A Two-Level Aggregation-Based Newton-Krylov-Schwarz Method for Hydrology *

E. W. Jenkins[a], R. C. Berger[b], J. P. Hallberg[b], S. E. Howington[b], C. T. Kelley[a], J. H. Schmidt[b], A. K. Stagg[b], and M. D. Tocci[c]

[a]North Carolina State University
Center for Research in Scientific Computation and Department of Mathematics
Box 8205, Raleigh, N. C. 27695-8205, USA
(ewjenkin@eos.ncsu.edu, Tim_Kelley@ncsu.edu)

[b]US Army Waterways Experiment Station
Coastal And Hydraulics Laboratory and Information Technology Laboratory
3909 Halls Ferry Road Vicksburg, Mississippi 39180, USA
(berger@hl.wes.army.mil, pettway@juanita.wes.army.mil,
stacy@hwy61.wes.army.mil, schmidt@href.wes.army.mil,
stagg@rusty.wes.hpc.mil)

[c]Department of Mathematical Sciences
Worcester Polytechnic Institute Worcester, MA 01609, USA
(mdtocci@wpi.edu)

Abstract

Newton-Krylov-Schwarz methods solve nonlinear equations by using Newton's method with a Schwarz domain decomposition preconditioned Krylov method to approximate the Newton step. In this paper we discuss the design and implementation of Newton-Krylov-Schwarz solvers in the context of the implicit temporal integration on an unstructured three-dimensional mesh of the Navier-Stokes equations for modeling flow in a river bend.

1. INTRODUCTION

In this paper we discuss the design and implementation of a Newton-Krylov-Schwarz solver for the implicit temporal integration on an unstructured three-dimensional spatial mesh of time-dependent partial differential equations. The novel feature of this approach is the formation of a coarse mesh problem using aggregation methods from algebraic multigrid [14]. The solver was tested within the **Ad**aptive **H**ydrology (ADH) Model, a

*This research was supported in part by Army Research Office contract DAAD19-99-1-0186, US Army contract DACA39-95-K-0098, National Science Foundation grant DMS-9700569, a Cray Research Corporation Fellowship, and a Department of Education GAANN fellowship. Computing activity was partially supported by an allocation from the North Carolina Supercomputing Center.

finite element code being developed by the Army Corps of Engineers Waterways Experiment Station (WES) that is designed to solve a variety of hydrology problems including surface water flow.

In [10] we will apply this approach to the solution of Richards' equation for groundwater flow in the unsaturated zone, and in [7] we discuss an application to surface water-groundwater interaction. In this paper we report on the performance of the method in a Navier-Stokes simulation.

The Navier-Stokes equations in terms of velocity $u(x,t)$ and pressure $p(x,t)$, can be written as

$$\begin{aligned} \rho\left(\frac{\partial u}{\partial t} + u \cdot \nabla u\right) - \nabla \cdot \sigma &= 0 \\ \nabla \cdot u &= 0 \end{aligned} \tag{1}$$

where

$$\begin{aligned} \sigma &= -pI + \tau, \\ \tau &= 2\mu D, \\ D_{ij} &= \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \text{ and} \\ \mu &= \text{ kinematic velocity.} \end{aligned} \tag{2}$$

These equations are applicable within the domain $\Omega \subset \mathcal{R}^3$ with the following boundary conditions:

$$\begin{aligned} u &= \hat{u} \text{ on } \Gamma_g \\ \sigma &= \hat{\sigma} \text{ on } \Gamma_h \\ u \cdot n &= h \text{ on } \Gamma_h. \end{aligned} \tag{3}$$

The boundary of Ω is denoted as Γ, and n is the outward normal. Γ_g and Γ_h represent non-overlapping subregions of Γ such that

$$\Gamma = \overline{\Gamma_g \cup \Gamma_h}.$$

In § 3 we give numerical results for a test case. These results show that our preconditioners have good scalability and that our coarse grid formulation is performing well.

2. NEWTON-KRYLOV-SCHWARZ

The weak formulation of the Navier-Stokes equations as given in [1] leads to implicit temporal integration. The discretization of the weak formulation leads to a system of nonlinear equations that must be solved at each time step. These equations are solved via Newton-Krylov-Schwarz (NKS) methods, which are described below.

NKS methods [11] use a Krylov subspace method to determine the Newton step s in

$$F'(u_c)\, s = -F(u_c)$$

where $F'(u_c)$ is the Jacobian at the current iteration. A Schwarz type preconditioner is used to accelerate the performance of the Krylov solver. The Krylov method used in the ADH Model is the Bi-CGStab method [13].

Both one-level additive Schwarz [3] and two-level additive Schwarz preconditioners [3] [4] are currently implemented in the ADH Model. Both of these preconditioners are domain decomposition type preconditioners, which means that the original domain is split into several subdomains, and the solutions of the original problem restricted to the subdomains are combined to form the preconditioner for the original system. If we define a matrix R_i to be the restriction matrix for subdomain i so that $R_i = [0 \;\; I \;\; 0]$, where I is an $n_i \times n_i$ identity matrix and n_i is the size of subdomain i, then the one-level additive Schwarz preconditioner can be written as

$$M = \sum_{i=1}^{p} R_i^T \left(R_i A R_i^T \right)^{-1} R_i$$

where p is the number of subdomains.

