

Acceleration of Iterative Algorithms on Highly Clustered Grids

Kwang-Yoon Choi*

Samsung Aerospace Industries, Ltd., Taejeon 305600, Republic of Korea
and

George S. Dulikravich†

Pennsylvania State University, University Park, Pennsylvania 16802

A family of new methods has been developed to accelerate the convergence rate of iterative algorithms for obtaining a steady-state solution as an asymptotic limit of an unsteady second-order partial differential equation or a system of such equations. It was assumed that a central differencing has been used for spatial discretization. The new acceleration methods are based on the sensitivity of the future residual at every grid point to the change in the solution vector components at the neighboring grid points used in the local discretization approximation. The acceleration parameters introduced in the methods have been optimized with the objective to minimize the future global residual. The new sensitivity-based methods have been applied to finite difference codes for two- and three-dimensional, laminar, incompressible flow Navier-Stokes equations; two-dimensional, turbulent, incompressible flow Navier-Stokes equations; and two-dimensional, compressible flow Euler equations. The new sensitivity-based acceleration methods demonstrated superior performance in all test cases that involved severe grid clustering and grid nonorthogonality and included laminar and turbulent flows with closed and open flow separation.

I. Introduction

TO accurately and reliably resolve details of field problems where variables experience very sharp gradients, it is necessary to use locally highly clustered grids. Nevertheless, implicit and explicit iterative algorithms for the integration of systems of nonlinear partial differential equations suffer from a slow convergence rate on such grids. This is because local time steps required for stability are proportional to the local grid spacing. Several attempts have been made to accelerate the convergence of such algorithms using local time stepping,^{1,2} implicit residual smoothing,³ multigriding,⁴⁻⁷ and preconditioning.⁸⁻¹⁰ The multigrid method is effective in reducing low- and high-frequency errors yielding impressive convergence rates. The preconditioning methods are very powerful in alleviating the slow convergence associated with a stiff system for solving the low-Mach-number compressible flow equations. However, these methods have not been shown to perform universally well on highly clustered, nonorthogonal, three-dimensional grids.

Optimum extrapolation techniques represent an entirely different type of convergence acceleration algorithms that belongs to the general class of Krylov subspace methods. One such technique is the minimal residual method (MRM), which uses an equal optimal weight for the corrections to every equation¹¹ in a system of partial differential equations¹² that is solved. The general nonlinear minimal residual (GNLMR) method,^{13,14} allows for an identical sequence of optimized relaxation parameters to be applied to each component of the solution vector in a system that is solved. The distributed minimal residual (DMR) method¹⁵⁻¹⁷ allows each component of the solution vector in a system of equations to have its own sequence of optimized relaxation parameters. The application of the DMR method requires that solutions from three or four consecutive iterations be stored. This is an improvement over the general minimal residual (GMRES) method,¹⁸ which requires storing a large number of solutions. However, none of these methods

have been shown to accelerate the iterative convergence rates for arbitrary systems of partial differential equations on highly clustered, nonorthogonal, three-dimensional grids.

Thus, the objective of the present study is to develop a numerical method that is capable of accelerating the convergence of arbitrary iterative algorithms for the integration of a general system of evolutionary (hyperbolic or parabolic) partial differential equations. Our objective is also to investigate the performance of the newly developed scheme on a variety of fluid dynamic problems with a special emphasis on the acceleration of convergence on highly clustered and nonorthogonal grids. A four-stage Runge-Kutta (RK) time integration method with local time stepping will be used as a basic iterative algorithm, and spatial derivatives will be approximated by a second-order-accurate central differencing¹ throughout the study. Characteristic boundary conditions will be used at the inlet, and nonreflecting boundary conditions will be used at the exit boundary of the flow domain. We will be seeking accurate steady-state solutions only.

II. Global Sensitivity-Based Minimum Residual Method

A system of unsteady partial differential equations can be written as $\mathbf{R} = -\partial \mathbf{Q} / \partial t = \mathbf{L}(\mathbf{Q})$ where \mathbf{Q} is the solution vector, t is the time, \mathbf{L} is the differential operator, and \mathbf{R} is the residual vector. Conventional iterative algorithms update \mathbf{Q} by calculating the corrections, $\Delta \mathbf{Q}$, without optimizing their influence on the future residual.

Our sensitivity-based minimum residual (SBMR) method¹⁹⁻²³ uses the fact that the future residual at a grid point depends upon the changes in \mathbf{Q} at the neighboring grid points used in the local finite difference approximation. The sensitivities are determined by taking partial derivatives of the finite difference approximation of R_r ($r = 1, \dots, r_{\max}$ where r_{\max} is the number of equations in the system) with respect to each component of Q_{ms} ($m = 1, \dots, M$, where M is the number of unknowns; and $s = 1, \dots, S$, where S is the number of surrounding grid points directly involved in the local discretization scheme). This information is then utilized to effectively extrapolate \mathbf{Q} so as to minimize the future residual. Nine grid points (Fig. 1) are used to formulate the global SBMR method for a two-dimensional problem when using central differencing,¹ compared with 19 grid points (Fig. 1) for a three-dimensional case. This approach is different from

Received Sept. 23, 1994; revision received April 26, 1995; accepted for publication May 6, 1995. Copyright © 1995 by Kwang-Yoon Choi and George S. Dulikravich. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Project Leader, Department of Aerodynamics. Member AIAA.

†Associate Professor, Department of Aerospace Engineering, 233 Hammond Bldg. Associate Fellow AIAA.

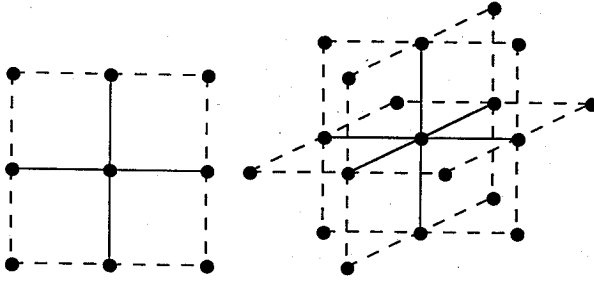


Fig. 1 Grid points involved in two-dimensional and three-dimensional SBMR.

our earlier DMR¹⁵⁻¹⁷ method where the analytical form of R was differentiated.

Suppose that we are performing an iterative solution of an arbitrary evolutionary system using an arbitrary iteration algorithm. Suppose we know the solution vectors Q^t and Q^{t+n} at iteration levels t and $t+n$, respectively. Here n is the number of regular iterations performed by the original nonaccelerated algorithm. Then ΔQ between the two iteration levels is given as $Q^{t+n} = Q^t + \Delta Q$. Using the first two terms of a Taylor series expansion in the artificial time direction, the residual for each of the equations in the system after n iterations is

$$R_r^{t+n} = R_r^t + \sum_m^M \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s} \quad (1)$$

Notice that the total number of equations in the system is the same as the total number of unknown components of Q , that is, $r_{\max} = M$. If we introduce convergence rate acceleration coefficients $\alpha_1, \alpha_2, \dots, \alpha_M$ multiplying corrections $\Delta Q_{1,s}, \Delta Q_{2,s}, \dots, \Delta Q_{M,s}$, respectively, the future solution vector component $Q_{m,s}^{(t+n)+1}$ at the grid point s can be extrapolated as

$$Q_{m,s}^{(t+n)+1} = Q_{m,s}^t + \alpha_m \Delta Q_{m,s} \quad (2)$$

This can be applied at every grid point in the domain, but it results in a huge system having $i_{\max} \times j_{\max} \times k_{\max} \times M$ unknown acceleration coefficients α . If each of the various α is assumed to have the same value over the entire domain D , the number of unknown α is reduced to M . We call this pragmatic approach the global SBMR method.¹⁹⁻²³ It requires solution of a much smaller $M \times M$ matrix. The future residual at the iteration level $(t+n)+1$ can, therefore, be approximated by

$$R_r^{(t+n)+1} = R_r^t + \sum_m^M \left[\alpha_m \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s} \right] \quad (3)$$

