# High-Performance Spectral Element Algorithms and Implementations* 

Paul F. Fischer and Henry M. Tufo<br>Mathematics and Computer Science Division<br>Argonne National Laboratory, Argonne, IL 60439 USA<br>\{fischer, tufo\}@mcs.anl.gov<br>http://www-unix.mcs.anl.gov/appliedmath/Flow/cfd.html


#### Abstract

We describe the development and implementation of a spectral element code for multimillion gridpoint simulations of incompressible flows in general two- and three-dimensional domains. Parallel performance is present on up to 2048 nodes of the Intel ASCI-Red machine at Sandia.


## 1 Introduction

We consider numerical solution of the unsteady incompressible Navier-Stokes equations,

$$
\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u} \cdot \nabla \mathbf{u}=-\nabla p+\frac{1}{R e} \nabla^{2} \mathbf{u}, \quad-\nabla \cdot \mathbf{u}=0
$$

coupled with appropriate boundary conditions on the velocity, u. We are developing a spectral element code to solve these equations on modern large-scale parallel platforms featuring cache-based nodes. As illustrated in Fig. 1, the code is being used with a number of outside collaborators to address challenging problems in fluid mechanics and heat transfer, including the generation of hairpin vortices resulting from the interaction of a flat-plate boundary layer with a hemispherical roughness element; modeling the geophysical fluid flow cell space laboratory experiment of buoyant convection in a rotating hemispherical shell; Rayleigh-Taylor instabilities; flow in a carotid artery; and forced convective heat transfer in grooved-flat channels.

This paper discusses some of the critical algorithmic and implementation features of our numerical approach that have led to efficient simulation of these problems on modern parallel architectures. Section 2 gives a brief overview of the spectral element discretization. Section 3 discusses components of the time advancement procedure, including a projection method and parallel coarse-grid solver that are applicable to other problem classes and discretizations. Section 4 presents performance results and Section 5 gives a brief conclusion.

## 2 Spectral Element Discretization

The spectral element method is a high-order weighted residual technique developed by Patera and coworkers in the '80s that couples the tensor product efficiency of global spectral methods with the geometric flexibility of finite elements [9, 11]. Locally, the mesh is structured, with the solution, data, and geometry expressed as sums of $N$ th-order tensor product Lagrange polynomials based on the Gauss or Gauss-Lobatto (GL) quadrature points. Globally, the mesh

[^0]

Figure 1: Recent spectral element simulations. To the right, from the top: hairpin vortex generation in wake of hemispherical roughness element ( $R e_{\delta}=850$ ); spherical convection simulation of the geophysical fluid flow cell at $R a=1.1 \times 10^{5}, T a=1.4 \times 10^{6}$; two-dimensional Rayleigh-Taylor instability; flow in a carotid artery; and temporal-spatial evolution of convective instability in heat-transfer augmentation simulations.
is an unstructured array of $K$ deformed hexahedral elements and can include geometrically nonconforming elements. The discretization is illustrated in Fig. 2, which shows a mesh in $\mathbb{R}^{2}$ for the case $(K, N)=(3,4)$. Also shown is the reference $(r, s)$ coordinate system used for all function evaluations. The use of the GL basis for the interpolants leads to efficient quadrature for the weighted residual schemes and greatly simplifies operator evaluation for deformed elements.

For problems having smooth solutions, such as the incompressible Navier-Stokes equations, exponential convergence is obtained with increasing $N$, despite the fact that only $C^{0}$ continuity is enforced across elemental interfaces. This is demonstrated in Table 1, which shows the computed growth rates when a small-amplitude Tollmien-Schlichting wave is superimposed on plane Poiseuille channel flow at $R e=7500$, following [6]. The amplitude of the perturbation is $10^{-5}$, implying that the nonlinear Navier-Stokes results can be compared with linear theory to about five significant digits. Three error measures are computed: error ${ }_{1}$ and error $_{2}$ are the relative amplitude errors at the end of the first and second periods, respectively, and error ${ }_{g}$ is the error in the growth rate at a convective time of 50 . From Table 1, it is clear that doubling the number of points in each spatial direction yields several orders of magnitude reduction in error, implying that just a small increase in resolution is required for very good accuracy. The


Figure 2: Spectral element discretization in $\mathbb{R}^{2}$ showing GL nodal lines for $(K, N)=(3,4)$.
significance of this is underscored by the fact that, in three dimensions, the effect on the number of gridpoints scales as the cube of the relative savings in resolution.