The coarse mesh component of the preconditioner is formed by defining one aggregate element per subdomain. The resulting coarse mesh basis function is constant except in the elements shared between subdomains. The contribution of the subdomain to the coarse matrix is computed locally and then communicated to all of the processors. Thus every processor solves the coarse mesh problem. The subdomain solves are performed using a profile solver [5] and the coarse grid problem is solved using a dense LU factorization. The two-level additive Schwarz preconditioner is formed by adding the coarse mesh problem to the one-level preconditioner, so that

$$M = R_0^T \left(R_0 A R_0^T \right)^{-1} R_0 + \sum_{i}^{p} R_i^T \left(R_i A R_i^T \right)^{-1} R_i$$

where R_0 and R_0^T are the restriction and interpolation operators from the fine to coarse meshes, respectively.

3. NUMERICAL EXPERIMENTS

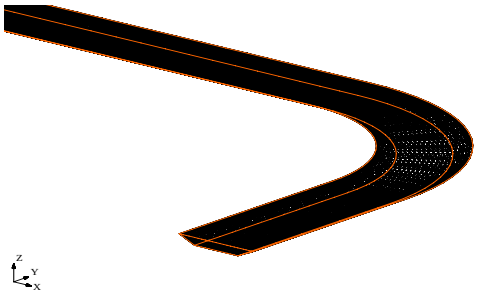
In this section we report on a Navier-Stokes simulation of flow through a river bend. The purpose of this test was to investigate the performance of the preconditioners near steady state. Riprap refers to a foundation or wall of stones or similar material used on riverbanks to prevent erosion. The "riprap" problems were designed to aid in the placement and size determination of riprap material along natural rivers and channels. The models only incorporate a short section of the test facility, and this short section does contain a river bend.

The riprap model has a trapezoidal cross-section, and riprap material is placed along the sidewalls. The flow in the cross-sectional area of the bend is in the form of a spiral, and the net flow from side to side in the cross-section is to the outside of the bend at the surface of the water and to the inside of the bend close to the bottom. There is more sediment at the bottom of the river bend than at the top, so normally the outside bend is eroded while a bar is developed on the inside of the bend. These riprap model results are being used to evaluate three-dimensional models of the river bend.

The equations were discretized on unstructured tetrehedral meshes in three space dimensions. We used the piecewise constant in time and piecewise linear in space finite element discretizations from [1]. These discretizations are implicit in time and therefore a discretized nonlinear elliptic problem must be solved at each time step. The Galerkin least squares methods of [6], [8], [9], and [12] were used to stabilize the discretization.

The meshes were generated using GMS [2]. Initially a straight channel with a trapezoidal cross-section was generated using the GMS tool. GMS created an element connection table, which was then used to move to nodes and construct the bend. The mesh is shown in Figure 1.

Figure 1. Riprap Mesh



After the grid was generated, the nodes were renumbered. The numbering of the nodes plays an important role in the performance of the preconditioner because of the way node allocation is currently done. As the present time, the nodes are divided among processors sequentially; i.e., the first n nodes go to processor 0, the next n to processor 1, and so forth. The number of subdomains per processor is an input parameter to the code. Once the nodes have a processor assignment, they receive a subdomain assignment. This assignment occurs in the same sequential fashion as the processor assignment. The coarse grid elements are defined on each processor, so the nodes that are assigned to that processor should be as physically close to one another as possible. The renumbering algorithm numbers the nodes in the vertical direction in the innermost loop, in the lateral direction in the first outer loop, and longitudinally in the final outer loop. This ensures that the numbering occurs across an entire cross-section before moving to the next cross-section, and it also ensures that the numbering is done with respect to the aspect ratios present in the mesh.

The nonlinear solver terminates when

$$\|F(u)\|_\infty < 10^{-5} \tag{4}$$

and the criterion for termination of the linear solver is

$$\|F'(u)\, s + F(u)\|_\infty < 10^{-7}. \tag{5}$$

The initial conditions were near steady state. Four time steps were taken for the smaller of the two meshes and sixteen time steps were taken for the larger mesh. The smaller

mesh had 5881 nodes and 30720 elements and the larger mesh had 43889 nodes and 245760 elements. In this way the mesh width was roughly halved. For a regular grid, the condition number of the linearized problem is $1 + O(\delta t/h^2)$, where δt is the time step and h the spatial mesh width and the accuracy is $O(\delta t + h^2)$. Motivated by these estimates, we reduced the time step was reduced by a factor of four for the larger problem, keeping the condition number roughly constant and increasing the temporal accuracy along with the spatial accuracy.

