JOURNAL OF COMPUTATIONAL PHYSICS 111, 324-335 (1994)

On Accelerating the Convergence of Nonlinear Iterative Algorithms

D. W. Black and A. P. Rothmayer

Department of Aerospace Engineering and Engineering Mechanics, Iowa State University, Ames, Iowa 50010

Received February 8, 1993

The underlying theory of vector sequence extrapolation methods for linear and nonlinear problems is examined. It is shown that nonlinearity limits savings in total number of iterations to 50% for strongly nonlinear problems when linear-based extrapolation methods are used. In support of this conclusion, convergence behaviors of solutions of Burgers' equation and an interacting boundary layer problem are examined. Supporting evidence from the work of previous researchers is also presented. A possible means of circumventing this limitation by including nonlinear terms in the extrapolation is suggested. Results of this method for a scalar model problem and a solution to Burgers' equation are given, showing that the 50% limitation can be circumvented.

INTRODUCTION

The differenced form of the governing equations of fluid dynamics can usually be written

$$\hat{B}_{ijk}x_jx_k + \hat{A}_{ij}x_j + \hat{c}_i = 0, \tag{1}$$

where the x_i's are the variables describing velocity, pressure, temperature, etc. To solve (1), many times a point-iterative scheme of the form

$$x_i^{(n)} = g_i(x_j^{(n-1)}) \tag{2}$$

is used. Such schemes are often convergent but they tend to converge slowly. Acceleration schemes which increase this convergence rate come in many varieties. The standard methods of successive relaxation can show significant improvement in convergence rate at very little computational expense. Due to the problem dependence of the relaxation parameter, however, often times the savings are not maximized. Recently, optimal relaxation schemes such as the *distributed minimal residual* method of Lee and Dulikravich [9] have been developed that tailor relaxation parameters to the problem. This method is of the same form as the *reduced rank extrapolation* method of Eddy [6] and Mešina [11], described in detail by Sidi [14]. Thus, the optimal relaxation methods can be shown to be a subclass of vector sequence extrapolation techniques.

Vector sequences are formed by the repeated application of (2) to some initial vector, $x_i^{(0)}$. The extrapolation techniques are used to predict the converged result of the sequence based on a relatively small number of iterations. In this paper, we examine the underlying rationale of these methods, the implications of applying them to nonlinear problems, and some sample results to illustrate the effectiveness and limitations that can be expected. In particular, we will show that individual acceleration schemes can only be expected by yield 50% savings for strongly nonlinear iterative problems. Such savings have been observed by researchers attempting to accelerate strongly nonlinear systems. (See, e.g., Sidi and Celestina [13] and Cheung, Cheer, Hafez, and Flores [4]). Although this does not eliminate the usefulness of such schemes, this result suggests that the greatest savings in strongly nonlinear problems will be gained by improving the convergence rates of the base algorithm if linear-based extrapolation methods are used. We will also suggest a nonlinear extrapolation method that may allow for greater than 50% savings in convergence rate without significantly greater computational effort.

LINEAR THEORY

Vector sequence extrapolation schemes operate on the series of residual vectors

$$u_i^{(n)} = x_i^{(n+1)} - x_i^{(n)} \tag{3}$$

to predict the converged solution vector, $x_i^{(\infty)}$. To see how this is accomplished, it is necessary to understand how linear mappings converge or diverge. The point iterative scheme (2), in linear form, is

$$x_i^{(n)} = A_{ij}x_j^{(n-1)} + b_i, \tag{4}$$

where the components of A_{ij} and b_i are constant and repeated indices imply summation. The residual of the resulting vector sequence is then

$$u_i^{(n)} = A_{ij}u_j^{(n-1)}. \tag{5}$$

0021-9991/94 $6.00

So, the residual at any iteration level, n, is simply

$$u_i^{(n)} = A_{ij}^n u_j^{(0)}, \tag{6}$$

where the initial residual is a function of the initial vector of the sequence. It is important to note the notational difference between a superscript enclosed in parentheses and one that is not. A superscript in parentheses denotes the value of the quantity at that iteration. A superscript alone denotes raising the quantity to that power, or in the case of matrices, multiplying the matrix by itself that many times.