Subtracting Eq. (1) from Eq. (3) yields

$$R_r^{(t+n)+1} = R_r^{t+n} + \sum_m^M (\alpha_m - 1) a_{rm} \quad (4)$$

where

$$a_{rm} = \sum_s^S \frac{\partial R_r^t}{\partial Q_{m,s}} \Delta Q_{m,s}$$

The various optimum α are determined such that the sum of the L_2 norm of the future residuals over the entire domain D will be minimized:

$$\sum_D \sum_r^{r_{\max}} \frac{\partial [R_r^{(t+n)+1}]^2}{\partial \alpha_m} = 2 \sum_D \sum_r^{r_{\max}} R_r^{(t+n)+1} \frac{\partial R_r^{(t+n)+1}}{\partial \alpha_m} = 0 \quad (5)$$

for $m = 1, \dots, M$. With the help of Eq. (4), the system (5) becomes

$$\sum_D \left[\sum_r^{r_{\max}} \left\{ R_r^{t+n} + \sum_m^M a_{rm} (\alpha_m - 1) \right\} a_{r1} \right] = 0$$

$$\sum_D \left[\sum_r^{r_{\max}} \left\{ R_r^{t+n} + \sum_m^M a_{rm} (\alpha_m - 1) \right\} a_{r2} \right] = 0 \quad (6)$$

$$\vdots$$

$$\sum_D \left[\sum_r^{r_{\max}} \left\{ R_r^{t+n} + \sum_m^M a_{rm} (\alpha_m - 1) \right\} a_{rM} \right] = 0$$

In Eq. (6), the various R and a are known from the preceding iteration levels. Since each α is assumed to have the same value over the entire computational domain, Eq. (6) gives a tractable system of M simultaneous algebraic equations for optimum $\alpha_1, \alpha_2, \dots, \alpha_M$:

$$\left[\sum_D \left(\sum_r^{r_{\max}} a_{r1} a_{r1} \right) \right] (\alpha_1 - 1) + \left[\sum_D \left(\sum_r^{r_{\max}} a_{r2} a_{r1} \right) \right] \times (\alpha_2 - 1) + \dots + \left[\sum_D \left(\sum_r^{r_{\max}} a_{rM} a_{r1} \right) \right] (\alpha_M - 1) = - \sum_D \left(\sum_r^{r_{\max}} R_r^{t+n} a_{r1} \right)$$

$$\left[\sum_D \left(\sum_r^{r_{\max}} a_{r1} a_{r2} \right) \right] (\alpha_1 - 1) + \left[\sum_D \left(\sum_r^{r_{\max}} a_{r2} a_{r2} \right) \right] \times (\alpha_2 - 1) + \dots + \left[\sum_D \left(\sum_r^{r_{\max}} a_{rM} a_{r2} \right) \right] (\alpha_M - 1) = - \sum_D \left(\sum_r^{r_{\max}} R_r^{t+n} a_{r2} \right) \quad (7)$$

$$\vdots$$

$$\left[\sum_D \left(\sum_r^{r_{\max}} a_{r1} a_{rM} \right) \right] (\alpha_1 - 1) + \left[\sum_D \left(\sum_r^{r_{\max}} a_{r2} a_{rM} \right) \right] \times (\alpha_2 - 1) + \dots + \left[\sum_D \left(\sum_r^{r_{\max}} a_{rM} a_{rM} \right) \right] (\alpha_M - 1) = - \sum_D \left(\sum_r^{r_{\max}} R_r^{t+n} a_{rM} \right)$$

For the general case of a system composed of M partial differential equations with M unknowns, the system (7) will become a full $M \times M$ symmetric matrix for M unknown optimum α . For the rest of this study, the global SBMR method^{19,20} will be called SBMR method to differentiate it from its descendants, such as line SBMR (LSBMR), plane SBMR (PSBMR), or alternating direction plane SBMR (APSBMR) methods, that will be designed for special purposes.

III. Line SBMR Method

It is plausible that for nonuniform computational grids and rapidly varying dependent variables optimum α should not necessarily be the same over the whole computational domain. A modification of the SBMR method called line sensitivity-based minimal residual (LSBMR) method was developed to allow various α to have different values from one grid line to another. The resulting system will have, for example, $j_{\max} \times M$ unknown α , which is quite

tractable, although it is more complex to implement than the SBMR method. The LSBMR formulation²⁰⁻²³ will be explained using the two-dimensional, incompressible flow Navier-Stokes equations as an example. The original RK algorithm uses Chorin's artificial compressibility method²⁴ that adds an artificial pressure-based unsteady term to mass conservation. The system has three equations ($r_{\max} = 3$) and three unknowns ($M = 3$). These are pressure p and the local velocity vector Cartesian components u and v . The acceleration coefficients for those unknowns are α_p , α_u , and α_v , respectively. If the grid lines are clustered in the j direction, then each $j = \text{const}$ grid line will be assigned its own set of constant α . The residual at a grid point (i, j) incorporates various α at the neighboring grid points plus the point itself (Fig. 1). For the given Navier-Stokes system, this yields

$$\begin{aligned} R_r^{(t+n)+1} = & R_r^t + \sum_{s=i-1}^{i+1} \left[\frac{\partial R_r^t}{\partial p_{s,j-1}} \Delta p_{s,j-1} \alpha_p^{j-1} \right. \\ & + \frac{\partial R_r^t}{\partial p_{s,j}} \Delta p_{s,j} \alpha_p^j + \frac{\partial R_r^t}{\partial p_{s,j+1}} \Delta p_{s,j+1} \alpha_p^{j+1} \Big] \\ & + \sum_{s=i-1}^{i+1} \left[\frac{\partial R_r^t}{\partial u_{s,j-1}} \Delta u_{s,j-1} \alpha_u^{j-1} + \frac{\partial R_r^t}{\partial u_{s,j}} \Delta u_{s,j} \alpha_u^j \right. \\ & + \frac{\partial R_r^t}{\partial u_{s,j+1}} \Delta u_{s,j+1} \alpha_u^{j+1} \Big] + \sum_{s=i-1}^{i+1} \left[\frac{\partial R_r^t}{\partial v_{s,j-1}} \Delta v_{s,j-1} \alpha_v^{j-1} \right. \\ & + \frac{\partial R_r^t}{\partial v_{s,j}} \Delta v_{s,j} \alpha_v^j + \frac{\partial R_r^t}{\partial v_{s,j+1}} \Delta v_{s,j+1} \alpha_v^{j+1} \Big] \end{aligned} \quad (8)$$

for $j = 1, \dots, j_{\max}$. For a given $j = \text{const}$ grid line, α^j appears only in $R_m^{(t+n)+1}(i, j-1)$, $R_m^{(t+n)+1}(i, j)$, and $R_m^{(t+n)+1}(i, j+1)$ when using central differencing in the j direction. Therefore, summation over the entire domain leaves only the terms α^{j-2} , α^{j-1} , α^j , α^{j+1} , and α^{j+2} summed along each $j = \text{const}$ grid line. Thus, on each $j = \text{const}$ grid line, the M values of various constant α are determined so as to minimize the $L-2$ norm of the future global residual summed over the entire domain D . The minimization results in $(j_{\max}) \times M$ algebraic equations for the same number of unknown optimum α :

$$\begin{aligned} 2 \sum_{i=1}^{i_{\max}} \sum_{r=1}^3 & \left[R_r^{(t+n)+1}(i, j-1) \frac{\partial R_r^{(t+n)+1}(i, j-1)}{\partial \alpha_m^j} \right] \\ & + 2 \sum_{i=1}^{i_{\max}} \sum_{r=1}^3 \left[R_r^{(t+n)+1}(i, j) \frac{\partial R_r^{(t+n)+1}(i, j)}{\partial \alpha_m^j} \right] \\ & + 2 \sum_{i=1}^{i_{\max}} \sum_{r=1}^3 \left[R_r^{(t+n)+1}(i, j+1) \frac{\partial R_r^{(t+n)+1}(i, j+1)}{\partial \alpha_m^j} \right] = 0 \end{aligned} \quad (9)$$

