COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 63 (1987) 15–36
NORTH-HOLLAND

FAST ITERATIVE ALGORITHMS BASED ON OPTIMIZED EXPLICIT TIME STEPPING

Chung-Yuan HUANG
Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712, U.S.A.

George S. DULIKRAVICH
Department of Aerospace Engineering, Penn State University, University Park, PA 16802, U.S.A.

Received 9 May 1986
Revised manuscript received 26 September 1986

The Generalized Nonlinear Minimal Residual (GNLMR) method is shown to consistently accelerate and stabilize iterative algorithms for solving nonlinear problems by using the optimized explicit multistepping. The examples presented in this paper illustrate the beneficial effects of the optimized multistep algorithm on the computational efficiency and the convergence rate as applied to several nonlinear problems in fluid dynamics. The significant reduction in computing time when using the multiple optimized acceleration factors is only negligibly weighed down by the computation costs due to the requirements for additional computer storage.

1. Introduction

The relaxation factor used in accelerating an iterative method to obtain the converged solution plays the same role as the time step size in advancing the transient solution to the steady-state solution for a time-dependent problem. The classical analyses for the stability of numerical schemes for solving time-dependent problems neglect boundary conditions and assume a uniform computational grid. Furthermore, these analyses are based on linear equations with constant coefficients and the assumptions of small perturbations and applicability of Fourier analysis [1, 2]. However, Cheng [2] pointed out that the perturbation of the error in the finite difference calculations may not be small and that the error in the finite difference calculations may not satisfy the conditions for Fourier series expansion. In addition, Mitchell and Griffiths [1] pointed out that the errors due to approximate or additional boundary conditions are represented by modes which are not of Fourier type. Thus, the linear stability analysis usually results in overly restrictive and even incorrect conclusions.

The numerical experiments performed by Kennon and Dulikravich [3] and Kennon [4] using the NonLinear Minimal Residual (NLMR) method showed that the usual Courant–Friedrichs–Lewy (CFL) number limitation for both linear and nonlinear problems can be significantly exceeded. The NLMR method provided a simple analytic way to determine the

optimal acceleration factors for both linear and nonlinear problems. However, the elementary time steps used for obtaining the corrections still follow the CFL number limitation concluded from the linear analysis.

The generalized nonlinear minimal residual (GNLMR) method developed by Huang, Kennon and Dulikravich [5] provided a practical analytical tool to determine the exact stability conditions for both linear and nonlinear problems in arbitrary domains. If accurate time evolution is required when solving an unsteady problem, the limitation on the time step size can be analytically determined by using the GNLMR method. If transient behavior is of no interest, the GNLMR method can be applied to determine the optimal value of the time step size (optimal acceleration factor) to minimize the number of time steps (number of iterations) for obtaining the steady-state converged solution.

The main objective of this paper is to investigate the effects of the optimized multistep algorithm on the computational efficiency and on the monotonicity of convergence rate of the GNLMR method. The analytic investigation is confirmed on four nonlinear test cases: the one-dimensional and two-dimensional viscous Burgers' equations and the two-dimensional incompressible and compressible Stream-Function-Coordinate (SFC) equations [6].

2. Theoretical aspects

2.1. Multistep minimal residual method for linear problems

Let us first consider a well-posed linear initial value problem:

$$\begin{aligned} &\partial\varphi/\partial\tau = L\varphi - F \quad \text{in } \Omega\,, \\ &\varphi = \varphi_B \qquad\quad \text{on } \partial\Omega\,, \\ &\varphi = \varphi_0 \qquad\quad \text{at } \tau = \tau_0\,. \end{aligned} \tag{1}$$

Define

$$r^t = l\varphi^t - f \tag{2}$$

as the residual vector at time level t. Here, l denotes the scheme-dependent discrete analog of L, f is the discrete analog of F and also includes boundary terms.

Assume that M steps are used to iterate at each time level t. Using the Einstein summation convention where repeated subscripts are summed, the multistep algorithm for (1) is then defined as follows:

$$\varphi^{t+1} = \varphi^t + \omega_m \delta_m\,, \quad m = 1, 2, \ldots, M\,, \tag{3}$$

where

$$\begin{aligned} &\delta_1 = l\varphi^t - f\,, \\ &\delta_m = l^{m-1}(\delta_1)\,, \quad m > 1\,, \end{aligned} \tag{4}$$

are the corrections at step m. Coefficients ω_m are the corresponding relaxation factors to be determined by minimizing the L_2 norm of the residual at time level $(t+1)$. With the definition

of residual vector (2) and the help of (3), the following relation can easily be verified:

$$r^{t+1} = r^t + \omega_m l\delta_m = r^t + \omega_m l^m r^t \,. \tag{5}$$

The L_2 norm of the residual vector at time level $(t+1)$ is then

$$\|r^{t+1}\|^2 = \|r^t\|^2 + 2\omega_m(r^t, l\delta_m) + (l\delta_m, l\delta_n)\omega_m\omega_n \,, \quad m, n = 1, 2, \ldots, M \,. \tag{6}$$

It should be pointed out that the boundary condition for the corrections δ_m in (3) is apparently zero. However, the boundary conditions for the residual vector r^t and corrections δ_m in (5) can be determined either by extrapolation from the interior points or simply by setting them equal to zero. The residual norm will then converge to the norm of the truncation error of the difference scheme if the first method is applied and to the machine accuracy if the second method is applied.

The highest rate of convergence is possible when ω_m are the solutions of the following system of linear equations:

$$\partial\Gamma/\partial\omega_m = 0 \quad \text{or} \quad (r^t, l\delta_m) + (l\delta_m, l\delta_n)\omega_n = 0 \,, \tag{7}$$

where the rate of convergence Γ is defined as

$$\Gamma = -\log(\|r^{t+1}\| / \|r^t\|) \,. \tag{8}$$

Multiplying (7) by ω_m, it follows that

$$(r^t, l\delta_m)\omega_m + (l\delta_m, l\delta_n)\omega_m\omega_n = 0 \,. \tag{9}$$

Subtracting (9) from (6) and using (7) results in

$$\|r^{t+1}\|^2 - \|r^t\|^2 = (r^t, l\delta_m)\omega_m = -(l\delta_m, l\delta_n)\omega_m\omega_n = -\int_\Omega (\omega_m l\delta_m)^2 \, d\Omega < 0 \,. \tag{10}$$

Thus, the residual norms for the multistep minimum residual method show a monotone convergence behavior which guarantees the stability of the iterative scheme and produces the highest rate of its convergence.

2.2. *Optimization of the Euler scheme for nonlinear problems*

For clarity, we consider two-dimensional problems and equations in conservative form only. The extension to multidimensional problems and nonconservative equations is then straightforward.

The conservative form of the governing equations for most engineering problems can be written as:

$$\partial\varphi/\partial\tau = L_\nu N^\nu(\varphi, \varphi_x, \varphi_y) - F \,, \tag{11}$$

where the operators are

$$L_1 = \partial/\partial x\,, \qquad L_2 = \partial/\partial y\,,$$

and N^ν is the nonlinear differential operator in coordinates x_ν. Using the Euler one-step, time-consistent, explicit scheme, the finite difference form of (11) can be written as:

$$\varphi^{t+1} = \varphi^t + \Delta\tau r^t\,, \tag{12}$$

where

$$r^t = l_\nu N^\nu(\varphi^t, \varphi^t_x, \varphi^t_y) - f \tag{13}$$

is defined as the residual at time level t. Therefore, the residual at time level $(t+1)$ can be expressed as

$$r^{t+1} = l_\nu N^\nu(\varphi^{t+1}, \varphi^{t+1}_x, \varphi^{t+1}_y) - f\,. \tag{14}$$

After expanding the nonlinear discretized operator N^ν in a Taylor series, it follows that

$$r^{t+1} = l_\nu\{N^\nu(\varphi^t, \varphi^t_x, \varphi^t_y) + [(\partial N^\nu/\partial\varphi^t)r^t + (\partial N^\nu/\partial\varphi^t_x)(r^t)_x + (\partial N^\nu/\partial\varphi^t_y)(r^t)_y]\Delta\tau + O(\Delta\tau^2)\} - f\,. \tag{15}$$

