Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 204 (2005) 520–532
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Algebraic multigrid for higher-order finite elements

J.J. Heys [a,*], T.A. Manteuffel [b], S.F. McCormick [b], L.N. Olson [c]

[a] Chemical and Materials Engineering Department, Arizona State University, Tempe, AZ 85287-6006, USA
[b] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, USA
[c] Division of Applied Mathematics, Brown University, USA

Received 15 July 2004; received in revised form 7 October 2004; accepted 19 October 2004
Available online 19 November 2004

Abstract

Two related approaches for solving linear systems that arise from a higher-order finite element discretization of elliptic partial differential equations are described. The first approach explores direct application of an algebraic-based multigrid method (AMG) to iteratively solve the linear systems that result from higher-order discretizations. While the choice of basis used on the discretization has a significant impact on the performance of the solver, results indicate that AMG is capable of solving operators from both Poisson's equation and a first-order system least-squares (FOSLS) formulation of Stoke's equation in a scalable manner, nearly independent of basis order, p, for $3 < p \leqslant 8$. The second approach incorporates preconditioning based on a bilinear finite element mesh overlaying the entire set of degrees of freedom in the higher-order scheme. AMG is applied to the operator based on bilinear finite elements and is used as a preconditioner in a conjugate gradient (CG) iteration to solve the algebraic system derived from the high-order discretization. This approach is also nearly independent of p. Although the total iteration count is slightly higher than using AMG accelerated by CG directly on the high-order operator, the preconditioned approach has the advantage of a straightforward matrix-free implementation of the high-order operator, thereby avoiding typically large computational and storage costs.

Keywords: Multigrid; Algebraic multigrid; Higher-order; Finite elements; Poisson; Stokes

1. Introduction

One of the goals in approximating the solution of elliptic partial differential equations is to formulate a method in which the computational cost scales optimally. More precisely, we seek a method for which the

* Corresponding author. Tel.: +1 303 492 2646; fax: + 1 303 492 4066.
E-mail addresses: jheys@asu.edu, heys@asu.edu (J.J. Heys), tmanteuf@colorado.edu (T.A. Manteuffel), stevem@colorado.edu (S.F. McCormick), lolson@dam.brown.edu (L.N. Olson).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.10.021

number of operations is proportional to the number of discrete unknown values. For elliptic problems with low-order discretizations (finite element or finite difference), this was achieved by using the multigrid algorithm developed by, for example, Brandt [1] and Hackbusch [2]. Multigrid, and other more recent multilevel algorithms (e.g. [3]), are often optimal due to their ability to properly handle different scales present in the problem – effectively reducing both smooth and oscillatory error.

The increased interest in developing optimal solution schemes for elliptic problems is paralleled by an increased interest in using higher-order discretization schemes. Standard isoparametric polynomial approximation theory [4] states that, under certain assumptions, the approximation error in the L_2-norm is bounded according to

$$\|u - u_h\|_0 \leqslant Ch^{p+1}\|u\|_{H^{p+1}(\Omega)}, \tag{1}$$

where $\|\cdot\|_{H^p}$ is the standard H^p Sobolev norm [4], $\|\cdot\|_0$ is the H^0 or L_2-norm, u is the exact solution of the PDE, and u_h is the solution of the discrete level h problem. Thus, under certain assumptions, if $u \in H^{p+1}(\Omega)$ and if a finite element basis of at least degree p is used, then we achieve $O(h^{p+1})$ convergence. This potentially high accuracy for sufficiently smooth problems has lead to extensive development of higher-order finite element methods, as e.g. spectral element methods [5].

Much of the current research in efficient solution methods for solving the linear system arising from higher-order discretizations relies on Schwarz-based methods, which consist of local block solves and a coarse-grid solve. The work of Widlund [6,7], Mandel [8], and others demonstrated that solver optimality is indeed achievable in that setting. However, these methods may still be computationally expensive since the local blocks are typically dense and solved with a direct method. More importantly, these methods often require a coarse-grid solve that can become relatively expensive, particularly since the coarse grid must be fairly dense. On the other hand, the work of Fischer [5,9–11] and others has overcome some of the high computational cost associated with such methods for certain classes of problems. Lottes and Fischer [12] have also conducted a comparison between Schwarz-based methods and multigrid methods.

Attempts have been made to apply multilevel ideas directly to higher-order finite element discretizations without using a Schwarz-based approach. The earliest approaches used coarse grids based on lower-order polynomials. For example, if a basis of polynomial order p is used, the coarse grids would consist of polynomials of order $p/2$, $p/4$, etc. Once the polynomial order was reduced to $p = 1$, the traditional geometric multigrid algorithm could then be used. The early results on 1-D problems are encouraging and show optimal solver performance [13,14]. However, the extension to higher dimensions results in non-optimal performance due to the dependence of the convergence factor, ρ, of the multigrid iteration using a Jacobi smoother on the order of the polynomial basis, p. More precisely, $\rho = 1 - c/p$, where c is a constant independent of p [13]. It was shown that using a Chebyshev acceleration scheme resulted in the convergence factor being $1 - c/\sqrt{p}$, but still not independent of p.

