Contemporary Mathematics
Volume 157, 1994

Implicit Domain Decomposition Algorithms for Steady, Compressible Aerodynamics

WILLIAM D. GROPP, DAVID E. KEYES, AND JON S. MOUNTS

ABSTRACT. To render tractable the linear systems to be solved at each time step, implicitly discretized compressible aerodynamics codes conventionally employ an approximate factorization that introduces an error of first-order in Δt. For steady-state problems, in which it would otherwise be desirable to allow Δt to approach infinity, this factorization error may impose a more stringent limitation on Δt (or the Courant number) than arises from any other physical or numerical consideration. Furthermore, conventional factorizations yield sparse block triangular or block tridiagonal systems not ideally suited for large-scale parallelism. Despite the dual requirements of high-order spatial accuracy and high Reynolds numbers in aerodynamic applications, "delta" forms of the governing equations with low-order upwind implicit parts can be effectively preconditioned in a domain-parallel fashion and accelerated by using methods of conjugate gradient type. We report comparisons of a variety of global and domain decomposition-based preconditioned iterative methods on Jacobians drawn from a two-dimensional compressible Euler problem.

1. Introduction

Computations of steady transonic flow are often approached through a process of semi-implicit pseudo-transient continuation: a time derivative is appended to the governing Euler or Navier-Stokes equations, and the discretized time step is chosen according to several interrelated criteria, including

- the ability of the algorithm to follow stably a nonlinear solution trajectory from a conveniently specifiable initial condition to steady state,

1991 *Mathematics Subject Classification.* Primary 65N55, 65N22, 76H05.

The work was supported in part by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38 (WDG), by the National Science Foundation under contract number ECS-8957475 and the State of Connecticut (DEK), and by the United Technologies Research Center (JSM).

This paper is in final form and no version of it will be submitted elsewhere.

- the stability of the algorithm with respect to lagged boundary conditions and any terms handled explicitly,
- the rate of convergence and overall cost of the iterative linear solver employed to invert the implicit part of the operator at each step, and
- the accuracy of the "converged" update.

The size of the time step is often chosen locally in terms of a local Courant number, $C = a\Delta t/\Delta x$, where Δx is a local representative cell dimension and a a local signal propagation speed. Grid Courant numbers of order 10 are typical in "production run" computations.

In this paper, we examine the degree to which sensitivity to the third and fourth criteria above can be weakened or removed by replacement of a conventional directionally split approximate factorization (AF) linear solver with a multidomain Krylov solver. The ultimate practical value of such a replacement depends on whether these criteria are the operative bottlenecks for Δt, which will be problem-dependent, in general. The utility of powerful Krylov iterative methods relative to conventional relaxation solvers in nonlinear elliptic systems arising in subsonic reacting flows has recently been considered in [**5**], where it was found that they can be significantly more efficient (e.g., threefold) in overall execution time. In the context of transonic aerodynamics, a twofold improvement in overall execution time was reported in [**21**] when ADI was replaced by a conjugate gradient-square method employing ADI as a preconditioner. This improvement was obtained by raising the local Courant number from a maximum tolerable value of approximately 13, when ADI was used alone, to about 40.

A major ultimate motivation of the present research, apart from allowing the nonlinear solver to employ large time steps, is parallelism. Approximate factorization solvers are notoriously sequential. A large variety of domain-decomposed Krylov algorithms have been explored in [**2, 3, 7, 8, 12**]. These algorithms, many of which are multigrid-like, have been developed for elliptic problems and endowed with a rich convergence theory that includes convection-diffusion problems in the diffusively dominated limit. In this paper, we explore the performance of a generic domain decomposition algorithm, additive Schwarz (without a coarse grid), in a convectively dominated limit.

A brief description of the physical background and mathematical structure of primitive variable transonic computational aerodynamics problems is provided in Section 2. This is followed in Section 3 by a description of the domain-decomposed preconditioned Krylov methods we employ on the discrete linear systems. Section 4 includes the specifications of a classical computational aerodynamics test problem on which the methods of Section 3 are compared. Some prospects are offered in Section 5.

