

KNOLL, D.A.; MCHUGH, P.R.; KEYES, D.E.

Domain Decomposition Methods Applied to a System of Convection-Diffusion-Reaction Equations

Obtaining efficient and robust solutions to realistic fluid flow problems with chemical reactions is a challenging computational task. This is due to the potential wide disparity in time and space scales and the severe nonlinearities. However, there is significant industrial application for such a capability in areas such as energy production, propulsion, and materials processing. These applications are often characterized by regions of low Mach number flow. This study investigates Newton-Krylov algorithms with domain-based preconditioning for the solution of these types of problems. Domain-based preconditioners may offer improvements in robustness and parallelism over more standard ILU type preconditioners. Results are presented comparing additive and multiplicative Schwarz preconditioners to more standard global ILU preconditioners on a low Mach number combustion problem.

1. Introduction

Mathematically, the time dependent compressible Navier-Stokes equations become stiff at low values of the reference Mach number. This stiffness is a result of the disparity in time scales between convection and acoustic waves. Including combustion further complicates the problem by contributing additional time scales and nonlinearities. In this paper we consider domain decomposition preconditioning, namely additive and multiplicative Schwarz methods [1], for Newton-Krylov algorithms applied to a low Mach number combustion problem. GMRES [2] is used as the Krylov algorithm and a matrix-free Newton-Krylov implementation [3] is employed. Both additive and multiplicative Schwarz methods are considered, along with different subdomain solvers and partitioning strategies.

2. Newton-Krylov-Schwarz Methods

Domain-based additive and multiplicative Schwarz methods are used to precondition the linear systems arising on each step of a pseudo-transient Newton-GMRES(40) algorithm. The right preconditioned linear systems are of the form

$$\left(\frac{V}{\Delta t^n} + J^n\right)P^{-1}(P\delta x^n) = -F(x^n). \quad (1)$$

$F(x^n)$ is the residual vector at the n^{th} Newton iteration representing the system of discrete algebraic equations. δx^n is the vector update used to obtain the new solution approximation (i.e., $x^{n+1} = x^n + s\delta x^n$, where s is a damping coefficient). J is the numerically evaluated Jacobian matrix, P is the preconditioning matrix, and V is a diagonal matrix with entries equal to cell volumes. Δt^n is the time step used in the pseudo-transient continuation. This time step is initially set near the explicit stability limit, but is increased according to a function that is inversely proportional to the infinity norm of the steady-state nonlinear residual, $\|F(x^n)\|_\infty$. An inexact Newton's method [3] is employed in that the convergence tolerance of the GMRES algorithm is related to the current residual of the nonlinear problem. The convergence tolerance of the nonlinear problem is based on both $\|\delta x^n\|_\infty$ and $\|F(x^n)\|_2$. In results presented here we require that they both be less than 10^{-5} .

The Newton-Krylov implementation used here replaces the Jacobian-vector products in the GMRES algorithm with finite difference projections [4]. Consequently, terms of the form $JP^{-1}v$ are computed from,

$$JP^{-1}v \approx \frac{F(x + \epsilon P^{-1}v) - F(x)}{\epsilon}. \quad (2)$$

Here v is a GMRES vector and ϵ is a small perturbation constant. The advantage of using this implementation is that it approximates the action of the Jacobian without explicitly computing the full Jacobian matrix. This feature is significant for combustion-type problems where the formation of the Jacobian can account for a large fraction of the CPU time. Typically, however, the Jacobian is needed periodically during the outer Newton iteration in order to generate an effective preconditioner for the Krylov iteration. In this situation, the primary advantage of this implementation is the ability to amortize the cost of these periodic Jacobian and preconditioner evaluations over several Newton iterations. In the results presented in this paper, the Jacobian and preconditioner are formed every

five Newton iterations. However, this evaluation frequency could be made adaptive as is often done with standard adaptive modified Newton’s method [5].