A third multilevel approach for solving higher-order discretizations is the multi-p preconditioner [15]. This technique is based on using the basis

$$\phi_0(\xi) = \frac{1-\xi}{2}, \tag{2}$$

$$\phi_1(\xi) = \frac{1+\xi}{2}, \tag{3}$$

$$\phi_j(\xi) = \sqrt{\frac{2j-1}{2}} \int_{-1}^{\xi} P_{j-1}(t)\,\mathrm{d}t, \tag{4}$$

where $P_{j-1}(t)$ is the Legendre polynomial of degree $j-1$, $j = 2,3,\ldots,p$. The modal basis functions generate an almost diagonal stiffness matrix in 1-D. While the numerical experiments based on this method

indicate that using the preconditioner results in an operator condition number that is independent of p, the total computational cost is not scalable. This is likely due to the fact that the multi-p V-cycle uses $p-1$ coarse grids, with each level having a grid corresponding to a polynomial order of $p-1$, $p-2,\ldots 1$. Relaxation on several relatively dense levels results in a computationally expensive algorithm.

A fourth method for solving matrices from higher-order discretizations, which is related to multilevel methods, is to precondition the higher-order operator, A_s, using a lower-order operator, A_f, obtained by overlaying the higher-order nodes with a lower-order finite element mesh. The idea of lower-order preconditioning was originally proposed by Orszag [16] and has been studied by others [17–19]. Numerical results by Fischer [10] show that the condition number of $A_f^{-1}A_s$ is bounded and that the bound is particularly low when an approximate, lumped mass matrix is used. Alternatively, Deville and Mund [18] show that bilinear finite elements are preferable to linears or triangles if the mass matrix is not lumped. For a single spectral element and $p = 40$, the condition number is 7.02 without mass-matrix lumping and only 2.35 with lumping. In either case, the preconditioned operator should be easily handled by a bounded number of Krylov iterations.

The final method for combining higher-order discretizations and multilevel solvers is to use an algebraic multigrid (AMG) solver. AMG is based on applying multigrid ideas directly to the discrete operator [20] and is especially helpfully in the case of unstructured or highly irregular grids. Early attempts at applying AMG directly to the high-order discretization operator, A_s, are discouraging [21]. However, as we show in this paper, this poor performance is the result of several obstacles that can be overcome with relatively modest adjustments, including the use of unevenly space nodes for the preconditioning basis. The goal of this paper is to explore this avenue and other aspects of the use of AMG on operators resulting from high-order discretizations (i.e., $1 < p \leqslant 8$).

2. Discretizations

The standard benchmark problem for studying the performance of linear solvers on matrices derived from the discretization of 2-D elliptic partial differential equations is Poisson's equation:

$$-\nabla^2 u = f. \tag{5}$$

Using the Galerkin finite element model, the weak form of (5) for a chosen basis $\mathscr{V} = (\psi_1, \psi_2, \ldots, \psi_n)$ is given by:

$$a(u_h, v_h) = (f, v_h) \quad \forall v_h \in \mathscr{V}, \tag{6}$$

where

$$a(u_h, v_h) = \int_\Omega \nabla u_h \cdot \nabla v_h, \tag{7}$$

$$(f, v_h) = \int_\Omega f v_h. \tag{8}$$

The weak problem (6) is equivalent to solving the following system of algebraic equations:

$$\mathbf{A}_h \mathbf{u}_h = \mathbf{f}_h, \tag{9}$$

where

$$\mathbf{A}_h = \left[a(\psi_j, \psi_i)\right]_{i,j=1}^{n}, \tag{10}$$

$$\mathbf{f}_h = \left[\int_\Omega f\psi_i\right]_{i=1}^{n}. \tag{11}$$

Matrix $\mathbf{A_h}$ is typically called the stiffness matrix. Likewise, we refer to

$$\mathbf{M}_h = \left[(\psi_j, \psi_i)\right]_{i,j=1}^{n} \tag{12}$$

as the mass matrix.

There are a variety of choices for the approximation basis, $\mathscr{V}$, which can loosely be divided into two groups: nodal and modal. The nodal basis is constructed using a Lagrangian interpolant such that $\psi_i(\xi_j) = \delta_{ij}$, where δ_{ij} is the Kronecker delta. The choice of node location does not typically affect the final approximation, since all bases of the same order span the same space, but the location greatly affects the condition of the discrete system. Choosing evenly spaced nodes is appealing since it enables straightforward adaptive *hp*-refinement. However, the resulting matrix can be extremely ill-conditioned, particularly as the polynomial order increases past 4 [22]. A second choice for node locations is the Gauss–Lobatto–Legendre (GLL) quadrature points. This choice is popular since the now approximate mass matrix becomes diagonal and the GLL points can be used for both quadrature and node locations. Unfortunately, the integration by quadrature is often not computed exactly, introducing additional error into the numerical approximation. (This aspect is beyond the scope of this paper, so exact quadrature is used throughout for both uniform and GLL distributions of points.) A third choice for node locations is the Chebyshev–Legendre (CL) points, given by

$$\xi_j = \cos\frac{\pi j}{N}, \quad 0 \leqslant j \leqslant N. \tag{13}$$

The location of the CL points is similar to the location of the GLL points and, in both cases, ill-conditioning of the matrix due to high-order polynomials is avoided as much as possible. The advantage of the CL points is that the points are somewhat nested. For example, the points associated with a 4th degree polynomial are a subset of those associated with an 8th degree polynomial. This property makes adaptive *p*-refinement more straightforward. For any of the nodal bases described, it is possible to overlay the higher-order element with a set of lower-order (bilinear) finite elements, as shown in Fig. 1.

