SIAM J. SCI. COMPUT.
Vol. 41, No. 5, pp. S2–S18

SCALABLE LOW-ORDER FINITE ELEMENT PRECONDITIONERS FOR HIGH-ORDER SPECTRAL ELEMENT POISSON SOLVERS*

PEDRO D. BELLO-MALDONADO† AND PAUL F. FISCHER†

Abstract. Low-order finite element (FE) systems are considered as preconditioners for spectral element (SE) discretizations of the Poisson problem in canonical and complex domains. The FE matrices are based on the same mapped set of Gauss–Lobatto–Legendre (GLL) points as the SE discretization. Three different versions of the preconditioner based on combinations of the low-order stiffness and mass matrices are tested for 2D and 3D geometries. When building the preconditioning operators, a new meshing approach that allows elements to overlap without needing to fill out the volume of the mesh is explored and shown to be better than traditional schemes. These preconditioners are robust with respect to cell aspect ratio, demonstrate bounded iteration counts with h- and p-refinement, and have lower iteration counts than scalable hybrid-Schwarz multigrid schemes currently used in production-level SE simulations. Overall costs for large-scale parallel applications are dependent on fast and robust solvers for the sparse FE systems. Algebraic multigrid is shown to offer a pathway to realizing a robust and fast preconditioning strategy in this context.

Key words. spectral element method, finite element method, high-order, low-order, preconditioner, algebraic multigrid

AMS subject classifications. 65F08, 65M70, 65M60, 65M55, 76D05, 65Y05

DOI. 10.1137/18M1194997

1. Introduction.

Since the seminal work of Orszag [29] and Patera [31], application of high-order spectral methods for general domains has gained prominence as one of the most effective means for simulation of transport phenomena such as turbulence, thermal transport, and electromagnetics. High-order spatial discretizations for simulation of problems having scale interactions spanning orders of magnitude offer significant computational savings over equivalent low-order discretizations at the same resolution (number of gridpoints) as noted by Kreiss and Oliger in the early 1970s [21] and later by Gottlieb and coworkers [22].

In $d = 3$ space dimensions, the spectral multidomain methods of Orszag [29] and the spectral element method (SEM) of Patera lead to system matrices that nominally have $\mathcal{O}(EN^6)$ nonzeros for polynomial expansion order N in each of E subdomains (elements). Crucially, for partial differential equations in $\mathbb{R}^3$, the fastest solvers are typically based on iterative schemes that do not require explicit computation of the systems matrices. Rather, they require only the *action* of the matrix on a vector—one need only return the product $\underline{w} = A\underline{p}$ when solving a system of the form $A\underline{x} = \underline{b}$. In this matrix-free context, Orszag [29] observed that the true advantage of Fourier- or Chebyshev-based monodomain spectral methods derived from their tensor-product structure, which reduced nominal matrix-vector product costs from $\mathcal{O}(N^6)$ to $\mathcal{O}(N^4)$ through the use of tensor-product factorizations. The use of fast transforms could further reduce the cost to near optimal complexity of $\mathcal{O}(N^3 \log N)$, but the savings was relatively small—only $\mathcal{O}(N/\log N)$ compared with the $\mathcal{O}(N^2)$ savings derived from tensor product forms. In [29], Orszag showed that these savings were applica-

*Received by the editors June 19, 2018; accepted for publication (in revised form) December 12, 2018; published electronically October 29, 2019.
https://doi.org/10.1137/18M1194997

†Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (belloma2@illinois.edu, http://belloma2.web.engr.illinois.edu/, fischerp@illinois.edu, http://fischerp.cs.illinois.edu/).

ble even in the case of deformed (nonseparable) domains. He further showed that, for the Poisson problem (but not the advection problem), effective *sparse* preconditioners could be built upon low-order discretizations. If the solutions for the low- and high-order discretizations were based on the same nodal points, then, under certain circumstances, the condition number of the preconditioned system was bounded independent of N, with asymptotic behavior $\kappa_P \sim \pi^2/4$.

This bounded condition number was subsequently observed by many authors under a variety of spectral and low-order discretization pairs, most notably in early work by Deville and Mund [6, 7, 8], who suggested basing the low-order discretization on the finite element method (FEM). Canuto and Quarteroni [3] and later Kim and Parter [23, 24] rigorously established the $\pi^2/4$ bound for the Legendre-based spectral Laplace operator preconditioned by low-order finite differences. Fischer [16] demonstrated the importance of using P_1-type or Q_1-type finite elements with numerical integration (NI), i.e., lumped mass matrix, to attain the $\pi^2/4$ result. More recently, Canuto, Gervasio, and Quarteroni [1] investigated a variety of P_1 and Q_1 configurations, with and without NI, in both strong and weak form, for $d = 2$ and $d = 3$ space dimensions. They identified which cases yielded the lowest condition numbers and also introduced the notion of an *iterative condition number*, taken to be the ratio of the maximum to minimum modulus eigenvalues. Olson [28] worked on preconditioning high-order discretizations by low-order collocation methods coupled with algebraic multigrid (AMG), as we consider here. He also studied meshing techniques based on Delaunay triangulation in order to handle the low-order collocation for the case of high-order triangular and tetrahedral elements. His work differs from ours in two respects. First, while he studied the case of rectangular and hexahedral high-order elements, his collocation scheme relied on Q_1 elements, which were found to be underperforming in [1]. Our approach relies on P_1 discretizations, which yields a lower condition number as demonstrated in [1]. Second, our method proposes a new meshing technique specifically targeted for rectangular and hexahedral elements that has not been proposed before.

Despite the allure of a bounded condition number (and hence, bounded iteration count), low-order preconditioning has not been widely adopted in production codes. The principal reason is that the low-order system is global and unstructured, albeit less dense than its spectral element (SE) counterpart. In $\mathbb{R}^3$, one must therefore use iterative methods to solve the low-order systems, which generally implies using either domain decomposition-based preconditioners or AMG (e.g., [16, 28]). Incomplete factorizations such as ILU and ICCG have also been considered [1, 17], but these are generally not effective in $\mathbb{R}^3$ where performance is most critical. As noted in [26], one of the reasons that iterative solution of the sparse problems is so challenging is that the tensor-product of the Gauss–Lobatto–Legendre (GLL) points (which form the nodal basis for the Nth-order expansions in the SEM) leads to high-aspect ratio cells that defeat any pointwise smoothing strategies for geometric multigrid. Indeed, the choice of smoothing is critical to the success of spectral multigrid methods as noted by Heinrichs [20] and Shen [32], who proposed line smoothers, and by Fischer and coworkers [16, 18, 19, 26], who proposed block smoothers based on the fast diagonalization method [4, 27]. Optimal coarsening strategies in the context of AMG with GLL point distributions are discussed in [25].

In this work, we extend the study of Canuto, Gervasio, and Quarteroni [1] to investigate optimal low-order stencil choices and to explore the potential of highly optimized AMG codes for rapid solution of the preconditioning problem, with a particular emphasis on large unstructured problems in $\mathbb{R}^3$. Our tests are all in the context

of preconditioned Krylov subspace iteration for Poisson problems in complex domains. Most of the problems have homogeneous Neumann conditions on all but a small subset of the domain boundary, which is a feature that is characteristic of pressure-Poisson subproblems in solution of the incompressible Navier–Stokes equations, which are our primary application of interest (and a fundamental driver for using high-order discretizations). Because we are using a projection-based contractor (e.g., GMRES or conjugate gradients) for solution of the high-order system, $A\underline{x}=\underline{b}$, we do not need to solve the low-order system exactly. In our studies, preconditioning with a single AMG V-cycle generally yields optimal performance and is used throughout unless otherwise indicated.

