Ph.D. Dissertation Prospectus
Matrix-Free Block Preconditioner for Spectral Element Discretization of the Navier Stokes Equations

P. Aaron Lott

March 23, 2007

1 Introduction

My dissertation research focuses on the construction, implementation and study of efficient parallel matrix-free Navier Stokes preconditioners and solvers that combine ideas and techniques from the Spectral Element and Finite Element communities. The Navier Stokes Equations govern the dynamics of many common fluids, such as water, oil, and blood. Numerical simulation of these equations allows us to better understand the dynamics of fluids for vast ranges of length and time scales. The construction of efficient and scalable parallel algorithms is paramount to being able to utilize new and upcoming generations of supercomputers to simulate complex flows. The efficiency of our method is centered on two main ideas, the first is the use of an accurate high order matrix-free discretization to construct accurate discrete solutions while minimizing memory requirements; the second is the use of fast iterative solvers, via effective preconditioners that take into account parallel architecture and efficient cache use to improve processor performance.

2 Spectral Element Discretization

The spectral element method is a Galerkin method based on the method of weighed residuals, in which a weak form integral equation is solved. When multiple elements are used, the integral equation is broken up into a summation of the integrals on each element. The element based integrals are then approximated by performing numerical quadrature. In particular, velocities and pressure are represented by a basis of high order Legendre polynomials. By choosing the degree of the pressure basis to be two less than the degree of the velocity basis, we form a staggered mesh that results in a stable discretization of the Navier-Stokes Equations. We use Gauss-Legendre-Lobatto and Gauss-Legendre quadrature to integrate the velocity and pressure terms respectively. The resulting system of non-linear matrix equations numerically represent the original integral equation on each element. Inter-element coupling of these matrix equations ensures continuity along elemental boundaries. These inter-element couplings can be enforced by either constructing a fully coupled system of equations, or by performing a gather-scatter operation that averages the solution along element boundaries after element-based matrix-vector products are performed. This gather-scatter operation together with a tensor product representation of the elemental matrices, yields a matrix-free discretization, in which only matrices associated with one-dimensional phenomena need to be stored.

Compared to low order methods, spectral methods require about half as many degrees of freedom in each spatial dimension to accurately resolve a flow. The trade-off for this low memory requirement is an increase in computational cost per degree of freedom. By using the Spectral Element Method, we retain the accuracy of spectral methods and gain the flexibililty of a matrix-free discretization to invoke cache efficient element-based matrix-matrix calculations [8]. These element-based calculations provide improved parallelism over spectral and low order methods via reduced global communication and small surface to volume ratios on each element. Together, these computational efficiencies offset the added cost per degree of freedom, thus making spectral elements a competitive choice for discretizing the Navier-Stokes equations.

3 Incompressible Navier Stokes Equations

The Navier Stokes equations govern the flow of an incompressible fluid by enforcing the conservation of mass, and momentum on a fluid volume. They can be written as

$$\frac{\partial \vec{u}}{\partial t} - \nu \nabla^2 \vec{u} + (\vec{u} \cdot \nabla)\vec{u} + \nabla p = f, \qquad \nabla \cdot \vec{u} = 0 \tag{1}$$

where u represents velocity, p represents pressure, f represents body forces, t represents time and ν represents the fluid's viscosity. Following discretization strategy discussed in section 2 we discretize the spatial components of (1) and obtain the system of nonlinear equations

$$\begin{bmatrix} F(u) & -D^T \\ -D & 0 \end{bmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} Mf \\ 0 \end{pmatrix} \tag{2}$$

where $F(u)$ is a non-symmetric, nonlinear advection-diffusion operator, D is a discrete divergence operator and D^T is a discrete gradient operator. Solving this system requires a nonlinear iteration scheme. This is done using a combination of Picard and Newton iterations. Since Picard iteration has a large domain of convergence but converges linearly, it is used to obtain a starting point for a Newton iteration, which has a narrower range of convergence but converges quadratically. Inside each nonlinear iteration, the non-symmetric block system must be solved. This is solved using an iterative scheme, namely, preconditioned Flexible GMRES [7]. In section 5 we discuss the preconditioning methods used to expedite the convergence of GMRES, but first we introduce a fast solver for tensor product based computations.

4 Fast Diagonalization Method (FDM)

The spectral element discretization enables us to write the Navier Stokes equations as sums of tensor products on each element. This is particularly useful when performing matrix-vector products, and when solving certain local systems of equations. The Fast Diagonalization Method (FDM) [6] allows for the inversion of $n^d \times n^d$ tensor product based matrices in $O(n^{d+1})$ operations, where d represents the number of spatial dimensions and n represents the number of quadrature points used on a single element.