The other option for defining the basis, as mentioned above, is the so-called modal basis. The most common approach is given by equations (2)–(4). This finite element approach is commonly referred to as the *p*-version of finite elements, and it has the advantage of producing a stiffness matrix that is nearly diagonal in 1-D due to the orthogonality of the derivatives of the basis functions. There are two main drawbacks to the method. First, the 1-D mass matrix is not diagonally dominant, and the 2-D stiffness matrix (produced as a tensor product of the 1-D mass and stiffness matrices) is not diagonal but still less dense than the equivalent nodal stiffness matrix. Second, because the solution only represents coefficients of the polynomial representing the solution, post-processing and visualization are less straightforward.

The first benchmark problem tested in this paper is the model Poisson equation.

Example 1.

$$-\nabla\cdot\nabla u = 0 \quad \text{in } \Omega, \tag{14a}$$

$$u = 0 \quad \text{on } \Gamma_1, \tag{14b}$$

$$u = \sin(\pi x) \quad \text{on } \Gamma_2, \tag{14c}$$

where $\Omega = [0,1]^2$, Γ_1 is the North, West and East boundaries of Ω, and Γ_2 is the South boundary. To maintain consistency with Lottes and Fischer [12], a regular spectral element mesh with 16×16 elements is used, unless otherwise noted.

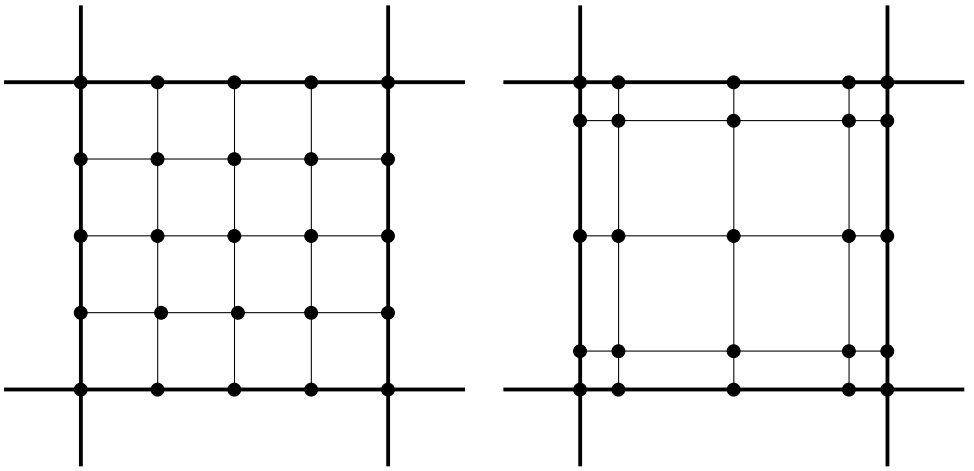

Fig. 1. Higher-order finite elements with a nodal basis – evenly spaced nodes (left) and approximately GLL nodes (right). In both cases, the spectral element, shown with bold lines, is overlaid with a set of bilinear finite elements, shown with thin lines and bold lines.

The second benchmark problem tested in this paper is Stokes equation for low Reynold's number flow.

Example 2.

$$-\nabla p + \nabla^2 \mathbf{u} = 0 \quad \text{in } \Omega, \tag{15a}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \tag{15b}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma, \tag{15c}$$

where p is the pressure scaled by viscosity and $\mathbf{u}$ is the velocity vector. The Galerkin finite element [5] model does not produce an H^1-elliptic form and multigrid schemes typically perform very poorly. An alternative formulation is based on reformulating the Stokes equation as a first-order system and minimizing the least-squares norm of residual equations in the resulting system – the so-called, first-order system least-squares (FOSLS) approach. Defining a matrix of new variables, $\mathbf{U}$, the first-order system for Stokes equation is

$$\mathbf{U} - \nabla \mathbf{u} = 0 \quad \text{in } \Omega, \tag{16a}$$

$$-\nabla p + \nabla \cdot \mathbf{U} = 0 \quad \text{in } \Omega, \tag{16b}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega. \tag{16c}$$

Because $\mathbf{U}$ is the gradient of $\mathbf{u}$ and we are solving a minimization problem, we can augment the first-order system of equations by the following consistent equations:

$$\nabla(tr(\mathbf{U})) = 0, \tag{17}$$

$$\nabla \times \mathbf{U} = 0, \tag{18}$$

where $tr(\mathbf{U}) = U_{11} + U_{22}$. The resulting least-squares functional is then

$$G(\mathbf{u}, \mathbf{U}, p) := \|\mathbf{U} - \nabla \mathbf{u}\|_0^2 + \| - \nabla p + \nabla \cdot \mathbf{U}\|_0^2 + \|\nabla \cdot \mathbf{u}\|_0^2 + \|\nabla(tr(\mathbf{U}))\|_0^2 + \|\nabla \times \mathbf{U}\|_0^2. \tag{19}$$

Using the FOSLS formulation, the minimization problem is H^1-elliptic [23], yielding optimal discretization error estimates and optimal multigrid convergence estimates. The Dirichlet boundary conditions (e.g. $\mathbf{u} = \mathbf{g}$) on the original problem (15a,15b) may be enforced weakly by adding them to the functional (19), or they may be strongly imposed on the finite element space. In either case, it is beneficial to include the consistent boundary condition: $\tau \cdot \mathbf{U} = \partial \mathbf{g}/\partial \tau$, where τ is the vector tangential to the boundary.

