[3]Bohl, D. G., and Foss, J. F., "Enhancement of Passive Mixing Tabs by the Addition of Secondary Tabs," AIAA Paper 96-0545, Jan. 1996.

[4]Yu, S. C. M., Yip, T. H., and Liu, C. Y., "The Mixing Characteristics of Forced Mixers with Scalloped Lobes," *Journal of Propulsion and Power*, Vol. 13, No. 2, 1997, pp. 305–311.

[5]Zaman, K. B. M. Q., Reeder, M. F., and Samimy, M., "Control of an Axisymmetric Jet Using Vortex Generators," *Physics of Fluids*, Vol. 6, No. 2, 1994, pp. 778–793.

[6]Hou, Y. X., "Velocity Measurements Downstream of a Square Jet with Primary and Secondary Tabs," Postgraduate Research Progress Rept., Thermal and Fluids Engineering Div., School of Mechanical and Production Engineering, Nanyang Technological Univ., Singapore, Sept. 1996.

R. W. Wlezien
Associate Editor

Application of Newton–Krylov–Schwarz Algorithm to Low-Mach-Number Compressible Combustion

P. R. McHugh*
Semitool, Inc., Kalispell, Montana 59901
D. A. Knoll†
Los Alamos National Laboratory, Los Alamos, New Mexico 87545
and
D. E. Keyes‡
Old Dominion University, Norfolk, Virginia 23529-0162

Introduction

THIS Note investigates the efficacy of Newton–Krylov–Schwarz algorithms in solving low-Mach-number compressible flow problems with combustion. Two low-Mach-number chemically reacting flow model problems are considered. The first problem is low-speed combusting flow through a channel, representing an idealized laminar diffusion flame. The second problem considers an expansion channel, representing an idealized combustion chamber, and introduces low-Mach-number recirculation zones. The results presented here build on recent work,[1] which has demonstrated the effectiveness of Newton–Krylov methods for solving low-Mach-number combustion problems. Additionally, this Note is a condensed version of Ref. 2. Here we study domain-based additive Schwarz preconditioners,[3] which may offer certain advantages over more conventional incomplete lower–upper (ILU) factorization type preconditioning.[4] This study draws attention to an important issue that affects the performance of these domain-based preconditioners, namely, subdomain solvers. We also demonstrate the ability to generate effective preconditioners with a low-order spatial discretization.

Presented as Paper 96-0911 at the AIAA 34th Aerospace Sciences Meeting, Reno, NV, Jan. 15–18, 1996; received March 24, 1997; revision received Sept. 5, 1997; accepted for publication Oct. 24, 1997.

*Staff Engineer, Thermal Products Division. Member AIAA.

†Technical Staff Member, Applied Theoretical and Computational Physics Division. Member AIAA.

‡Professor, Department of Computer Science; also Senior Research Associate, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23681-0001. Member AIAA.

Newton–Krylov–Schwarz Method

The pseudotransient Newton–Krylov algorithm used here follows the implementation outlined in previous work.[1] Specifically, the Jacobian is evaluated only periodically to generate a new preconditioner, the cost of which is then amortized over many Newton steps. However, and this is a very important point, the effect of the current Jacobian is captured within the Krylov iteration by approximating Jacobian-vector products with finite difference projections.[5] The performance of the Krylov algorithm in solving these linear systems is improved via matrix preconditioning. With right preconditioning, the linear systems that are iteratively solved on each Newton step are of the form

$$[(V/\Delta t^n + J^n)P^{-1}][P\delta x^n] = -F(x^n) \tag{1}$$

where $F(x^n)$ is the residual vector at the nth Newton iteration representing the system of discrete algebraic equations, J^n is the numerically evaluated Jacobian matrix, P is the preconditioning matrix, V is a diagonal matrix with entries equal to cell volumes, and Δt^n is the pseudotime step. Also, δx^n is the vector update used to obtain the new solution approximation according to $x^{n+1} = x^n + \alpha\delta x^n$. The damping coefficient α is described in Ref. 1.