The structure of the paper is as follows. In section 2 we outline the spectral element formulation for the Poisson problem in $\mathbb{R}^3$, as well as describe several FEM-based preconditioners and their construction. Numerical and performance results for a range of two-dimensional (2D) and three-dimensional (3D) problems are demonstrated in section 3, and a brief summary is presented in section 4.

2. Methods. We consider the Poisson equation in $\mathbb{R}^d$, $d = 2$ or $d = 3$,

$$-\nabla^2 u = f \quad \text{for } u, f : \Omega \subset \mathbb{R}^d \to \mathbb{R}. \tag{2.1}$$

Our primary application of interest is the simulation of incompressible flows in which the pressure substep requires a Poisson solve such as (2.1). This pressure solve is typically the most expensive substep in the Navier–Stokes time advancement and is characterized by a Poisson problem with Neumann conditions everywhere or almost everywhere. In the former case, which corresponds to a closed domain, one seeks solutions having zero mean. In the latter case, typically only a small subset of the domain boundary, $\partial\Omega_D \subset \partial\Omega$, corresponding to a flow outlet, has Dirichlet conditions. Thus, an important consideration in developing preconditioners for flow applications is to test domains where Neumann conditions predominate. For an iterative solution of such problems, the homogeneous error equations do not benefit from having the error pinned to zero on $\partial\Omega$, and the error decay rates are consequently diminished, which makes these problems significantly harder than their all-Dirichlet counterparts.

The SE discretization of (2.1) is based on the weak form: *Find* $u \in X_0^N$ *such that*

$$(\nabla v, \nabla u)_N = (v, f)_N \quad \forall\, v \in X_0^N, \tag{2.2}$$

where $(\cdot,\cdot)_N$ denotes the discrete L^2 inner product, and X_0^N is the finite-dimensional approximation space comprising the SE basis functions, $\{\phi_j(\mathbf{x})\}$, $j = 1, \ldots, n$, which vanish on $\partial\Omega_D$. Using these basis functions to represent $u(\mathbf{x}) = \sum u_j \phi_j(\mathbf{x})$ leads to the matrix form of (2.2),

$$A\underline{u} = B\underline{f}, \tag{2.3}$$

with respective stiffness and mass matrices, $A_{ij} := (\nabla\phi_i, \nabla\phi_j)_N$ and $B_{ij} := (\phi_i, \phi_j)_N$.

Within this variational framework, the SEM is characterized by the choice of basis functions and quadrature rule. For $d = 2$, Ω is tessellated by nonoverlapping isoparametric quadrilateral elements, Ω^e, $e = 1, \ldots, E$, that are images of $\hat{\Omega} := [-1, 1]^2$, given by $\mathbf{x}^e(r, s) = \sum_{ij} \mathbf{x}_{ij}^e h_i(r) h_j(s)$, where the summation ranges are 0 to N. The local one-dimensional (1D) basis functions, $h_i(r)$ and $h_j(s)$, are Nth-order Lagrange polynomials based on the GLL quadrature points $\xi_j \in [-1, 1]$. The scalar fields v and

u take a similar form. That is, on Ω^e, $u(\mathbf{x}) = u^e(\mathbf{x}^e) = \sum_{ij} u^e_{ij} h_i(r) h_j(s)$. C^0 continuity across element interfaces is enforced by requiring $u^e_{ij} = u^{\hat{e}}_{\hat{i}\hat{j}}$ whenever $\mathbf{x}^e_{ij} = \mathbf{x}^{\hat{e}}_{\hat{i}\hat{j}}$. Following standard finite element assembly procedures [5, 33], this requirement leads to $A = Q^T A_L Q$, where Q is a Boolean matrix mapping global degrees-of-freedom to their elemental counterparts, and A_L=block-diag(A^e) is a block-diagonal matrix comprising local stiffness matrices, A^e, $e = 1, \ldots, E$. The mass matrix has a similar form, $B = Q^T B_L Q$. Typically, the SE quadrature is based on the GLL nodal points, which results both in reduced overhead for matrix-free evaluation of local products $A^e \underline{u}^e$ and in a diagonal mass matrix, B. We remark that, for deformed elements, the local stiffness matrices A^e are completely full if formed, which precludes direct use of AMG for (2.3). Fortunately, tensor-product sum factorization [29] implies that $A\underline{u}$ can be evaluated in only $O(EN^{d+1})$ work and $O(EN^d) = O(n)$ storage (with small constants). As noted in [1], the SEM is a combination of a stable (GLL) nodal basis for Q_N elements with nodal integration; full details may be found elsewhere (e.g., [2, 5]).

2.1. Preconditioners. In addition to A and B, we introduce finite element–based matrices A_F and B_F that have entries

$$A_{F,ij} := (\nabla\psi_i, \nabla\psi_j), \quad B_{F,ij} := (\psi_i, \psi_j), \tag{2.4}$$

where $\psi_j(\mathbf{x})$ is a linear finite element basis function based on a triangulation of the mapped GLL points, $\mathbf{x}^e_{ij}$, from the SE discretization, and $(\cdot, \cdot)$ is the standard (exact) L^2 inner product for the linear basis functions. We will also consider the diagonal (lumped) mass matrix $B_d := diag(\sum_j B_{F,ij})$.

We are interested in developing a preconditioner M that will reduce the iterative condition number $\kappa_P := \kappa(M^{-1}A)$ at a reasonable cost for large unstructured SE formulations for $d = 2$ and $d = 3$. We consider three such systems:

$$M^{-1} = A_F^{-1} \qquad \text{Weak preconditioner, } P^w, \tag{2.5}$$

$$M^{-1} = A_F^{-1} B_F B^{-1} \qquad \text{Strong preconditioner, } P^s, \tag{2.6}$$

$$M^{-1} = A_F^{-1} B_d B^{-1} \qquad \text{Strong diagonal preconditioner, } P^{sd}. \tag{2.7}$$

With these preconditioners, the bulk of the effort is shifted to (repeated) solutions in A_F. Because B is diagonal, the cost for applying the strong preconditioners is almost the same as for P^w. In particular, the extra cost of the strong diagonal preconditioner is effectively nil since $B_d B^{-1}$ is diagonal.

The advantage of A_F over A is in that it is sparse, with a fixed number of nonzeros per row, in contrast to A, which nominally has $O(N^d)$ nonzeros per row. AMG or other scalable sparse solvers can thus readily be applied to solve systems in A_F. As discussed in section 1, κ_P is in many cases bounded, which implies that a fixed number of preconditioned conjugate gradient or (nonrestarted) GMRES iterations are required to solve (2.2). In section 3 we further demonstrate that effective preconditioning can be realized by replacing the system solves A_F^{-1} with a single AMG V-cycle.