The computational efficiency of spectral element methods derives from the use of tensorproduct forms. Functions in the mapped coordinates are expressed as

$$
\begin{equation*}
\left.u\left(\mathbf{x}^{k}(r, s)\right)\right|_{\Omega^{k}}=\sum_{i=0}^{N} \sum_{j=0}^{N} u_{i j}^{k} h_{i}^{N}(r) h_{j}^{N}(s), \tag{1}
\end{equation*}
$$

where $u_{i j}^{k}$ is the nodal basis coefficient; $h_{i}^{N}$ is the Lagrange polynomial of degree $N$ based on the GL quadrature points, $\left\{\xi_{j}^{N}\right\}_{j=0}^{N}$; and $\mathbf{x}^{k}(r, s)$ is the coordinate mapping from the reference domain, $\hat{\Omega}:=[-1,1]^{2}$, to $\Omega^{k}$. With this basis, the stiffness matrix for an undeformed element $k$ in $\mathbb{R}^{2}$ can be written as a tensor-product sum of one-dimensional operators,

$$
\begin{equation*}
A^{k}=\widehat{B}_{y} \otimes \hat{A}_{x}+\widehat{A}_{y} \otimes \widehat{B}_{x} \tag{2}
\end{equation*}
$$

where $\widehat{A}_{*}$ and $\widehat{B}_{*}$ are the one-dimensional stiffness and mass matrices associated with the respective spatial dimensions. If $\underline{u}^{k}=u_{i j}^{k}$ is the matrix of nodal values on element $k$, then a typical matrix-vector product required of an iterative solver takes the form

$$
\begin{align*}
\left(A^{k} \underline{u}^{k}\right)_{l m} & =\sum_{i=0}^{N} \sum_{j=0}^{N}\left(\widehat{B}_{y, m j} \hat{A}_{x, l i} \underline{u}_{i j}^{k}+\widehat{A}_{y, m j} \hat{B}_{x, l i} \underline{u}_{i j}^{k}\right)  \tag{3}\\
& =\widehat{A}_{x} \underline{u}^{k} \widehat{B}_{y}^{T}+\widehat{B}_{x} \underline{u}^{k} \hat{A}_{y}^{T} .
\end{align*}
$$

Similar forms result for other operators and for complex geometries. The latter form illustrates how the tensor-product basis leads to matrix-vector products ( $A \underline{u}$ ) being recast as matrixmatrix products, a feature central to the efficiency of spectral element methods. These typically account for roughly $90 \%$ of the work and are usually implemented with calls to DGEMm, unless hand-unrolled F77 loops prove faster on a given platform.

Table 1: Spatial convergence, O-S problem: $K=15, \Delta t=.003125$

| $N$ | $E\left(t_{1}\right)$ | error $_{1}$ | $E\left(t_{2}\right)$ | error $_{2}$ | error $_{g}$ |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 7 | 1.11498657 | 0.003963 | 1.21465285 | 0.037396 | 0.313602 |
| 9 | 1.11519192 | 0.003758 | 1.24838788 | 0.003661 | 0.001820 |
| 11 | 1.11910382 | 0.000153 | 1.25303597 | 0.000986 | 0.004407 |
| 13 | 1.11896714 | 0.000016 | 1.25205855 | 0.000009 | 0.000097 |
| 15 | 1.11895646 | 0.000006 | 1.25206398 | 0.000014 | 0.000041 |

