Parallel Computational Fluid Dynamics – Practice and Theory
P. Wilders, A. Ecer, J. Periaux, N. Satofuka and P. Fox (Editors)
© 2002 Elsevier Science B.V. All rights reserved.

A Parallel Solenoidal Basis Method for Incompressible Fluid Flow Problems*

Sreekanth R. Sambavaram and Vivek Sarin
Department of Computer Science
Texas A&M University
College Station, TX 77843

The convergence of iterative methods used to solve the linear systems arising in incompressible flow problems is sensitive to flow parameters such as the Reynolds number, time step and the mesh width. Incompressibility of the fluid makes the systems indefinite, and poses difficulty for the iterative solvers. This paper outlines a class of solenoidal basis methods that use local solenoidal functions to restrict fluid velocity to divergence-free subspace. An optimal preconditioner based on the Laplace operator is used to solve the resulting ill-conditioned reduced system. Experimental results for two and three dimensional problems show that the convergence of the proposed algorithm is optimal across the range of flow parameter variation. Scalability of the algorithm is suggested by the experiments on the SGI Origin 2000.

1. INTRODUCTION

Large-scale simulation of incompressible flow is one of the most challenging application. Realistic simulations are possible only with the use of sophisticated modeling techniques, preconditioned iterative methods and advanced parallel architectures. The motivation for this work is to develop an effective approach for solving the linear systems arising in incompressible flows with high efficiency on a multi-processor platform.

The principles of classical mechanics, thermodynamics, and laws of conservation of mass, momentum, and energy govern the motion of the fluid. Law of conservation of momentum for incompressible, viscous flow in a region Ω with boundary $\partial\Omega$ is captured by Navier-Stokes equation given by

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{R}\Delta \mathbf{u}, \tag{1}$$

where $p = p(\mathbf{x}, t)$ is the pressure, R is the Reynolds number, and $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity vector at $\mathbf{x}$. The law of conservation of mass for incompressible fluids gives rise to the continuity equation

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega. \tag{2}$$

*This work has been supported in part by NSF under the grants NSF-CCR 9984400, NSF-CCR 9972533, and NSF-CTS 9873236. Computational resources were provided by NCSA, University of Illinois at Urbana-Champaign.

Appropriate boundary conditions may be specified for fluid velocity. Suitable discretization and linearization of the equations (1)–(2) result in the following linear system

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \tag{3}$$

where B^T is the discrete divergence operator and A is given by

$$A = \frac{1}{\Delta t} M + C + \frac{1}{R} L, \tag{4}$$

in which M is the mass matrix, L is the Laplace matrix, and C is the matrix arising from the convection term. When operator splitting is used to separate the linear and non-linear terms, we obtain the generalized Stokes problem (GSP) with a symmetric positive definite A given as

$$A = \frac{1}{\Delta t} M + \frac{1}{R} L. \tag{5}$$

The linear system (3) is large and sparse. Although direct methods can be used to solve this system, they require prohibitively large amount of memory and computational power. The inherent sequential nature of these techniques limits the efficiency on parallel architectures. In contrast, iterative methods require significantly less memory and are well suited for parallel processing. These methods can be made more reliable by using preconditioning techniques which accelerate convergence to the solution. In order to make iterative methods more competitive, one must devise robust preconditioning techniques that are not only effective but parallelizable as well.

This paper presents a preconditioned solenoidal basis method to solve the linear system (3) arising in the generalized Stokes problem. Section 2 describes the solenoidal basis method and section 3 outlines the preconditioning scheme. Experiments for the driven cavity problem in 2D and 3D are presented in section 4.

2. A SOLENOIDAL BASIS METHOD

Incompressible fluid flow can be viewed as compressible flow with additional constraint that fluid velocity should be divergence free. This incompressibility constraint in (3) $B^T u = 0$, makes the linear system indefinite. This indefinite nature is the main cause of difficulty in solving the system via preconditioned iterative methods. The degree of difficulty also depends upon the nature of matrix A which is affected by the Reynolds number R and the choice of time step Δt and mesh width.