The FOSLS formulation results in 7 degrees of freedom (DOF) per node. Thus, the size of the finite element mesh for the test problem is reduced to an 8×8 element mesh. The domain is the unit square, $\Omega = (0,1)^2$, and the velocity components are set to zero on the North and South boundaries. On the West and East boundaries, the tangential velocity is set to zero and a parabolic profile is specified for the normal velocity. With these boundary conditions, the exact solution is in the finite element space if elements of degree 2 or greater are used. One advantage of the FOSLS formulation is that we are free to pick the finite element basis for velocity and pressure independently (i.e., there is no inf–sup condition). We thus choose the same space for velocity and pressure in all cases because the algorithm development is simplified.

3. AMG solver

The preconditioning strategy we follow is a multilevel approach in that some error components are reduced on the fine grid, while other error components are accounted for on successively coarser, and thus less expensive, grids. The classic multigrid scheme is *geometric* in nature, meaning coarse grids are selected based on the geometric location of the degrees of freedom. In a geometric multigrid (GMG) method, the coarse grids are know a priori, enabling one to choose intergrid transfer operators–interpolation and restriction–accordingly. Error components are described in a geometrical sense as well, usually as oscillatory or smooth. The goal in GMG is to properly define a relaxation strategy (e.g. Gauss–Seidel) to reduce the oscillatory errors on a given grid and rely on the predetermined interpolation to effectively represent the remaining smooth error components on coarser levels.

We employ an *algebraic* based multigrid method, where the underlying philosophy is quite different. The most popular published implementation of the algorithm, by Ruge and Stüben (RS) [24], is the implementation used here. AMG does not rely on the notion of an underlying ''physical'' grid. Instead, the multilevel process is realized from a purely algebraic standpoint in that only the connections in the matrix graph are used to determine intergrid transfer operators and to define coarse grids. Central to AMG is that we fix the relaxation strategy and define coarse grids and transfer operators on-the-fly to handle various components of the error. For a fixed relaxation process, such as point-wise Gauss–Seidel, some error is effectively eliminated, while other components of the error are reduced quite slowly. Smooth error, in the case of AMG, is then defined to be any error not reduced by relaxation. Using the nonzero structure of the matrix to determine the adjacency relationships between the unknowns, we can relate this idea of smooth error to strongly connected degrees of freedom. We say that u_i is connected to u_j if $a_{ij} \neq 0$. The magnitude of a_{ij} indicates how strong the connection is and how much influence the error at j has on the error in i in relaxation. We follow the classic RS [24] and define u_i to be *strongly connected* to u_j if

$$-a_{ij} \geqslant \theta \max_{k \neq i}(-a_{ik}). \tag{20}$$

Throughout the paper, a threshold of $\theta = 0.25$ is used. This value has been used successfully in a number of applications [25] and is a typical choice when other motivation is not present. The RS algorithm also allows us to control a number of other parameters in an effort to more accurately define strong connections, while maintaining a reasonable level of complexity. For example, so-called long range connections are not used in the results in this paper, meaning connections to connections are not considered: only direct connections may be used to define interpolation. The interpolation weights are defined as in the original RS [24] imple-

mentation: the problem is not considered to be a scalar problem, strong connections are distributed along all connections, and weak connections are distributed to the diagonal. We refer to [24,25] for a more in-depth discussion of these details, only mentioning that we adhere the RS approach for nearly all choices in defining interpolation. For systems of PDEs, we also note that the coarsening process is separate for each unknown (unlike the strict scalar case, where coarsening is based on only one variable). We choose the RS implementation due to its proven robustness and generality, noting that one of the main differences between the RS algorithm and other, newer algorithms is the method by which the coarse grid points are selected.

The AMG algorithm has setup costs associated with the automatic selection of coarse grid operators that other multilevel methods do not have in general. While this automatic selection by the algorithm makes implementing the algorithm into a given application simpler, it does add extra computations to every simulation. Unfortunately, it is not possible to get realistic a priori estimates of the setup costs for AMG, especially for general unstructured grids [26]. Experience has shown that in single processor environments, the setup costs are typically 1–10 V-cycles [26,27], and all AMG setup costs for simulations in the paper were in that range (typically close to 2 V-cycles). In parallel (distributed memory) environments, the setup phase can require significant communication between nodes and can require a larger fraction of that total computing time. However, estimates are even less reliable in the distributed memory setting.

AMG is also commonly used as a preconditioner for a Krylov subspace method such as the conjugate gradient (CG) or the generalized minimal residual (GMRES) method. The advantage to this scheme is that the Krylov method reduces the error in eigenmodes that are not being effectively reduced by multigrid. For this reason, the approach of using AMG with CG is often referred to as "CG accelerated" multigrid (AMG/CG). One variation is to base the preconditioning operator on a matrix other than the discretization matrix, that is, perform the AMG preconditioning on a different operator from that being solved by CG. For example, in this paper, we use the AMG preconditioner on an operator, A_f, obtained by overlaying the spectral element nodes with a bilinear finite element mesh. The operator from the original spectral element problem, A_s, is treated by CG using AMG on A_f as a preconditioner.

