping factors in two dimensions with block Jacobi relaxation scheme

			η/η_0	
Scheme	p	1/4	1	4
LDG, $\beta = 0$	2	0.70	0.59	0.79
LDG, $\beta = 1/2$, $\eta = 0$	2	——	0.63	——
Bassi et al.[16]	2	u	0.61	0.77
LDG, $\beta = 0$	4	0.86	0.68	0.75
LDG, $\beta = 1/2$, $\eta = 0$	4	——	0.73	——
Bassi et al.	4	u	0.73	0.85
LDG, $\beta = 0$	8	0.95	0.88	0.84
LDG, $\beta = 1/2$, $\eta = 0$	8	——	0.89	——
Bassi et al.	8	u	0.86	0.94

Table 11 Multigrid damping factors in two dimensions with block Gauss–Seidel relaxation scheme

Scheme	η	p	Factor
LDG, $\beta = 0$	4	2	0.44
LDG, $\beta = 1/2$	0	2	0.48
Bassi et al.[16]	1	2	0.42
LDG, $\beta = 0$	4	4	0.51
LDG, $\beta = 1/2$,	0	4	0.58
Bassi et al.	1	4	0.52
LDG, $\beta = 0$	4	8	0.74
LDG, $\beta = 1/2$	0	8	0.73
Bassi et al.	1	8	0.69

note that when applying block Jacobi without any underrelaxation, it is critical to use the algebraic approach to evaluate the coarse grid matrices. When the rediscretization approach is used, the iteration is often unstable. However, when an underrelaxation parameter of $\frac{1}{2}$ is used, both the rediscretization and algebraic approaches perform similarly, but considerably slower than the cases shown here.

Because block Jacobi is such a strong relaxation scheme, we also investigate the case in which we move directly to $p_c = 1$. Table 7 gives these results. Compared to the damping factors given in Table 6, there is little change if we move directly to $p = 1$. This may recover some of the additional cost of block Jacobi compared to the simpler relaxation schemes. We have also tried moving directly to $p = 0$, but this again causes a large degradation in performance.

D. Gauss–Seidel

As mentioned earlier, any of the preceding relaxation schemes can be used with a Gauss–Seidel approach of updating the solution. We find that there is little improvement to either the mass matrix preconditioner or the Jacobi preconditioner results when the Gauss–Seidel terms are added. There is obviously no improvement to the block Jacobi approach for the LDG one-sided scheme because this is a direct solution anyway. For the two other DG formulations, the block Jacobi preconditioner with Gauss–Seidel gives the improved results shown in Table 8. Because the additional computational cost of evaluating the Gauss–Seidel terms is much less than the cost of inverting the diagonal blocks, it is definitely beneficial to add the Gauss–Seidel terms to the iteration.

VIII. Two-Dimensional Results

In this section, we investigate the performance of p-multigrid in multiple dimensions. We begin with Jacobi preconditioning using the spectral-element basis and then look at block Jacobi relaxation. In the last part of the section, we examine Gauss–Seidel iteration and line relaxation schemes. The results are obtained for both isotropic and high-aspect-ratio grids. The reason the line relaxation schemes are introduced is that they have been shown to be effective in finite volume solvers for high-aspect-ratio grids.

A. Jacobi

Table 9 shows the damping factors for various values of η. Compared to the one-dimensional results shown in Table 5, these results have damping factors closer to one. This behavior also occurs when using p-multigrid to solve continuous spectral-element formulations.[9] The explanation that is usually given for this is that a high-order finite element simulation corresponds to a discretization that is on a high-aspect-ratio mesh; the spacing of Gauss Legendre points near $\xi = -1$ or 1 goes as $1/p^2$ as compared to $1/p$ for a uniform spacing. This causes the same aspect-ratio difficulties that occur for geometric multigrid on high-aspect-ratio meshes; at the high wave numbers, there is a directional dependence in the eigenvalues, which results in poor damping of some of the high wave number modes. Although the performance decreases with p, at moderate values of p, the damping rates are still moderately good at ≈0.9.