Global matrix products, $A \underline{u}$, also require a gather-scatter step to assemble the elemental contributions. Since all data is stored on an element-by-element basis, this amounts to summing nodal values shared by adjacent elements and redistributing the sums to the nodes. The parallel implementation of this follows the standard message-passing-based SPMD model, in which contiguous groups of elements are distributed to processors and data on shared interfaces is exchanged and summed. A stand-alone MPI-based utility has been developed for this operation that has an easy-to-use interface requiring only two calls:

$$
\text { handle }=\text { gs-init }(\text { global-node-numbers }, n) \text { and } \quad \mathbf{i e r r}=\operatorname{gs-op}(u, o p, h a n d l e),
$$

where global-node-numbers() associates the $n$ local values contained in the vector $u()$ with their global counterparts, and op denotes the reduction operation performed on shared elements of $u()$ [14]. The utility supports a general set of commutative/associative operations as well as a vector mode for problems having multiple degrees of freedom per vertex. Communication overhead is further reduced through the use of a recursive spectral bisection based element partitioning scheme to minimize the number of vertices shared among processors [12].

## 3 Time Advancement and Solvers

The Navier-Stokes timestepping is based on the second-order operator splitting methods developed in $[1,10]$. The convective term is expressed as a material derivative, and the resultant form is discretized using a stable second-order backward difference formula:

$$
\frac{\tilde{\mathbf{u}}^{n-2}-4 \underline{\tilde{\mathbf{u}}}^{n-1}+3 \underline{\mathbf{u}}^{n}}{2 \Delta t}=S\left(\underline{\mathbf{u}}^{n}\right)
$$

where $S\left(\underline{\mathbf{u}}^{n}\right)$ is the linear symmetric Stokes problem to be solved implicitly, and $\tilde{\mathbf{u}}^{n-q}$ is the velocity field at time step $n-q$ computed as the explicit solution to a pure convection problem. The subintegration of the convection term permits values of $\Delta t$ corresponding to convective CFL numbers of $2-5$, thus significantly reducing the number of (expensive) Stokes solves.

The Stokes problem is of the form

$$
\left[\begin{array}{rr}
\mathbf{H} & -\mathbf{D}^{T} \\
-\mathbf{D} & 0
\end{array}\right]\left(\begin{array}{l}
\frac{\mathbf{u}^{n}}{\underline{p}^{n}}
\end{array}\right)=\binom{\mathbf{B} \underline{\mathbf{f}}}{\underline{0}}
$$

and is also treated by second-order splitting, resulting in subproblems of the form

$$
H \underline{u}_{i}^{n}=\underline{f}_{i}^{n}, \quad E \underline{p}^{n}=\underline{g}^{n}
$$

for the velocity components, $u_{i}^{n},(i=1, \ldots, 3)$, and pressure, $\underline{p}^{n}$. Here, $H$ is a diagonally dominant Helmholtz operator representing the parabolic component of the momentum equations and is readily treated by Jacobi-preconditioned conjugate gradients; $E:=\mathbf{D} \mathbf{B}^{-1} \mathbf{D}^{T}$ is the Stokes Schur complement governing the pressure; and $\mathbf{B}$ is the (diagonal) velocity mass matrix.
$E$ is a consistent Poisson operator and is effectively preconditioned by using the overlapping additive Schwarz procedure of Dryja and Widlund [2, 6, 7]. In addition, a high-quality initial guess is generated by projecting the solution onto the space of previous solutions. The procedure is summarized in the following steps

$$
\begin{equation*}
\underline{\bar{p}}=\sum_{i=1}^{l} \alpha_{i} \underline{\tilde{p}}_{i}, \quad \alpha_{i}:=\underline{\tilde{p}}_{i}^{T} \underline{g}^{n} \tag{i}
\end{equation*}
$$

(ii) Solve : $E \Delta \underline{p}=\underline{g}^{n}-E \underline{p}$ to tolerance $\epsilon$.