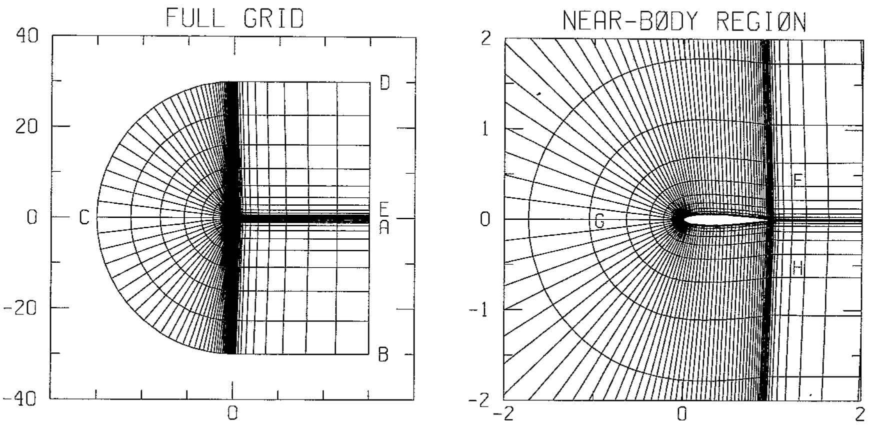

FIGURE 1. Global grid system for NACA0012 tests and enlargement in airfoil region. Labeled points correspond to Fig. 2.

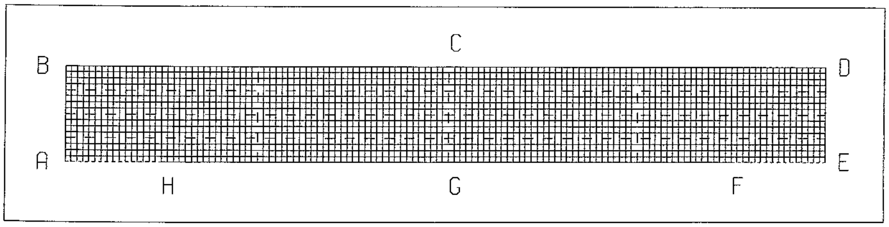

FIGURE 2. Global grid system in transformed coordinates, illustrating a 4 × 4 decomposition. Labeled points correspond to Figure. 1. Dotted wake segments AH and EF coincide.

2. Approximate Factorization Euler Solvers

Our test Jacobians come from a two-dimensional transonic airfoil problem modeled using the EAGLE Flow Solver [**14**], which is more generally capable of three-dimensional, multiblock thin-layer Navier-Stokes analysis. EAGLE employs a finite volume discretization over a body-fitted coordinate grid. For brevity, the only grid considered herein is the C-grid of 128 × 16 cells shown in Figure. 1. The dimensions were chosen to allow balanced repeated bisection into subdomains in both coordinate directions, not by truncation error criteria. However, both the number and placement of the gridpoints are typical of Euler meshes for this problem. (For Navier-Stokes simulations, much denser resolution would be required normal to the airfoil in its vicinity.) Figure. 2 shows the logical tensor-product character of the grid, and the apportionment of grid cells between the wing surface itself and the wake region.

The Euler equations for dependent variable vector $Q \equiv [\rho, \rho u, \rho v, e]^T$ are expressed in strong conservation curvilinear coordinate form as

$$\tilde{Q}_\tau + (\tilde{F})_\xi + (\tilde{G})_\eta = 0, \tag{1}$$

where $\tilde{Q}$ and the contravariant flux vectors, $\tilde{F}$ and $\tilde{G}$, are defined in terms of the Cartesian fluxes and the Jacobian determinant of the coordinate system transformation, $J = x_\xi y_\eta - y_\xi x_\eta$, through

$$\begin{aligned} \tilde{Q} &= J^{-1}Q \\ \tilde{F} &= J^{-1}\left(\xi_t Q + \xi_x F + \xi_y G\right) \\ \tilde{G} &= J^{-1}\left(\eta_t Q + \eta_x F + \eta_y G\right). \end{aligned}$$

The flux terms are split into two parts: an implicit part chosen for convenience in the inversion of the left-hand side operator, and an explicit remainder, denoted R below. For this implementation, the treatment of the explicit and implicit operators is patterned after [**13**]. The flux-vector split scheme of Steger and Warming [**17**] is employed for the implicit operators, and for the explicit operators we use the modified flux-difference split scheme of Roe [**15**]. Characteristic variable boundary conditions are employed at farfield boundaries using an explicit, first-order accurate formulation.