In summary,

$$r^{t+1} = r^t + a_p(\Delta\tau)^p\,, \quad 1 \leqslant p \leqslant P\,, \tag{16}$$

where P is the degree of the nonlinearity of the operator N. Equation (16) indicates that the residual at time level $(t+1)$ is a polynomial (henceforth called Residual Polynomial [3] or RP) of the time step size, $\Delta\tau$. Thus, the L_2 norm of the residual at time level $(t+1)$ can be expressed as

$$\|r^{t+1}\|^2 = \|r^t\|^2 + 2(r^t, a_p)(\Delta\tau)^p + (a_p, a_q)(\Delta\tau)^p(\Delta\tau)^q\,, \quad 1 \leqslant p\,,\ q \leqslant P\,. \tag{17}$$

Equation (17) implies that the residual norm at time level $(t+1)$ is a positive polynomial (henceforth called Minimizing Polynomial [3] or MP) of the time step size $\Delta\tau$, which is to be determined. Thus, the convergence of scheme (12) will be guaranteed provided that $\Delta\tau$ is chosen in such a way that $\Gamma > 0$. The highest rate of convergence can be achieved only when $\Delta\tau$ is chosen as the optimizer of the minimizing polynomial (17) such that $\|r^{t+1}\|$ is an infimum. However, the determination of the optimizer needs special numerical techniques [7]. To avoid this difficulty, the linearized operator [3–5] of N^ν may be applied. If N^ν is truncated to the first order in $\Delta\tau$ (linearized operator), the approximate residual vector is

$$r^{*t+1} = r^t + a_1\Delta\tau\,, \tag{18}$$

where

$$a_1 = l_\nu[(\partial N^\nu/\partial\varphi^t)r^t + (\partial N^\nu/\partial\varphi^t_x)(r^t)_x + (\partial N^\nu/\partial\varphi^t_y)(r^t)_y]\,. \tag{19}$$

Then, the approximate MP is

$$\|r^{*t+1}\|^2 = \|r^t\|^2 + 2(a_1, r^t)\Delta\tau + (a_1, a_1)(\Delta\tau)^2 . \tag{20}$$

The optimal time step for the explicit Euler scheme can be easily found as

$$(\Delta\tau)_{\text{opt.}} = -(a_1, r^t)/\|a_1\|^2 . \tag{21}$$

2.3. The generalized nonlinear minimum residual (GNLMR) method

The GNLMR method actually is the application of the methods described in the previous sections. The multistep algorithm for nonlinear problems is defined as

$$\varphi^{t+1} = \varphi^t + \omega_m \delta_m + \mathrm{O}(\omega_m^2) , \quad m = 1, 2, \ldots, M , \tag{22}$$

where repeated indices are summed. The correction at the first step if defined as

$$\delta_1 = r^t = l_\nu N^\nu(\varphi^t, \varphi_x^t, \varphi_y^t) - f . \tag{23}$$

The correction at step $m > 1$ is defined as

$$\delta_m = l_\nu[(\partial N^\nu/\partial\varphi^t)\delta_{m-1} + (\partial N^\nu/\partial\varphi_x^t)(\delta_{m-1})_x + (\partial N^\nu/\partial\varphi_y^t)(\delta_{m-1})_y] . \tag{24}$$

The coefficients of the higher-order terms of ω_m can be obtained by Taylor-series expansion. If only linear terms of ω_m are retained, the residual polynomial (RP) at time level $(t+1)$ can be expressed by Taylor-series expansion as

$$\begin{aligned} r^{t+1} &= l_\nu N^\nu(\varphi^{t+1}, \varphi_x^{t+1}, \varphi_y^{t+1}) - f \\ &= l_\nu N^\nu[\varphi^t + \omega_m\delta_m, \varphi_x^t + \omega_m(\delta_m)_x, \varphi_y^t + \omega_m(\delta_m)_y] - f \\ &= r^t + l_\nu\{[(\partial N^\nu/\partial\varphi^t)\delta_m + (\partial N^\nu/\partial\varphi_x^t)(\delta_m)_x + (\partial N^\nu/\partial\varphi_y^t)(\delta_m)_y]\omega_m + \mathrm{O}(\omega_m^2)\} . \end{aligned} \tag{25}$$

Therefore, the minimizing polynomial (MP) at time level $(t+1)$ can be determined as

$$\|r^{t+1}\|^2 = \|r^t\|^2 + g(\omega_m) , \tag{26}$$

where $g(\omega_m)$ is a polynomial in ω_m. For a highly nonlinear differential equation, g will be a complicated multivariable polynomial that depends on the total number of intermediate steps M that were used and the degree of the nonlinearity of the differential operator N^ν. Thus, a fast and accurate procedure of determining the optimizer of MP is required for the GNLMR method to guarantee the highest rate of convergence. If the linearized operator of N^ν is used, the method that was described in Section 2.1 can be applied to determine the approximate optimizer of (26).

The GNLMR method requires $(M+1)$ times larger computer storage to save the correc-

tions from M intermediate steps than does the single-step nonaccelerated scheme. Some additional algebraic operations are also required to determine the coefficients of the MP which are obtained by integrating the corrections over the entire domain. The storage requirement of the GNLMR method is quite acceptable when compared with the excessive storage required by the GMRES method [8, 9]. It should be pointed out that the GMRES method also needs a large number of arithmetic operations not only for orthonormalizing search directions but also for determining the optimal weighing parameters in updating the iterative solutions.

3. Numerical examples

Four test cases were used to demonstrate the application, the computational efficiency, and the monotone convergence behavior of the GNLMR method. Since it was found [5] that the linearized residual polynomial still guarantees a relatively high convergence rate, it will be used for all test cases. The first two cases representing the one-dimensional and two-dimensional viscous Burgers' equation were solved by the time-dependent technique as described in Section 2.

The last two test cases, the two-dimensional incompressible and compressible stream-function-coordinate (SFC) equations [6] were solved in their steady-state and nonconservative form. Liebman's or Gauss–Seidel's method was applied to determine the correction at each intermediate iterative step m.

Details about the control parameters such as grid size, stopping criteria, and number of acceleration factors used for each test case are summarized in Table 1. For all test cases, comparisons are based on the relative improvement of computational efficiency that can be

Table 1
Summary of the control parameters for numerical test cases

Test case	Max. no. of ω used	Boundary conditions for residual and corrections	Stopping criteria	Grid size
Case 1: 1D Burgers' equation	8	Extrapolation from interior data Zero	$\Gamma \leq 10^{-8}$ $\|\|r'\|\| \leq 10^{-8}$	41
Case 2: 2D Burgers' equation	8	Extrapolation from interior data Zero	$\Gamma \leq 10^{-8}$ $\|\|r'\|\| \leq 10^{-8}$	51×51
Case 3: 2D incomp. SFC equation	8	Zero	$\|\|r'\|\| \leq 10^{-8}$	47×11
Case 4: 2D comp. SFC equation	8	Zero	$\|\|r'\|\| \leq 10^{-8}$	61×11

obtained using different total number of intermediate steps M. The relative improvement of computational efficiency η_M is defined as

$$\eta_M = T_0/T_M \,, \tag{27}$$

where T_0 denotes the computing time spent for the nonaccelerated method and T_M denotes the computing time spent for the acceleration method based on the stopping criteria as described in Table 1. The results are summarized in Fig. 9.

3.1. Burgers' equations

According to the notations defined in Section 2, the one-dimensional, viscous Burgers' equation can be written as

$$\partial\varphi/\partial\tau = \partial/\partial x[N(\varphi, \varphi_x)] \,, \tag{28}$$

where

$$N(\varphi, \varphi_x) = -\tfrac{1}{2}\varphi^2 + \nu\varphi_x \,, \tag{29}$$

and ν is the viscosity coefficient. In this example $\nu = 0.07$ is used. The initial and the boundary conditions are chosen as follows:

$$\varphi(1, \tau) = 0 \,, \qquad \varphi(0, \tau) = 1 \,, \qquad \varphi(x, 0) = 1 - x \,. \tag{30}$$

The two-dimensional, nonlinear, viscous Burgers' equation solved by Ghia et al. [10] was chosen in the presented test case. According to the notations defined in Section 2, it can be expressed as

$$\partial u/\partial\tau = \partial/\partial x[N^1(u, u_x)] + \partial/\partial y[N^2(u, u_y)] \,, \tag{31}$$

where

$$N^1(u, u_x) = u_x - \lambda y(\tfrac{1}{2}u^2 - uU) \,, \qquad N^2(u, u_y) = u_y - \lambda x(\tfrac{1}{2}u^2 - uU) \,. \tag{32}$$

Here, λ is a parameter and U is a constant. The values of U and λ used in this test case are 0.5 and 2.0, respectively.