$$
\begin{equation*}
\underline{\tilde{p}}_{l+1}=\left(\Delta \underline{p}-\sum_{i=1}^{l} \beta_{i} \underline{\tilde{p}}_{i}\right) /\left\|\Delta \underline{p}-\sum_{i=1}^{l} \beta_{i} \underline{\tilde{p}}_{i}\right\|_{E}, \quad \beta_{i}:=\underline{\tilde{p}}_{i}^{T} E \Delta \underline{p} . \tag{4}
\end{equation*}
$$

The first step computes an initial guess, $\underline{p}$, as a projection in the $E$-norm $\left(\|\underline{p}\|_{E}:=\left(\underline{p}^{T} E \underline{p}\right)^{\frac{1}{2}}\right)$ of $\underline{p}^{n}$ onto an existing basis, $\left(\underline{\tilde{p}}_{1}, \ldots, \underline{\tilde{p}}_{l}\right)$. The second computes the remaining (orthogonal) perturbation to a specified absolute tolerance, $\epsilon$. The third augments the approximation space


Figure 3: Iteration count (left) and residual history (right) with and without projection for the $1,658,880$ degree-of-freedom pressure system associated with the spherical convection problem of Fig. 1.
with the most recent (orthonormalized) solution. The approximation space is restarted once ( $l>L$ ) by setting $\underline{\underline{p}}_{1}:=\underline{p}^{n} /\left\|\underline{p}^{n}\right\|_{E}$. The projection scheme (steps (i) and (iii)) requires two matrix-vector products per timestep, one in step (ii) and one in step (iii). (Note that it is not possible to use $\underline{g}^{n}-E \underline{\bar{p}}$ in place of $E \Delta \underline{p}$ in (iii) because (ii) is satisfied only to within $\epsilon$.)

As shown in [4], the projection procedure can be extended to any parameter-dependent problem and has many desirable properties. It can be coupled with any iterative solver, which is treated as a black box (4ii). It gives the best fit in the space of prior solutions and is therefore superior to extrapolation. It converges rapidly, with the magnitude of the perturbation scaling as $O\left(\Delta t^{l}\right)+O(\epsilon)$. The classical Gram-Schmidt procedure is observed to be stable and has low communication requirements because the inner products for the basis coefficients can be computed in concert. Under normal production tolerances, the projection technique yields a two- to fourfold reduction in work. This is illustrated in Fig. 3, which shows the reduction in residual and iteration count for the buoyancy-driven spherical convection problem of Fig. 1, computed with $K=7680$ elements of order $N=7(1,658,880$ pressure degrees of freedom). The iteration count is reduced by a factor of 2.5 to 5 over the unprojected ( $L=0$ ) case, and the initial residual is reduced by two and one-half orders of magnitude.

The perturbed problem (4ii) is solved using conjugate gradients, preconditioned by an additive overlapping Schwarz method [2] developed in [6, 7]. The preconditioner,

$$
M^{-1}:=R_{0}^{T} A_{0}^{-1} R_{0}+\sum_{k=1}^{K} R_{k}^{T} \tilde{A}_{k}^{-1} R_{k},
$$

requires a local solve ( $\tilde{A}_{k}^{-1}$ ) for each (overlapping) subdomain, plus a global solve ( $A_{0}^{-1}$ ) for a coarse-grid problem based on the mesh of spectral element vertices. The operators $R_{k}$ and $R_{k}^{T}$ are simply Boolean restriction and prolongation matrices that map data between the global and local representations, while $R_{0}$ and $R_{0}^{T}$ map between the fine and coarse grids. The method is naturally parallel because the subdomain problems can be solved independently. Parallelization of the coarse-grid component is less trivial and is discussed below. The local subdomain solves exploit the tensor product basis of the spectral element method. Elements are extended by a single gridpoint in each of the directions normal to their boundaries. Bilinear finite element Laplacians, $\tilde{A}_{k}$, and lumped mass matrices, $\tilde{B}_{k}$, are constructed on each extended element, $\tilde{\Omega}^{k}$, in a form similar to (2). The tensor-product construction allows the inverse of $\tilde{A}_{k}^{-1}$ to be expressed as