Similar equations result for $j = 1, \dots, j_{\max}$ and $m = p, u, v$. In this example the simultaneous system of Eq. (9) yields a block pentadiagonal matrix equation for $(j_{\max}) \times 3$ optimum α where each block is a 3×3 matrix. In the general case of a two-dimensional system having M partial differential equations, the block pentadiagonal system (9) will have blocks of size $M \times M$. For example of pressure, when $m = p$, the terms inside the summation sign of Eq. (9) can be written as follows:

$$\begin{aligned} & R_r^{(t+n)+1}(i, j-1) \frac{\partial R_r^{(t+n)+1}(i, j-1)}{\partial \alpha_p^j} \\ = & \left[R_r^t(i, j-1) + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial p_{s,j-2}} \Delta p_{s,j-2} \alpha_p^{j-2} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial p_{s,j-1}} \Delta p_{s,j-1} \alpha_p^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial p_{s,j}} \Delta p_{s,j} \alpha_p^j \right. \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial u_{s,j-2}} \Delta u_{s,j-2} \alpha_u^{j-2} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial u_{s,j-1}} \Delta u_{s,j-1} \alpha_u^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial u_{s,j}} \Delta u_{s,j} \alpha_u^j \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial v_{s,j-2}} \Delta v_{s,j-2} \alpha_v^{j-2} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial v_{s,j-1}} \Delta v_{s,j-1} \alpha_v^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial v_{s,j}} \Delta v_{s,j} \alpha_v^j \Big] \\ & \times \left[\sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j-1)}{\partial p_{s,j}} \Delta p_{s,j} \right] \end{aligned} \quad (10)$$

$$\begin{aligned} & R_r^{(t+n)+1}(i, j) \frac{\partial R_r^{(t+n)+1}(i, j)}{\partial \alpha_p^j} \\ = & \left[R_r^t(i, j) + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial p_{s,j-1}} \Delta p_{s,j-1} \alpha_p^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial p_{s,j}} \Delta p_{s,j} \alpha_p^j + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial p_{s,j+1}} \Delta p_{s,j+1} \alpha_p^{j+1} \right. \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial u_{s,j-1}} \Delta u_{s,j-1} \alpha_u^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial u_{s,j}} \Delta u_{s,j} \alpha_u^j + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial u_{s,j+1}} \Delta u_{s,j+1} \alpha_u^{j+1} \\ & + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial v_{s,j-1}} \Delta v_{s,j-1} \alpha_v^{j-1} + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial v_{s,j}} \Delta v_{s,j} \alpha_v^j + \sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial v_{s,j+1}} \Delta v_{s,j+1} \alpha_v^{j+1} \Big] \\ & \times \left[\sum_{s=i-1}^{i+1} \frac{\partial R_r^t(i, j)}{\partial p_{s,j}} \Delta p_{s,j} \right] \end{aligned} \quad (11)$$

with a similar expression for the $(i, j + 1)$ grid line and similar expressions for $m = u$ and $m = v$.

Substituting these equations into Eq. (10) and collecting like terms with the α_p , α_u , and α_v coefficients, we can construct the first row of each block in the block pentadiagonal matrix. Similar expressions for $m = u$ and $m = v$ give the second and third row of each block, respectively. The overall structure of the block pentadiagonal matrix is

$$\begin{bmatrix} [C^1] & [D^1] & [E^1] & & & & \\ & \ddots & & & & & \\ & & \ddots & & & & \\ \dots & [A^{j-1}] & [B^{j-1}] & [C^{j-1}] & [D^{j-1}] & [E^{j-1}] & \\ & & [A^j] & [B^j] & [C^j] & [D^j] & [E^j] \\ & & & [A^{j+1}] & [B^{j+1}] & [C^{j+1}] & [D^{j+1}] & [E^{j+1}] & \dots \\ & & & & \ddots & & & \\ & & & & & \ddots & & \\ [A^{j\max}] & [B^{j\max}] & [C^{j\max}] & & & & \end{bmatrix} \begin{Bmatrix} \{\alpha^1\} \\ \vdots \\ \{\alpha^{j-2}\} \\ \{\alpha^{j-1}\} \\ \{\alpha^j\} \\ \{\alpha^{j+1}\} \\ \{\alpha^{j+2}\} \\ \vdots \\ \{\alpha^{j\max}\} \end{Bmatrix} = \begin{Bmatrix} \{F^1\} \\ \vdots \\ \{F^{j-2}\} \\ \{F^{j-1}\} \\ \{F^j\} \\ \{F^{j+1}\} \\ \{F^{j+2}\} \\ \vdots \\ \{F^{j\max}\} \end{Bmatrix} \quad (12)$$

where

$$\{\alpha^j\} = \begin{Bmatrix} \alpha_p^j \\ \alpha_u^j \\ \alpha_v^j \end{Bmatrix}$$

and $\{F^j\}$ is the vector of terms that do not contain various α . The row elements of the blocks $[A]$, $[B]$, $[C]$, $[D]$, and $[E]$ are coefficients of α^{j-2} , α^{j-1} , α^j , α^{j+1} , and α^{j+2} , respectively. Each column of the blocks $[A]$, $[B]$, $[C]$, $[D]$, and $[E]$ corresponds to α_p , α_u , and α_v , respectively.

Application of the LSBMR method to a three-dimensional problem would result in a large, unstructured sparse block matrix rather than a block pentadiagonal matrix as for the two-dimensional LSBMR. For this reason it is simpler and more economical to use the plane SBMR (PSBMR) and alternating direction SBMR (APSBMR) methods for three-dimensional problems. The PSBMR method assumes that each grid plane has its own set of various α . Suppose that the $j = \text{const}$ grid plane has a fixed set of various α . Then the PSBMR method yields

$$\begin{aligned} & 2 \sum_{i=1}^{i\max} \sum_{k=1}^{k\max} \sum_{r=1}^{r\max} \left[R_r^{(t+n)+1}(i, j-1, k) \frac{\partial R_r^{(t+n)+1}(i, j-1, k)}{\partial \alpha_m^j} \right] \\ & + 2 \sum_{i=1}^{i\max} \sum_{k=1}^{k\max} \sum_{r=1}^{r\max} \left[R_r^{(t+n)+1}(i, j, k) \frac{\partial R_r^{(t+n)+1}(i, j, k)}{\partial \alpha_m^j} \right] \\ & + 2 \sum_{i=1}^{i\max} \sum_{k=1}^{k\max} \sum_{r=1}^{r\max} \left[R_r^{(t+n)+1}(i, j+1, k) \frac{\partial R_r^{(t+n)+1}(i, j+1, k)}{\partial \alpha_m^j} \right] \\ & = 0 \end{aligned} \quad (13)$$

for $m = 1, \dots, M$. A variation of the PSBMR method, called APSBMR (alternating direction plane SBMR) method can be used by applying the PSBMR method alternately on the $j = \text{const}$ and $k = \text{const}$ planes. This could be useful when the grids are clustered in both j and k directions.

IV. Computational Results

A. Two-Dimensional Flows

Convergence histories of the basic four-stage RK scheme with local time stepping and four acceleration methods [implicit residual smoothing¹ (IRS), DMR, SBMR, and LSBMR] were compared for a steady, incompressible, viscous flow in a two-dimensional straight channel and a U-shaped turnaround channel. The inlet pressure was iteratively computed by enforcing characteristic boundary conditions, whereas at the exit boundary, we used nonreflecting

boundary conditions^{25,26} since there was a recirculating flow at the exit of the U-shaped channel. No artificial dissipation was used in any of these test cases. The original RK algorithm used Chorin's artificial compressibility method.²⁴ The Courant–Friedrichs–Lewy number ($\text{CFL} = c\Delta t/\Delta x^2 = 2.8$, where c is the equivalent speed of sound^{16,17}) and the von Neumann number ($\sigma = \nu\Delta t/\Delta x^2 = 0.4$, where ν is the kinematic viscosity) had maximum possible values

for the four-stage RK scheme in all test cases discussed in this paper to maximize the performance of the basic RK scheme.