4. Results

Table 1 summarizes the convergence factors obtained by solving Poisson's equation in 2-D on the unit square with a 16×16 spectral element grid (Example 1). The convergence factor is defined as the average fractional change in the residual for a multigrid cycle. For example, a convergence factor of 0.1 indicates that the residual decreases by one order of magnitude each multigrid cycle on average over all the computed cycles. Four different finite element bases were used, including three nodal bases: GLL points, CL points, evenly spaced nodes, and a modal basis based on (2)–(4). V(1,1)-cycles (1 pre- and 1 post-smoothing sweep)

Table 1
Average AMG convergence factors on spectral element operator from Poisson's equation

P	DOF	GLL	CL	Even	Modal
2	1089	0.11 (0.04)	0.11 (0.04)	0.11 (0.04)	0.69 (0.26)
3	2401	0.14 (0.06)	0.11 (0.06)	0.20 (0.08)	0.75 (0.29)
4	4225	0.11 (0.04)	0.22 (0.10)	0.28 (0.10)	0.68 (0.32)
5	6561	0.08 (0.03)	0.12 (0.05)	0.78 (0.40)	0.69 (0.37)
6	9409	0.13 (0.06)	0.18 (0.08)	0.95 (0.67)	0.71 (0.38)
7	12769	0.16 (0.05)	0.38 (0.14)	0.98 (0.86)	0.79 (0.42)
8	16641	0.17 (0.06)	0.19 (0.09)	0.98 (0.94)	0.81 (0.45)

Different finite element bases are used: GLL points, CL points, evenly spaced nodes, and a modal basis. The numbers in parentheses indicate convergence factors using CG acceleration.

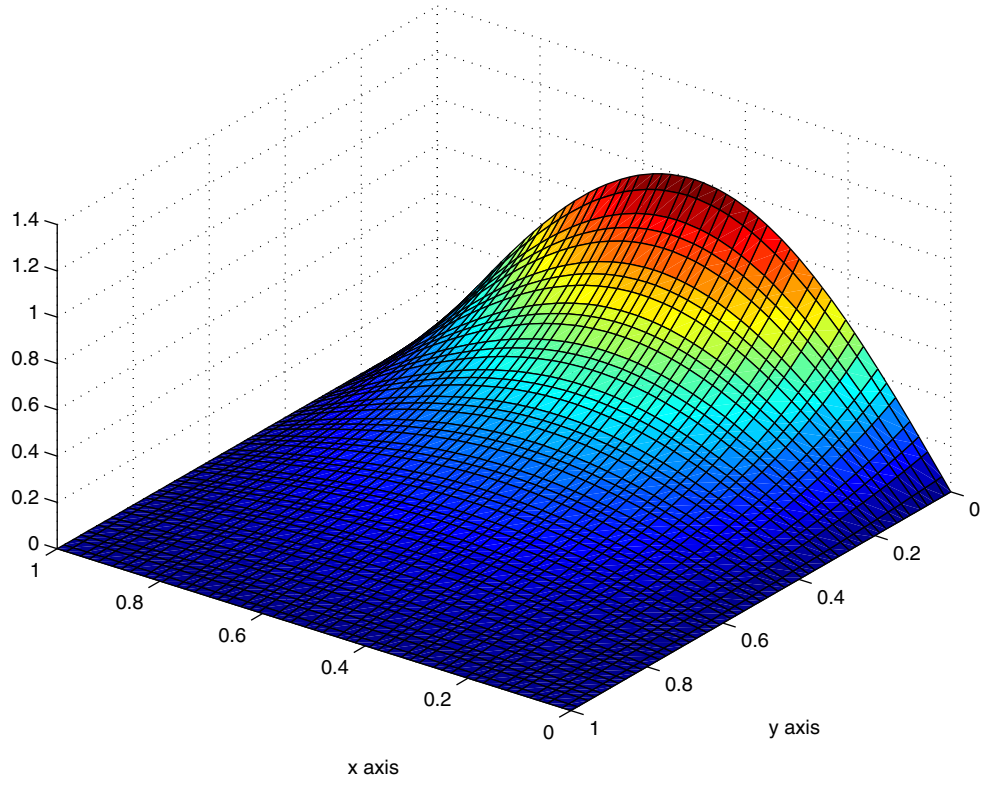

Fig. 2. Poisson test problem using a cubic ($p = 3$) nodal basis on the GLL nodes.

are used in the AMG algorithm in each case. As shown in Table 1, AMG performance on the spectral element operator using both GLL and CL nodal distributions is very good. In fact, convergence factors are approximately independent of p and, when coupled with CG acceleration, the results are even more convincing. Fig. 2 shows the approximate solution using the GLL nodal basis and $p = 3$. The nodal basis using evenly spaced nodes resulted in poor AMG performance due to the operator being ill-conditioned. The modal basis also performed poorly, likely a result of the lowest, smooth eigenmode not being represented by an approximately constant vector in this modal basis – a fundamental assumption in AMG.

The approximate independence of the convergence factors to p in Table 1 for the GLL and CL basis function is not consistent with past efforts at applying multilevel solvers to higher-order finite elements. Specifically, it is not consistent with methods which chose coarse grids by halving the basis order on each element, i.e., course grids of $p/2$, $p/4$, etc. Fig. 3 shows the coarse grids chosen by AMG for a 2×2 element Poisson problem with $p = 4$ and 5. While nothing conclusive can be drawn from such a small test problem, two observation can be made. First, the coarse grids chosen for the $p = 5$ case are more regular (i.e., every other point is chosen) than those chosen for the $p = 4$ case. This may help explain the better performance of the solver in the $p = 5$ case. Second, when comparing the two finest grids (neglecting boundary points), AMG uses approximately 50% of the total finer grid points in the coarser grid (cf. Fig. 3). This is significantly more than using $p/2$ for the coarser grid, in which only 25% of the points are retained. This may help explain greater p independence of AMG.