In Table 1 we report iteration counts for the small problem on 8 processors with 8 subdomains per processor, the larger problem on 64 processors and 4 subdomains per processor with the time step reduced by a factor of 4. The reduction in the number of subdomains per processor for the larger problem was necessary because the communication costs of assembling the coarse mesh problem with 8 subdomains per processor was too high. For the coarse mesh problem, using 8 subdomains per processor gave the lowest execution time because the subdomain factorization cost dominated that for communication of coarse mesh data.

The iteration counts provided in Table 1 are the total number of linear iterations used during the simulation. The timings are given in seconds and are the timings for the entire simulation, including the initialization, calculation of the Jacobian, the construction and application of the preconditioners, and the nonlinear solves. We have measured performance in this manner because we are not solely interested in the performance of the preconditioner, but its impact on the entire application. We compare Jacobi and two-level Schwarz iteration. One-level Schwarz did not perform as well as Jacobi in our experiments.

For the runs labeled Schwarz, the coarse grid was computed, factored, and stored once for every 10 nonlinear iterations. This means the coarse grid was computed and factored only once for the smaller problem and twice for the larger problem. This lag factor appears to be much more important for the larger problems. We are in the process of determining when lag factors are necessary and what the optimal lag factor is.

All of the numerical results were calculated on an IBM SP2 located at the WES Major Shared Resource Center. The SP2 has 255 processors, with an aggregate memory size of 256 Gbytes. The operating system is the SP/135 AIX, Version 4.1.5.x, and the IBM parallel operating environment (poe), Version 2.1.0.24, is used for batch processing. The compiler is C for AIX with message passing, Version 3.1.4.

Table 1
Riprap Problem Iterations

	Small		Large	
	Jacobi	Schwarz	Jacobi	Schwarz
Linear Iterations	12398	1467	16279	1397
Time (seconds)	1633	953	4807	2663
Linear Iterations/time step	3099	367	1017	87

4. CONCLUSIONS

While the preconditioners were originally designed for use on groundwater problems, they perform well for Navier-Stokes simulations. Our use of a coarse mesh problem reduces the number of iterations, while lagging the coarse problem maintains the reduction in iteration counts while simultaneously reducing the execution time for the simulation. Our formulation of the coarse mesh with aggregate elements led to an easy construction of the coarse-mesh problem and trivial intergrid transfers.

The performance will be improved in our future work by lagging the fine-mesh subdomain factorizations, using more efficient subdomain solvers or incomplete factorizations, and updating preconditioning information adaptively based on the behavior of the solution. We observed that the number of subdomains per processor has a significant effect on the performance of the preconditioner and optimization of this as a function of problem size and computer architecture is an open problem.

5. ACKNOWLEDGEMENTS

The authors wish to thank David Keyes, Jun Zou, Carol Woodward, Van Henson, and Jim Jones for many helpful discussions.

REFERENCES

1. M. Behr and T. E. Tezduyar. Finite element solution strategies for large–scale flow simulations. *Computer Methods in Applied Mechanics and Engineering*, 112:3–24, 1994.
2. Brigham Young University. *GMS – The Department of Defense Groundwater Modeling System Reference Manual.* Brigham Young University Engineering Computer Graphics Laboratory, Provo, Utah, 1994.
3. M. Dryja and O. B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, Courant Institute, 1987. also Ultracomputer Note 131.
4. M. Dryja and O. B. Widlund. Some domain decomposition algorithms for elliptic problems. In D. R. Kincaid and L. J. Hayes, editors, *Iterative Methods for Large Linear Systems.* Academic Press, San Diego, 1990.
5. I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices.* Oxford University Press, New York, NY, 1986.
6. L. P. Franca and T. J. R. Hughes. Two classes of mixed finite element methods. *Computer Methods in Applied Mechanics and Engineering*, 69:88–129, 1988.
7. S. E. Howington, R. C. Berger, J. P. Hallberg, J. F. Peters, A. K. Stagg, E. W. Jenkins, and C. T. Kelley. A model to simulate the interaction between groundwater and surface water, 1999. Proceedings of the High Performance Computing Users' Group Meeting, Monterrey, CA, June 7–10.
8. T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the Babuška – Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accomodating equal–order

interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.

9. T. J. R. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the Galerkin/least squares method for advective–diffusion equations. *Computer Methods in Applied Mechanics and Engineering*, 73:173–189, 1989.
10. E. W. Jenkins, Joseph H. Schmidt, Alan Stagg, Stacy E. Howington, R. C. Berger, J. P. Hallberg, C. T. Kelley, and M. D. Tocci. Newton-Krylov-Schwarz methods for Richards' equation. In preparation.
11. D. E. Keyes. Aerodynamic applications of Newton-Krylov-Schwarz solvers. to appear in Proc. of 14th International Conference on Num. Meths. in Fluid Dynamics. (R. Narishima et al. eds), Springer, NY 1995.
12. T. E. Tezdoyar, S. Mittal, S. R. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal–order–interpolation velocity–pressure elements. *Computational Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.
13. H. A. van der Vorst. Bi–CGSTAB: A fast and smoothly converging variant to Bi–CG for the solution of nonsymmetric systems. *SIAM J. Sci. Statist. Comput.*, 13:631–644, 1992.
14. P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.