First, laminar flow in a straight channel with mild symmetric grid clustering towards the walls [maximum cell aspect ratio $AR_{\max} = (\Delta x/\Delta y)_{\max} = 100$] was tested. The LSBMR method was applied every $\text{freq} = 100$ iterations by utilizing information from two iterations, $n = 20$ iterations apart. It was found to converge faster than the SBMR method with $\text{freq} = 30$ and $n = 10$ (Fig. 2). This was an expected result since the LSBMR method allows various different optimum α in the clustered grid region (Fig. 3) rather than enforcing fixed acceleration parameters over the whole computational domain as required by the SBMR. The computational results compared^{22,23} with analytical solutions to within machine accuracy. Convergence histories in terms of computational time consistently mirrored the convergence histories in term of the number of iterations. Both SBMR and LSBMR methods outperformed the RK

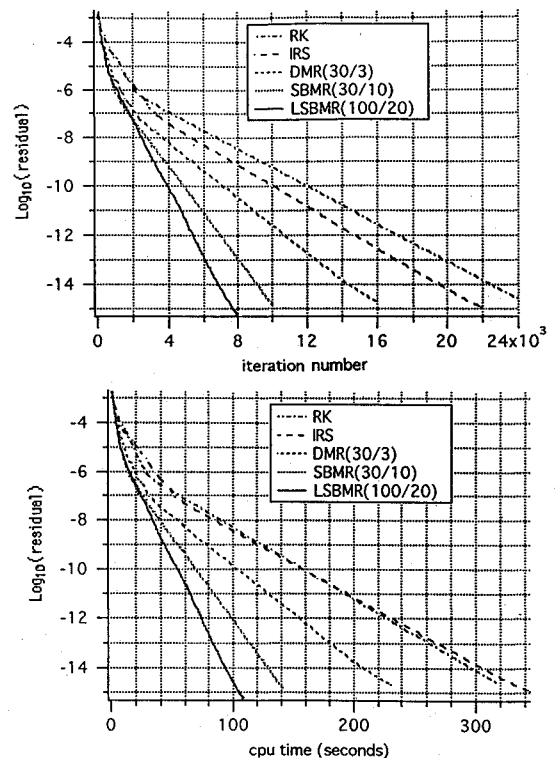


Fig. 2 Convergence histories for a straight channel incompressible flow ($Re = 1.6 \times 10^3$, $L/H = 5$, 60×60 grid cells, $AR_{\max} = 100$, $\beta = 5$, $\text{CFL} = 2.8$, and $\sigma = 0.4$).

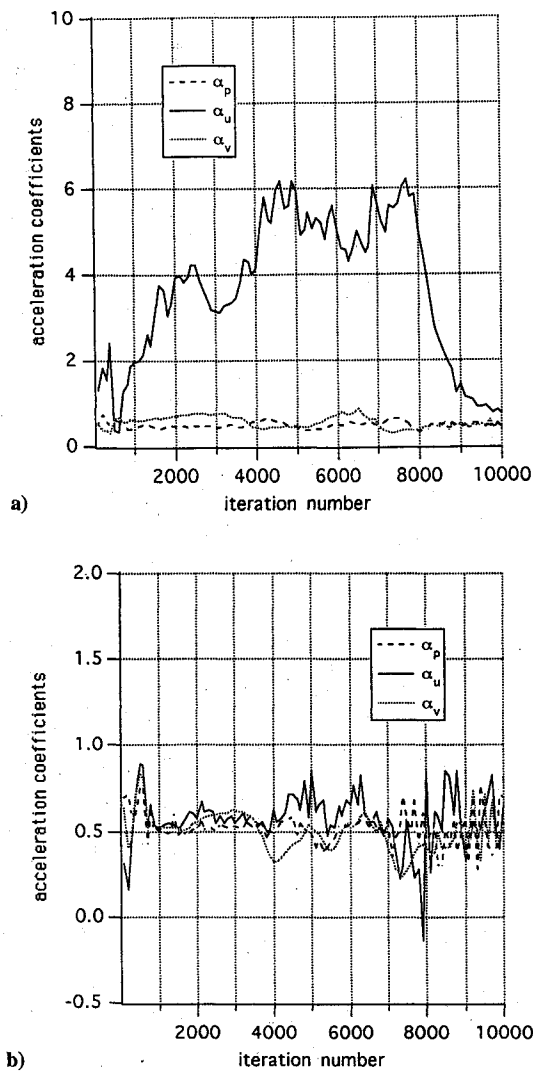


Fig. 3 Evolution of optimum acceleration parameters for LSBMR method applied to a straight channel laminar incompressible flow: a) in a clustered grid region and b) in a coarse grid region.

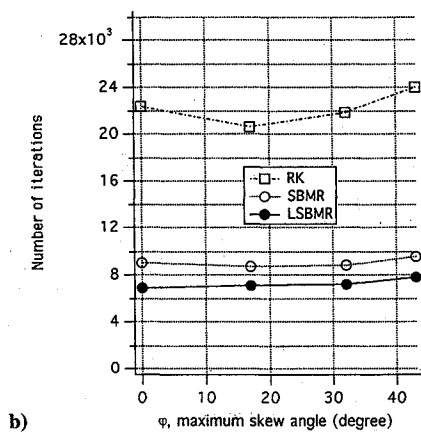
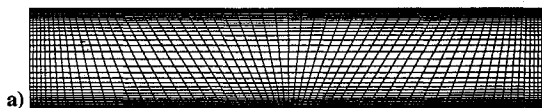


Fig. 4 Grid nonorthogonality effect on RK, SBMR, and LSBMR convergence: a) a nonorthogonal clustered grid and b) sensitivity to nonorthogonality angle.

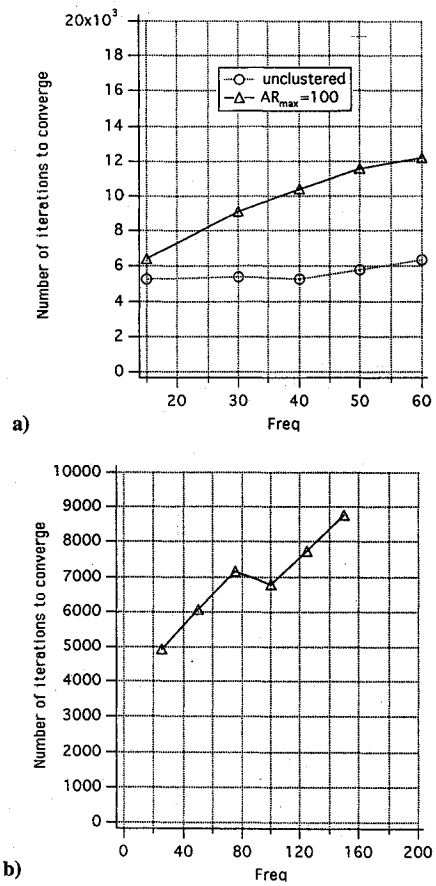


Fig. 5 Effect of frequency of application of a) SBMR and b) LSBMR on convergence.

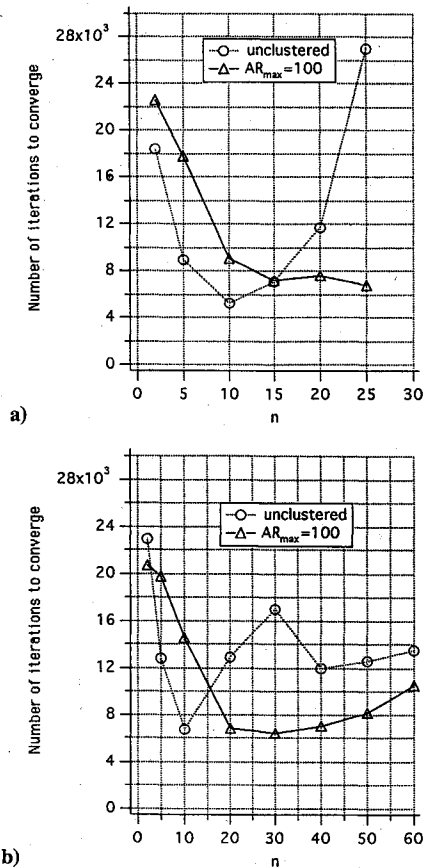


Fig. 6 Effect of the number of iterations apart, n , appearing in the SBMR and LSBMR formulations on the convergence of a) SBMR and b) LSBMR.