The FTCS scheme is applied to discretize (28) and (31). If the linearized form of operator N is used with M steps at each time level t, the residual polynomial is truncated up to its first order as

$$\mathrm{RP} = r^{t+1} = r^t + a_m\omega_m \,, \tag{33}$$

where

$$a_m = \partial/\partial x[-\varphi^t\delta_m + \nu(\delta_m)_x] \tag{34}$$

for the one-dimensional case and

$$a_m = \partial/\partial x[(\partial N^1/\partial u)\delta_m + (\partial N^1/\partial u_x)(\delta_m)_x] + \partial/\partial y[(\partial N^2/\partial u)\delta_m + (\partial N^2/\partial u_y)(\delta_m)_y] \tag{35}$$

for the two-dimensional case. The corrections δ_m at each intermediate step m can be determined by (23), (24). The minimizing polynomial MP for both cases is then

$$\mathrm{MP} = \|r^{t+1}\|^2 = \|r^t\|^2 + 2(r^t, a_m)\omega_m + (a_m, a_n)\omega_m\omega_n \,. \tag{36}$$

Thus, the optimal acceleration parameters ω_m can be easily determined by solving the following system of linear equations:

$$A_{mn}\omega_n = b_m \,, \tag{37}$$

where

$$A_{mn} = (a_m, a_n) \,, \tag{38}$$

$$b_m = -(r^t, a_m) \,, \tag{39}$$

and A_{mn} is a symmetric matrix of order M.

The boundary conditions of the residual and corrections in (33) were determined by extrapolating them from the interior points (Case a) or by explicitly enforcing them to be zero (Case b). The stopping criteria for all cases were such that the computations were terminated when the asymptotic rate of convergence (Case a) or the norm of the residual (Case b) approached the machine accuracy.

It must be mentioned that the norm of the truncation error represents the maximum attainable accuracy of a numerical scheme and is obviously scheme-dependent. It can be seen from Figs. 2(a) and 4(a) that under the same stopping criteria the residual norms for all cases converge to the values corresponding to the respective norms of the truncation error. Moreover, Figs. 1(a) and 2(a) illustrate that the accuracy of the nonaccelerated scheme can be improved by applying the GNLMR method.

Since the linearized operators were used in these two test cases, the convergence history shown in Figs. 1–4 exhibit a similar behavior as in the linear problems as solved in our earlier works [5]. It is obvious that if the GNLMR method is applied, the number of iterations and the computing time required to achieve the asymptotic rate of convergence are considerably lowered as compared to the nonaccelerated schemes (Figs. 1(a), 2(a), 3(a) and 4(a)). Moreover, the time required by the GNLMR method for marching the solution from the asymptotic state to a fully converged solution is much shorter than with the nonaccelerated method. The improvement of the computational efficiency that can be obtained using a different number of intermediate steps M is summarized in Fig. 9.

Although the computational efficiency increases significantly with the increasing number of intermediate steps M, the improvement becomes less pronounced and even shows a reverse trend after approximately $M = 5$ in the one-dimensional case. The reason for this unexpected result is that when using a multistep algorithm, an $M \times M$ matrix has to be inverted (directly) at each time level, t. The number of operations and computing time required for the direct inversion of a matrix grows very fast with the increase of the matrix size, thus countering the benefits of adding more intermediate steps in the multistep procedure especially for one-dimensional problems.

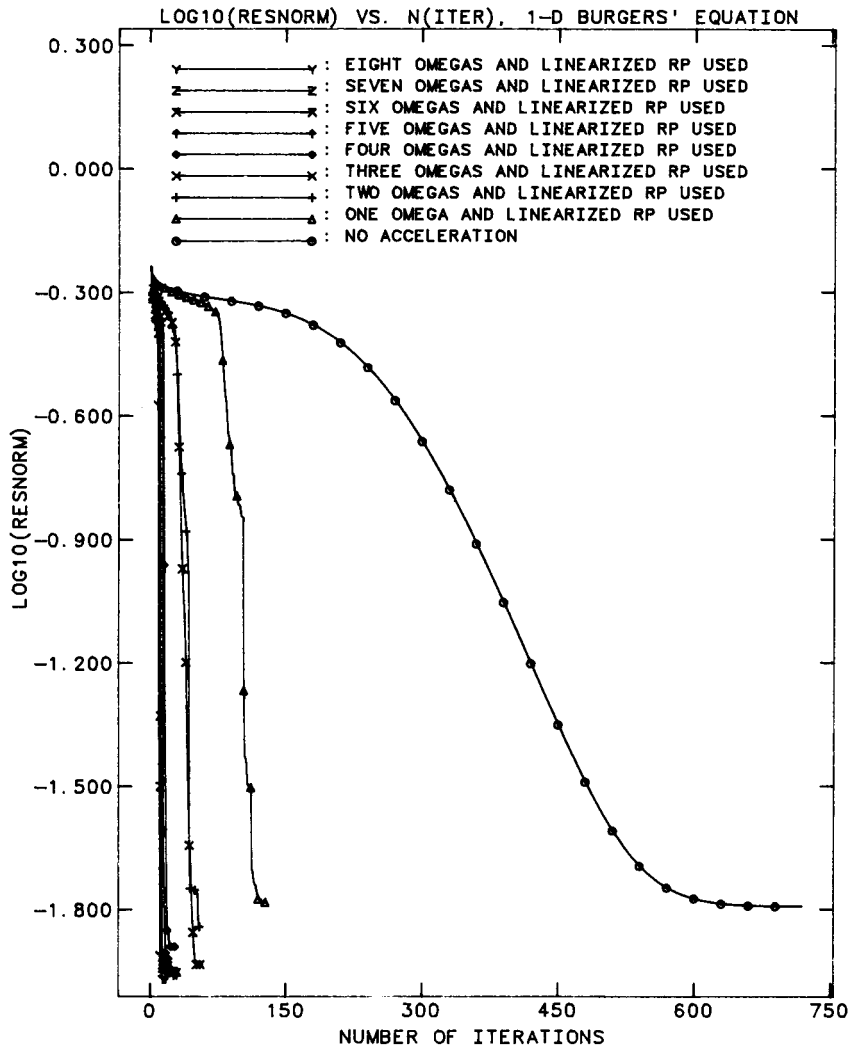

Fig. 1(a). Residual norm versus the number of iterations, 1D Burgers' equation.

3.2. *Stream-function-coordinate (SFC) equations*

The two-dimensional stream-function-coordinate (SFC) equation for an irrotational, inviscid, steady flow derived by Huang and Dulikravich [6] is given by:

$$(y_\psi^2 - \sigma)y_{xx} - 2y_x y_\psi y_{x\psi} + (1 + y_x^2)y_{\psi\psi} = 0\,, \tag{40}$$

where σ represents the compressibility and is equal to zero for incompressible flows. It is defined as

$$\sigma = (\rho^* a^*/\rho a)^2 = [\tfrac{1}{2}(\gamma + 1) - \tfrac{1}{2}(\gamma - 1)M^{*2}]^{-(\gamma+1)/(\gamma-1)}\,, \tag{41}$$