Solenoidal basis methods are a class of techniques that use a divergence-free or solenoidal basis to represent velocity. A discrete solenoidal basis can be obtained by computing the null space of the divergence operator B^T. A matrix $P \in R^{n \times (n-m)}$ that satisfies the condition $B^T P = 0$ is used to compute divergence-free velocity via the matrix-vector product $u = Px$, for an arbitrary $x \in R^{(n-m)}$. Clearly, such a velocity satisfies the continuity constraint $B^T u = 0$. By restricting u to the column space of P and pre-multiplying the first block of (3) by P^T, we get the following *reduced system*

$$P^T A P x = P^T f, \tag{6}$$

which may be solved by a suitable iterative method such as the conjugate gradients (CG) method, GMRES, etc. (see, e.g., [6]). Once x has been calculated, velocity is computed as

$$u = Px, \tag{7}$$

and pressure is recovered by solving the least squares problem iteratively

$$Bp \approx f - APx. \tag{8}$$

The success of the solenoidal basis method depends on a number of factors. First, the matrix-vector product with the reduced system must be computed efficiently. Second, one must develop a robust and effective preconditioner for the reduced system. Finally, these computations must be implemented efficiently on a parallel processor.

At each iteration, the matrix-vector product with P^TAP is computed as a series of three matrix-vector products with P, A, and P^T, respectively, in that order. Each column of P represents a solenoidal function with a localized region of influence on the mesh. As an example, consider a uniform 3D mesh to discretize a driven cavity problem via the Marker-and-Cell (MAC) scheme. The MAC scheme assigns pressure unknowns to each node and velocity unknowns to each edge. One can construct a local circulating flow by assigning appropriate velocity to the edges forming a face of a given cell in this mesh. Such a flow is represented as a vector, and the set of these vectors form the columns of P. The localized nature of these flows leads to a sparse P with a nonzero structure resulting from the underlying mesh. This may be used to compute Px and P^Ty efficiently in parallel. Furthermore, one can apply P and P^T to a vector without actually constructing P itself. This feature has been exploited to develop a matrix-free implementation.

We proposed the use of local solenoidal functions for 2D flows in [7], where we presented a scheme to construct a solenoidal basis derived from circulating flows or vortices on uniform meshes. We also outlined an optimal preconditioning technique for the generalized Stokes problem. In [8], we introduced a linear algebraic technique to construct a hierarchical basis of solenoidal functions which is applicable to the generalized Stokes problem on arbitrary meshes. This approach was successfully applied to 2D particulate flow problems using structured meshes [3,5,9] and was extended to unstructured meshes [3]. Details of a distributed memory parallel implementation were presented in [3]. Several schemes have been proposed for computing discrete solenoidal functions [2,1]. Unlike other schemes, our approach can be formulated as a linear-algebraic method which is applicable to arbitrary discretization schemes including finite element and finite volume methods.

In this paper, we extend the solenoidal basis method to 3D problems defined on uniform meshes. In 3D, the solenoidal basis P constructed from local solenoidal functions turn out to be rank-deficient due to linear dependence between the local circulating flows. However, it can be shown that the space of discrete solenoidal functions is contained within the column space of P. Since the reduced system is consistent despite the rank-deficiency of the system matrix, it can be solved by preconditioned CG or GMRES. Iterative solution of the driven cavity problem via the MAC scheme exhibits optimal convergence rate. The parallel implementation demonstrates good speed improvement on a medium-sized multiprocessor.

3. ACCELERATING CONVERGENCE BY PRECONDITIONING

Effective preconditioning of the reduced system is critical to the overall success of the solenoidal basis method. The design of the preconditioner becomes challenging because the reduced system matrix P^TAP is ever explicitly formed. One can take advantage of the analogy between matrix vector products involving P in the solenoidal basis method with vortex methods to construct the preconditioner.