The two-dimensional Euler equations, being a system of four hyperbolic partial differential equations, have associated with them at each point in the domain four characteristic velocities. Numerical schemes for such systems are designed to ensure that information is propagated in the local characteristic directions. The characteristic velocities of this system are determined from the quasilinear form of the implicit part of the operator in Eq. (1),

$$\tilde{Q}_\tau + A\tilde{Q}_\xi + B\tilde{Q}_\eta,$$

where the flux Jacobian matrices A and B are given by

$$A = \frac{\partial \tilde{F}^{impl}}{\partial \tilde{Q}} \quad \text{and} \quad B = \frac{\partial \tilde{G}^{impl}}{\partial \tilde{Q}},$$

and the components of the characteristic velocities in the ξ and η directions are the eigenvalues of A and B, respectively.

Discretization then yields the form

$$\left[I + \Delta\tau(\delta_\xi A_\bullet^+ + \delta_\eta B_\bullet^+ + \delta_\xi A_\bullet^- + \delta_\eta B_\bullet^-)\right]\Delta Q^n = -\Delta\tau R^n, \tag{2}$$

where δ is the first-order spatial difference operator, superscripts ± denote the characteristic (upwind) direction in which the differencing occurs, and the bullets signify that each spatial differencing is carried out on the entire product to the right, for example, δ_ξ on $(A^+\Delta Q^n)$. Note that, for a natural ordering of the unknowns, A^+ and B^+ are sparse block lower triangular and A^- and B^- are upper. The standard solver employs a two-pass lower-upper approximate factorization:

$$\begin{aligned} \left[I + \Delta\tau(\delta_\xi A_\bullet^+ + \delta_\eta B_\bullet^+)\right] X &= -\Delta\tau R^n, \\ \left[I + \Delta\tau(\delta_\xi A_\bullet^- + \delta_\eta B_\bullet^-)\right] \Delta Q^n &= X. \end{aligned}$$

Other approximate factorizations of the left-hand operator of Eq. (2) sometimes employed include block tridiagonals:

$$\left[I + \Delta\tau(\delta_\xi A_\bullet^+ + \delta_\xi A_\bullet^-)\right]\left[I + \Delta\tau(\delta_\eta B_\bullet^+ + \delta_\eta B_\bullet^-)\right],$$

and block bidiagonals:

$$\left[I + \Delta\tau(\delta_\xi A_\bullet^+)\right]\left[I + \Delta\tau(\delta_\eta B_\bullet^+)\right]\left[I + \Delta\tau(\delta_\xi A_\bullet^-)\right]\left[I + \Delta\tau(\delta_\eta B_\bullet^-)\right].$$

3. Domain Decomposition Krylov Algorithms

The domain decomposition methods of primary interest here are preconditioned Krylov methods for solving an $n \times n$ linear system, $Au = f$, that arises from discretization of a differential equation. Krylov methods find the best approximation of the solution u in a subspace $\{v_1, v_2, ..., v_m\}$, with $m \leq n$ that is built up from successive powers of the matrix A on the initial residual, $r_0 = f - Au_0$. For Krylov methods to be effective, we must have $m \ll n$, which is accomplished through preconditioning. For instance, instead of the original problem, we may solve $M\bar{u} = f$ by a Krylov method, where $M = AB^{-1}$ is closer to the identity than A, and then $u = B^{-1}\bar{u}$. The matrix B (or its inverse) is called a preconditioner, and a variety of parallel preconditioners can be induced by decomposing the domain of the underlying PDE, finding the representation of A on each subdomain, inverting each piece locally, and combining the results.