2.2. Choice of FEM spaces. The effectiveness of A_F as a preconditioner depends strongly on the choice of space $X_0^h := \text{span}\{\psi_j\}$. For example, κ_P is lower for Q_1 elements based on numerical integration (Q_1-NI) rather than exact integration for P^w preconditioners, but is not for P^s preconditioners [1, 16]. For $d = 2$, Q_1-NI leads to 5-point "star-type" stencils rather than 9-point "box" stencils. (Because of the GLL quadrature, the SE stencils are also "star-type" whenever $\mathbf{x}^e(r, s)$ is affine.) The

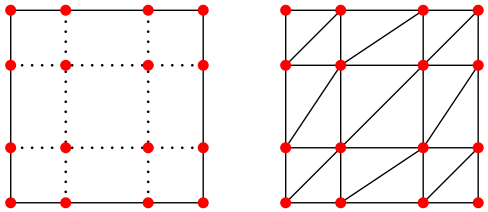


FIG. 1. *GLL points (left) for $N = 2$ and corresponding aligned triangulation (right) for the P_1 basis functions.*

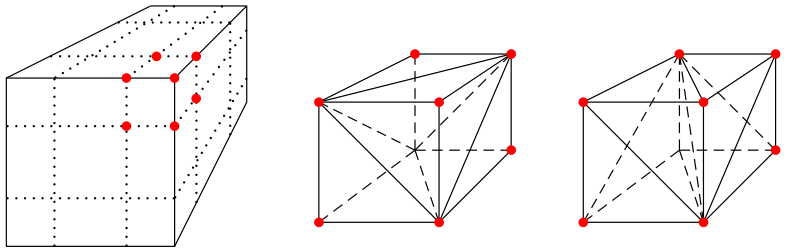


FIG. 2. *Tetrahedral discretization of a hex element with five tets (center) and six tets (right) for a single hex subelement of the complete SE (left).*

superiority of the 5-point stencil suggests the use of P_1 triangular elements, which lead to the same 5-point stencil in the affine case and which have the added benefit of readily accommodating geometric deformation induced by more general mappings $\mathbf{x}^e$.

Figures 1 and 2 show examples of collocated meshes using triangular and tetrahedral elements (i.e., P_1 elements). Q_1 elements such as rectangles and hexahedrals are another alternative for the construction of the low-order preconditioner but, as mentioned before, they are not as efficient in terms of reducing the number of iterations. Notice that the orientation of the triangles and tetrahedrals given here is not the only alternative, and multiple orientations are possible. Given that the number of iterations in Krylov-subspace projection methods scales as $\sqrt{\kappa_P}$, it is important to choose an orientation that minimizes the condition number of the preconditioned system. In particular, the 6-tet configuration of Figure 2 (right) was shown in [1] to give bounded values of κ_P versus N, whereas the 5-tet configuration did not. For this reason, we consider only 6-tet configurations in our so-called "fill-out" examples below.

Intuitive meshing schemes for FEM discretizations suggest meshing the domain so the elements fill out the whole geometry. Our approach, however, is novel in that it relaxes these constraints in terms of how the mesh represents the domain so that a preconditioner can be built in a more systematic way without the need to worry about the orientation of the elements while retaining the sparsity of the 6-tet configuration. Given that one of the leading indicators in AMG cost is the amount of fill in successive levels, we also strive for a starting point that is as sparse as possible. The basic idea is to discretize the PDE using a single triangle or tet per vertex in each of the d-dimensional cells that make up the SE point distribution. Each d-simplex connects

vertices to the vertices immediately adjacent to it, thus constraining the orientation of the elements to only one option while mimicking a star stencil. Figures 3 and 4 show what the low-order elements look like in two dimensions and three dimensions, respectively. For $d = 2$, one assembles four 3×3 matrices, one for each triangle as shown in Figure 3, into a 4×4 matrix. For each spectral element, there are N^2 such 4×4 matrices which are assembled, and these are in turn assembled into the global stiffness matrix A_F. For $d = 3$, one has eight 4×4 matrices, one for each tet as shown in Figure 4, which are assembled into an 8×8 matrix. As is the case for $d = 2$, these are ultimately assembled into A_F.

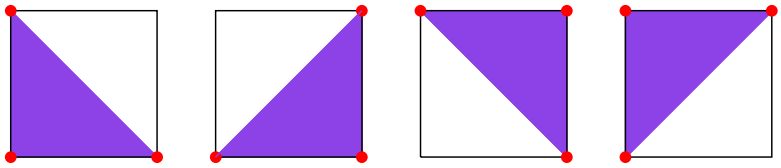


FIG. 3. *FEM meshing of a rectangular element with one triangle per vertex for a total of four low-order elements.*

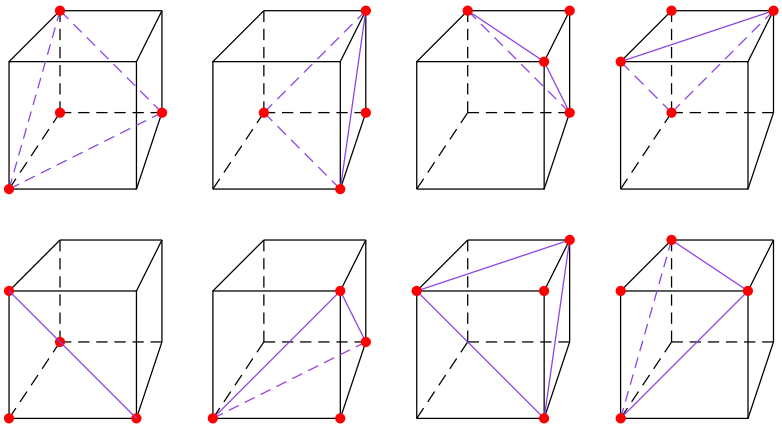


FIG. 4. *FEM meshing of a hexahedral element with one tet per vertex for a total of eight low-order elements.*

2.3. Preconditioning a single element. Tables 1 and 2 illustrate the effectiveness of the one-per-vertex strategy for the weak preconditioner $A_F^{-1}A$ for $d = 2$ and $d = 3$, respectively. For each table, three cases are considered. *All Dirichlet* shows κ_P when Dirichlet conditions are applied to the boundary of $\Omega = \hat{\Omega} := [-1, 1]^d$ for $N = 2, 4, \ldots, 20$. The Standard column corresponds to the P_1 configuration of Figure 1 (right) for $d = 2$ and to the 6-tet configuration of Figure 2 (right) for $d = 3$. The New column corresponds to the one-per-vertex configurations of Figures 3 and 4. The *Dirichlet–Neumann* case is similar, with $\Omega = \hat{\Omega}$, save that Dirichlet conditions are applied only at $x = -1$ in the 2D case and only at $z = -1$ in three dimensions; Neumann conditions are applied at all remaining surfaces. The *Deformed-Neumann* case retains the same Dirichlet surfaces (resp., $x = -1$ and $z = -1$) with Neumann elsewhere, but differs in that the unit square and unit cube are deformed as indicated in Figure 5.