If A_{ij} is a diagonalizable matrix, then (6) can be written

$$u_i^{(n)} = T_{ij} \Lambda_{jk}^n S_{kl} u_l^{(0)}, \tag{7}$$

where the matrix T_{ij} is the matrix of right eigenvectors, S_{ij} is the inverse of T_{ij}, and Λ_{ij} is the diagonal matrix of the eigenvalues of A_{ij}:

$$\Lambda_{ij} = \begin{cases} \lambda_i & i = j \\ 0 & i \neq j. \end{cases} \tag{8}$$

These eigenvalues are ordered such that $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_N|$. Therefore, from (7), the residual can also be written in the form

$$u_i^{(n)} = E_{i1} \lambda_1^n + E_{i2} \lambda_2^n + \cdots, \tag{9}$$

or, with a summation convention,

$$u_i^{(n)} = E_{ij} \lambda_j^n, \tag{10}$$

where the components of E_{ij} are constants depending on the eigenvectors of A_{ij} and the initial vector. Since we have chosen to order the eigenvalues by decreasing magnitude, as the number of iterations increases, the number of eigenvalues that significantly contribute to the convergence behavior decreases. In fact, in the limit of $n \to \infty$, the residual will behave like

$$u_i^{(n)} = E_{i1} \lambda_1^n. \tag{11}$$

Logarithmic plots of this residual will become straight lines in the limit as the iteration number becomes large. Figure 1 illustrates this type of convergence. This is a fairly common convergence mode for both linear and nonlinear systems.

This log-linear limit behavior is actually one of three main possibilities and correspond to the behavior for λ_1 real with the modulus of all other eigenvalues less than $|\lambda_1|$. It is also possible that the second eigenvalue could be the negation of the real first eigenvalue, $\lambda_2 = -\lambda_1$. In this case, the residual will alternate between two straight lines, resulting in a saw-toothed behavior, as illustrated by Fig. 2. Finally, λ_1 and λ_2 could form a complex conjugate pair:

$$\lambda_{1,2} = |\lambda_1| \, e^{\pm i\phi}, \tag{12}$$

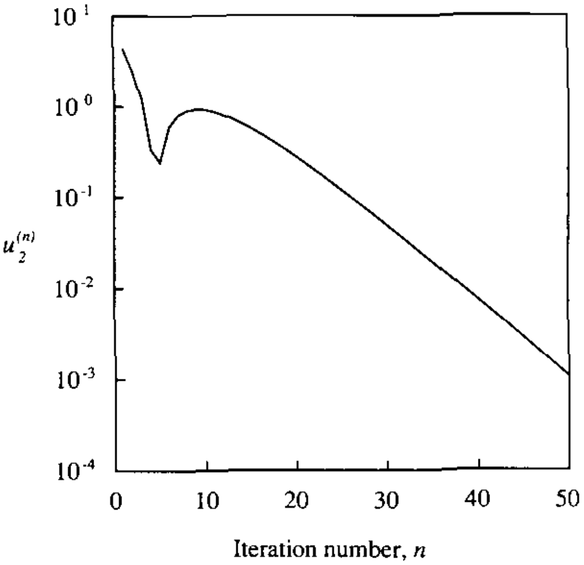

FIG. 1. Log-linear convergence.

in which case, the residuals will converge like

$$u_i^{(n)} = e_i \, |\lambda_1|^n \cos(\phi n + \omega_i). \tag{13}$$

Such convergence is of a scalloped form where the upper bound is a straight line of slope log $|\lambda_1|$. Figure 3 depicts this type of behavior. The residuals plotted in these three figures are those obtained by iterating on three different 3 × 3 linear systems in a Gauss–Seidel fashion. The coefficients were adjusted in each case so as to yield the various behaviors. It is interesting to note that the complicated behaviors seen in nonlinear systems of very high order can