To test the performance of AMG on higher-order elements and a system of equations, a FOSLS formulation of Stokes equation (Example 2) is used with a nodal basis based on the GLL nodes and a mesh of

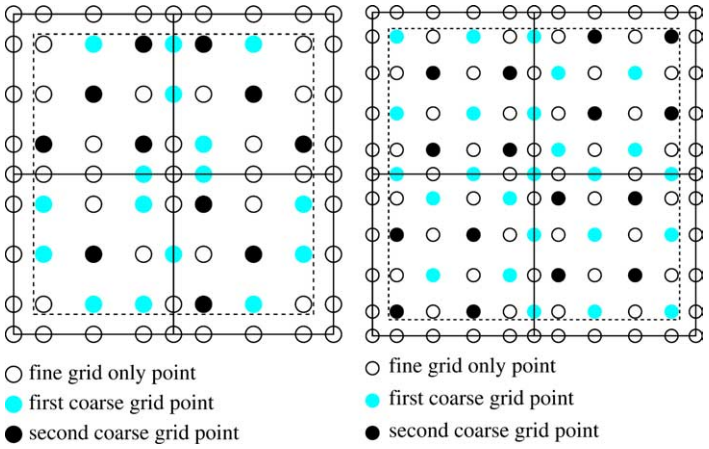

Fig. 3. The coarse grids selected by the AMG algorithm for $p = 4$ (left) and $p = 5$ (right) for a 2×2 mesh and the Poisson problem. AMG uses two levels of coarsening for this small problem. Points outside the dotted line are on the boundary and not in the operator.

8×8 spectral elements on the unit square. The AMG algorithm is modified slightly from the original RS algorithm so that coarsening can be applied to each of the seven unknowns separately. The results are summarized in Table 2. Fig. 4 shows the (exact) solution using bicubic basis functions. Without CG acceleration, the AMG convergence factors are independent of p, but the convergence is slow. However, to put this in context, geometric multigrid applied to this FOSLS formulation of Stokes equation with bilinear elements yields convergence factors in the range of 0.75–0.85 [28]. With CG acceleration, the convergence factors remain independent of p, and the convergence is much faster. This large benefit in using CG acceleration is consistent with what has been traditionally observed even using low-order discretizations. The final column in Table 2 is the operator complexity (OC), which is used to describe the size of all the coarse grid operators combined and compare that size to the fine grid operator. The OC is defined to be the ratio of the size (number of non-zeros) of all operators (coarse plus fine operators) to the size of the fine grid operator. A complexity of 1 would imply only storing the fine grid operator. A complexity of 2 implies that the size of all coarse operators combined is equal to the size fine grid operator.

Compared to low-order discretizations, the operator from higher-order discretizations is expensive both in the number of computations required to build and in the storage costs. Both costs typically scale as $O(p^4)$ in 2-D. A matrix-free approach to AMG is not achievable for the RS implementation. However, as mentioned above, AMG preconditioning can be applied to a low-order operator, A_f, thus minimizing the cost of

Table 2
AMG performance on a FOSLS formulation of Stokes equation

P	DOF	Avg. conv. factor	Avg. conv. factor with CG acceleration	Operator complexity
1	567	0.89	0.45	1.4
2	2023	0.94	0.58	1.7
3	4375	0.94	0.59	2.1
4	7623	0.95	0.62	1.7
5	11767	0.95	0.65	1.7
6	16807	0.95	0.66	1.6
7	22743	0.95	0.66	1.6
8	29575	0.95	0.67	1.6

The mesh is 8×8 spectral elements with a nodal basis defined by the GLL nodes.

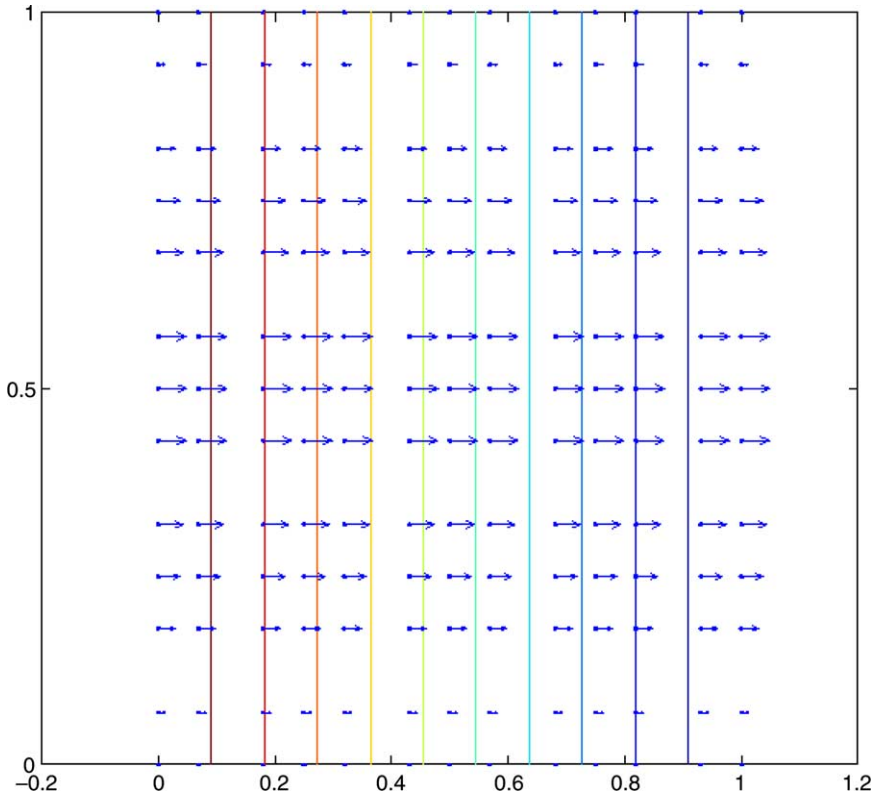

Fig. 4. Stokes test problem using a cubic ($p = 3$) nodal basis on the GLL nodes. 4×4 spectral elements were used to generate the plot.