In effect, we seek to approximate the inverse of A by a sum of local inverses:

$$B^{-1} = \sum_k P_k A_k^{-1} R_k, \tag{3}$$

where, R_k is a restriction operator and P_k a prolongation operator that respectively take vectors spanning the entire space into and out of the smaller dimensional subspace in which A_k is defined. To understand the potential effectiveness of B^{-1} so defined, recall that if A possesses a complete orthonormal set of eigenvectors v_k with corresponding nonzero eigenvalues λ_k, and we take $R_k = v_k^T$, $A_k^{-1} = \lambda_k^{-1}$, and $P_k = v_k$, then Eq. 3 gives an exact inverse to A. The key feature of Eq. 3 for parallelism is that each term may be computed independently of the rest. For arbitrary operators A it is difficult to find orthogonal subspaces whose union is the complete solution space, but the subspaces induced by domain decomposition provide a practical parallelizable compromise.

The simplest domain decomposition preconditioner is block Jacobi. Each grid point in the domain of the PDE is associated with a single subdomain. For a subdomain containing n_k unknowns, R_k is an $n_k \times n$ matrix of all zeros, except for a 1 in each row, unique in its column. A_k is the corresponding $n_k \times n_k$ diagonal block of A, the discretization of the homogeneous Dirichlet problem on the k^{th} subdomain. P_k is the transpose of R_k.

The convergence rate of block Jacobi can be improved, at the price of a higher cost per iteration, with subdomain overlap. Though each grid point retains an association with a unique subdomain that will set its value, the data at border grid points is exchanged with neighbors. The n_k of the preceding paragraph is replaced with a larger n'_k. In this paper we consider only subdomain overlaps of one mesh width, a limiting case of the additive Schwarz method [4] without a coarse grid.

The convergence rate of additive Schwarz can be improved, at the price of a lower parallel granularity, by enforcing a multicolored sequentiality between the subdomain solves, just as Gauss-Seidel improves on Jacobi. See [**1**] for algorithmic details. On a sequential computer, the most natural form of Schwarz iteration is multiplicative, with as many colors as subdomains.

A large variety of domain decomposition algorithms exist, many of which that share the preconditioned Krylov framework are surveyed in [**11**]. In the preliminary studies reported here we have yet to incorporate two enhancements found valuable in related investigations. In [**10**], in the context of multicomponent reacting flow, we replaced the A_k^{-1} with less expensive block-ILU approximations, $\tilde{A}_k^{-1}$. In [**1**], in numerical experiments on scalar convection-diffusion problems with large numbers of subdomains, we employed a coarse grid in the preconditioner, adding a term $R_0^T A_k^{-1} R_0$ to the sum in Eq. 3 to render the convergence rate asymptotically independent of the number of grid points and subdomains [**4**].

4. Numerical Results

To obtain a representative Jacobian/residual pair on which to test the methods of the preceding section, we ran a demonstration case from the EAGLE code manual, a NACA0012 airfoil at an angle of attack of 1.25 degrees and a freestream Mach number of 0.8, for 1,000 time steps at the recommended local Courant number of 15, and used the resulting semi-converged result as the Q about which to linearize. The 2-norm of the steady-state nonlinear residual had by this time decreased by a factor of just under 50 from its initial value, and the residual curve had yet to attain terminal monotonicity. However, in the grid of 2,048 cells, 127 were supersonic, and this indicator had stabilized for over 400 time steps. Plots of C_p over the upper and lower surfaces of the wing were qualitatively close to accepted results. At this stage, it would be useful to begin taking larger time steps en route to a full Newton method on the steady-state equations. However, as shown in Table 1, the AF linear solver was delivering less than a factor of two residual reduction at a Courant number of 10, and at Courant numbers of 100 or 1000 it was not improving at all on the initial iterate of zero.

In the unstructured grid literature, it is established that preconditioned Krylov methods can exploit high Courant number to outperform stationary iterative methods. For example, in [**20**], an Euler calculation of flow over a NACA0012 airfoil at identical conditions was solved in approximately one-tenth the num-

TABLE 1. Residual reductions (from an initial iterate of zero for ΔQ^n) produced by one step of the AF method in solving Eq. (3) at five different Courant numbers.