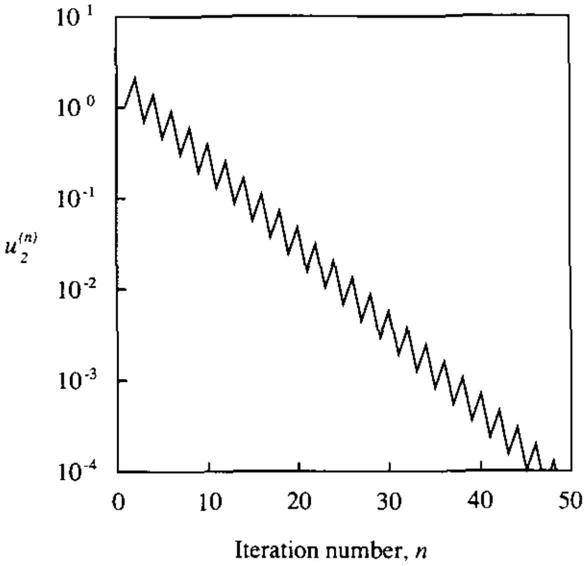

FIG. 2. Sawtooth convergence.

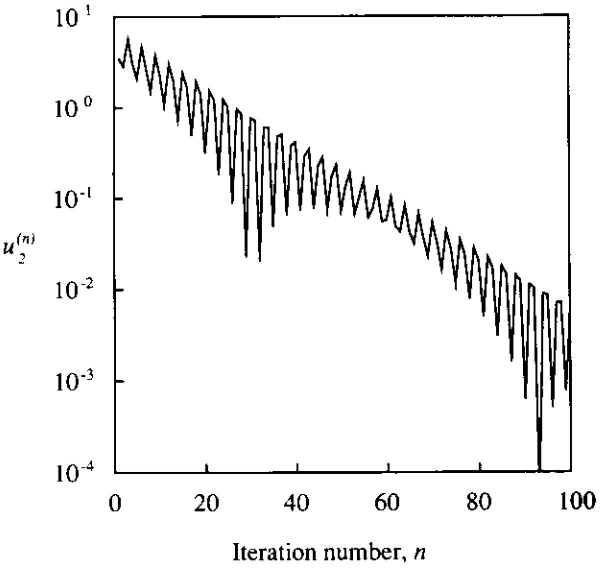

FIG. 3. Scalloped convergence.

also be seen in such simple low-order linear systems. Other behaviors, such as that arising from two sets of complex eigenvalues of equal magnitude but differing phase are possible. However, these are less likely to occur in practice.

EXTRAPOLATION METHODS

By noting that at least some point iterative schemes converge like (10) with λ_1 real, it is possible to predict the converged value of the vector sequence to first order. This is done by noting that (10) implies that

$$\lim_{n \to \infty} \frac{u_i^{(n)}}{u_i^{(n-1)}} = \lambda_1 \tag{14}$$

and that

$$x_i^{(\infty)} - x_i^{(n)} = \sum_{m=1}^{\infty} u_i^{(n+m)} = \frac{\lambda_1}{1-\lambda_1} u_i^{(n)}. \tag{15}$$

By substitution, we obtain the very simple relation:

$$x_i^{(\infty)} = \frac{(x_i^{(n-1)})^2 - x_i^{(n)} x_i^{(n-2)}}{2x_i^{(n-1)} - x_i^{(n)} - x_i^{(n-2)}}. \tag{16}$$

This is the acceleration method described by Black and Rothmayer [2]. It is also essentially the method developed as long ago as 1926 by Aitken [1] and in 1948 by Lyusternik [10].