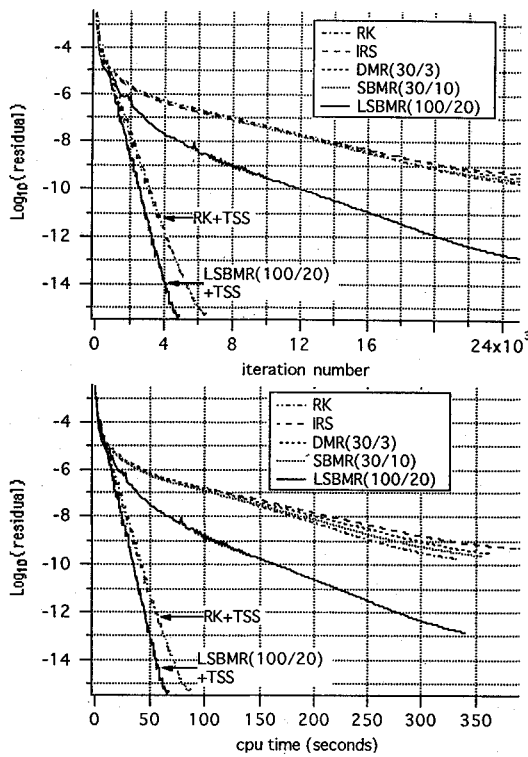


Fig. 7 Convergence histories for a straight channel incompressible flow ($Re = 1.6 \times 10^3$, $L/H = 5$, 60×60 grid cells, $AR_{max} = 10,000$, $\beta = 5$, $CFL = 2.8$, and $\sigma = 0.4$).

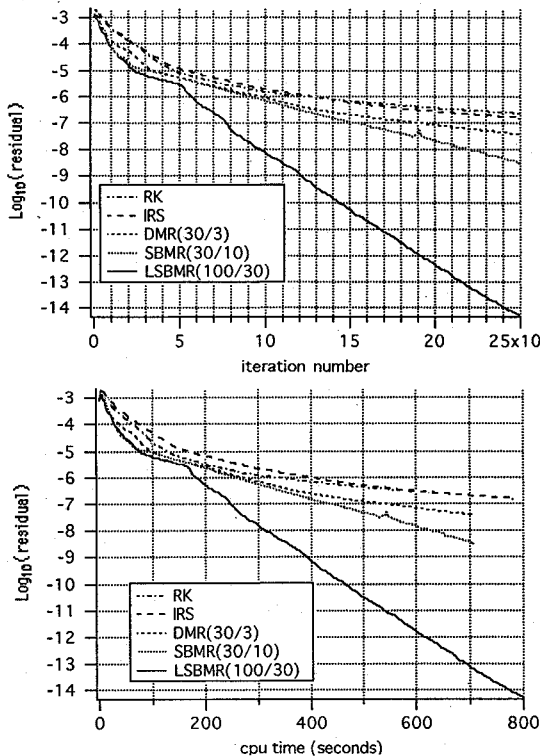


Fig. 8 Convergence histories for a straight channel incompressible turbulent flow ($Re = 1.6 \times 10^6$, $L/H = 10$, 60×120 grid cells, $AR_{max} = 1000$, $\beta = 5$, $CFL = 2.0$, and $\sigma = 0.4$).

scheme over a range of grid nonorthogonality angles (Fig. 4). Both methods depended on the user-specified frequency freq of their application (Fig. 5) and on the number of iterations, n , that separate the iteration levels contributing the information (Fig. 6). When the grid clustering was increased to $AR_{max} = 10,000$, the convergence rates clearly slowed down for all methods tested (Fig. 7). The LSBMR method, however, maintained a considerably faster convergence especially when combined with the simple time-step scaling²² (TSS) method. When this method was used, the continuity equation was

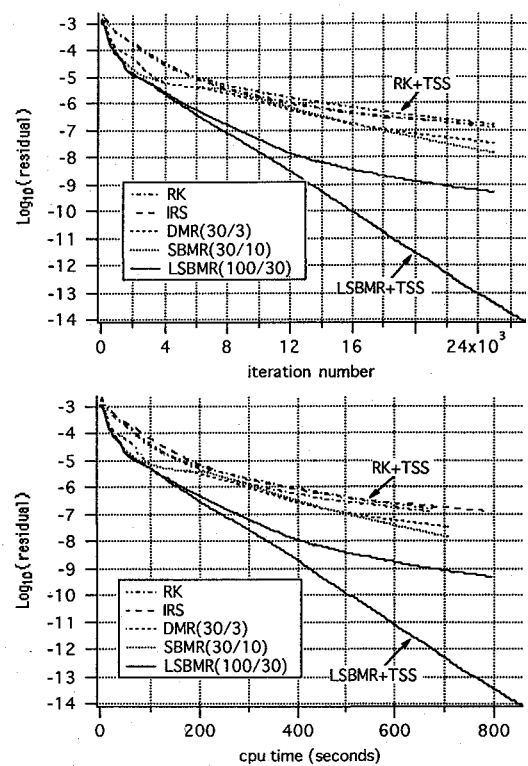


Fig. 9 Convergence histories for a straight channel incompressible turbulent flow ($Re = 1.6 \times 10^6$, $L/H = 10$, 60×120 grid cells, $AR_{max} = 10,000$, $\beta = 5$, $CFL = 2.0$, and $\sigma = 0.4$).

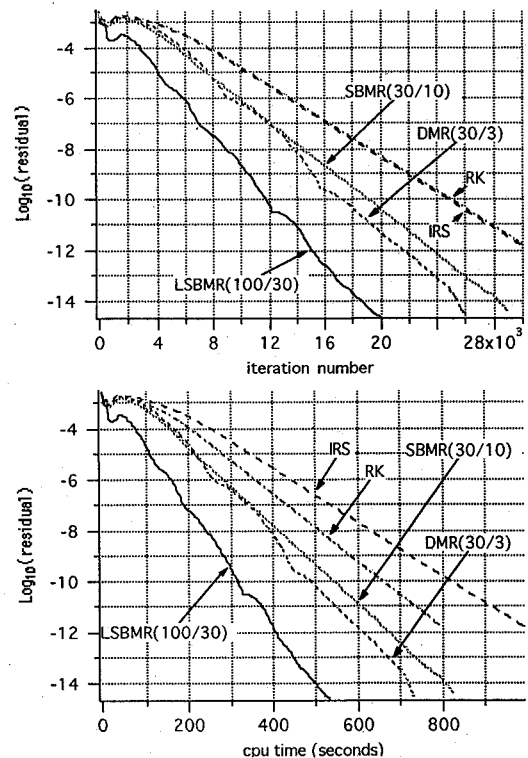


Fig. 10 Convergence histories for a U-shaped channel incompressible flow ($Re = 2 \times 10^2$, 129×80 grid cells, $AR_{max} = 200$, $\beta = 10$, $CFL = 2.8$, and $\sigma = 0.4$).

scaled such that $J \Delta \tau$ in the clustered regions had the same order of magnitude as that in the coarse grid regions. Here J is the determinant of the local Jacobian transformation matrix between the physical (x, y) and the curvilinear nonorthogonal boundary-conforming computational (ξ, η) coordinate system, while $\Delta \tau$ is the local time step computed using standard formulation.²⁷