The discrete system represented by $F(x^n)$ is obtained from a finite volume discretization of the governing equations using a staggered, Cartesian mesh. Second-order central differencing is used for derivative terms other than convection. Convection is treated using Leonard's quadratic upstream interpolation for convective kinematics (QUICK) scheme[6] to compute convected quantities at cell faces. To preserve monotonicity, the QUICK scheme is flux limited. The flux limiting strategy selected here was first proposed by Gaskel and Lau[7] in describing their SMART scheme. Convergence of the outer Newton iteration is based on the maximum relative Newton update

$$\max_{\text{all } m}\left[\frac{|\delta x_m|}{\max(|x_m|, 1)}\right]$$

and the infinity norm of the residual, $\|F(x^n)\|_\infty$, being less than 1×10^{-6}.

Jacobian-vector products within the Krylov algorithm are replaced with finite difference projections.[5] The generalized minimal residual (GMRES) algorithm[8] is selected as the Krylov solver based on previous observations concerning this type of implementation.[9] Thus, terms of the form $JP^{-1}v$ appearing within the GMRES algorithm are computed from

$$JP^{-1}v \approx \frac{F(x + \varepsilon P^{-1}v) - F(x)}{\varepsilon} \tag{2}$$

where v is a general Krylov vector and ε is a small perturbation.

The advantage of using Eq. (2) is that it approximates the action of the Jacobian without explicitly computing the full Jacobian matrix. This implementation allows the number of expensive numerical Jacobian evaluations to be reduced considerably, while maintaining strong Newton-like nonlinear convergence. Explicit formation is needed only periodically to refresh the preconditioner used during the Krylov iteration. Often a less costly approximate Jacobian, such as the one arising from a low-order discretization, may suffice in generating a new preconditioner.[10]

The one-level additive Schwarz preconditioner used in this work is described in detail by Dryja and Widlund.[3] The goal is to construct a global preconditioner that incorporates solutions of p subdomain problems. The operation of the one-level additive Schwarz preconditioner on a general Krylov vector, such as the one appearing in Eq. (2), can be expressed as

$$P^{-1}v = \sum_{i=1}^{p} R_i^T \left\{ \left(P_i^{\text{sub}}\right)^{-1} [R_i v] \right\} \tag{3}$$

where $(P_i^{\text{sub}})^{-1}$ is an approximate inverse to the ith subdomain contribution to the matrix $(V/\Delta t^n + J^n)$. The restriction matrix R_i extracts the portions of the v vector corresponding to the ith

subdomain. The outer action of the restriction matrix transpose serves to map the resulting subdomain vector back to the global domain. This study compares different choices for the subdomain solver, i.e., different methods for computing $(P_i^{\text{sub}})^{-1}$. In this work, both complete and incomplete factorizations of P_i^{sub} are considered. The complete factorizations use LINPACK banded Gaussian elimination. ILU factorizations[4] are constructed from either a pointwise basis, i.e., ILU(k), where k denotes the level of fill in, or an analogous block ILU (BILU) basis, i.e., BILU(k), where the blocks are defined by the variables and equations within a finite volume. Schwarz preconditioners are ideal for the cache-based, distributed-memory multiprocessors dominating the scientific marketplace. Their application possesses concurrency on the granularity of the subdomains by construction, and they induce a blocking on standard preconditioners, such as ILU, that leads to much improved cache locality and, therefore, higher computation rates, even in serial.

Results and Discussion

The nondimensional conservation equations for chemically reacting, compressible, laminar flow with variable transport coefficients expressed in two-dimensional Cartesian coordinates constitute the physical model. The problems considered here contain three species continuity equations (fuel, oxidizer, and product with finite rate chemistry), two momentum equations, and one thermal energy equation, resulting in a system of six coupled equations. The details of the physical model and the geometry of problem 1 are given in Ref. 1, and the geometry of problem 2 is given in Ref. 2. Numerical solutions to the second model problem (idealized combustion chamber) were observed to be considerably more challenging than the first problem, which did not exhibit the recirculating flow regions.