building and storing a matrix for solving by AMG. The spectral operator, A_s, can then be solved using a PCG algorithm in a matrix-free approach, avoiding the high cost of storing the full spectral matrix. Table 3 summarizes the numerical performance when using AMG as a preconditioner on A_f and then performing CG iterations on A_s for the Poisson test problem (Example 1) with 16×16 elements. The convergence factors appear nearly independent of p, with the exception of the $p = 4$ basis. The convergence factors are not as low as applying AMG/CG directly to the spectral operator, a reflection of using a less effective preconditioner, but the benefit of not having to build and store the higher-order operator may outweigh the cost of additional iterations. All iteration counts are based on the number of iterations required to reduce the

Table 3
Performance of CG on Poisson's equation with and without an AMG preconditioner based on a $p = 1$ discretization

P	DOF	Avg. conv. factor	AMG/CG iterations	Operator complexity	CG iterations without AMG
1	289	0.03	4	1.3	15
2	1089	0.25	10	1.4	46
3	2401	0.30	12	3.6	74
4	4225	0.57	23	2.5	110
5	6561	0.34	12	3.3	147
6	9409	0.34	12	3.0	187
7	12769	0.38	14	3.8	234
8	16641	0.39	14	3.7	279

Table 4
Performance of CG on Poisson's equation with and without an AMG preconditioner based on a $p = 1$ discretization

P	DOF	Avg. conv. factor	AMG/CG iterations	Operator complexity	CG iterations without AMG
1	567	0.46	16	1.4	98
2	2023	0.69	34	1.5	308
3	4375	0.72	39	2.9	496
4	7623	0.84	69	2.4	699
5	11767	0.77	47	3.0	917
6	16807	0.79	52	3.4	1150
7	22743	0.79	54	3.8	1397
8	29575	0.81	57	3.8	1651

The mesh size is 8×8 spectral elements with a GLL nodal basis.

residual by a factor of 1×10^{-6}. This criterion is reasonable because, in practice, one needs to reduce the residual further for high-order elements than for low-order elements in order to reach the accuracy level of the approximation.

The choice between method (1), which applies AMG/CG directly to the higher-order operator, A_s, and method (2), which applies the AMG preconditioning to A_f and applying PCG to A_s, is based on memory available and polynomial order. The cost of a V-cycle in AMG scales as $O(p^4)$ in method (1) and $O(p^2)$ in method two, but method (1) usually requires few total V-cycles. Looking at computational time, we found that for the Poisson test problem, the solve time was lower for method (1) when $p < 4$. However, for $p > 4$, method (2) required less solve time even though the total number of V-cycles was larger. In both cases, CG acceleration was used, and the computational time was equal (2.4 s on 1.8 GHz Pentium 4) with $p = 4$. Method (2) also has the potential for requiring less memory, so it may be the only choice on smaller memory systems, independent of p.

The second method can also be applied to the FOSLS formulation of Stokes equation (Example 2). Table 4 summarizes the performance of CG on the Stokes system with and without preconditioning by AMG on the operator A_f. Again, the convergence factor is independent of p once $p > 3$, but it is slower than simply using an AMG/CG cycle on the full spectral matrix as the preconditioner (cf., Table 2). The operator complexities are large, but that is partially due to the lower-order operator being much sparser than the higher-order operator used in Table 2. Further, new coarse-grid selection algorithms are being developed that maintain low convergence factors while reducing the complexity [29,30].

5. Conclusions

The use of an AMG algorithm to solve linear problems associated with higher-order finite element discretizations is both computationally efficient and relatively simple to implement considering the widespread availability of AMG algorithms. Two different methods were explored in the paper: (1) AMG and AMG/CG were applied directly to the operator associated with a higher-order finite-element discretization; and (2) CG was applied to the higher-order operator, A_s, while AMG was used as a preconditioner on the lower-order operator, A_f, assembled using the same nodes as the higher-order operator. In both cases, the results were independent or nearly independent of the basis order, p, for $3 < p \leqslant 8$. The choice between the methods is a function of memory available and p where method (2) is faster at larger p ($p > 4$ for the Poisson test problem).

There are other methods that achieve convergence factors that are independent of p and, in fact, many of them have demonstrated independence over a much larger range of p [12]. The number of iterations required by Schwarz type methods is similar to that required by our methods for these simple test problems

(cf., [12]). The main disadvantages of the methods presented here are the requirement of assembling and storing an entire operator. Since AMG builds coarse operators based on the fine-grid operator, current techniques require that the global operator be assembled, but, as we have shown, only building a low-order operator may be necessary. A second disadvantage of AMG is the overhead cost associated with picking coarse operators and building interpolation operators. However, in many cases, these disadvantageous are more than offset by the simplicity of this approach and the flexibility it provides, especially in cases of adaptive refinement and unstructured grids.