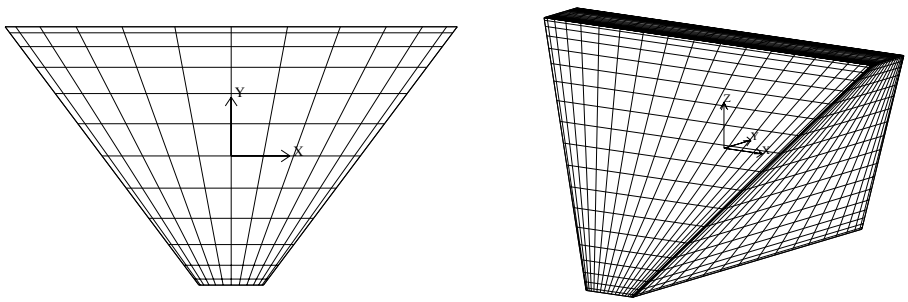


FIG. 5. *Single element tests: (left) 2D distorted domain with $N = 12$; (right) 3D distorted domain with $N = 12$ for Tables 1 and 2.*

In Tables 1 and 2 we see that $\kappa_P \sim \pi^2/4$ for the *All Dirichlet* cases, in accordance with the findings of [1]. The fact that κ_P converges to $\pi^2/4$ from *below* is particularly advantageous given that production-level simulations typically use $N = 5$–15. We see that the Standard and New results are identical in this case, as should be expected for the regular tessellations. In two dimensions, Table 1 indicates that the Standard and New configurations are also identical for the *Dirichlet–Neumann* case, but that κ_P is reduced by roughly one-half when the one-per-vertex discretization is used for the *Deformed-Neumann* case for $N = 20$. The 3D results of Table 2 are similar, save that the one-per-vertex savings is manifest even in the Dirichlet–Neumann case, which is of particular relevance to Navier–Stokes applications. Again, a roughly two-fold reduction in κ_P is observed at $N = 20$ for the deformed case.

TABLE 1
$\kappa(A_F^{-1}A)$ for 2D case, single spectral element.

	All Dirchlet		Dirichlet–Neumann		Deformed-Neumann	
N	Standard	New	Standard	New	Standard	New
2	1.000	1.000	2.495	2.495	3.478	3.087
4	1.554	1.554	2.692	2.692	5.925	4.110
6	1.803	1.803	2.920	2.920	7.738	4.857
8	1.945	1.945	3.073	3.073	9.086	5.304
10	2.037	2.037	3.180	3.180	10.150	5.593
12	2.101	2.101	3.259	3.259	11.034	5.794
14	2.148	2.148	3.319	3.319	11.789	5.942
16	2.185	2.185	3.367	3.367	12.447	6.055
18	2.214	2.214	3.406	3.406	13.029	6.144
20	2.237	2.237	3.438	3.438	13.550	6.216

3. Multielement results. To test the proposed preconditioners under more realistic conditions, we consider several applications for $d = 2$ and $d = 3$. All cases feature multiple elements in either structured or unstructured meshes with mixed boundary conditions comprising Dirichlet, Neumman, and periodic types. The objectives of the tests are two-fold. First, we are trying to determine which of the low-order preconditioning strategies is most effective for realistic configurations. Second, we are verifying whether it is possible for the AMG-based low-order preconditioning strategy

TABLE 2
$\kappa(A_F^{-1}A)$ *for 3D case, single SE.*

	All Dirchlet		Dirichlet–Neumann		Deformed-Neumann	
N	Standard	New	Standard	New	Standard	New
2	1.000	1.000	6.228	5.082	7.831	6.498
4	1.555	1.554	6.624	5.000	10.215	6.722
6	1.804	1.803	7.112	5.347	10.911	7.231
8	1.945	1.945	7.456	5.601	12.126	7.686
10	2.037	2.037	7.703	5.784	13.134	8.004
12	2.101	2.101	7.887	5.921	14.189	8.289
14	2.148	2.148	8.028	6.027	15.193	8.547
16	2.185	2.185	8.140	6.111	16.105	8.764
18	2.214	2.214	8.231	6.179	16.929	8.951
20	2.237	2.237	8.306	6.235	17.676	9.115

presented here to outperform the hybrid Schwarz (HS) multigrid solver [14, 26] that is currently used in the spectral element code Nek5000 [15] for parallel settings.

The HS scheme is effectively a p-multigrid scheme that uses overlapping Schwarz for a smoother on the fine ($N_2 = N$) and intermediate ($N_1 = N/2$ or $N_1 = 3$) levels, and a global coarse-grid solve at the coarsest ($N_0 = 1$) level [9, 10, 26]. The block-oriented Schwarz smoothers are implemented using separable approximations to the local Poisson problems, which can be inverted in less time than the forward operator by using fast diagonalization methods [4, 27]. The HS scheme is consequently very inexpensive to apply. Its effectiveness, however, does degrade in the presence of high-aspect ratio cells as noted in [16] and observed in the tests below. As we shall see, the low-order schemes proposed here are much more robust in the presence of mesh deformation and thus viable candidates for many challenging applications.

Details regarding the simulation components for each test are outlined below.

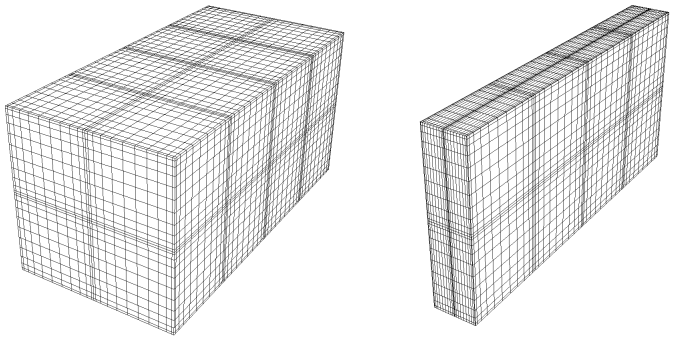
- Assembly of FEM matrices using two different strategies:
  - Our proposed one-element-per vertex approach.
  - Elements that fill-out the volume as done traditionally.

- Preconditioned GMRES iterative solver:
  - Convergence tolerance of 10^{-7}.
  - Restarting when projection reaches 30 vectors.

- AMG implemented in Hypre [11] with the following setup:
  - HMIS coarsening.
  - 0.25 strength of connection.
  - Extended+i interpolation ($p_{\max} = 4$).
  - SOR relaxation (one sweep going down, one sweep going up).
  - Only one V-cycle for preconditioning.
  - Non-Galerkin grid transfer operators with no dropping in the first level.

- Comparison with the production-level HS Multigrid preconditioner used in Nek5000 [15, 14, 26].
- All parallel simulations were run on the IBM BG/Q *Cetus* at the Argonne Leadership Computing Facility in -c32 mode (i.e., 32 ranks per node).

All coding was done using a combination of Fortran 77 and C. For the SEM dis-



(a) Aspect ratio 1. (b) Aspect ratio 4.