Different Subdomain Partitions and Solvers: Problem 1

This subsection discusses algorithm performance data accumulated from steady-state solutions to the first model problem on a 90×30 uniform grid. The starting point for these calculations was a poor initial guess. The initial pseudotime step Δt^0 for these calculations was set at 0.05, representing an acoustic Courant–Friedrichs–Lewy (CFL) number of approximately 2. The maximum allowed pseudotime step size $\Delta t^{\max}$ was 100, yielding a maximum acoustic CFL number of approximately 3800. Table 1 presents algorithm performance data and preconditioner memory requirements as a function of various subdomain blocking strategies and solvers. The first column indicates the selected blocking strategy (x blocks $\times$ y blocks), whereas the first row indicates the selected subdomain solver, i.e., the manner in which P_i^{sub} is factored. Performance data listed for each subdomain solver include the number of nonlinear Newton steps n, the average number of GMRES iterations per Newton step $\bar{m}$ along with the number of instances where the GMRES iteration reached the upper bound of 50 (number of hits), the Hewlett Packard-735 CPU time, and the computer memory required for storage of the preconditioner. The ILU(0) subdomain solver was generally ineffective for this problem and so its data were omitted from Table 1. The loss of diagonal dominance of the Jacobian in very low-Mach number flows has been observed to favor Schwarz methods with exact subdomain solves[1] over global ILU (too weak) and global exact solves (too expensive). With respect to the stripwise partitions, cuts across the flow channel, e.g., 3×1 and 6×1, typically produced a more effective preconditioner (measured with lower values for $\bar{m}$) than did cuts along the flow channel, e.g., 1×3 and 1×6, an observation consistent with previous work.[11]

Preconditioning with a Low-Order Discretization: Problem 1

This subsection investigates the advantages of using a less expensive, low-order discretization, i.e., first-order upwinding, in the evaluation of the preconditioner. Table 2 presents algorithm performance data using this approach. These data should be compared with those in Table 1, which used the higher-order discretization in the

Table 1 Algorithm performance data and preconditioner memory requirements using additive Schwarz preconditioning with four different subdomain solvers, problem 1

	LINPACK banded Gaussian elimination				ILU(1) (156 nonzero diagonals)				BILU(0) (13 block diagonals)				BILU(1) (21 block diagonals)			
Blocking	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB
1×1	141	5–0	2.73	142.4	142	14–0	1.62	20.2	141	26–0	1.69	10.1	142	14–0	1.25	16.3
1×3	142	13–0	1.50	49.1	141	23–0	2.22	20.2	152	30–22	2.18	10.1	144	23–3	1.85	16.3
3×1	141	10–0	3.28	142.4	162	19–10	2.25	20.2	141	25–9	1.76	10.1	142	17–0	1.49	16.3
1×6	141	24–2	1.80	25.8	141	24–2	1.98	15.7	153	36–43	2.56	10.1	143	23–2	1.77	16.3
6×1	141	14–0	3.59	72.4	166	24–37	2.70	20.2	141	27–12	1.87	10.1	142	20–0	1.61	16.3
9×3	142	24–1	2.17	49.1	—	—	—	20.2	152	33–33	2.35	10.1	142	29–9	2.15	16.3

Table 2 Algorithm performance data and preconditioner memory requirements using a Jacobian from a low-order discretization to generate a preconditioner, problem 1

	LINPACK banded Gaussian elimination				ILU(1) (113 nonzero diagonals)				BILU(0) (9 block diagonals)				BILU(1) (13 block diagonals)			
Blocking	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB
1×1	141	5–0	1.29	72.5	141	13–0	1.23	14.6	142	19–1	1.30	7.0	142	14–0	1.11	10.1
3×1	142	10–0	1.72	72.5	142	19–10	1.63	14.6	148	21–1	1.46	7.0	142	17–0	1.30	10.1
6×1	141	13–0	2.03	37.5	141	20–1	1.71	14.6	142	24–4	1.59	7.0	142	20–0	1.41	10.1
9×3	142	23–0	1.76	25.8	148	29–21	2.36	14.6	141	32–27	1.95	7.0	141	19–13	1.88	10.1