Such first-order extrapolations do significantly accelerate solutions but they can only be applied in the log-linear convergence region. Furthermore, they are not valid without modification for saw-toothed or scalloped convergence. In order to make the extrapolation effective earlier as well as to account for cases where more than one eigenvalue contributes, higher order extrapolation techniques are needed. Several different types of such methods exist. Among these are the *minimal polynomial extrapolation* of Cabay and Jackson [3], *reduced rank extrapolation* of Eddy [6] and Mešina [11] (both described in detail by Smith, Ford, and Sidi [15]), and the *distributed residual method* described by Lee and Dulikravich [9]. In a different vein are the "dominant eigenvalue annihilation methods" such as that described by Jespersen and Bunning [8]. Although they differ in details, the overall intent is the same for all of them. All these methods hope to improve the quality of the predicted solution by fitting the residual behavior to some truncated form of (10). To illustrate, the basic methods of one of these schemes, minimal polynomial extrapolation (MPE), will be outlined.

MPE works by assuming that the converged value of the solution can be found from a weighted average of some finite number, k, of consecutive x_i's. In other words,

$$x_i^{(\infty)} = \sum_{j=1}^{k} \gamma_j x_i^{(n+j)}, \tag{17}$$

where

$$\sum_{j=1}^{k} \gamma_j = 1. \tag{18}$$

The weighting coefficients, γ_j are found by solving the overdetermined set of equations,

$$c_0 u_i^{(n)} + c_1 u_i^{(n+1)} + \cdots + c_{k-1} u_i^{(n+k-1)} = -c_k u_i^{(n+k)}, \tag{19}$$

and choosing $c_k = 1$. The other coefficients are then found by a linear least squares method such as that of Sidi [14]. If the residual can be assumed to converge like (10) truncated to the first k eigenvalues, then the c_j's and the γ_j's are related by

$$\gamma_j = \frac{c_j}{\sum_{m=1}^{k} c_m}. \tag{20}$$

NONLINEAR SYSTEMS

By going to higher orders, it should be possible to extrapolate to the final solution much earlier. Indeed, for linear systems this is the case. For instance, Sidi [14] was able to reduce iteration numbers by a factor of 10 for a liner system of dimension 1000 using an extrapolation of order 10. In fact, if the computations could be performed with infinite precision, it would be possible to extrapolate to the exact solution by letting the order of the fit, k, equal the dimension of the system, N. Of course, $N+2$ iterations would have to pass before there was enough data to conduct the fit. And at that point, it would be necessary to invert a $N \times N$ matrix to find the weighting coefficients, which is as difficult a problem as the original. However, relatively low order fits ($k = 10$–15) give astonishing savings for linear systems. It is therefore natural to attempt to apply these methods to nonlinear problems such as those of fluid dynamics. This has been done Sidi and Celestina [13] and Lee and Dulikravich [9], for example. Here the results have been good but not as remarkable. In this section, we offer an extrapolation of why that may be.

First, it is necessary to understand why extrapolation techniques developed for linear systems would be considered appropriate for nonlinear ones. If the nonlinear operator, g_i of (2) is expanded about its stationary point, the value of $x_i^{(n+1)}$ can be written

$$x_i^{(n+1)} = b_i + A_{ij} x_j^{(n)} + B_{ij} x_j^{(n)} x_k^{(n)} + \cdots, \tag{21}$$

which, when $x_i^{(n)}$ is sufficiently close to $x_i^{(\infty)}$, would appear to be approximated by

$$x_i^{(n+1)} = b_i + \bar{A}_{ij} x_j^{(n)}. \tag{22}$$

Thus, for solutions near the converged value, linear behavior is expected to result. Yet, some degree of approximation is inherent in assuming a form like (22). The amount of error incurred by approximating in this way can be assessed by examining what (21) implies for the calculated eigenvalues.