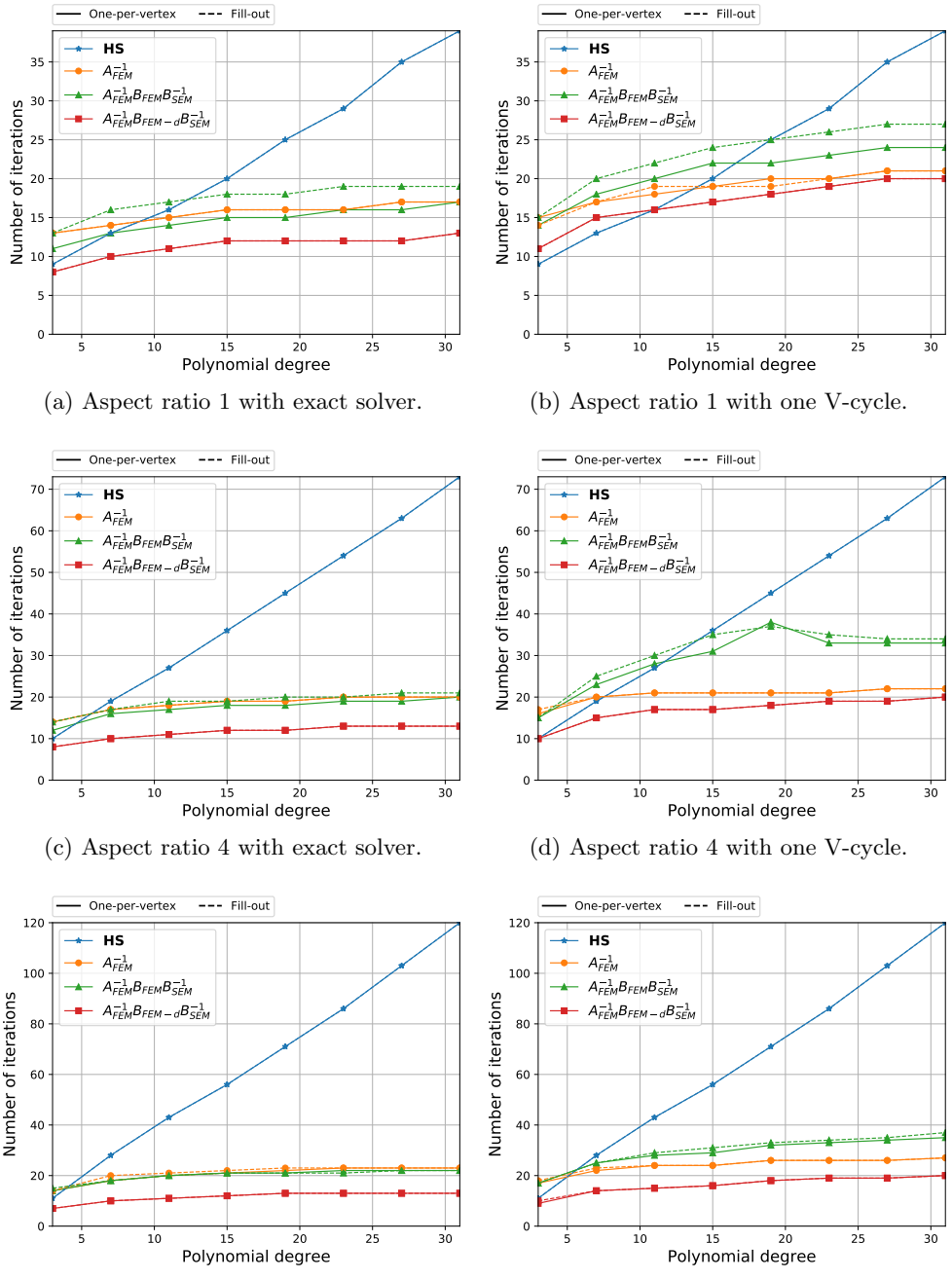
FIG. 6. *Hexahedral mesh with $E=16$ elements and polynomial degree $N=11$ for different aspect ratios in the xy-plane.*

cretization, Nek5000 (see [15]) was used and updated to implement the low-order preconditioners. 64-bit floating point arithmetic was used for all calculations.

3.1. Impact of aspect ratio. The first problem we present is used to study the preconditioning properties of our method under a very simple geometry with varying aspect ratios. A mesh with $(E_x, E_y, E_z) = (2 \times 2 \times 4)$ elements (i.e., a total of 16 cubic elements) shown in Figure 6 is compressed in the x-direction so that the aspect ratio, w_y/w_x, of each element in the xy-plane is increased. We consider $w_y/w_x = 1$, 4, and 16. As mentioned before, one of the main goals of our low-order preconditioner is to minimize the meshing effects, such as high-aspect ratio cells, which degrade the HS preconditioner. Using these meshes we can study the behavior of our method for a simple problem that is representative of more realistic meshes such as boundary layer meshes.

Another aspect of our proposed method that needs to be addressed is that of minimizing the computational time needed to apply the preconditioner. With this in mind we also explore the effects of applying the AMG solver as a full precision solver (i.e., performing as many V-cycles as needed to reduce the residual to machine precision) versus just applying one V-cycle. In the former case, the preconditioning step is more accurate at a higher computational cost, while in the latter, the approximation is inexact but relatively cheap to compute. We present in Figure 7 the experimental results for all our preconditioners, meshing strategies, and solver conditions. The different colors and markers represent the type of preconditioner, and the shape of the lines represent the meshing strategy.

It is clear from the figures that as the aspect ratio increases, the HS iteration count also increases. Furthermore, we also see a dependency on the polynomial degree in that with increasing degree, the iteration count increases too. This is not the case for the low-order preconditioners in both accounts. First, the aspect ratio of the elements has almost no effect on the iteration count for a given polynomial degree. Second, as the polynomial degree is increased, the iteration count eventually reaches a plateau. This tells us that the condition number is independent of the polynomial degree when using this type of preconditioner. We also observe that the preconditioner with the lumped mass matrix achieves the highest iteration reduction. As we will see in the next problem, this is not the case for more realistic problems in which the elements are deformed and/or curvilinear. Finally, applying AMG as a full solver does reduce



(a) Aspect ratio 1 with exact solver.

(b) Aspect ratio 1 with one V-cycle.

(c) Aspect ratio 4 with exact solver.

(d) Aspect ratio 4 with one V-cycle.

(e) Aspect ratio 16 with exact solver.

(f) Aspect ratio 16 with one V-cycle.

FIG. 7. *Preconditioning behavior under p-refinement of HS and low-order preconditioners for the synthetic problem with varying aspect ratios.*

the iteration count more than using 1 V-cycle, but at a cost that is too high to be used in production runs for the relative small gain in iteration reduction. Therefore, for the rest of this paper we will focus on using AMG with only 1 V-cycle.

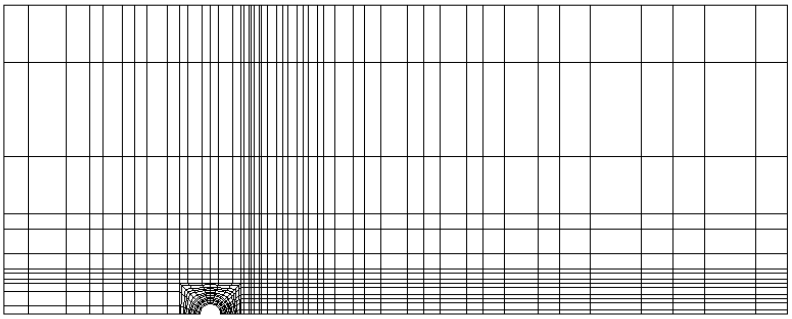


FIG. 8. *Two-dimensional mesh from* [16] *used to simulate a flow past a cylinder using homogeneous Dirichlet pressure conditions at the outlet (right) boundary and Neumann conditions elsewhere. Mesh lines are shown for* $(E, N) = (93, 3)$.