The two domain decomposition algorithms investigated here are the additive and multiplicative Schwarz [1] methods. The preconditioners for these options are defined by:

$$\text{Additive Schwarz: } P_{AS}^{-1} = J_1^{-1} + J_2^{-1} + \dots + J_p^{-1},$$

$$\text{Multiplicative Schwarz: } P_{MS}^{-1} = I - (I - J_1^{-1}J) \dots (I - J_p^{-1}J),$$

where the action of the multiplicative Schwarz preconditioner, i.e. $w = P_{MS}^{-1}v$ can be computed by [3],

$$\begin{aligned} v_1 &= J_1^{-1}v \\ v_j &= v_{j-1} + J_j^{-1}(v - Jv_{j-1}), \quad \text{for } j = 2, \dots, p \\ w &= v_p. \end{aligned}$$

Here J_1, \dots, J_p are specific subdomain contributions to the full Jacobian matrix J . We will consider both exact and approximate factorizations for efficiently calculating terms of the form $J_i^{-1}v$.

3. Model Problem

We solve a two-dimensional system of conservation equations representing compressible, laminar, chemically reacting flow with variable transport coefficients. Each equation in our system has a general convection-diffusion-reaction form closely related to:

$$\frac{\partial \phi_i}{\partial t} + \nabla \cdot (\vec{V}\phi_i - D(\phi)\nabla\phi_i) = S_i(\phi). \quad (3)$$

\vec{V} is the velocity field which is obtained by solving the momentum equations. The transport coefficients, D (species diffusion, viscosity, and thermal conductivity), and the source/sink terms, S , can be nonlinear functions of the dependent variables ϕ_i . Specifically, S in the continuity equations and energy equation contain reaction rates of the Arrhenius form, which means they exhibit an exponential dependence on temperature. A simple combustion system is solved consisting of three chemical species, fuel (A), oxidizer (B), and product (C), and one forward reaction, $A + 2B \rightarrow C$. This system requires the solution of three continuity equations, along with two momentum equations and one energy equation, which results in a system of six coupled, nonlinear convection-diffusion-reaction equations. Details on systems of combustion equations in general can be found in [7], and on our specific system in [8]. We discretize these equations using finite volumes on a staggered mesh. First order upwind techniques are used for the convective terms in this paper.

Our model problem represents an idealized non-premixed laminar diffusion flame. The geometry, boundary conditions, and temperature contours on a 90×30 uniform grid are depicted in Figure 1. The inlet Reynolds number is 30 and the inlet Mach number is 0.14. The chemistry rate is set such that the fuel mass fraction drops below 0.1 by $x = 10$, and the peak temperature has risen to 5, from an inlet value of 1. These solution features indicate that the chemistry time scale is faster than the convection time scale. Note the very strong cross-flow temperature gradient near the inlet where combustion is initiated.

4. Results and Discussion

Table 1 compares the performance of additive and multiplicative Schwarz preconditioners without overlap using three different partitioning strategies and three different subdomain solvers. The data presented are the inner (GMRES) iteration counts per Newton step, averaged over ten Newton steps. These ten Newton steps occurred during the middle of a pseudo-transient calculation, i.e. the time step was at some intermediate level in route to an infinite value. The subdomain solvers considered are point ILU(1) [9,10], block ILU(0) (BILU(0)), and LINPACK banded Gaussian elimination [11]. The BILU(0) preconditioner is constructed by lumping all conservation equations at a control volume into individual blocks, performing a complete factorization on these small matrices, and then constructing an ILU(0) preconditioner at the block level. It should be mentioned that while ILU(0) did work as a global preconditioner, its performance was poor when used as a subdomain solver within a Schwarz preconditioning algorithm.

Under the “Partitioning” heading in Table 1, the first entry represents the number of partitions in the x