The mapping (21) implies that the residuals behave like

$$u_i^{(n)} = A_{ij} u_j^{(n-1)} + B_{ijk}(x_j^{(n)} x_k^{(n)} - x_j^{(n-1)} x_k^{(n-1)}) \tag{23}$$

or

$$u_i^{(n)} = A_{ij}^{(n)} u_j^{(n-1)}, \tag{24}$$

where the residual transition matrix, $A_{ij}^{(n)}$, is defined to be

$$A_{ij}^{(n)} = A_{ij} + B_{ijk}(2x_k^{(n-1)} + u_k^{(n-1)}) + \cdots. \tag{25}$$

Alternatively, the residual transition matrix can be defined recursively:

$$A_{ij}^{(n)} = A_{ij}^{(n-1)} + B_{ijk}(u_k^{(n-1)} + u_k^{(n-2)}) + \cdots. \tag{26}$$

This is the form the extrapolation actually sees. It is interesting to compare this form with the linear form, (5). Such comparison reveals the essential difference between linear and nonlinear vector sequences. While the residual transition matrix in linear systems is not dependent on previous residuals, in the nonlinear system this history is retained. The errors that arise in applying linear-based models to nonlinear problems stem from this essential difference.

An estimate of the error incurred by assuming linearity can be found by comparing the eigenvalues of $A_{ij}^{(n)}$ and $A_{ij}^{(n-1)}$. If the residual transition matrix is written in similar form,

$$A_{ij}^{(n-1)} = T_{ik}^{(n-1)} \Lambda_{kl}^{(n-1)} S_{lj}^{(n-1)}, \tag{27}$$

and each of the matrices is assumed to be perturbed by some small amount, $\varepsilon \ll 1$, then the transition matrix can be shown to be

$$A_{ij}^{(n)} = A_{ij}^{(n-1)} + \varepsilon C_{ij}^{(n)} + \cdots. \tag{28}$$

Since B_{ijk} is $O(1)$, in general, and the residuals converge like $|\lambda_1|^n$ in the limit, comparison of (26) and (28) yields $\varepsilon \sim O(|\lambda_1|^n)$. This implies that the stepwise behavior of the eigenvalues of the residual transition matrix can be expanded in the form

$$\lambda_i^{(n)} = \lambda_i^{(n-1)} + \zeta_i(n)\,|\lambda_1|^n + \cdots, \tag{29}$$

where $\zeta_i(n)$ are unknown functions of the iteration level that depend on the higher derivatives of the mapping. If the original nonlinear mapping is quadratic, these functions are constant. For higher order mappings, they are complicated functions of n.

What does this imply for vector sequence extrapolation? Such methods rely on the assumption that, for a kth-order fit, the residuals decay like

$$u_i^{(n)} = E_{1i}\lambda_1^n + E_{2i}\lambda_2^n + \cdots + O(|\lambda_{k+1}|^n). \tag{30}$$

We now see, however, that the first term will actually behave like

$$\begin{aligned}(\lambda_1^{(n)})^n &\sim \lambda_1^n(1 + n\zeta_i(n)\,\lambda_1^n) + \cdots \\ &\sim \lambda_1^n + \bar{\zeta}_i(n)\,\lambda_1^{2n} + \cdots.\end{aligned} \tag{31}$$

So, in general, a nonlinear algorithm can only be considered linear to the extent that the eigenvalues considered are greater in magnitude than $|\lambda_1|^2$. This presents a means of classifying how nonlinear a system is. The *order* of nonlinearity can be defined as the number of eigenvalues of the system that are greater than the square of the magnitude of the dominant eigenvalue.

The order of nonlinearity is independent of the *strength* of nonlinearity. While the order of nonlinearity determines the asymptotic behavior as the iteration approaches convergence, strength determines the degree to which nonlinearity affects the solution early in the iteration.