Suppose a system of equations can be written as

$$C = A \otimes B + B \otimes A. \tag{3}$$

Then if A is symmetric and B is symmetric positive definite, a set of eigenvalues $\{\lambda_i\}$ and eigenvectors $\{v_i\}$ can be found such that

$$V^T A V = \Lambda, \qquad V^T B V = I. \tag{4}$$

This allows for one to inexpensively apply the action of the inverse of C, via

$$C^{-1} = (V \otimes V)(I \otimes \Lambda + \Lambda \otimes I)^{-1}(V^T \otimes V^T), \tag{5}$$

which only depends on the inverse of a diagonal matrix. We have extended the Fast Diagonalization Method to allow for non-symmetric A if B is diagonal, which is the case in the Spectral Element Discretization of the advection-diffusion equation with constant advection coefficient. We will exploit this in the application of our block preconditioner as discussed in the next section.

5 Block Preconditioning Navier-Stokes Equations

Block Preconditioning consists of two important parts, choosing an appropriate block form to allow for the eigenvalues of the system to be clumped in a few small intervals, and employing methods to efficiently perform subsidiary solves while retaining a similar eigenstructure to the original problem. In [2] a block preconditioning strategy was introduced, in which the block form looked like

$$P = \begin{bmatrix} P_F & -D^T \\ 0 & -P_s \end{bmatrix}. \tag{6}$$

By choosing a P_F that cheaply retains the spectral properties of $F(u)$ in (2), and a P_s that cheaply mimics the Schur complement $DF(u)^{-1}D^T$, one may obtain an effective preconditioner. To perform matrix-vector products involving P_F^{-1}, previous authors have used multigrid with stream-wise diffusion and splitting. Due to our matrix-free constraint we only perform the action of F on a vector, and so we cannot perform standard splitting techniques on F. However, we wish to retain the accelerated convergence of the Krylov subspace methods obtained via splitting by instead formulating P_F using element-wise approximations of $F(u)$ and locally approximating the advection coefficient by a constant vector. With this local approximation, we can use the Fast Diagonalization Method to perform element-wise solves, and combine these local approximations with a coarse grid solution using a non-overlapping Additive Schwarz method similar to that described in [8]. The advection-diffusion preconditioner can now be written as

$$P_F^{-1} = R_0 F_0^{-1}(w) R_0^T + \sum_{i=1}^{M} R_i^T F_{\Omega_i}^{-1}(w) R_i. \tag{7}$$

We use a similar technique for performing P_s^{-1}, but first we approximate $DF^{-1}D^T$ using the "Least-Squares-Commutator" as proposed by [1]. Using this approximation, we get

$$P_s^{-1} = (DM^{-1}D^T)^{-1}(DM^{-1}FM^{-1}D^T)(DM^{-1}D^T)^{-1}. \tag{8}$$

This inverse entails two Poisson type solves of the form $(DM^{-1}D^T)^{-1}$. We also use the Fast Diagonalization Method to perform each of these solves element-wise.

6 Contributions

The primary contributions of this research include:

- Development of fast steady Navier-Stokes equation solver
- Development of fast subsidiary solves for the Advection-Diffusion equation and Least Squares Commutator
- Creation of parallel, object-oriented software to implement these methods
- Implementation of Poisson, Advection-Diffusion, Least-Squares Commutator and Navier-Stokes solvers and preconditioners within a matrix-free spectral element framework
- Comparison of convergence results from our method with those from preconditioning methods described in [1], [2], [3], and [5].

References

[1] H. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, *Block preconditioners based on approximate commutators*, SIAM J. Sci. Comput., 27 (2006), pp. 1651–1668.

[2] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[3] P. Fischer, *An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations*, Journal of Computational Physics, 133 (1997), pp. 84–101.

[4] W. Gropp, D. Keyes, L. McInness, and M. Tidriri, *Globalized Newton-Krylov-Schwarz algorithms and software for parallel implicit CFD*, The International Journal of High Performance Computing Applications, 14 (2000), pp. 102–136.

[5] J. Lottes and P. Fischer, *Hybrid multigrid/schwarz algorithms for the spectral element method*, J. Sci. Comput., 24 (2005), pp. 613–646.

[6] R. Lynch, J. Rice, and D. Thomas, *Direct solution of partial difference equations by tensor product methods*, Numer. Math., 6 (1964), pp. 185–199.

[7] Y. Saad, *A flexible inner-outer preconditioned gmres algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[8] H. Tufo and P. Fischer, *Terascale spectral element algorithms and implementations*, in Supercomputing '99: Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM), New York, NY, USA, 1999, ACM Press, p. 68.