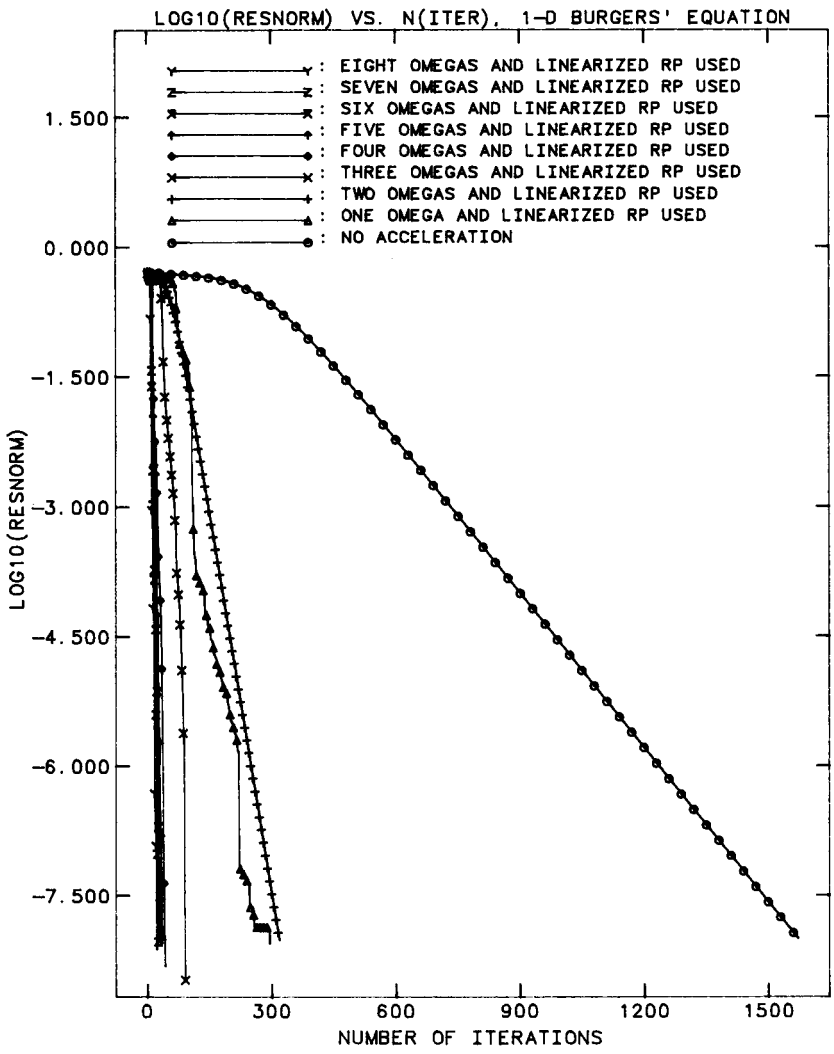

Fig. 1(b). Residual norm versus the number of iterations, 1D Burgers' equation.

where ρ and a denote the local density and the local speed of sound, respectively, and γ is the ratio of the specific heats. The superscript terms denote the characteristic quantities of the flow. It can be shown [6] that σ is an implicit function of y_x and y_ψ, that is

$$(1 + y_x^2)/y_\psi^2 = [(\gamma + 1)\sigma^{(\gamma-1)/(\gamma+1)} - 2]/[(\gamma - 1)\sigma] . \tag{42}$$

Let us define

$$\begin{aligned} &c_1 = y_x , \quad && c_2 = y_\psi , \quad && c_3 = y_{xx} , \\ &c_4 = y_{x\psi} , \quad && c_5 = y_{\psi\psi} . \end{aligned} \tag{43}$$

Then (40) can be rewritten as

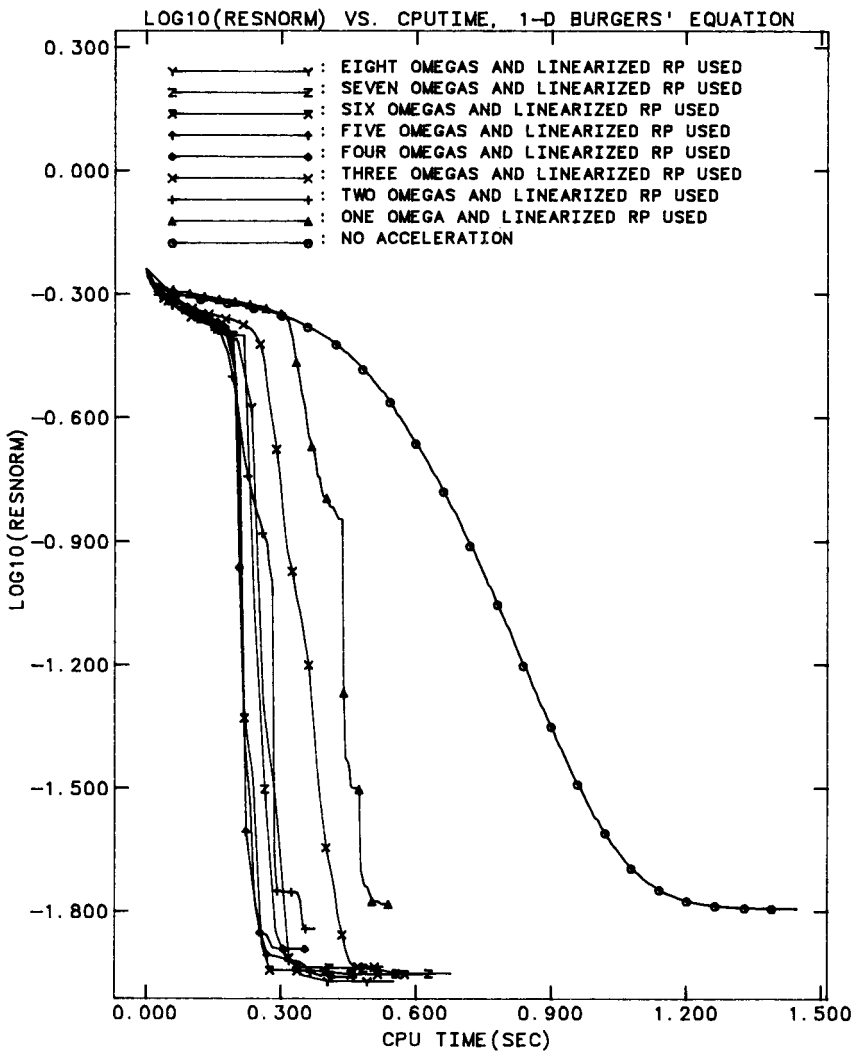

Fig. 2(a). Residual norm versus the computing time, 1D Burgers' equation.

$$N(c_1, c_2, c_3, c_4, c_5) = 0\,, \tag{44}$$

or

$$[c_2^2 - \sigma(c_1, c_2)]c_3 - 2c_1c_2c_4 + (1 + c_1^2)c_5 = 0\,. \tag{45}$$

Assume that a uniform computational grid is used (both Δx and $\Delta\psi$ are constant) and central differencing is applied to discretize all derivatives. The finite difference approximation of the SFC equation can be expressed as

$$y_{i,j} = [(y_\psi^2 - \sigma)/\Delta x^2 + (1 + y_x^2)/\Delta\psi^2]^{-1}[(y_\psi^2 - \sigma)(y_{i+1,j} + y_{i-1,j})/(2\Delta x^2) + (1 + y_x^2)(y_{i,j+1} + y_{i,j-1})/(2\Delta\psi^2) - y_x y_\psi y_{x\psi}]\,. \tag{46}$$

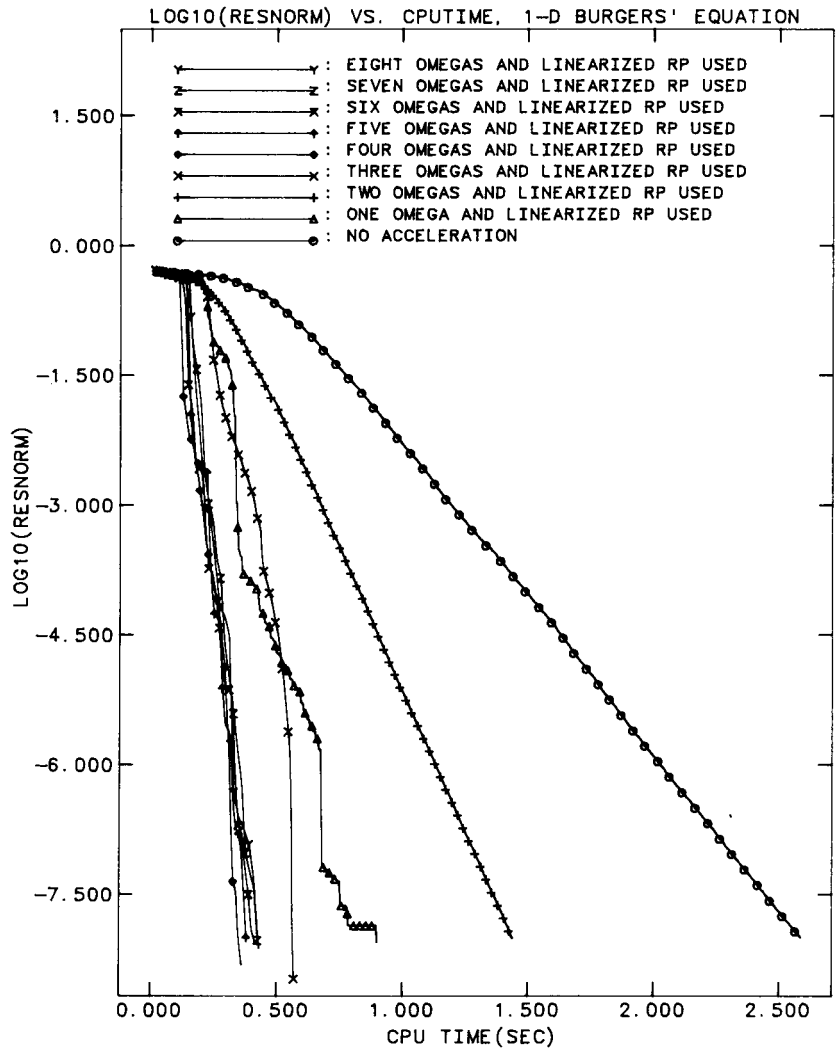

Fig. 2(b). Residual norm versus the computing time, 1D Burgers' equation.