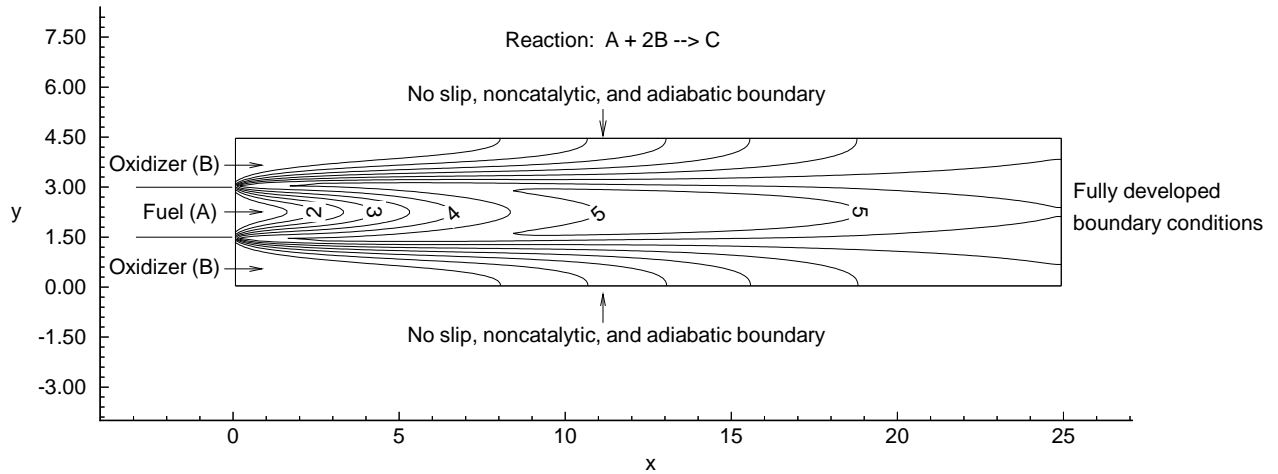


Figure 1: Geometry, boundary conditions and temperature contours for model problem.

direction, while the second entry represents the number of partitions in the y direction. The table shows that the 6×1 partitioning clearly outperforms the 1×6 partitioning in terms of inner (GMRES) iterations. The 6×3 partitioning, in many cases, also outperformed the 1×6 partitioning, even though it has three times as many subdomains. That the 6×1 partitioning performs best for this problem is consistent with previous domain decomposition algorithm research on a similar combustion model problem [12]. An important point is that the optimal blocking strategy for a given problem can be very sensitive to the physics of the problem. Also note from Table 1 that, in general, multiplicative outperforms additive Schwarz, although this advantage is diminished when using an approximate subdomain solver. It is also clear that banded Gaussian elimination outperforms the approximate solvers in terms of inner iteration counts. However, it requires the most memory, and it can also result in higher CPU times as is shown below. (It reduces inner iteration counts but not sufficiently to overcome the high CPU cost of forming the preconditioner).

Table 1: Average GMRES iterations per Newton iteration 90×30 grid, (global ILU(0)=19, global ILU(1)=9, global BILU(0)=12).

Partitioning	Additive			Multiplicative		
	ILU(1)	BILU(0)	Banded G.E.	ILU(1)	BILU(0)	Banded G.E.
6×1	13.9	16.1	11.1	10.7	12.1	4.7
1×6	21.2	31.6	21.2	9.7	16.5	9.8
6×3	21.8	24.5	17.6	12.8	15.1	7.0

Figure 2 is a convergence plot for the 90×30 grid problem using an interpolated 45×15 grid solution as an initial guess. The structure of the convergence plots reflects the Jacobian and preconditioner being formed every five pseudo-transient Newton iterations, as well as an increase in the time step. Figure 2 compares global ILU(1) and global BILU(0) preconditioners with a multiplicative Schwarz preconditioner (6×1 partitioning) with no overlap using ILU(1) (MS-ILU(1)), BILU(0) (MS-BILU(0)), and banded Gaussian elimination (MS-LINPACK) as subdomain solvers. For this specific solution, the global BILU(0) preconditioner required the least amount of CPU time. However, MS-BILU(0) required only 28 percent more CPU time, while being more amenable to parallelization with an appropriate coloring scheme. Also note that these two preconditioning choices required the least amount of computer memory (7.0 MB). Though the per processor memory requirements of the MS-BILU(0) could be considerably less than that of global BILU(0) preconditioning within a distributed parallel implementation.

As seen in Figure 2, the MS-LINPACK required the most CPU time in spite of the smallest average inner iteration counts. Note that this preconditioning choice requires excessive computer memory (39.8 MB) compared with the other preconditioner options. The global ILU(1) preconditioner performed better than only the MS-LINPACK preconditioner in terms of CPU time and memory requirements. Recall that the use of ILU(1) as a subdomain solver was ineffective for this problem. This failure was a result of a loss of GMRES convergence as the time step was increased past the value used to obtain the data in Table 1.