Local Courant Number	10^{-1}	1	10^1	10^2	10^3
Initial Residual	7.30(-5)	7.30(-4)	7.30(-3)	7.30(-2)	7.30(-1)
Final Residual	4.56(-7)	8.05(-5)	4.49(-3)	9.14(-2)	2.37(0)
AF Reduction	0.0062	0.11	0.61	1.25	3.25

ber of time-steps and one-quarter the execution time when SSOR was replaced by ILU/GMRES. The former allowed Courant numbers of up to 25, while the latter permitted 10^6. In Navier-Stokes turbulent flow over an RAE2822 airfoil, the same authors had no practical convergence for SSOR, but were able to use Courant numbers of up to 25,000 with ILU/GMRES. For this pilot study, we considered Courant numbers of 10^{-1} up to 10^3. Results for $C = 1$ and 100 with a relative residual tolerance of 10^{-5} are tabulated in Table 2 for iteration count and Table 3 for execution time on a workstation with a SPARC chip.

Tests were run using a package of sparse linear solver routines developed at Argonne National Laboratory by Gropp and Smith [9] and designed for portability to distributed-memory parallel computers. The package is, in fact, being ported to a variety of parallel architectures, but our use of it in the tables below is restricted to a workstation. We tested GMRES [16], BiCGSTAB [18], and TFQMR [6] as alternative Krylov accelerators. Differences between these three methods in execution time were small, however, so we present only the GMRES results. The maximum dimension of the Krylov vector space in GMRES was set at the narrow dimension of the grid. Therefore, for the problem under consideration, all iteration counts above 16 represent restarted GMRES (see [16]). To produce the action of A_k^{-1} on each subdomain we used direct elimination with a nested dissection ordering. Thus, the 1×1 rows in Tables 2 and 3 correspond to direct nested dissection solutions on the entire domain, with a small amount of extra overhead in setting up unnecessary iteration. Nested dissection is superior to natural ordering, and much less sensitive than natural ordering to the large aspect ratios present in some of the stripwise decompositions.

Table 2 illustrates the generally superior convergence rates of additive and multiplicative Schwarz (with an overlap thickness of one grid cell) over commonly employed block Jacobi, particularly in the case of many subdomains. Stripwise decompositions in both directions and boxwise decompositions are tested. An interesting anomaly occurs in the boxwise decompositions for additive Schwarz on the $C = 1$ problem, where block Jacobi converges in fewer iterations than its overlapped relative.

The execution times of Table 3 include both the significant preprocessing time of the direct factorizations on each subdomain and the cost of iterating. In applications, the preprocessing time would ordinarily be amortized over several

TABLE 2. Iteration counts for transonic flow Jacobians at local Courant numbers of 1 and 10^2, for various preconditioner/decomposition pairs.

Precond.	Block Jacobi		Add. Schwarz		Mult. Schwarz	
C	1	10^2	1	10^2	1	10^2
1×1	1	1	1	1	1	1
1×2	5	13	3	7	2	4
1×4	5	19	4	11	3	7
1×8	7	31	6	17	3	8
1×16	11	56	6	21	4	10
2×1	5	20	4	11	2	6
4×1	5	20	4	12	2	7
8×1	5	22	4	15	2	9
16×1	6	31	4	23	2	17
32×1	6	60	5	45	4	37
64×1	8	109	6	56	4	40
128×1	13	207	7	80	4	50
2×2	4	14	7	14	2	7
4×4	4	18	7	17	3	8
8×8	5	28	10	23	3	8

time steps during which the linearized operator was frozen. Simple block Jacobi is frequently the best method, in spite of its greater number of iterations. We emphasize, however, that there are some inefficiencies in the handling of overlap in current version of the code that will eventually be removed. We have italicized any multidomain entry that costs more execution time than the corresponding monodomain problem at the head of its column. The multidomain methods are often less expensive than nested dissection on the full domain because the cost of the direct solver grows superlinearly in the discrete subdomain size and iterations beyond the first are inexpensive and (sometimes) few. Furthermore, the multidomain methods have built-in parallelism in the preconditioner. As an example, the 8×8 block Jacobi method appears attractive on the $C = 10^2$ problem. With its 28 iterations in 21.5s, it requires virtually the same time as the direct method, but its potential speedup is as high as 64. Communication in the inner products of GMRES and in subdomain border updating will prevent realization of this full factor, of course, but it does perform without the need for any global coarse-grid solution, which is the usual parallel bottleneck for domain decomposition and multigrid methods.