Equation (46) will be referred to as the iterative equation. Most iterative schemes for solving (46) can be expressed as

$$y_{i,j}^{t+1} = y_{i,j}^{t} + \omega \delta_{i,j}^{t}, \tag{47}$$

where t represents the iteration level, ω is the relaxation factor, and $\delta_{i,j}^{t}$ is the correction at iteration level t. It is defined as

$$\delta_{i,j}^{t} = \bar{y}_{i,j}^{t+1} - y_{i,j}^{t}. \tag{48}$$

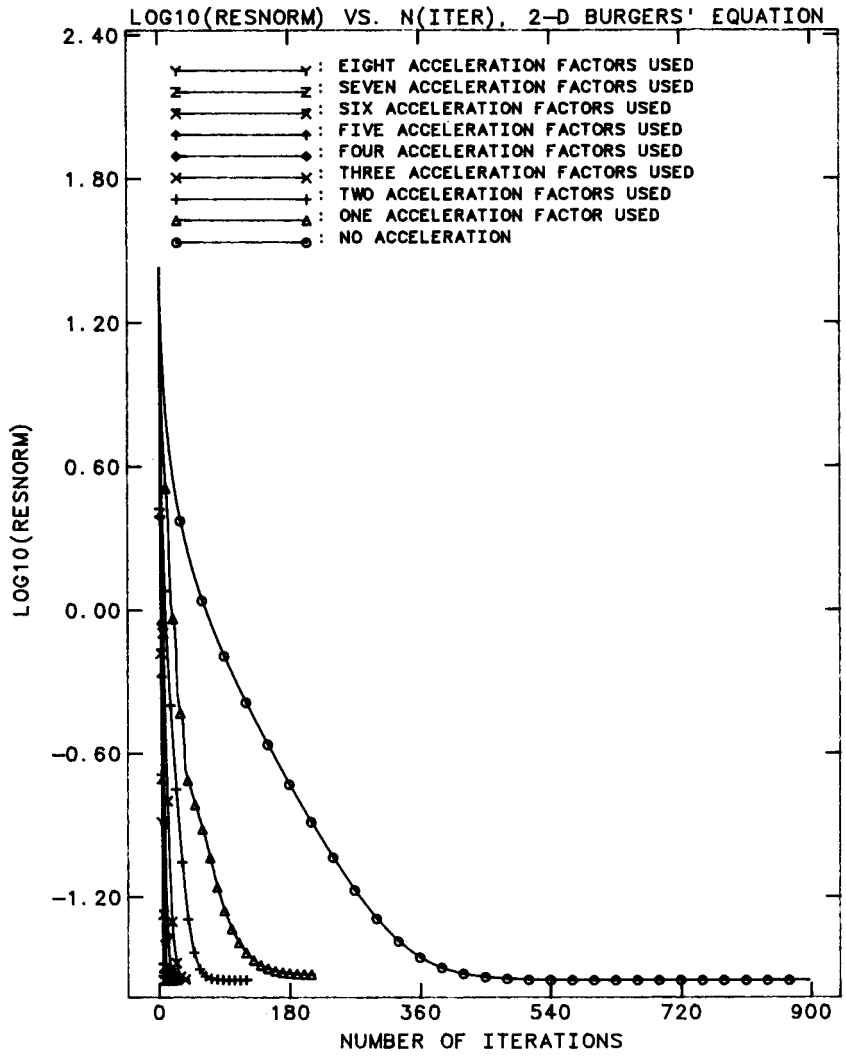

Fig. 3(a). Residual norm versus the number of iterations, 2D Burgers' equation.

Here, $\bar{y}_{i,j}^{t+1}$ is the temporary value of y at iteration level $(t+1)$ obtained by applying (46) with any fundamental iterative scheme. In the presented studies, Liebman's method ($\omega = 1$) was used as the fundamental iterative scheme and will be henceforth referred to as the nonaccelerated method. Assume that M steps are used in the GNLMR method. The solution is then updated by using

$$y_{i,j}^{t+1} = y_{i,j}^{t} + \omega_m \delta_m , \quad m = 1, 2, \ldots, M , \tag{49}$$

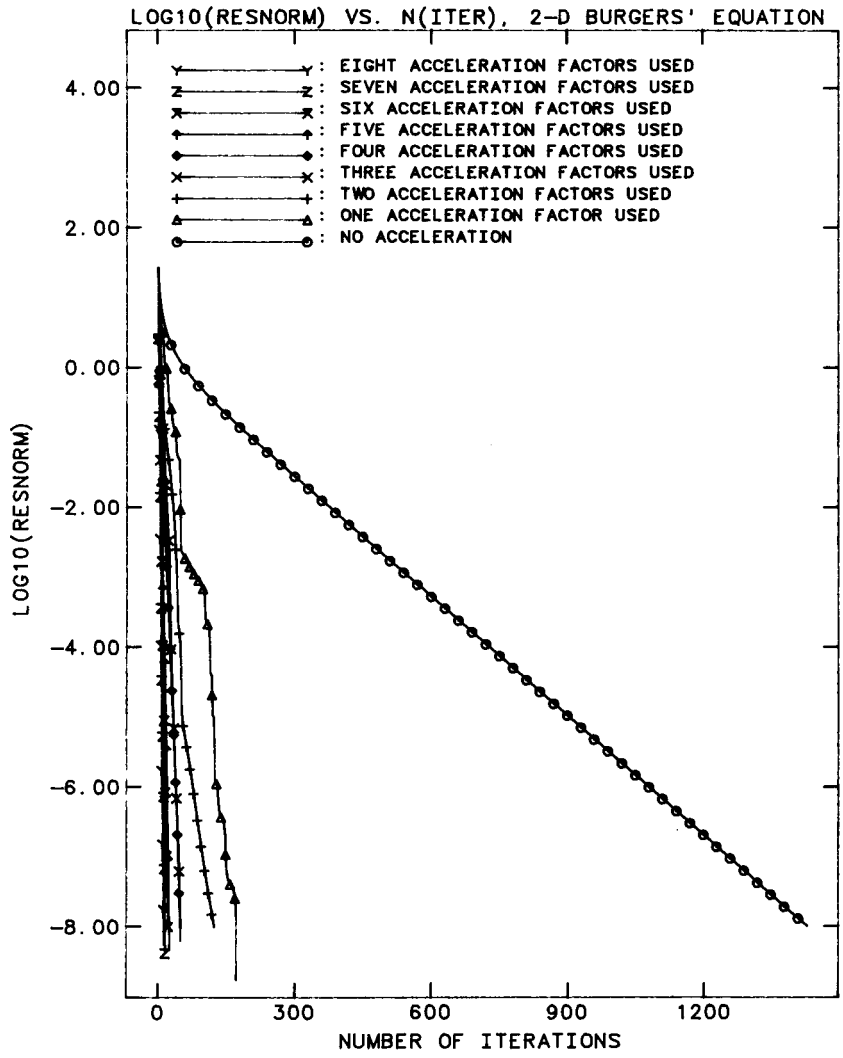

Fig. 3(b). Residual norm versus the number of iterations, 2D Burgers' equation.

where δ_m are the corrections at each intermediate step m. They are obtained by successively applying Liebman's method. The optimal values of ω_m based on a linearized RP can be determined by solving (37) with

$$r^t = N(c_1, c_2, c_3, c_4, c_5)^t , \tag{50}$$

$$a_m = [(\partial N/\partial c_1)(\delta_m)_x + (\partial N/\partial c_2)(\delta_m)_\psi + (\partial N/\partial c_3)(\delta_m)_{x\psi} + (\partial N/\partial c_4)(\delta_m)_{xx} + (\partial N/\partial c_5)(\delta_m)_{\psi\psi}] . \tag{51}$$