3.2. Flow past a cylinder. Here, we consider the mesh of Figure 8 used to simulate flow past a cylinder. This mesh has elements with high-aspect ratios in the far field and was used as a test problem in the development of multilevel additive Schwarz methods for the SEM [16], which was a precursor to the current HS preconditioner. In [16], a significant slow-down in convergence was clearly attributable to modes in the high-aspect ratio cells that were not well approximated by either the local fine-scale or global coarse-grid Schwarz function spaces. There are thus two objectives in studying this particular problem. First, we want to compare the fill-out and one-per-vertex meshing strategies in terms of the total number of iterations with the different kinds of preconditioners proposed (i.e., weak, strong, strong-diagonal). Second, we want to study the robustness of these preconditioners under h- and p-refinement.

We constructed four meshes of different aspect ratios for this experiment. A starting mesh with a total of 93 elements is refined to produce other two meshes with 372 elements and 1488 elements. A fourth mesh with 1744 elements was generated by refining the elements only in the vertical direction, which lowered the number of high-aspect-ratio elements.

Tables 3 and 4 show the GMRES iteration counts for the solution of (2.3) with random right-hand side. Preconditioning tests employ both the fill-out (6-tet) and one-per-vertex meshing strategies. In both tables we compare the HS preconditioner with our scheme. Notice that the iteration count of the low-order preconditioners is always lower in Table 4 compared to the same preconditioners in Table 3, demonstrating the enhanced preconditioning properties of the one-per-vertex approach. The lower iteration count for the HS preconditioner when $E = 1744$ stems from the fact that this preconditioner is highly sensitive to high-aspect ratio elements, which are reduced for this mesh. As stated before, high-aspect ratio elements significantly degrade the spectrum of the operators, thus increasing the iteration count. Low-order preconditioners are effectively insensitive to the presence of such cells and thus expected to be relatively robust for complex problems.

From this point forward, based on the experimental results just shown, we present only results using the weak preconditioner, P^w, as we found this to be the best alternative both in terms of iteration reduction and cost per iteration.

3.3. Wire-coil insert. The next problem is motivated by the study of heat-transfer enhancement in pipes having helical wire-coil inserts. While the primary focus of this application is convective heat transfer, it is the pressure solve, which is

TABLE 3
GMRES iteration counts for fill-out elements.

	$N=3$				$N=7$				$N=11$			
E	HS	P^w	P^s	P^{sd}	HS	P^w	P^s	P^{sd}	HS	P^w	P^s	P^{sd}
93	25	19	27	21	43	21	33	25	78	25	38	30
372	40	20	26	20	60	23	38	25	102	29	34	33
1488	57	19	25	20	85	25	31	28	136	27	33	30
1744	34	21	27	19	44	27	32	30	54	28	39	32

TABLE 4
GMRES iteration counts for one-per-vertex elements.

	$N=3$				$N=7$				$N=11$			
E	HS	P^w	P^s	P^{sd}	HS	P^w	P^s	P^{sd}	HS	P^w	P^s	P^{sd}
93	25	17	21	17	43	18	24	19	78	18	27	21
372	40	17	20	17	60	17	27	18	102	20	27	22
1488	57	15	20	15	85	18	22	21	136	19	25	21
1744	34	16	21	16	44	19	24	21	54	20	31	22

required to advance the turbulent advecting field, that consumes most of the solution time. Fast Poisson solvers are thus of significant interest. The domain in this case comprises a 2D mesh that is extruded and twisted to give the proper void region surrounding the wire-coil insert. The resultant curved 3D mesh with $E = 5720$ elements is shown in Figure 9.

Figure 10 shows the execution time and GMRES iteration counts as a function of the number of MPI ranks, P. The SE discretization parameters are $(E, N) = (5720, 7)$ or about two million gridpoints ($n \approx EN^3$). For all plots, we see roughly 50% parallel efficiency at P=1024, which corresponds to $n/P \approx 2000$ points per MPI rank. It was established in [13] that 50% efficiency is expected when $n/P \approx 2000$ on this architecture, based on the communication-to-computation ratios realized by BG/Q.

As we can see from Figure 10, the HS preconditioner requires about four times the number of iterations to converge compared to the low-order preconditioner using one-per-vertex. The average number of iterations across the parallel runs for the first timestep is 393 for HS, 194 for fill-out, and 103 for one-per-vertex. The cost-per-iteration of the fill-out and one-per-vertex methods are roughly the same, so one-per-vertex is the preferred low-order approach. In terms of time-to-solution, one-per-vertex is on a par with the HS scheme.

Figure 11 shows the run history for a full Navier–Stokes plus heat transfer simulation for 500 timesteps. We note that the iteration counts drop dramatically after the start of the computation because we generate highly accurate initial guesses to the pressure by projecting the solution onto the space of a handful (typically 8–20) of prior pressure solutions [12]. It is thus important to test the preconditioners in this projected context where the solver spends the majority of its time. We can see that the one-per-vertex scheme requires about one-third the number of pressure iterations as HS. The total execution time for the full simulation was measured to be 2946.71 s

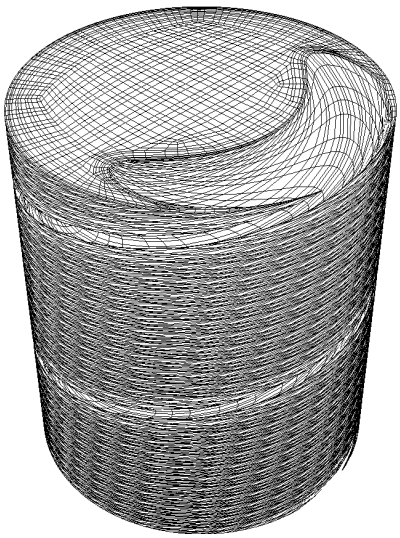
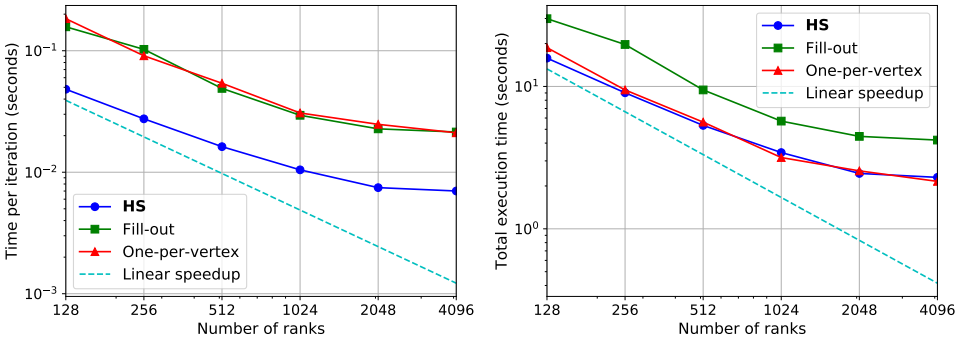
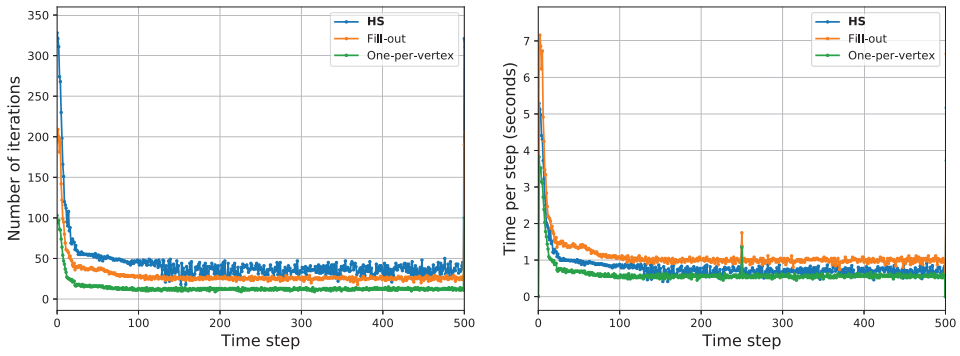