$$
\tilde{A}_{k}^{-1}=\left(S_{y} \otimes S_{x}\right)\left[I \otimes \Lambda_{x}+\Lambda_{y} \otimes I\right]^{-1}\left(S_{y}^{T} \otimes S_{x}^{T}\right),
$$

where $S_{*}$ is the matrix of eigenvectors, and $\Lambda_{*}$ the diagonal matrix of eigenvalues, solving the generalized eigenvalue problem $\tilde{A}_{*} \underline{z}=\lambda \tilde{B}_{*} \underline{z}$ associated with each respective spatial direction. The complexity of the local solves is consequently of the same order as the matrix-vector product evaluation $\left(O\left(K N^{3}\right)\right.$ storage and $O\left(K N^{4}\right)$ work in $\left.\mathbb{R}^{3}\right)$ and can be implemented as in (3) using fast matrix-matrix product routines. While the tensor product form (2) is not strictly applicable to deformed elements, it suffices for preconditioning purposes to build $\tilde{A}_{k}$ on a rectilinear domain of roughly the same dimensions as $\tilde{\Omega}^{k}[7]$.

The coarse-grid problem, $\underline{x}=A_{0}^{-1} \underline{b}$, is central to the efficiency of the overlapping Schwarz procedure, resulting in an eightfold decrease in iteration count in model problems considered in $[6,7]$. It is also a well-known source of difficulty on large distributed-memory architectures because the solution and data are distributed vectors, while $A_{0}^{-1}$ is completely full, implying a need for all-to-all communication [3, 8]. Moreover, because there is very little work on the coarse grid (typ. $\mathrm{O}(1)$ d.o.f. per processor), the problem is communication intensive. We have recently developed a fast coarse-grid solution algorithm that readily extends to thousands of processors [5, 13]. If $A_{0} \in \mathbb{R}^{n \times n}$ is symmetric positive definite and $X:=\left(\underline{\tilde{x}}_{1}, \ldots, \underline{\tilde{x}}_{n}\right)$ is a matrix of $A_{0}$-orthonormal vectors satisfying $\underline{\tilde{x}}_{i}^{T} A_{0} \tilde{\underline{x}}_{j}=\delta_{i j}$, then the coarse-grid solution is computed as

$$
\begin{equation*}
\underline{\bar{x}}:=\sum_{i=1}^{n} \alpha_{i} \tilde{\underline{x}}_{i}=X X^{T} \underline{b}, \quad \alpha_{i}:=\underline{\tilde{x}}_{i}^{T} \underline{b} . \tag{5}
\end{equation*}
$$

Since $\underline{\bar{x}}$ is the best fit in $\mathcal{R}(X) \equiv \mathbb{R}^{n}$, we have $\underline{\bar{x}}=\underline{x}$ and $X X^{T}=A_{0}^{-1}$. The projection procedure (5) is similar to (4i), save that the basis vectors $\left\{\underline{\tilde{x}}_{i}\right\}$ are chosen to be sparse. Such sparse sets can be readily found by recognizing that, for any gridpoint $i$ exterior to the stencil of $j$, there exists a pair of $A_{0}$-conjugate unit vectors, $\hat{\underline{e}}_{i}$ and $\underline{\hat{e}}_{j}$. For example, for a regular $n$-point mesh in $\mathbb{R}^{2}$ discretized with a standard five-point stencil, one can immediately identify half of the unit vectors (associated, e.g., with the "red" squares) in $\mathbb{R}^{n}$ as unnormalized elements of $X$. The remainder of $X$ can be created by applying Gram-Schmidt orthogonalization to the remainder of $\mathbb{R}^{n}$. In [5, 13], it is shown that nested dissection provides a systematic approach to identifying a sparse basis and yields a factorization of $A_{0}^{-1}$ with $O\left(n^{\frac{2 d-1}{d}}\right)$ nonzeros for $n$-point grid problems in $\mathbb{R}^{d}, d \geq 2$. Moreover, the required communication volume on a $P$-processor machine is bounded by $\overline{3} n^{\frac{d-1}{d}} \log _{2} P$, a clear gain over the $O(n)$ or $O\left(n \log _{2} P\right)$ costs incurred by other commonly employed approaches.