We point out that ILU preconditioners based on the global grid are uniformly more effective than block Jacobi or additive Schwarz methods relying on exact subdomain solves in these problems. For instance, ILU-GMRES on the $C = 10^2$ problem requires 21 iterations and only 9.4s with a residual reduction of 10^{-5}, compared with 28 iterations and 21.5s for 8×8-block Jacobi-GMRES. However, ILU is more difficult to parallelize, and further comparisons involve machine-specific issues. Based on the encouraging monodomain performance of ILU, we

TABLE 3. Execution times (in seconds) for transonic flow Jacobians at local Courant numbers of 1 and 10^2, for various preconditioner/decomposition pairs.

Precond.	Block Jacobi		Add. Schwarz		Mult. Schwarz	
C	1	10^2	1	10^2	1	10^2
1×1	21.2	20.9	23.0	22.8	23.0	22.8
1×2	14.2	19.6	17.0	19.8	17.3	18.7
1×4	9.0	17.2	15.1	20.6	15.6	19.1
1×8	6.9	18.6	17.5	*27.5*	16.7	21.3
1×16	6.5	*23.8*	22.1	*38.8*	22.2	*30.4*
2×1	20.6	*32.1*	21.2	*26.5*	20.0	*23.5*
4×1	17.4	*28.3*	18.1	*24.0*	17.3	21.3
8×1	13.9	*25.5*	15.7	*23.6*	14.9	20.7
16×1	9.8	*24.4*	13.5	*27.3*	13.2	*26.1*
32×1	7.3	*34.2*	12.6	*41.1*	13.2	*41.9*
64×1	6.0	*48.7*	14.2	*53.8*	14.3	*50.5*
128×1	6.5	*76.0*	18.6	*88.8*	18.1	*76.3*
2×2	13.6	20.6	20.5	*26.2*	17.8	21.9
4×4	9.5	18.4	19.1	*28.2*	17.1	22.1
8×8	8.3	*21.5*	24.4	*38.1*	19.6	*25.5*

recommend examining a hybrid method in which ILU is employed in place of nested dissection on the subdomains.

5. Outlook

The twin advents of parallelism and parallelizable implicit algebraic methods provide an incentive for computational aerodynamicists to seek alternatives to the communication-bottlenecked directional AF methods that lie at the heart of many multidimensional production codes. Domain splitting is more natural than general operator splitting on parallel computers, and has the added advantage of leading to more robust linear solvers that do not retard the nonlinear convergence with an error of first-order-in-Δt.

In this paper we have demonstrated the applicability of Krylov-accelerated domain decomposition algorithms to a classic two-dimensional Euler test problem. Relaxation of Courant number limitations necessary to exploit their potential can be obtained by moving the remainder of the discrete operator from the right-hand side to the left-hand side in a full Newton method approach. The desirability and feasibility of Newton-based approaches have been declared in the recent past [**19, 22**], but held up by the cost and sequentiality of the sparse linear system solution. Domain decomposition methods have a role to play in the lowering this barrier.

Our preliminary domain decomposition tests do not include a coarse-grid component in the preconditioner, nor relaxation of the exact subdomain matrix inversions to less expensive approximations. We anticipate improved effectiveness when these algorithmic options are incorporated.