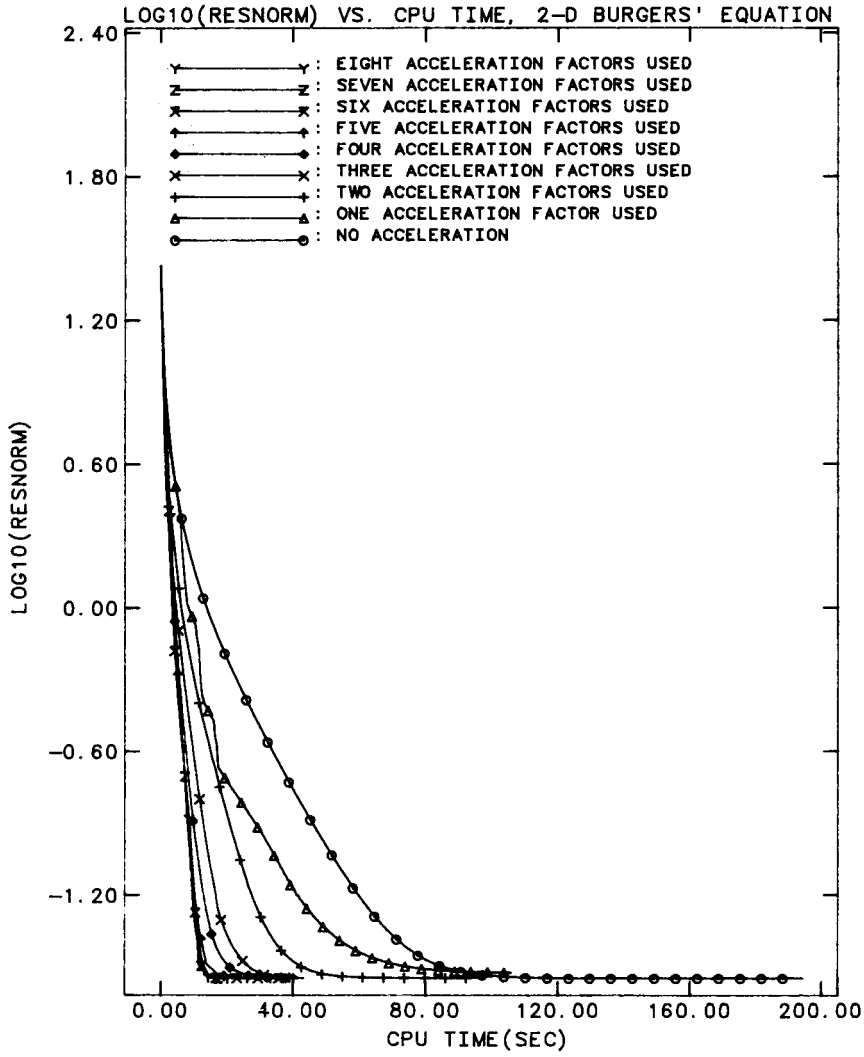

Fig. 4(a). Residual norm versus the computing time, 2D Burgers' equation.

For the incompressible case, a uniform flow around a cascade of doublets was solved [6]. For the compressible case, a subsonic flow with free-stream Mach number $M_\infty = 0.65$ around a NACA 0012 airfoil in a channel with height/chord ratio =3.6 was solved [6].

Since linearized operators were used in these two cases, and the boundary conditions for the residual and corrections in (33) were set to zero, the residual norm will converge to machine accuracy. Therefore, the stopping criteria for these two cases was chosen in such a

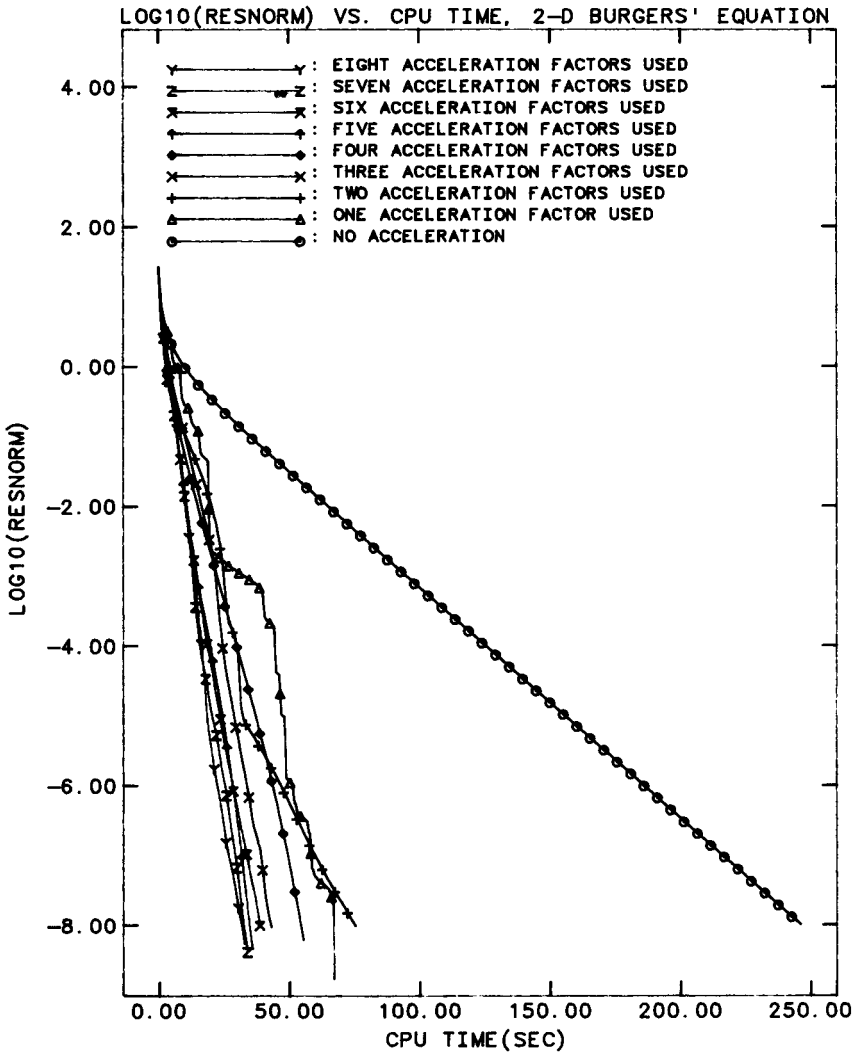

Fig. 4(b). Residual norm versus the computing time, 2D Burgers' equation.

way that as the residual norm approaches machine accuracy, the computation is forced to stop.

The improvement of the computational efficiency is summarized in Fig. 9. Both cases show that the computational efficiency is increased significantly by increasing the number of intermediate steps M.

The numerical results for these two cases are summarized in Figs. 5–8. Both cases exhibit a