Acknowledgments

This work was motivated by a talk with Paul Fischer at the 2004 Copper Mountain Conference. The work was sponsored by the Department of Energy under Grant Nos. DE-FG03-94ER25217 and DE-FC02-01ER25479, and the National Science Foundation under VIGRE Grant Nos. DMS-9810751 and DMS-9977371.

References

[1] A. Brandt, Multi-level adaptive solutions to boundary value problems, Math. Comput. 31 (1977) 333–390.
[2] W. Hackbusch, Multigrid Methods and ApplicationsComputational Mathematics, vol. 4, Springer, Berlin, 1985.
[3] P. Bjørstad, O. Widlund, Iterative methods for the solution of elliptic problems on regions partitioned in substructures, SIAM J. Numer. Anal. 23 (6) (1986) 1097–1120.
[4] S. Brenner, L. Scott, The Mathematical Theory of Finite Element Methods, second ed., Springer, New York, 2000.
[5] M. Deville, P. Fischer, E. Mund, High-Order Methods for Incompressible Fluid FlowCambridge Monographs on Applied and Computational Mathematics, vol. 9, Cambridge University Press, Cambridge, UK, 2002.
[6] M. Dryja, B. Smith, O. Widlund, Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions, SIAM J. Numer. Anal. 31 (1994) 1662–1694.
[7] L. Pavarino, O. Widlund, A polylogarithmic bound for an iterative substructuring method for spectral element in three dimensions, SIAM J. Numer. Anal. 33 (1996) 1303–1335.
[8] J. Mandel, Two-level domain decomposition preconditioning for the *p*-version finite element method in three dimensions, Int. J. Numer. Meth. Eng. 29 (1990) 1095–1108.
[9] P. Fischer, E. Ronquist, Spectral element methods for large-scale parallel Navier–Stokes calculations, Comput. Meth. Appl. Mech. Eng. 116 (1994) 69–76.
[10] P. Fischer, An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations, J. Comput. Phys. 133 (1997) 84–101.
[11] H. Tufo, P. Fischer, Fast parallel direct sovers for coarse grid problems, J. Parallel Distr. Comput. 61 (2001) 151–177.
[12] J. Lottes, P. Fischer, Hybrid multigrid/Schwarz algorithms for the spectral element method, J. Sci. Comp. (to appear).
[13] Y. Maday, R. Muñoz, Numerical analysis of a multigrid method for spectral approximations, in: M. Hussaini, D. Dwoyer, R. Voigt (Eds.), Proceedings of the 11th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, vol. 323, Springer, New York, 1989, pp. 389–394.
[14] E. Rønquist, A. Patera, Spectral element multigrid. I. formulation and numerical results, J. Sci. Comp. 2 (1987) 389–406.
[15] N. Hu, X.-Z. Guo, I. Katz, Multi-*p* preconditioners, SIAM J. Sci. Comput. 18 (1997) 1676–1697.
[16] S. Orszag, Spectral methods for problems in complex geometries, J. Comput. Phys. 37 (1980) 70–92.
[17] M. Deville, E. Mund, Cheybshev pseudospectral solution of second-order elliptic equations with finite element preconditioning, J. Comput. Phys. 60 (1985) 517–522.
[18] M. Deville, E. Mund, Finite-element preconditioning for pseudospectral solutions of elliptic problems, SIAM J. Sci. Stat. Comput. 11 (1990) 311–342.
[19] S. Beuchler, Multigrid solver for the inner problem in domain decomposition methods for *p*-fem, SIAM J. Numer. Anal. 40 (2002) 928–944.
[20] A. Brandt, S. McCormick, J. Ruge, Algebraic Multigrid (AMG) for Sparse Matrix Equations, Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.

[21] J. Ruge, AMG for higher-order discretizations of second-order elliptic problems, in: Abstracts of the Eleventh Copper Mountain Conference on Multigrid Methods. Available from: <http://www.mgnet.org/mgnet/Conferences/CopperMtn03/Talks/ruge.pdf>, 2003.
[22] P. Šolín, K. Segeth, I. Doležel, Higher-Order Finite Element Methods, Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, FL, 2004.
[23] Z. Cai, T. Manteuffel, S. McCormick, First-order system least squares for the Stokes equations, with application to linear elasticity, SIAM J. Numer. Anal. 34 (5) (1997) 1727–1741.
[24] J. Ruge, K. Stüben, Algebraic multigrid, in: S.F. McCormick (Ed.), Multigrid Methods, Frontiers in Applied Mathematics, vol. 3, SIAM, Philadelphia, 1987, pp. 73–130.
[25] W. Briggs, V. Henson, S. McCormick, A Multigrid Tutorial, second ed., SIAM, Philadelphia, 2000.
[26] K. Stüben, Algebraic multigrid (AMG): experiences and comparisons, Appl. Math. Comput. 13 (1983) 419–451.
[27] A. Cleary, R. Falgout, V.E. Henson, J. Jones, T. Manteuffel, S. McCormick, G. Miranda, J. Ruge, Robustness and scalability of algebraic multigrid, SIAM J. Sci. Comput. 21 (2000) 1886–1908.
[28] J. Heys, T. Manteuffel, S. McCormick, J. Ruge, First-order systems least squares (FOSLS) for coupled fluid-elastic problems, J. Comput. Phys. 195 (2) (2004) 560–575.
[29] K. Stüben, A review of algebraic multigrid, J. Comput. Appl. Math. 128 (2001) 281–309.
[30] H.D. Sterck, U. Yang, Reducing complexity in algebraic multigrid, in: Abstracts of the Eighth Copper Mountain Conference on Iterative Methods, 2004.