FIG. 9. *Mesh for wire-coil insert with* $E = 5720$ *elements and polynomial degree* $N = 3$.



(a) Average time per iteration. (b) Total execution time.

FIG. 10. *Strong scaling of* P^w *preconditioning for the helical copper wire problem with* $(E, N) = (5720, 7)$ *and matrix size* $n = 2{,}002{,}910$.



(a) GMRES iteration history. (b) Time per Navier–Stokes step history.

FIG. 11. *Run history for helical copper wire problem with* $(E, N) = (5720, 7)$.

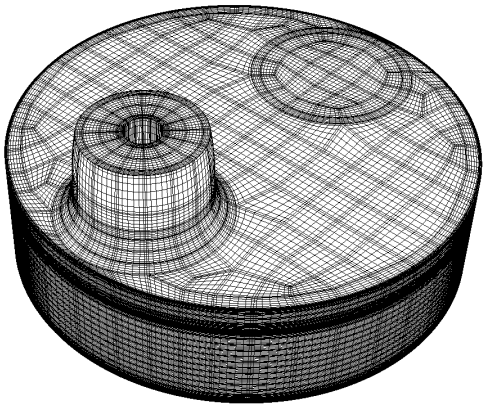


FIG. 12. *Cylinder mesh with* $E = 6784$ *and polynomial degree* $N = 7$.

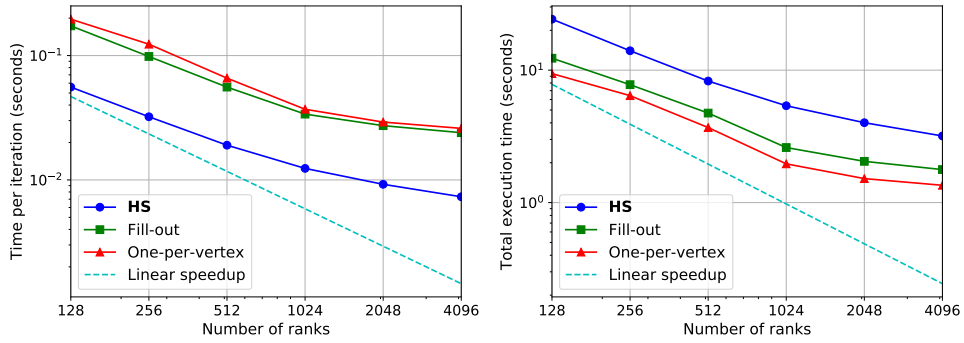
for HS, 4067.542 s for fill-out, and 2615.87 s for one-per-vertex, thus achieving a 12% reduction in overall computation time.

3.4. Cylinder-valve simulation. Our final problem considers simulation of the intake stroke for a moving valve-piston configuration used in modeling of internal combustion engines. Figure 12 shows the exterior of the mesh with E =6784 elements. The geometry is deforming with time using the arbitrary Lagrangian–Eulerian (ALE) formulation described in [30]. The preconditioner is constant throughout each restarted simulation. That is, while A and B are functions of time because of the mesh motion, A_F is taken to be fixed at its initial instantiation at the start of each restarted computation. A particular challenge for this problem is that the timestep is quite large, corresponding to a Courant number of $C \approx 4$, which is enabled through the ALE+characteristics-based timestepper developed in [30]. Higher Courant numbers typically lead to higher pressure iteration counts because the initial residual scales with timestep size, even in the context of time-projection [12]. Thus the baseline iteration count for this example is high, which provides more opportunity for improved preconditioning strategies.

Strong-scaling results for this mesh are presented in Figure 13 for both the time-per-iteration and the total execution time for the first timestep. The number of iterations on this step was 435 for HS, 77 for fill-out, and 52 for one-per-vertex. As in the preceding case, we observe the anticipated 50% parallel efficiency when $n/P \approx 2000$.

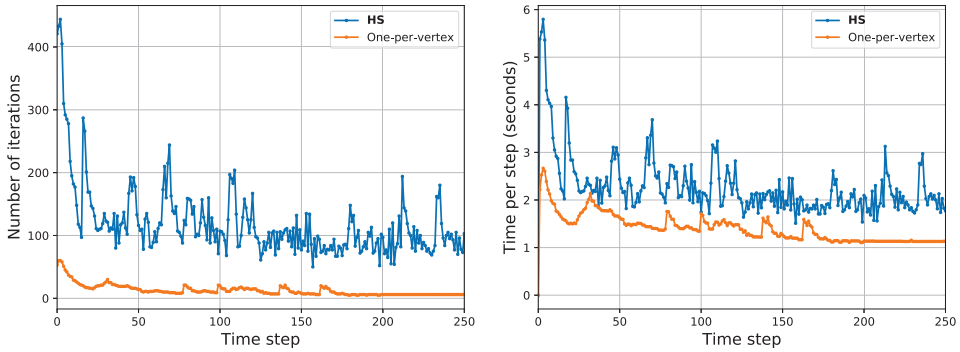
Figure 14 shows the iteration history for 250 time steps. Here, the eight-fold reduction in iteration count for the one-per-vertex algorithm translates into a significant reduction in time-to-solution compared to the HS method. For the HS preconditioner this time was $2.82\,{}^{\text{s}}/_{\text{step}}$ versus $1.54\,{}^{\text{s}}/_{\text{step}}$ for the one-per-vertex case, thus achieving an overall improvement of 46% in the time spent at each Navier–Stokes step.

4. Conclusions. In this work we explored a new way to construct low-order finite element preconditioners for spectral element discretizations of the Poisson problem. We tested three different preconditioners from weak and strong operators built using two different approaches: one-per-vertex elements and elements that fill-out the geometry. While intuition tells us that elements that don't overlap and cover the whole domain are the best alternative, we showed this is not necessarily the case for preconditioning purposes. Our proposed one-per-vertex approach proved to be the



(a) Average time per iteration. (b) Total execution time.

FIG. 13. *Strong scaling of P^w preconditioning for the piston problem with $(E, N) = (6784, 7)$ and matrix size $n = 2{,}366{,}862$.*



(a) GMRES iteration history. (b) Time per Navier–Stokes step history.