B. Block Jacobi

Table 10 shows the damping factors for the block Jacobi relaxation scheme. Again, we see a significant reduction in performance in two dimensions. In one dimension, the modes that are undamped by block Jacobi are well represented in the coarse space. In two dimensions, there are modes that are high wave number in one direction and low wave number in the other that are not damped well by block Jacobi but not represented well in the coarse space either. These modes cause the performance to degrade. Because of the greater computational cost of block Jacobi, it may be more efficient to use the Jacobi iteration in two dimensions.

C. Gauss–Seidel

As is the case in one dimension, the Gauss–Seidel terms do little to improve the damping factors of the Jacobi iteration, but the block Gauss–Seidel iteration, shown in Table 11, again

Table 12 Multigrid damping factors in two dimensions with effect of aspect ratio for various iterative schemes and polynomial degree[a]

		$\Delta x/\Delta y$		
Scheme	p	1/10	1	10
Jacobi	2	0.997	0.90	0.997
Block Jacobi	2	0.978	0.58	0.988
Block Gauss–Seidel	2	0.96	0.44	0.96
Line solve	2	0.28	0.53	0.989
Line Gauss–Seidel	2	0.16	0.36	0.96
Jacobi	4	0.997	0.90	0.997
Block Jacobi	4	0.988	0.69	0.988
Block Gauss–Seidel	4	0.95	0.51	0.95
Line solve	4	0.51	0.61	0.987
Line Gauss–Seidel	4	0.32	0.43	0.95
Jacobi	8	0.9987	0.96	0.9987
Block Jacobi	8	0.991	0.85	0.991
Block Gauss–Seidel	8	0.96	0.73	0.96
Line solve	8	0.79	0.81	0.992
Line Gauss–Seidel	8	0.66	0.66	0.96

[a] DG formulation is the LDG, $\beta = 0$ scheme with $\eta = 4$.

improves significantly compared to results without the Gauss–Seidel terms (Table 10). Thus, in multiple dimensions, it is again beneficial to include the Gauss–Seidel terms in the block relaxation scheme.

D. Aspect Ratio Effects

The last issue we examine is the effect of mesh aspect ratio on the damping rates. The effect of aspect ratio on all of the schemes is nearly the same, and so we only show results for the LDG, $\beta = 0$ scheme. Table 12 shows the damping factors for various relaxation schemes as a function of aspect ratio. We have introduced two new relaxation schemes that are commonly used for high-aspect-ratio problems. The first is a block line solver that is a block tri- or pentadiagonal preconditioner that must be inverted along lines. It is pentadiagonal for the LDG, $\beta = 0$ scheme and tri-diagonal for the LDG one-sided scheme and the Bassi et al.[16] scheme. The lines are oriented in the x direction. The second scheme is similar except that the line updates are performed sequentially from bottom to top and each line uses the most recent information in its update, that is, it is a line solution in x and Gauss–Seidel in y.

Examining the results, we see that the line solvers are the only effective relaxation scheme when the mesh has a high aspect ratio. To get good performance, the lines must be aligned with the compressed direction of the mesh. The line solvers actually improve in performance as the mesh aspect ratio increases. Given that the cost of a line solution is much larger than evaluating Gauss–Seidel terms, it is highly beneficial to use the line Gauss–Seidel relaxation version. When the mesh is isotropic, the regular line solution actually performs worse than block Gauss–Seidel, which is much less expensive. For the $p = 8$ and $p = 4$ cases, we have checked that these results are reproducible using a V-cycle to $p = 1$. As was found in the one-dimensional case, the damping factors for the V-cycle are either identical or very close to those shown in Table 12.

IX. Conclusions

The p-multigrid is an effective way to solve DG formulations of the Poisson equation. A key finding is that the coarse space matrix operators must be derived by applying the restriction and prolongation operators to the fine space matrices. The rediscretization approach of reevaluating these matrices for each space often results in an unstable iteration.