Vortex methods are a class of techniques that solve the vorticity transport equation instead of the Navier-Stokes equation. Vorticity field ξ is expressed in terms of velocity $\mathbf{u}$ and velocity is in turn obtained by applying curl operator on scalar stream function ψ. In particular, vorticity is given as $\xi = \nabla \times \mathbf{u}$ and velocity is expressed as $\mathbf{u} = \nabla \times \psi$. The relation between ξ and ψ is given by the Poisson equation $\Delta\xi = -\psi$ The matrix vector product $u = Py$ computes the velocity vector function $u = \nabla \times \mathbf{y}$ and the matrix vector product $w = P^Tu$ computes the vorticity vector function $\mathbf{w} = \nabla \times \mathbf{u}$. Further more, y and w are analogous to the velocity potential ψ and ξ. The relation $\mathbf{w} = \nabla \times \nabla \times \mathbf{y}$ can be implemented via matrix vector product $w = P^TPy$. Since, w and y are assumed divergence-free, it is easy to show that $-w = \Delta y$, which is identical to the relation between ψ and ξ.

Observing that the product Px and P^Ty compute the discrete curl of the functions represented by x and y, respectively, it can be inferred that the product $y = P^TPw$ represents $\nabla \times \nabla \times \mathbf{w}$ in a discrete setting. Thus, the matrix P^TP can be shown to be equivalent to the Laplace operator on the solenoidal function space. This suggests the following preconditioner for the generalized Stokes problem:

$$G = \left[\frac{1}{\Delta t}M + \frac{1}{R}L_s\right] L_s, \tag{9}$$

where L_s is the Laplace operator for the local solenoidal functions. Since the preconditioned system is spectrally equivalent to a symmetric positive definite matrix, one can use preconditioned CG to solve the reduced system (6).

4. EXPERIMENTS

In this section, we present results of numerical experiments for the driven cavity problem. The preconditioned solenoidal basis method was used to solve the linear system arising in the generalized Stokes problem. The experiments were conducted for 2D unit square and 3D unit cube domains. In each case, the MAC scheme was used to discretize the domain. The linear system was solved under various physical conditions by changing the ratio $h^2R/\Delta t$ which determines the condition number of A. For the 3D driven cavity problem, the condition number of A is approximated by

$$\kappa(A) = \frac{h^2R/\Delta t + 12}{h^2R/\Delta t + h^2}. \tag{10}$$

Hence, $\kappa(A) < 2$ when $h^2R/\Delta t > 12$, and $\kappa(A) \approx h^{-2}$ when $h^2R/\Delta t \ll 12$. This ratio also captures the difficulty associated with solving the linear system when parameters such as mesh width (h), Reynolds number (R) and time step (Δt) are changed.

The first set of experiments highlights the effectiveness of the preconditioner in accelerating convergence of the CG method. The linear system in the preconditioning step was solved by CG as well, resulting in an inner-outer scheme. The iterations were terminated when the relative residual was reduced below 10^{-4}. A much larger tolerance (10^{-2}) was used for the inner iterations. Table 1 presents the iterations required by the preconditioned CG method for several instances of $h^2R/\Delta t$. The preconditioner ensures a stable convergence rate independent of the values of various parameters h, R, and Δt, suggesting optimality of the preconditioner. The overall computation time was reduced significantly by using a large threshhold for the inner CG iterations. This choice did not adversely effect convergence of the outer iterations.

Table 1
Convergence rate independent of mesh width, R and Δt.