The convergence acceleration schemes were then tested for a turbulent flow²⁸ with Reynolds number $Re = 1.6 \times 10^6$ in a straight

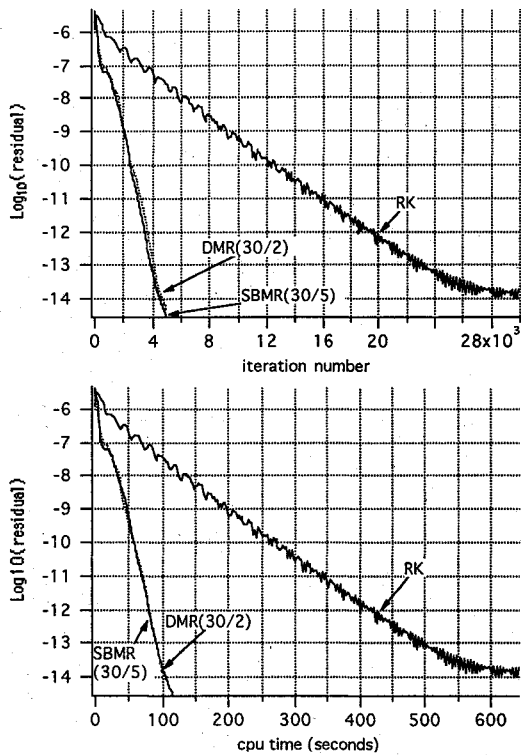


Fig. 11 Convergence histories for the compressible flow Euler equations ($M_\infty = 0.05$).

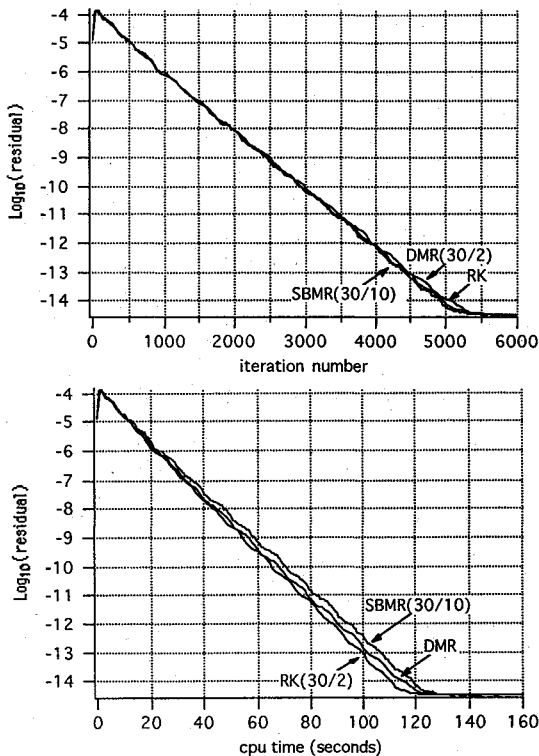


Fig. 12 Convergence histories for the compressible flow Euler equations ($M_\infty = 0.675$).

channel. Convergence histories for $AR_{\max} = 1000$ demonstrated (Fig. 8) that the LSBMR method consistently outperformed other acceleration schemes. When the grid clustering was increased to $AR_{\max} = 10,000$, the LSBMR converged consistently faster than all other acceleration schemes (Fig. 9). It is interesting to notice that TSS did not have any effect when applied to the basic RK scheme, but it achieved impressive acceleration (Fig. 9) when applied together with the LSBMR method. The acceleration methods were also tested for situations with flow separation that stretches through the exit plane.^{22,23} We computed laminar flow in a U-shaped chan-

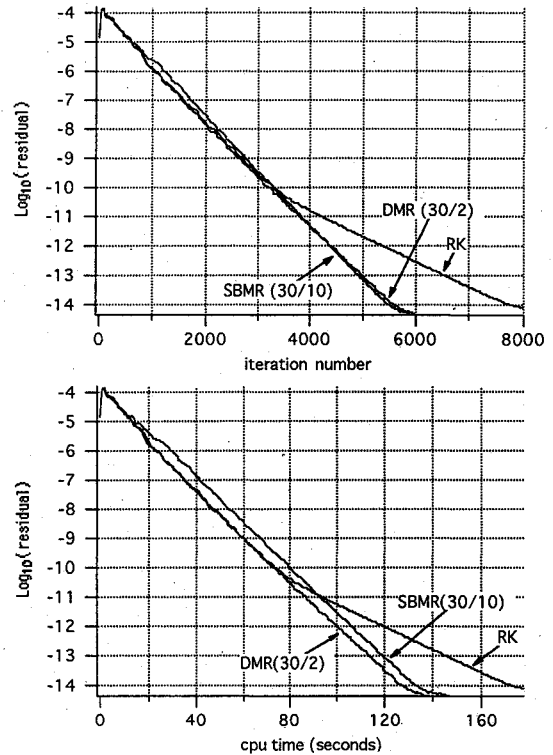


Fig. 13 Convergence histories for the compressible flow Euler equations ($M_\infty = 0.675$) with smaller artificial dissipation.

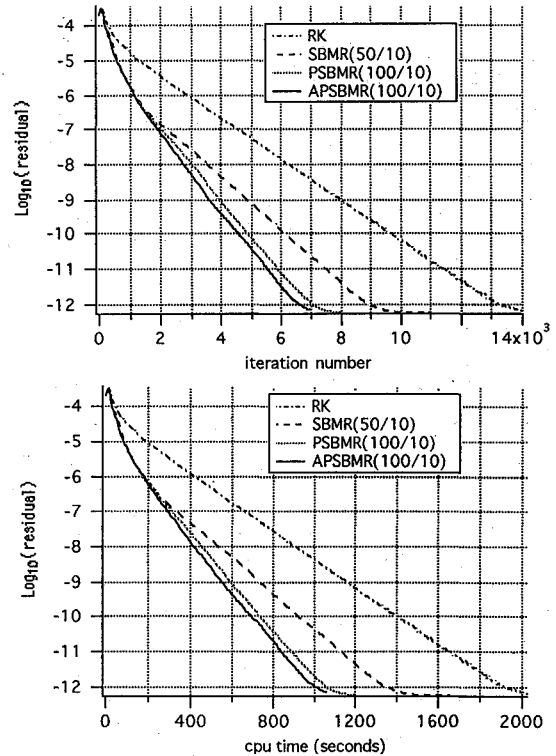


Fig. 14 Convergence histories for a straight square duct incompressible flow ($Re = 2 \times 10^3$, $24 \times 20 \times 20$ grid cells, and $AR_{\max} = 127$).

nel with computational grid symmetrically clustered towards the channel walls. With $AR_{\max} = 200$, the LSBMR method offered the fastest convergence (Fig. 10). In this test case the RK with $CFL = 2.8$ was faster than the RK-based IRS, although performance of the IRS was maximized by finding its coefficients using numerical experimentation.

Performance of the SBMR method was next evaluated against the DMR method and the RK scheme for inviscid compressible flows. Convergence rates of existing algorithms are very slow when integrating compressible flow equations at low Mach numbers¹⁵

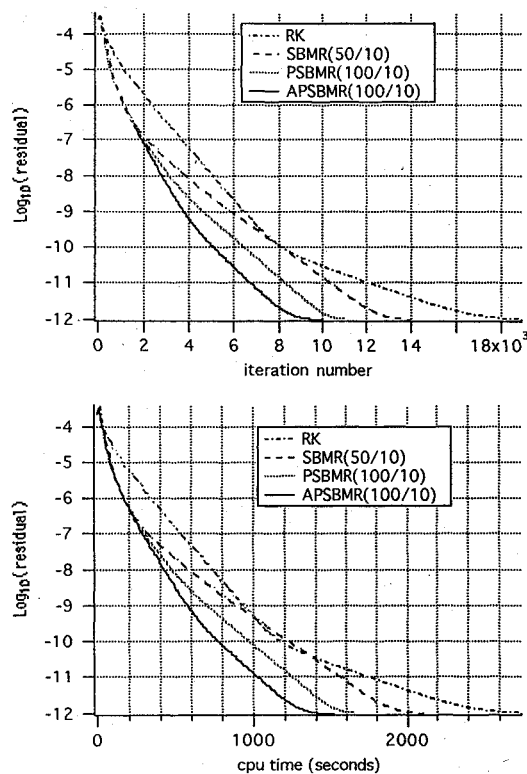


Fig. 15 Convergence histories for a straight square duct incompressible flow ($Re = 2 \times 10^3$, $24 \times 20 \times 20$ grid cells, and $AR_{max} = 318$).