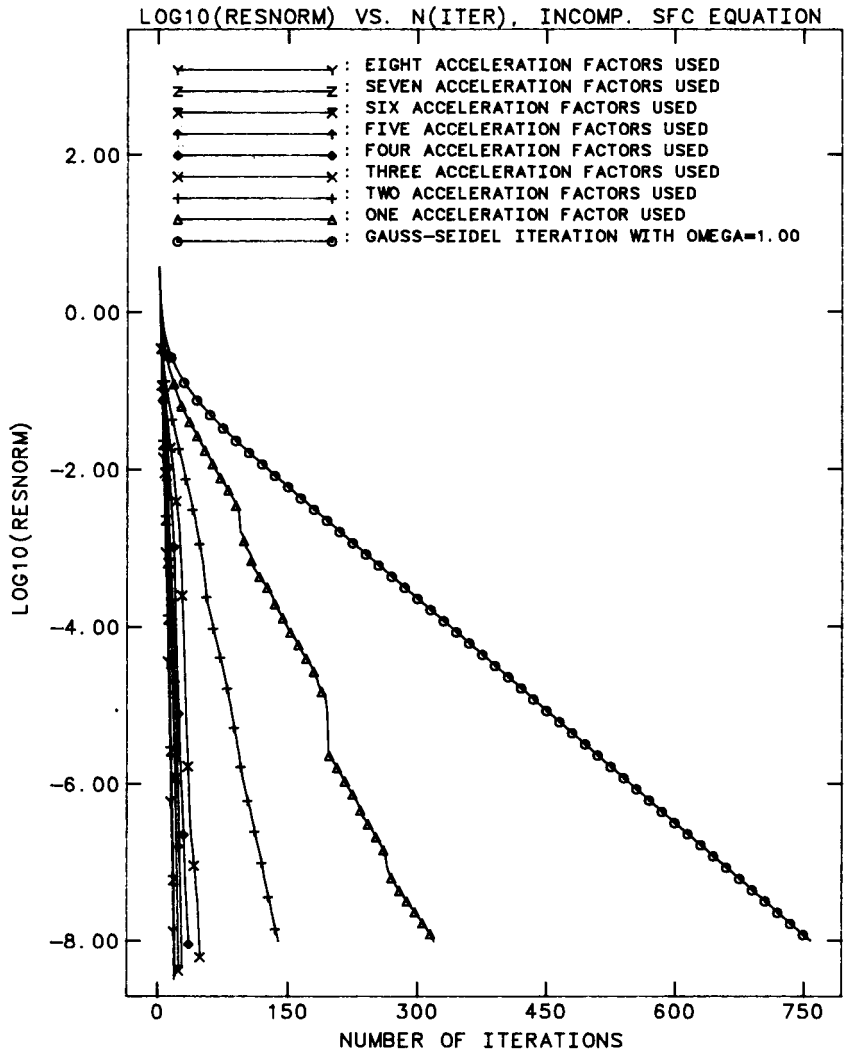

Fig. 5. Residual norm versus the number of iterations, incompressible SFC equation.

common feature that the monotonicity and the rate of convergence increase with the increase of the total number of intermediate steps M. Most importantly, Figs. 6 and 8 show that with a specified minimum computing time, the difference in the residual norms between the nonaccelerated method and the GNLMR method varies between one to eight orders of magnitude depending on the number of steps used in the GNLMR method. This fact strongly proves the computational efficiency that can be obtained using the GNLMR method [8].

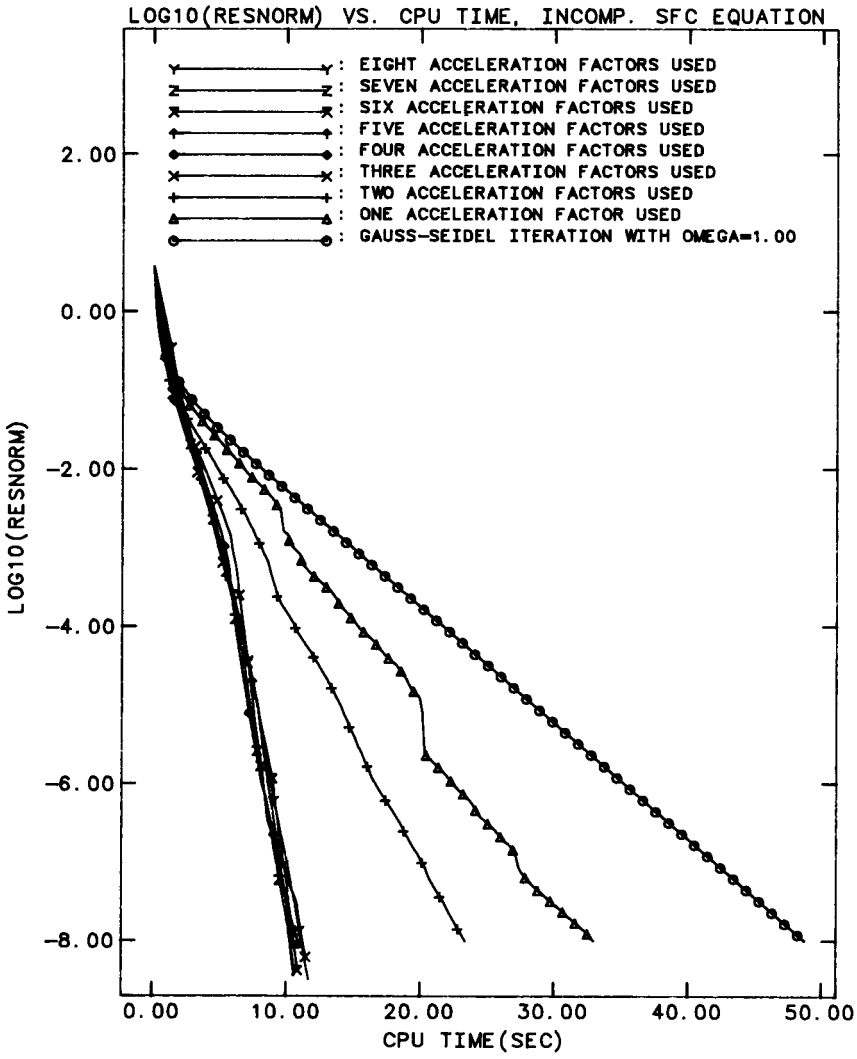

Fig. 6. Residual norm versus the computing time, incompressible SFC equation.

4. Concluding remarks

Four numerical test cases for nonlinear problems in fluid dynamics were presented to demonstrate the applicability, computational efficiency, and monotone convergence behavior of the GNLMR method. It was found that even though the theory of the GNLMR method is based on the evolution problems and equations in conservative form, the method can be applied equally successfully to the solutions of steady-state problems governed by equations in

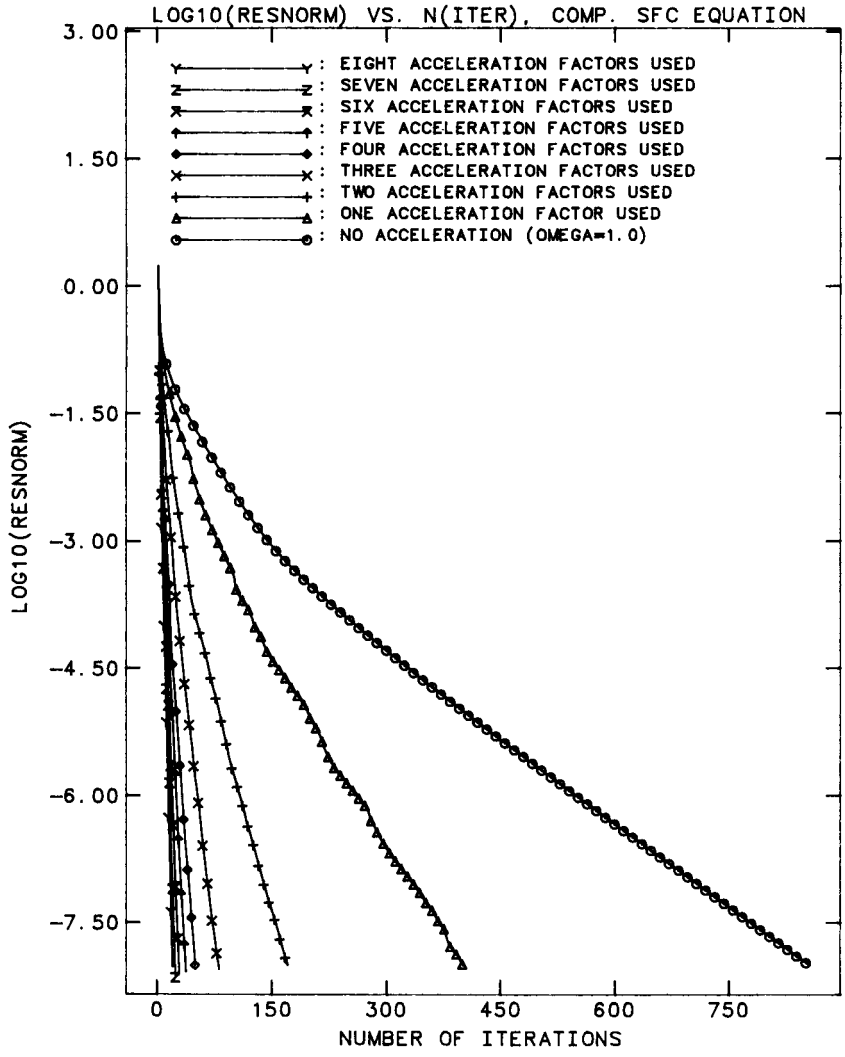

Fig. 7. Residual norm versus the number of iterations, compressible SFC equation.

nonconservative form. The results for all test cases show that when applying the GNLMR method to nonlinear problems, the number of iterations and the corresponding computer time are considerably lowered by increasing the number of intermediate time steps.