The performance of the $X X^{T}$ scheme on ASCI-Red is illustrated in Fig. 4 for a ( $63 \times 63$ ) and ( $127 \times 127$ ) point Poisson problem ( $n=3069$ and $n=16129$, respectively) discretized by a standard five-point stencil. Also shown are the times for the commonly used approaches of redundant banded- $L U$ solves and row-distributed $A_{0}^{-1}$. The latency $2 \log P$ curve represents a


Figure 4: ASCI-Red solve times for a 3969 (left) and 16129 (right) d.o.f. coarse grid problem.
lower bound on the solution time, assuming that the required all-to-all communication uses a contention free fan-in/fan-out binary tree routing. We see that the $X X^{T}$ solution time decreases until the number of processors is roughly 16 for the $n=3969$ case, and 256 for the $n=16129$ case. Above this, it starts to track the latency curve, offset by a finite amount corresponding to the bandwidth cost. We note that $X X^{T}$ approach is superior to the distributed $A_{0}^{-1}$ approach from a work and communication standpoint, as witnessed by the substantially lower solution times in each of the work- and communication-dominated regimes.

## 4 Performance Results

We have run our spectral element code on a number of distributed-memory platforms, including the Paragon at Caltech, T3E-600 at NASA Goddard, Origin2000 and SP at Argonne, ASCIBlue at Los Alamos, and ASCI-Red at Sandia. We present recent timing results obtained using up to 2048 nodes of ASCI-Red. Each node on ASCI-Red consist of two Zeon 333 MHz Pentium II processors that can be run in single- and dual-processor mode. The dual mode is exploited for the matrix-vector products associated with $H, E$, and $\tilde{A}_{k}^{-1}$ by partitioning the element lists on each node into two parts and looping through these independently on each of the processors. The timing results presented are for the timestepping portion of the runs only. During production runs, usually 14 to 24 hours in length, our setup and I/O costs are typically in the range of $2-5 \%$. The test problem is the transitional boundary layer/hemisphere calculation of Fig. 1 at $R e_{\delta}=1600$, using a Blasius profile of thickness $\delta=1.2 R$ as an initial condition. The mesh is an oct-refinement of the production mesh with $(K, N)=(8168,15)$ corresponding to $27,799,110$ grid points for velocity and 22,412,992 for pressure.

Figure 5 shows the time per step (left) and the iteration counts for the pressure and ( $x$ component) Helmholtz solves (right) over the first 26 timesteps. The significant reduction in pressure iteration count is due to the difficulty of computing the initial transients and the benefits gained from the pressure projection procedure. Table 2 presents the total time and sustained performance for the 26 timesteps using a combination of unrolled f 77 loops and assembly coded DGEMM routines. Two versions of DGEMM were considered: the standard version (csmath), and a specially-tuned version (perf) written by Greg Henry at Intel. We note that the average time per step for the last five steps of the 319 GF run is 17.5 seconds. Finally, the coarse grid for this problem has 10,142 distributed degrees of freedom and accounts for $4.0 \%$ of the total solution time in the worst-case scenario of 2048 nodes in dual-processor mode. If the $A^{-1}$ approach were used instead this would have increased to $15 \%$.


Figure 5: $P=2048$ ASCI-Red-333 dual-processor mode results for the first 26 time steps for $(K, N)=(8168,15):$ solution time-per-step (left) and number of pressure and ( $x$-component) Helmholtz iterations per-step (right).