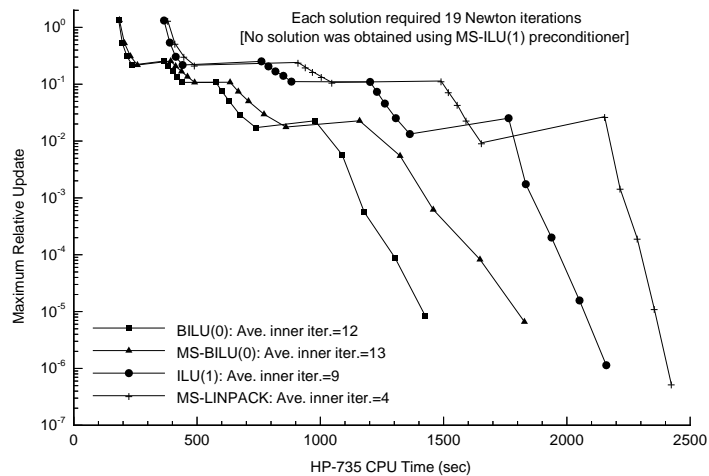


Figure 2: Nonlinear convergence behavior using various preconditioning options.

Acknowledgements

The work of DAK and PRM was supported by the U.S. Department of Energy under DOE Idaho Field Office Contract No. DE-AC07-94ID13223. The work of DEK was supported by the National Science Foundation under contract number ECS-8957475 and by the National Aeronautics and Space Administration under NASA contract NAS1-19480 while in residence at the Institute for Computer Applications in Science and Engineering. The authors wish to thank Youcef Saad for granting permission to use the GMRES algorithm from the SPARSKIT package.

5. References

- 1 DRYJA, M., WIDLUND, O.B. : Domain Decomposition algorithms with small overlap. SIAM J. Sci. Stat. Comput. **15** (1994), 604-620.
- 2 SAAD, Y., SCHULTZ, M.H. : GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7** (1986), 856-869.
- 3 DEMBO, R.S., EISENSTAT, S.C., STEIHAUG, T.: Inexact Newton methods. SIAM J. Numer. Anal. **19** (1982), 400-408.
- 4 BROWN, P.N., HINDMARSH, A.C. : Matrix-free methods for stiff systems of ODE's. SIAM J. Numer. Anal. **23** (1986), 610-638.
- 5 KNOLL, D.A., MCHUGH, P.R.: A fully implicit direct Newton's method for the steady-state Navier-Stokes equations. Int. J. Num. Meth. Fluids. **17** (1993), 449-461.
- 6 CAI, X.C., SAAD, Y.: Overlapping domain decomposition algorithms for general sparse matrices. Tech. Rep. Preprint 93-027, Army High Performance Computing Research Center, University of Minnesota, 1993.
- 7 WILLIAMS, F.A. : Combustion Theory. Addison-Wesley Publishing Company, Inc., New York (1988).
- 8 KNOLL, D.A., MCHUGH, P.R., KEYES, D.E. : Newton-Krylov methods for low Mach number combustion. AIAA paper 95-1672, (submitted to AIAA J.).
- 9 MEIJERINK, J.A., VAN DER VORST, H.A. : An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Mathematics of Computation. **31** (1977), 148-162.
- 10 WATTS, J.W. : A conjugate gradient-truncated direct method for the iterative solution of the reservoir simulation pressure equation. Soc. Pet. Eng. J. **21** (1981), 345-353.
- 11 DONGARRA, J.B., BUNCH, J., MOLER, C., STEWART, G. :LINPACK User's Guide. SIAM, Philadelphia, PA,(1979).
- 12 KEYES, D.E. :Domain decomposition methods for the parallel computation of reacting flows. Comput. Phys. Comm. **53** (1989), 181-200.

Addresses: D.A. Knoll, Idaho National Engineering Laboratory, P.O. Box 1625, Idaho Falls, ID USA 83415-3808, email: nol@inel.gov
P.R. McHugh, Idaho National Engineering Laboratory, P.O. Box 1625, Idaho Falls, ID USA 83415-3808, email: ugh@inel.gov
D.E. Keyes, ICASE NASA-LaRC, Hampton, VA USA 23681-0001, and Computer Science Dept. Old Dominion University, Norfolk, VA USA 23529-0162, email: keyes@icase.edu