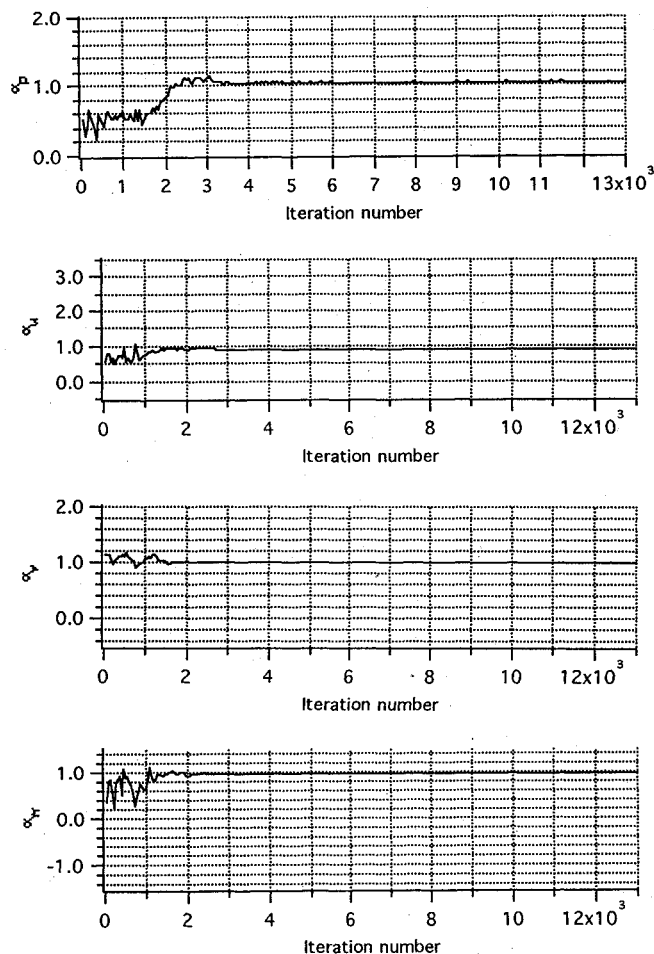


Fig. 16 Evolution of optimum acceleration parameters for SBMR method applied to a laminar incompressible flow in a straight square duct ($\beta = 5$, $AR_{max} = 318$, $freq = 50$, and $n = 10$).

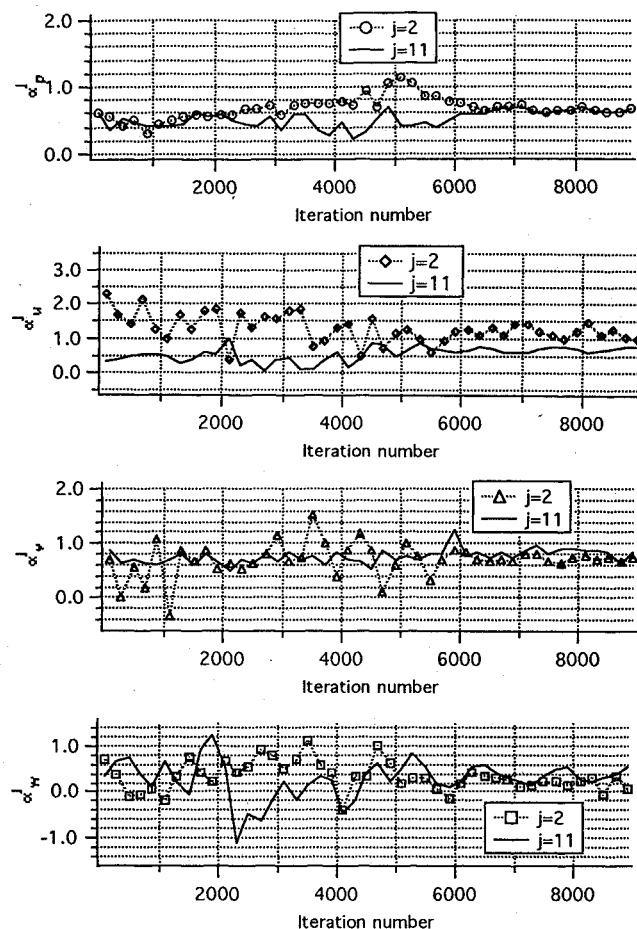


Fig. 17 Evolution of optimum acceleration parameters for APSBMR method applied to a laminar incompressible flow in a straight square duct ($\beta = 5$, $AR_{max} = 318$, $freq = 100$, and $n = 10$).

because of the excessive stiffness of the system. We used a second-order-accurate one-sided differencing at the boundary points. A combination of second- and fourth-order artificial dissipation¹ was used in a fully conservative form to eliminate even-odd decoupling. A flow tangency boundary condition was employed on the solid boundaries. The test geometry was a straight channel with the computational grid composed of 64×32 cells clustered towards the bottom wall having a circular arc bump.⁴ Grid cells in the main stream direction were clustered in the constant height sections of the channel towards the leading edge and the trailing edge of the arc, whereas a uniform grid spacing was used in the arc section. In all test cases, both SBMR and DMR methods were applied after every 30 iterations. The DMR method combined two consecutive iteration steps, whereas the SBMR method used the results of $n = 5$ iteration levels apart except for the case of inlet Mach number $M_\infty = 0.675$ where the results of $n = 10$ levels apart were used. The convergence histories for $M_\infty = 0.05$ (Fig. 11) are equally impressive for both SBMR and DMR methods. They reduced the number of iterations and the computing time by 84% for this stiff system. The actual CPU time overhead per iteration for the SBMR method was found to be 8.8%, whereas it was 2.4% for the DMR method for the test cases done here. The transonic ($M_\infty = 0.675$) test case required an artificial dissipation five times larger to stabilize the solution with a shock wave.²² Convergence of the basic RK iteration scheme was much faster than for the low subsonic case (Fig. 12), and no convergence acceleration was obtained by either the SBMR method or the DMR method. When the artificial dissipation was reduced,^{29,30} the SBMR method performed better than the RK scheme (Fig. 13). This was understandable since the present formulation of the SBMR method did not account for the artificial dissipation terms. It used various constant α at every grid point, and it was based on a central differencing, thus violating proper domain of dependence in locally supersonic flows.

B. Three-Dimensional Flow

The performances of the SBMR, PSBMR, and APSBMR, methods were compared for a fully developed laminar incompressible flow through a straight duct with a unit square cross section having a length of 10 units. The grid was clustered symmetrically toward the solid walls, and a uniform grid spacing was used in the axial x direction. Chorin's compressibility coefficient, $\beta = 5$, was used in this case. At the inlet boundary, a fully developed three-dimensional laminar velocity profile was given, pressure was computed from characteristic boundary conditions, and a nonreflecting boundary condition was used at the exit boundary.^{25,26} The SBMR method was applied after every 50 iterations utilizing the results $n = 10$ iteration levels apart. The convergence histories for $AR_{\max} = 127$ (Fig. 14) and $AR_{\max} = 318$ (Fig. 15) demonstrated that the PSBMR and especially the APSBMR method reduced the computing time by about 50%, although optimum acceleration parameters for SBMR rapidly stabilize (Fig. 16), whereas these parameters for APSBMR in clustered and coarse grid regions continue changing (Fig. 17) throughout the iterative process.