Table 2: ASCI-Red-333: total time and gflops, $K=8168, N=15$.

|  | single (csmath) |  | dual (csmath) |  | single (perf) |  | dual (perf) |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $P$ | time(s) | GFLOPS | time(s) | GFLOPS | time(s) | GFLOPS | time(s) | GFLOPS |
| 512 | 6361 | 47 | 4410 | 67 | 5969 | 50 | 3646 | 81 |
| 1024 | 3163 | 93 | 2183 | 135 | 2945 | 100 | 1816 | 163 |
| 2048 | 1617 | 183 | 1106 | 267 | 1521 | 194 | 927 | 319 |

## 5 Conclusion

We have developed a highly accurate spectral element code based on scalable solver technology that exhibits excellent parallel efficiency and sustains high mflops. It attains exponential convergence, allows a convective CFL of $2-5$, and has efficient multilevel elliptic solvers including a coarse-grid solver with low communication requirements.

## References

[1] J. Blatr Perot, "An analysis of the fractional step method", J. Comput. Phys., 108, pp. 51-58 (1993).
[2] M. Dryja and O.B. Widlund, "An additive variant of the Schwarz alternating method for the case of many subregions", Tech. Rep. 339, Dept. Comp. Sci., Courant Inst., NYU (1987).
[3] C. Farhat and P.S. Chen, "Tailoring Domain Decomposition Methods for Efficient Parallel Coarse Grid Solution and for Systems with Many Right Hand Sides", Contemporary Math., 180, pp. 401-406 (1994).
[4] P.F. Fischer, "Projection techniques for iterative solution of $A \underline{x}=\underline{b}$ with successive right-hand sides", Comp. Meth. in Appl. Mech., 163 pp. 193-204 (1998).
[5] P.F. Fischer, "Parallel multi-level solvers for spectral element methods", in Proc. Intl. Conf. on Spectral and High-Order Methods '95, Houston, TX, A.V. Ilin and L.R. Scott, eds., Houston J. Math., pp. 595-604 (1996).
[6] P.F. Fischer, "An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations", J. of Comp. Phys., 133, pp. 84-101 (1997).
[7] P.F. Fischer, N.I. Miller, and H.M. Tufo, "An overlapping Schwarz method for spectral element simulation of three-dimensional incompressible flows," ANL/MCS Preprint P730-1098.
[8] W.D. Gropp, "Parallel Computing and Domain Decomposition", in Fifth Conf. on Domain Decomposition Methods for Partial Differential Equations, T.F. Chan, D.E. Keyes, G.A. Meurant, J.S. Scroggs, and R.G. Voigt, eds., SIAM, Philadelphia, pp. 349-361 (1992).
[9] Y. Maday and A.T. Patera, "Spectral element methods for the Navier-Stokes equations", in State of the Art Surveys in Computational Mechanics, A.K. Noor, ed., ASME, New York, pp. 71-143 (1989).
[10] Y. Maday, A.T. Patera, and E.M. Rønquist, "An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow", J. Sci. Comput., 5(4), pp. 263-292 (1990).
[11] A.T. Patera, "A spectral element method for fluid dynamics: laminar flow in a channel expansion", J. Comput. Phys., 54, pp. 468-488 (1984).
[12] A. Pothen, H.D. Simon, and K.P. Liou, "Partitioning sparse matrices with eigenvectors of graphs", SIAM J. Matrix Anal. Appl., 113 (1990) pp. 430-452.
[13] H.M. Tufo and P.F. Fischer, "Fast parallel direct solvers for coarse-grid problems", J. Dist. Par. Comp. (accepted).
[14] H.M. Tufo, "Algorithms for Large-Scale Parallel Simulation of Unsteady Incompressible Flows in Three-Dimensional Complex Geometries", Ph.D. Thesis, Brown University (1998).


[^0]:    *This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