Since the explicit multistep algorithm was employed in developing the GNLMR method, the advantage of accelerating the convergence rate of the iterative process is partially offset by some extra costs. These are caused by the requirements for additional storage in order to save

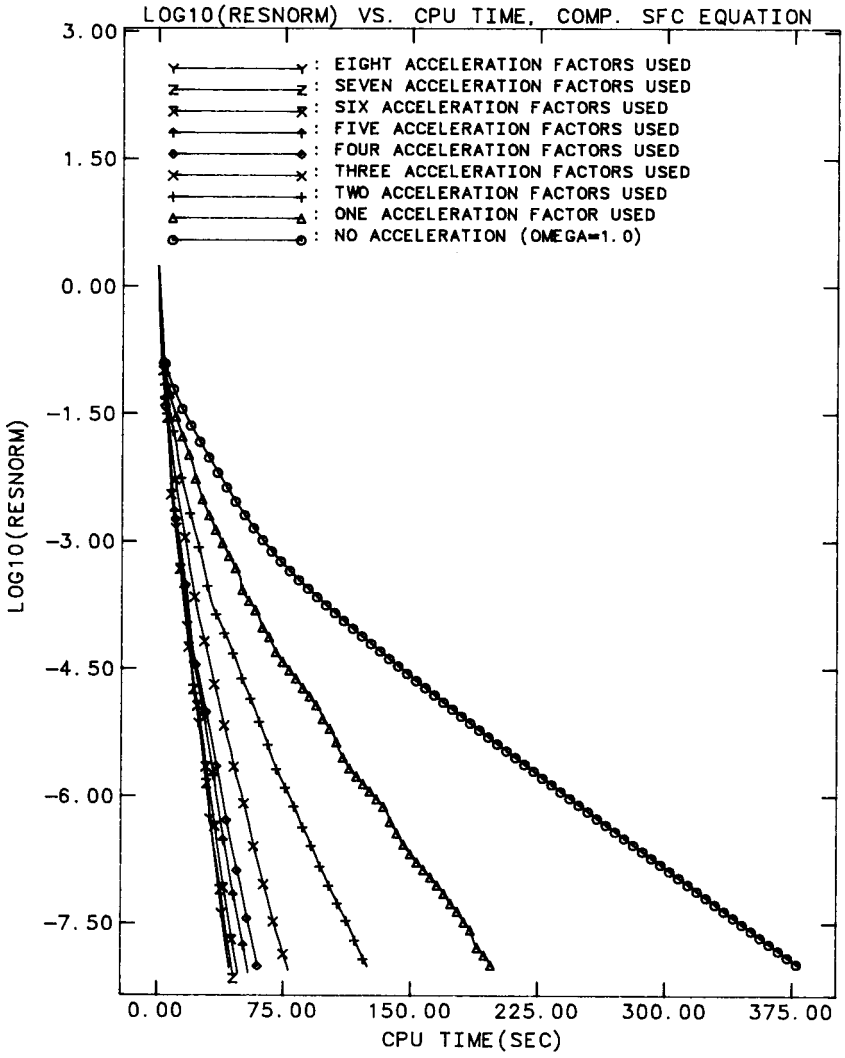

Fig. 8. Residual norm versus the computing time, compressible SFC equation.

corrections obtained from each intermediate step and by the additional arithmetic operations to determine the coefficients of the minimizing polynomial. In practice, a maximum gain in computational efficiency can be obtained with a moderate number (usually not more than five) of intermediate steps. The requirement for additional storage linearly increases with the number of intermediate time steps used and represents only a fraction of the computer storage required by the GMRES method [8].

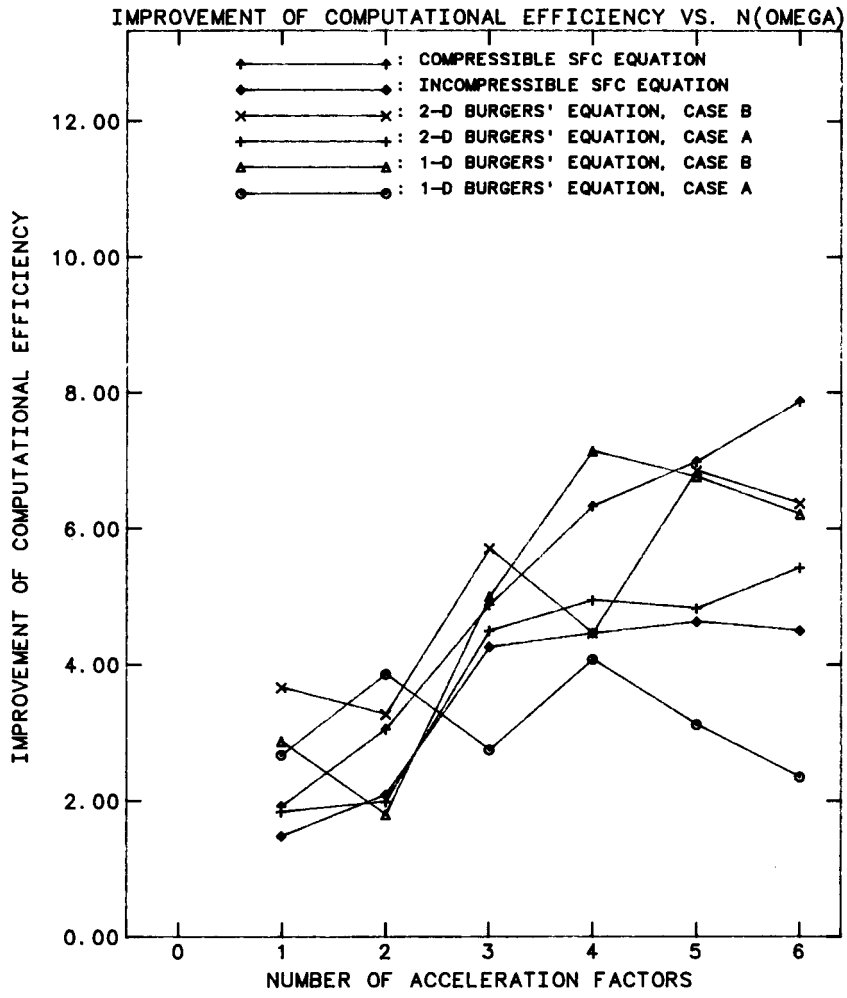

Fig. 9. Improvement of computational efficiency versus the number of acceleration factors.

Acknowledgment

This research work was funded by the grant 85-0052 of the USAFOSR/NM office under the supervision of Maj. John P. Thomas. Authors are thankful to Dr. Beam and Dr. Warming of NASA Ames Research Center for helpful discussions and to the reviewers for their constructive comments.

References

[1] A.R. Mitchell and D.F. Griffiths, The Finite Difference Method in Partial Differential Equations (Wiley, New York, 1980).

[2] S.I. Cheng, Computational Fluid Dynamics—Numerical Method for Partial Differential Equations (Scientific Publishing Company, Peking, 1984) (in Chinese).
[3] S.R. Kennon and G.S. Dulikravich, Optimal acceleration factors for iterative solution of linear and nonlinear differential systems, Comput. Meths. Appl. Mech. Engr. 47 (1984) 357–367.
[4] S.R. Kennon, Optimal acceleration factors for iterative solution of linear and nonlinear differential systems, AIAA Paper 85-0162, Reno, NV, 1985.
[5] C.-Y. Huang, S.R. Kennon and G.S. Dulikravich, Generalized non-linear minimal residual (GNLMR) method for iterative algorithms, Comput. Appl. Math. 16 (1986) 215–232.
[6] C.-Y. Huang and G.S. Dulikravich, Stream function and stream-function-coordinate (SFC) formulation for inviscid flow field calculations, Comput. Meths. Appl. Mech. Engrg. 59 (1986) 155–177.
[7] H. Ratschek and J. Rokne, Computer Methods for the Range of Functions (Ellis Horwood, Chichester, U.K., 1984).
[8] Y. Saad and M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, Research Rept. YALEU/DCS/RR-254, Yale University, New Haven, CT, 1983.
[9] L.B. Wigton, N.J. Yu and D.P. Young, GMRES acceleration of computational fluid dynamics codes, AIAA Paper 85-1494, 1985.
[10] K.N. Ghia, U. Ghia and C.T. Shin, Adaptive grid generation for flows with local high gradient regions, ASME FED-5, Houston, TX (1983) 35–47.