V. Conclusions and Recommendations

The sensitivity-based minimal residual method and its variations were found to be applicable to a general system of time-dependent second-order partial differential equations and systems of such equations. The performance of the SBMR and the LSBMR methods depended on how frequently these methods are applied during the basic iteration process. The performance also depended on the number of iterations performed with the basic iterative algorithm between the two iteration levels that are involved in the evaluation of the change of the solution vector. In the case of two-dimensional incompressible viscous flows without a severe pressure gradient, the SBMR and LSBMR methods significantly accelerated the convergence of iterative procedure on clustered grids with the LSBMR method becoming more efficient as grids become more clustered. For a two-dimensional incompressible laminar flow, the SBMR and the LSBMR methods maintained fast convergence for nonorthogonal grids and for flows with closed and open flow separation. The TSS method, used together with the RK scheme and the LSBMR method, greatly enhanced the convergence rates when solving two-dimensional incompressible laminar and turbulent flows through a straight channel with highly clustered grids. The TSS method warrants further study since its performance with the basic RK scheme deteriorated with the increase in grid clustering. The SBMR method accelerated the convergence of inviscid, low-Mach-number, compressible flows where the system was very stiff. However, the SBMR method did not offer any convergence acceleration for a transonic flow where there was a large amount of artificial dissipation that was not incorporated in the sensitivity analysis. The APSBMR method, a three-dimensional analogy of the LSBMR method, successfully reduces the computational effort for solving a three-dimensional, laminar flow through a straight duct without flow separation. The general formulation of our new acceleration methods is applicable to any iteration scheme (explicit or implicit) and it requires only two iteration levels to be stored. Future research should explore the possibilities for combining our methods with other iteration algorithms and with other acceleration methods to achieve a cumulative acceleration effect.

Acknowledgments

This work was supported by Grant NAS 8-38861 from NASA Marshall Space Flight Center, Huntsville, AL, monitored by Paul McConaughy and by the Penn State NASA Propulsion Engineering Research Center, Contract NAGW-1356, Supplement 8. Computing time was provided on the NASA Ames NAS Cray-90 facility by Louis Povinelli of NASA Lewis Research Center.

References

- Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.
- Pulliam, T. H., "Euler and Thin Layer Navier-Stokes Codes: ARC2D, ARC3D," Notes for Computational Fluid Dynamics User's Workshop. Univ. of Tennessee Space Inst., Tullahoma, TN, March 1984.
- Jameson, A., and Baker, T. J., "Solution of the Euler Equations for Complex Configurations," AIAA Paper 83-1929, July 1983.
- Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," AIAA Journal, Vol. 20, No. 11, 1982, pp. 1565-1571.
- Martinelli, L., Jameson, A., and Grasso, F., "A Multigrid Method for the Navier-Stokes Equations," AIAA Paper 86-0208, Jan. 1986.
- Caughey, D. A., "A Diagonal Implicit Multigrid Algorithm for the Euler Equations," AIAA Paper 87-0354, Jan. 1987.
- Chima, R. V., Turkel, E., and Schaffer, S., "Comparison of Three Explicit Multigrid Methods for the Euler and Navier-Stokes Equations," AIAA Paper 87-0602, Jan. 1987.
- Dannenhoffer, J. F., and Giles, M. B., "Accelerated Convergence of Euler Solutions Using Time Inclining," AIAA Journal, Vol. 28, No. 8, 1990, pp. 1457-1463.
- Turkel, E., "Review of Preconditioning Methods for Fluid Dynamics," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 92-47, NASA Langley Research Center, Hampton, VA, Sept. 1992.
- Choi, Y., and Merkle, C. L., "Time-Derivative Preconditioning for Viscous Flow," AIAA Paper 91-1652, June 1991.
- Kennon, S. R., and Dulikravich, G. S., "Optimum Acceleration Factors for Iterative Solution of Linear and Nonlinear Differential Systems," Computer Methods in Applied Mechanics and Engineering, Vol. 47, 1984, pp. 357-367.
- Hafez, M., Parlette, E., and Salas, M. D., "Convergence Acceleration of Iterative Solutions of Euler Equations for Transonic Flow Computations," AIAA Paper 85-1641, July 1985.
- Huang, C. Y., Kennon, S. R., and Dulikravich, G. S., "Generalized Non-Linear General Minimal Residual (GNLMR) Method for Iterative Algorithms," Journal of Computational and Applied Mathematics, Vol. 16, Nov. 1986, pp. 215-232.
- Huang, C. Y., and Dulikravich, G. S., "Fast Iterative Algorithms Based on Optimized Explicit Time-Stepping," Computer Methods in Applied Mechanics and Engineering, Vol. 63, Aug. 1987, pp. 15-36.
- Lee, S., and Dulikravich, G. S., "Acceleration of Iterative Algorithms for Euler Equations of Gasdynamics," AIAA Journal, Vol. 28, No. 5, 1990, pp. 939-942.
- Lee, S., and Dulikravich, G. S., "Distributed Minimal Residual (DMR) Method for Acceleration of Iterative Algorithms," Computer Methods in Applied Mechanics and Engineering, Vol. 86, 1991, pp. 245-262.
- Lee, S., and Dulikravich, G. S., "Accelerated Computation of Viscous Flow with Heat Transfer," Numerical Heat Transfer: Fundamentals, Pt. B, Vol. 19, June 1991, pp. 223-241.
- Saad, Y., and Schultz, M., "Conjugate Gradient-Like Algorithms for Solving Non-Symmetric Linear Systems," Mathematics of Computation, Vol. 44, No. 170, 1985, pp. 417-424.
- Choi, K.-Y., and Dulikravich, G. S., "Reliability Enhancement of Navier-Stokes Codes Through Convergence Enhancement," Proceedings of 5th Annual Symposium of the Penn State-NASA Propulsion Engineering Research Center, NASA Propulsion Engineering Research Center, University Park, PA, Sept. 1993.
- Choi, K.-Y., and Dulikravich, G. S., "Global Sensitivity-Based Minimal Residual (GSBMR) Method for Acceleration of Iterative Convergence Rates," Proceedings of the 14th International Conference on Numerical Fluid Dynamics (Bangalore, India), July 1994.
- Choi, K.-Y., and Dulikravich, G. S., "Convergence Rate Enhancement of Navier-Stokes Codes on Clustered Grids," Proceedings of 6th Annual Symposium of the Penn State-NASA Propulsion Engineering Research Center, NASA Lewis Research Center, Cleveland, OH, Sept. 1994.
- Choi, K.-Y., "Sensitivity-Based Methods for Convergence Acceleration of Iterative Algorithms," Ph.D. Thesis, Dept. of Aerospace Engineering, Pennsylvania State Univ., University Park, PA, Dec. 1994.
- Choi, K.-Y., and Dulikravich, G. S., "Sensitivity-Based Methods for Convergence Acceleration of Iterative Algorithms," Computer Methods in Applied Mechanics and Engineering, Vol. 123, 1995, pp. 161-172.
- Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," Journal of Computational Physics, Vol. 2, 1967, pp. 12-26.
- Thompson, K. W., "Time-Dependent Boundary Conditions for Hyperbolic Systems, II," Journal of Computational Physics, Vol. 89, 1990, pp. 439-461.
- Dulikravich, G. S., Ahuja, V., and Lee, S., "Modeling Three-Dimensional Solidification with Magnetic Fields and Reduced Gravity," International Journal of Heat and Mass Transfer, Vol. 37, No. 5, 1994, pp. 837-853.
- MacCormack, R. W., and Baldwin, B. S., "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Interaction," AIAA Paper 75-1, Jan. 1975.
- Baldwin, B. S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Jan. 1978.
- Dulikravich, G. S., Dorney, D. J., and Lee, S., "Numerical Versus Physical Dissipation in the Solution of Compressible Navier-Stokes Equations," AIAA Paper 89-0550, Jan. 1989.
- Dulikravich, G. S., and Dorney, D. J., "Artificial Dissipation Sensors for Computational Gasdynamics," AIAA Paper 89-0643, Jan. 1989.