Table 3 Algorithm performance data for 120×70 uniform grid solution for problem 2 at $Re = 60$

		BILU(0)				BILU(1)				BILU(2)			
Blocking	Convection discretization scheme used during preconditioner evaluation	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB	n	$\bar{m}$ number of hits	CPU, h	Memory, MB
1×1	Upwind	41	36–24	2.34	21.8	41	33–18	2.23	31.5	41	22–0	1.79	46.0
	QUICK	44	39–30	2.75	31.5	40	32–11	2.51	50.8	42	23–0	2.51	89.5
6×1	Upwind	42	37–28	2.39	21.8	41	36–24	2.48	31.5	41	32–6	2.35	46.0
	QUICK	54	41–41	3.63	31.5	40	36–24	2.73	50.8	41	35–20	3.33	89.5

preconditioner evaluation. An immediate benefit of this technique is a marked reduction in the preconditioner memory requirements. This is reflected in the reduced number of nonzero diagonals (or block diagonals) required for the same level of fill in. In all cases, the average GMRES iterations diminished or stayed the same, indicating that the overall preconditioner quality was not negatively impacted, whereas CPU efficiency was improved.

Geometry Effects: Problem 2

Performance data for the idealized combustion chamber model problem using a 120×70 uniform grid are presented in Table 3 for $Re = 60$ and $Ma = 0.14$. These calculations were initialized from an interpolated 60×35 grid solution. The pseudotime step control parameters were again set to the values used earlier so that $\mathrm{CFL}^0 \approx 1.4$ and $\mathrm{CFL}^{\max} \approx 2800$. The ILU subdomain solvers were not effective for any of the partition selections. As with the first model problem, stripwise cuts across the flow channel were more effective than cuts along the flow channel. As shown in Table 3, both the low-order and high-order discretizations were used in the preconditioner evaluation for comparison purposes. Using the first-order upwind discretization scheme in the preconditioner evaluation was more effective than using the flux limited QUICK scheme in terms of both computer memory requirements and CPU times. This was the case despite that, at $Re = 60$, the flow reattachment length of the flux limited QUICK solution is approximately 53% longer than that obtained using first-order upwinding.

Summary

This Note presents pseudotransient Newton–Krylov–Schwarz solutions to two idealized low-Mach-number combustion problems. Schwarz methods offer effective and flexible preconditioning options for problems of this type, while also providing an avenue toward parallelism. BILU subdomain solvers generally provided a good compromise between subdomain solver quality and memory requirements. Preconditioning with a low-order discretization proved to be a wise choice in terms of both computer memory requirements and algorithm performance.

Acknowledgments

This work was supported under the auspices of the U.S. Department of Energy under DOE Contract W-7405-ENG-36 at Los Alamos National Laboratory and Contract DE-AC07-94ID13223 at the Idaho National Engineering Laboratory. The work of the third author was supported in part by the National Science Foundation under DMS-9505110 and ECS-9527169 and in part by NASA under NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering. A significant portion of the research was performed while the first two authors were employed by the Idaho National Engineering Laboratory.

References

[1]Knoll, D. A., McHugh, P. R., and Keyes, D. E., "Newton–Krylov Methods for Low-Mach-Number Compressible Combustion," *AIAA Journal*, Vol. 34, No. 5, 1996, pp. 961–967.

[2]McHugh, P. R., Knoll, D. A., and Keyes, D. E., "Schwarz-Preconditioned Newton–Krylov Algorithm for Low Speed Combustion Problems," AIAA Paper 96-0911, Jan. 1996.