FIG. 14. *Run history for piston problem with $(E, N) = (6784, 7)$.*

best choice in all cases. In solving the linear system at the preconditioning phase, AMG was used with an implementation provided by the Hypre library at LLNL. By carefully choosing the setup parameters of the AMG solver, we were able to reduce the computation time below that of the hybrid-Schwarz multigrid preconditioner, even in the strong-scale limit of $n/P \approx 2000$ points per processor.

REFERENCES

[1] C. CANUTO, P. GERVASIO, AND A. QUARTERONI, *Finite-element preconditioning of G-NI spectral methods*, SIAM J. Sci. Comput., 31 (2010), pp. 4422–4451, https://doi.org/10.1137/090746367.

[2] C. CANUTO, M. Y. HUSSAINI, A. QUARTERONI, AND T. A. ZANG, *Spectral Methods: Fundamentals in Single Domains*, 1st ed., Springer-Verlag, Berlin, Heidelberg, 2006.

[3] C. CANUTO AND A. QUARTERONI, *Preconditioned minimal residual methods for Chebyshev spectral calculations*, J. Comput. Phys., 60 (1985), pp. 315–337.

[4] W. COUZY AND M. O. DEVILLE, *A fast Schur complement method for the spectral element discretization of the incompressible Navier-Stokes equations*, J. Comput. Phys., 116 (1995), pp. 135–142.

[5] M. O. DEVILLE, P. F. FISCHER, AND E. H. MUND, *High-Order Methods for Incompressible Fluid Flow*, 1st ed., Cambridge Monogr. Appl. Comput. Math. 9, Cambridge University Press, Cambridge, UK, 2002.

[6] M. O. Deville and E. H. Mund, *Chebyshev pseudospectral solution of second-order elliptic equations with finite element preconditioning*, J. Comput. Phys., 60 (1985), pp. 517–533.
[7] M. O. Deville and E. H. Mund, *Finite-element preconditioning for pseudospectral solutions of elliptic problems*, SIAM J. Sci. Comput., 11 (1990), pp. 311–342, https://doi.org/10.1137/0911019.
[8] M. O. Deville and E. H. Mund, *Fourier analysis of finite element preconditioned collocation schemes*, SIAM J. Sci. and Stat. Comput., 13 (1992), pp. 596–610, https://doi.org/10.1137/0913033.
[9] M. Dryja and O. Widlund, *An Additive Variant of the Schwarz Alternating Method for the Case of Many Subregions*, Tech. Report TR 339, Dept. Comp. Sci., Courant Inst., NYU, New York, NY, 1987.
[10] M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput., 15 (1994), pp. 604–620, https://doi.org/10.1137/0915040.
[11] R. D. Falgout and U. M. Yang, *Hypre: A Library of High Performance Preconditioners*, Lecture Notes in Comput. Sci. 2331, Springer, Berlin, Heidelberg, 2002, pp. 632–641.
[12] P. Fischer, *Projection techniques for iterative solution of $A\underline{x} = \underline{b}$ with successive right-hand sides*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 193–204.
[13] P. Fischer, K. Heisey, and M. Min, *Scaling limits for PDE-based simulation (invited)*, in 22nd AIAA Computational Fluid Dynamics Conference, AIAA Aviation, AIAA 2015-3049, 2015.
[14] P. Fischer and J. Lottes, *Hybrid Schwarz-multigrid Methods for the spectral element method: Extensions to Navier-Stokes*, in Domain Decomposition Methods in Science and Engineering Series, R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, eds., Springer, Berlin, 2005, pp. 35–49.
[15] P. Fischer, J. Lottes, and S. Kerkemeier, *Nek*5000: *Open Source Spectral Element CFD Solver*, http://nek5000.mcs.anl.gov and https://github.com/nek5000/nek5000, 2008.
[16] P. F. Fischer, *An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations*, J. Comput. Phys., 133 (1997), pp. 84–101.
[17] P. F. Fischer, L.-W. Ho, G. E. Karniadakis, E. M. Rønquist, and A. T. Patera, *Recent advances in parallel spectral element simulation of unsteady incompressible flows*, in Computational Structural Mechanics & Fluid Dynamics, Pergamon, Oxford, 1988, pp. 217–231.
[18] P. F. Fischer and J. W. Lottes, *Hybrid Schwarz-Multigrid Methods for the Spectral Element Method: Extensions to Navier-Stokes*, Lect. Notes Comput. Sci. Eng. 40, Springer, Berlin, 2005, pp. 35–49.
[19] P. F. Fischer, H. M. Tufo, and N. I. Miller, *An Overlapping Schwarz Method for Spectral Element Simulation of Three-Dimensional Incompressible Flows*, Springer, New York, 2000, pp. 159–180.
[20] W. Heinrichs, *Line relaxation for spectral multigrid methods*, J. Comput. Phys., 77 (1988), pp. 166–182.
[21] K. Heinz-Otto and O. Joseph, *Comparison of accurate methods for the integration of hyperbolic equations*, Tellus, 24 (1972), pp. 199–215.
[22] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral Methods for Time-Dependent Problems*, Cambridge Monogr. Appl. Comput. Math. 21, Cambridge University Press, Cambridge, 2007.
[23] S. D. Kim and S. V. Parter, *Preconditioning Chebyshev spectral collocation method for elliptic partial differential equations*, SIAM J. Numer. Anal., 33 (1996), pp. 2375–2400, https://doi.org/https://doi.org/10.1137/S0036142994275998.
[24] S. D. Kim and S. V. Parter, *Preconditioning Chebyshev spectral collocation by finite-difference operators*, SIAM J. Numer. Anal., 34 (1997), pp. 939–958, https://doi.org/10.1137/S0036142995285034.
[25] J. W. Lottes, *Independent Quality Measures for Symmetric Algebraic Multigrid Components*, 2011, https://www.mcs.anl.gov/papers/P1820-0111.pdf.
[26] J. W. Lottes and P. F. Fischer, *Hybrid multigrid/Schwarz algorithms for the spectral element method*, J. Sci. Comput., 24 (2005), pp. 45–78.
[27] R. E. Lynch, J. R. Rice, and D. H. Thomas, *Direct solution of partial difference equations by tensor product methods*, Numerische Mathematik, 6 (1964), pp. 185–199.
[28] L. Olson, *Algebraic multigrid preconditioning of high-order spectral elements for elliptic problems on a simplicial mesh*, SIAM J. Sci. Comput., 29 (2007), pp. 2189–2209, https://doi.org/10.1137/060663465.
[29] S. A. Orszag, *Spectral methods for problems in complex geometries*, J. Comput. Phys., 37 (1980), pp. 70–92.

[30] S. Patel, P. Fischer, M. Min, and A. Tomboulides, *A characteristic-based spectral element method for moving-domain problems*, J. Sci. Comput, submitted.
[31] A. T. Patera, *A spectral element method for fluid dynamics: Laminar flow in a channel expansion*, J. Comput. Phys., 54 (1984), pp. 468–488.
[32] J. Shen, *Efficient spectral-Galerkin method* I. *Direct solvers of second- and fourth-order equations using Legendre polynomials*, SIAM J. Sci. Comput., 15 (1994), pp. 1489–1505, https://doi.org/10.1137/0915089.
[33] G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, NJ, 1973.