Mesh	Reduced system size	$h^2R/\Delta t$		
		10^{-2}	10^{0}	10^{+2}
2D				
128 × 128	16,128	12	7	4
256 × 256	65,024	12	7	4
512 × 512	261,120	12	7	4
3D				
8 × 8 × 8	1,176	8	6	5
16 × 16 × 16	10,800	12	8	6
32 × 32 × 32	92,256	16	9	7

4.1. Parallel Performance

The solution methodology can be effectively parallelized. On a multi-processor machine with q processors, the domain is partitioned into q partitions, and the underlying mesh is distributed across processors. Parallelization of the computation of P and matrix vector products with P and P^T is fairly straight forward (see, e.g., [7]). Other operations such as vector additions, inner-products and matrix vector products with A are also easy to parallelize. The reader may refer to the texts [6,4]. The linear system in the preconditioning step can be solved via parallel versions of fast poisson solvers, domain decomposition, multi-grid, and multi-level methods.

A second set of experiments focused on the parallelization of the proposed algorithm. All the elementary matrix-vector products were parallelized by distributing the grid equally among the processors. The algorithm was parallelized using OpenMP. Table 2 indicates that the algorithm can be parallelized with high efficiency on a multi-processor platform such as the SGI Origin 2000.

5. CONCLUSIONS

This paper presents a high performance algorithm for solving the linear systems arising from incompressible flow problems. The proposed solenoidal basis method uses discrete

Table 2
A parallel implementation using OpenMP demonstrates good speed improvement on 16 processors of SGI Origin 2000.

	Mesh size = $256 \times 256 \times 256$	
Processors	Time	Speedup
1	2684.10	1.0
2	1552.95	1.7
4	961.71	2.8
8	473.67	5.7
16	278.88	9.5

local solenoidal functions to represent divergence-free velocity. A reduced system is solved in the divergence-free subspace via a preconditioned iterative scheme. An optimal preconditioner has been suggested which assures stable convergence regardless of parameters such as the mesh width, Reynolds number, and the time step. An inexpensive low accuracy iterative solve for the preconditioner appears to be sufficient for optimal convergence. The method is parallelizable with high efficiency.

REFERENCES

1. K. Gustafson and R. Hartman. Divergence-free bases for finite element schemes in hydrodyna mics. *SIAM J. Numer. Anal.*, 20(4):697–721, 1983.
2. C. A. Hall, J. S. Peterson, T. A. Porsching, and F. R. S. ledge. The dual variable method for finite element discretizations o f Navier-Stokes equations. *Intl. J. Numer. Meth. Engg.*, 21:883–898, 1985.
3. M. G. Knepley, A. H. Sameh, and V. Sarin. Design of large scale parallel simulations. In D. Keyes, A. Ecer, J. Périaux, N. Satofuka, and P. Fox, editors, *Parallel Computational Fluid Dynamics*. Elsevier, 2000.
4. V. Kumar, A. Grama, A. Gupta, and G. Karypis. Introduction to parallel computing: Design and analysis of algorithms. Benjamin/Cummings, 1994.
5. T.-W. Pan, V. Sarin, R. Glowinski, A. H. Sameh, and J. Périaux. A fictitious domain method with distributed Lagrange multipilers for the numerical simulation of particulate flow and its parallel implementation. In C. A. Lin, A. Ecer, N. Satofuka, P. Fox, and J. Périaux, editors, *Parallel Computational Fluid Dynamics, Development and Applications of Parallel Technology*, pages 467–474. North-Holland, Amsterdam, 1999.
6. Y. Saad. Iterative methods for sparse linear systems. PWS publishing company, 1996.
7. V. Sarin. Parallel linear solvers for incompressible fluid problems. In *Proceedings of the SIAM Conference on Parallel Processing for Scientific Computing*. Portsmouth, VA, Mar. 2001.
8. V. Sarin and A. H. Sameh. An efficient iterative method for the generalized Stokes problem. *SIAM Journal of Scientific Computing*, 19(1):206–226, 1998.
9. V. Sarin and A. H. Sameh. Large scale simulation of particulate flows. In *Proceedings of the Second Merged Symposium IPPS/SPDP*, pages 660–666, Apr. 1999.