[3]Dryja, M., and Widlund, O. B., "Domain Decomposition Algorithms with Small Overlap," *SIAM Journal on Scientific and Statistical Computing*, Vol. 15, No. 3, 1994, pp. 604–620.

[4]Meijerink, J. A., and 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*, Vol. 31, No. 137, 1977, pp. 148–162.

[5]Brown, P. N., and Saad, Y., "Hybrid Krylov Methods for Nonlinear Systems of Equations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 11, No. 3, 1990, pp. 450–481.

[6]Leonard, B. P., "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 19, No. 1, 1979, pp. 59–98.

[7]Gaskell, P. H., and Lau, A. K. C., "Curvature-Compensated Convective Transport: SMART, A New Boundedness-Preserving Transport Algorithm," *International Journal for Numerical Methods in Fluids*, Vol. 8, No. 6, 1988, pp. 617–641.

[8]Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 7, 1986, pp. 856–869.

[9]McHugh, P. R., and Knoll, D. A., "Comparison of Standard and Matrix-Free Implementations of Several Newton–Krylov Solvers," *AIAA Journal*, Vol. 32, No. 12, 1994, pp. 2394–2400.

[10]Keyes, D. E., "Aerodynamic Applications of Newton-Krylov-Schwarz Solvers," *Proceedings of the 14th Conference on Numerical Methods in Fluid Dynamics*, edited by S. M. Deshpande, Springer, Berlin, 1995, pp. 1–20.

[11]Keyes, D. E., "Domain Decomposition Methods for the Parallel Computation of Reacting Flows," *Computer Physics Communication*, Vol. 53, May 1989, pp. 181–200.

K. Kailasanath
Associate Editor

Improved Optimality Criterion Algorithm for Optimal Structural Design

Jianping Chen,* Weiming Ni,† and Wenliang Wang‡
Fudan University,
Shanghai 200433, People's Republic of China

I. Introduction

THE optimal design of structures with frequency constraints is extremely useful in manipulating dynamic characteristics in a variety of ways.[1] References 2 and 3 first pointed out that the singularity of eigenvalue derivatives with respect to the design vector does not allow use of the Kuhn–Tucker condition with multiple frequency constraints. Recently, Czyz and Lukasiewicz[4] put forward the optimality criteria using Lagrange multipliers. The strategy of these algorithms in overcoming the singularity is that the variation (or increment) $\delta \boldsymbol{h}$ of design vector $\boldsymbol{h}$ is confined to a subspace $\Delta \boldsymbol{H}$ in which the multiplicity of the eigenvalue does not change ($\delta \boldsymbol{h} \in \Delta \boldsymbol{H}$). Reference 5 takes damping into account, but once the repeated eigenvalues appear this method will be doomed to failure.

Various optimal structural methods under multiple eigenvalue constraints require knowledge of corresponding modes and sensitivities of the (progressively improved) structure. Hence, after almost every iteration improvement, modes and sensitivity would be reanalyzed. To save computational cost and hasten the optimal design, Refs. 6 and 7 posed the substructural sensitivity synthesis techniques; unfortunately, their formulas are confined to simple eigenvalues.

The purpose of this Note is to develop a highly efficient algorithm for structural optimization under multiple frequency constraints, making use of the generalized variations $\delta \boldsymbol{q}$ of design vectors $\boldsymbol{h}$ to automatically achieve the optimality search of $\delta \boldsymbol{h} \in \Delta \boldsymbol{H}$. This not only ensures that the Kuhn–Tucker condition holds but also excludes additional Lagrange multipliers. We use the improved substructural synthesis suitable for repeated eigenvalue sensitivity analysis of general nondefective vibratory systems[8,9] to quickly draw the required simple or repeated eigenvalue modes and their sensitivities of a complex structure after every iteration.

Received March 25, 1997; revision received Sept. 15, 1997; accepted for publication Sept. 30, 1997.

*Graduate Student, Department of Applied Mechanics.
†Doctoral Candidate, Department of Applied Mechanics.
‡Professor, Department of Applied Mechanics.