Of the relaxation schemes evaluated, the Jacobi relaxation scheme with a spectral-element basis gives reasonable results on isotropic meshes. An alternative is the block Gauss–Seidel scheme. This scheme gives damping factors on the order of 0.5 for $p = 4$ and 0.7 for $p = 8$ in two dimensions. On high-aspect-ratio meshes, the most effective scheme is the line Gauss–Seidel iteration. This scheme gives aspect-ratio-independent results and damping factors on the order of 0.4 for $p = 4$ and 0.6 for $p = 8$.

A problem that remains is that the multigrid efficiency degrades if the polynomial space is coarsened beyond $p = 1$. For moderate p discretizations, after coarsening to $p = 1$, the remaining system may still be fairly large and expensive to solve directly. Standard geometric multigrid techniques could be easily applied to the $p = 0$ discontinuous system, but there is no obvious generalization of these techniques to the $p = 1$ discontinuous system.

References

[1] Rönquist, 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.

[2] Maday, Y., and Munoz, R., "Spectral Element Multigrid Part 2: Theoretical Justification," ICASE, Technical Rept. 88-73, 1988.

[3] Helenbrook, B. T., "A Two-Fluid Spectral Element Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 191, Nos. 3–5, 2001, pp. 273–294.

[4] Brooks, A. N., and Hughes, T. J. R., "Streamline Upwind/Petrov–Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier–Stokes Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, Nos. 1–3, 1982, pp. 199–259.

[5] Fischer, P. F., and Lottes, J. W., "Hybrid Multigrid/Schwarz Algorithms for the Spectral Element Method," *Journal of Scientific Computing*, Vol. 24, No. 1, 2005, pp. 45–78.

[6] Fidkowski, K., "A High-Order Discontinuous Galerkin Multigrid Solver for Aerodynamic Applications," M.S. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, June 2004.

[7] Oliver, T. A., "Multigrid Solution for High-Order Discontinuous Galerkin Discretizations of the Compressible Navier–Stokes Equations," M.S. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, Aug. 2004.

[8] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid Solution of High-Order Discontinuous Galerkin Discretizations of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 207, No. 1, 2005, pp. 92–113.

[9] Helenbrook, B. T., Atkins, H. L., and Mavriplis, D. J., "Analysis of p-Multigrid for Continuous and Discontinuous Finite Element Discretizations," AIAA Paper 2003-3989, June 2003.

[10] Reed, W. H., and Hill, T. R., "Triangular Mesh Methods for the Neutron Transport Equation," Los Alamos National Lab., Technical Rept. LA-UR-73-479, Los Alamos, NM, 1973.

[11] Bassi, F., and Rebay, S., "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 131, No. 2, 1997, pp. 267–279.

[12] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems," *SIAM Journal on Numerical Analysis*, Vol. 39, No. 5, 2002, pp. 1749–1779.

[13] Brezzi, F., Manzini, M., Marini, D., Pietra, P., and Russo, A., "Discontinuous Finite Elements for Diffusion Problems," *Francesco Brioschi (1824–1897) Convegno di Studi Matematici. Incontro di Studio*, No. 16, Istituto Lombardo di Scienze e Lettere, 1999, pp. 197–217.

[14] Cockburn, B., and Shu, C.-W., "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1998, pp. 2440–2463.

[15] Douglas, J., Jr., and Dupont, T., "Interior Penalty Procedures for Elliptic and Parabolic Galerkin Methods," *Computing Methods in Applied Sciences*, Vol. 58, Lecture Notes in Physics, Springer-Verlag, Berlin, 1976.

[16] Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., and Savini, M., "A High-Order Accurate Discontinuous Finite Element Method for Inviscid and Viscous Turbomachinery Flows," *Proceedings of the 2nd European Conference on Turbomachinery, Fluid Dynamics, and Thermodynamics*, edited by R. Decuypere and G. Dibelius, Technologisch Inst., Antwerp, Belgium, 1997, pp. 99–108.

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

[18] Hughes, T. J. R., *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Prentice–Hall, Englewood Cliffs, NJ, 1987, p. 433.

[19] Atkins, H. L., and Shu, C.-W., "Analysis of Preconditioning and Relaxation Operators for the Discontinuous Galerkin Method Applied to Diffusion," AIAA Paper 2001-2554, June 2001.

P. Givi
Associate Editor