References

1. X.- C. Cai, W. D. Gropp, and D. E. Keyes, *A Comparison of Some Domain Decomposition and ILU Preconditioned Iterative Methods for Nonsymmetric Elliptic Problems*, J. Numer. Lin. Alg. Applics., to appear, 1993.
2. T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., *Proc. of the Third Intl. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1990.
3. T. F. Chan, G. Meurant, J. Périaux, and O. B. Widlund, eds., *Proc. of the Second Intl. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1989.
4. M. Dryja and O. B. Widlund, *An Additive Variant of the Alternating Method for the Case of Many Subregions*, Courant Institute, NYU, TR 339, 1987.
5. A. Ern, V. Giovangigli, D. E. Keyes and M. D. Smooke, *Towards Polyalgorithmic Linear System Solvers for Nonlinear Elliptic Problems*, SIAM J. Sci. Stat. Comp., to appear, 1993.
6. R. W. Freund, G. H. Golub and N. M. Nachtigal, *Iterative Solution of Linear Systems*, Acta Numerica, pp. 57-100, Cambridge Univ. Press, 1991.
7. R. Glowinski, G. H. Golub, G. Meurant and J. Périaux, eds., *Proc. of the First Intl. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1988.
8. R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., *Proc. of the Fourth Intl. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1991.
9. W. D. Gropp and B. F. Smith, *Simplified Linear Equations Solvers Users Manual*, ANL-93/8, Argonne National Laboratory, 1993.
10. D. E. Keyes, *Domain Decomposition Methods for the Parallel Computation of Reacting Flows*, Comp. Phys. Comm. **53**:181–200, 1989.
11. D. E. Keyes, *Domain Decomposition: A Bridge between Nature and Parallel Computers*, in *Adaptive, Multilevel and Hierarchical Computational Strategies*, A. K. Noor, ed., ASME, New York, pp. 293–334, 1992.
12. D. E. Keyes, T. F. Chan, G. Meurant, J. S. Scroggs, and R. G. Voigt, eds., *Proc. of the Fifth Intl. Symp. on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1992.
13. M.-S. Liou and B. van Leer, *Choice of Implicit and Explicit Operators for the Upwind Differencing Method*, AIAA-88-0624, Reno, Nevada, January 1988.
14. J. S. Mounts, D. M. Belk and D. L. Whitfield, *Program EAGLE User's Manual, Vol. IV – Multiblock Implicit, Steady-state Euler Code*, Air Force Armament Laboratory TR-88-117, Vol. IV, September 1988.
15. P. L. Roe, *Approximate Riemann Solvers, Parameter Vector, and Difference Schemes*, J. Comp. Phys. **43**:357–372, 1981.
16. Y. Saad and M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comp. **7**:865–869, 1986.
17. J. L. Steger and R. F. Warming, *Flux Vector Splitting of the Inviscid Gasdynamics Equations with Applications to Finite-Difference Methods*, J. Comp. Phys. **40**:263–293, 1981.
18. H. A. Van der Vorst, *Bi-CGSTAB: A More Smoothly Converging Variant of CG-S for the Solution of Nonsymmetric Linear Systems*, SIAM J. Sci. Stat. Comp., **13**:631–644, 1992.
19. V. Venkatakrishnan, *Newton Solution of Inviscid and Viscous Problems*, AIAA-88-0413, Reno, Nevada, January 1988.
20. V. Venkatakrishnan and D. J. Mavriplis, *Implicit Solvers for Unstructured Meshes*, ICASE TR 91-40, May 1991.
21. M. Vitaletti, *Solver for Unfactored Implicit Schemes*, AIAA J. **29**:1003–1005, 1991.
22. D. Whitfield and L. Taylor, *Discretized Newton-Relaxation Solution of High Resolution Flux-Difference Split Schemes*, AIAA-91-1539, Honolulu, Hawaii, June 1991.

Argonne National Laboratory, 9700 South Cass Ave., Argonne, IL 60439-4844.
E-mail address: gropp@mcs.anl.gov

Department of Mechanical Engineering, Yale University, New Haven, CT 06520-2159
E-mail address: keyes@cs.yale.edu

Department of Mechanical Engineering, Yale University, New Haven, CT 06520-2159 and United Technologies Research Center, East Hartford, CT 06108 *(deceased).*