Throughout the derivation above, we assumed that the effect of the nonlinear terms was $O(1)$. If the nonlinear terms were instead quite small with respect to the linear ones, perhaps due to a fortuitous choice of initial conditions, we would see that while the nonlinear behavior described above would hold in the limit, early in the iteration the convergence would in fact appear linear to the extrapolation techniques. The conclusion is that while nonlinear effects exist for all nonlinear systems, they may be negligible for those problems where the nonlinearity is weak. An example of this is the low Mach number Euler solutions of Lee and Dulikravich [9]. Here the convergence of the extrapolated sequence is faster than would be predicted above. However, for the case in question, the problem is essentially a linear potential flow problem. For general nonlinear systems, it may be assumed that the nonlinearity is not weak, else a linear approximation to the solution would suffice.

As an example of the nonlinear effects discussed thus far, we examine the numerical solution of a steady form of Burger's equation

$$vv_x = v_{xx}, \tag{32}$$

on the interval $[0, 1]$ with boundary conditions $v(0) = 0$ and $v(1) = 1$. This equation has an analytical solution given by

$$v(x) = \alpha \tan\left(\frac{\alpha x}{2}\right), \tag{33}$$

where the constant, $\alpha = 1.306542374$, satisfies the boundary conditions. This equation was solved on a 101-point grid in Gauss–Seidel fashion:

$$\frac{v_{i+1}^{(n)} - 2v_i^{(n+1)} + v_{i-1}^{(n+1)}}{\Delta x^2} = v_i^{(n+1)} \frac{v_{i+1}^{(n)} - v_{i-1}^{(n+1)}}{2\,\Delta x}. \tag{34}$$

To verify the conclusions above, it is desirable to study the convergence of this numerical scheme for a range of dominant eigenvalues. To do this, the grid could be refined or the boundary conditions could be changed. Both of these alternatives would change the basic character of the solution, however. Changing the boundary conditions changes the final solution while refining the grid changes the relative strength of the nonlinearity. It was decided to affect changes in the eigenvalues by using successive relaxation at the end of each grid sweep. The relaxation took the form:

$$x_i^{(n+1)} = \omega g_i(x_j^{(n)}) + (1 - \omega)\, x_i^{(n)}. \tag{35}$$

This relaxation was done only to alter the eigenvalues, not to stabilize or accelerate the calculation. The altered eigenvalues of the relaxed form are

$$\lambda_i(\omega) = \omega\hat{\lambda}_i + (1 - \omega), \tag{36}$$

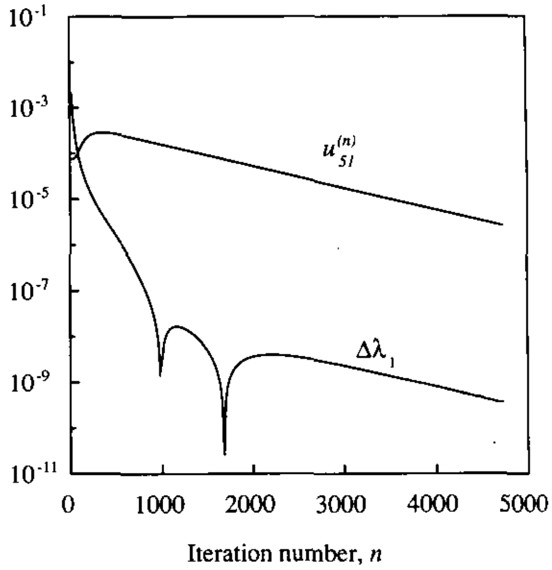

FIG. 4. Comparison of residual and eigenvalue residual of the nonlinear Burger's equation with $\omega = 1$.

where $\hat{\lambda}_i$'s are the eigenvalues for $\omega = 1$. At the end of each sweep, the dominant eigenvalue as predicted by MPE was calculated. If the nonlinear theory is correct, then stepwise changes in this dominant eigenvalue should behave like

$$\lambda_1^{(n)} - \lambda_1^{(n-1)} = \Delta\lambda_1^{(n)} \sim \lambda_1^n. \tag{37}$$

So the slope of the log plot of the change in λ_1 should be the same as that of the residual itself in the limit of large n if the system is first-order nonlinear. Figure 4 shows that this is the case for a relaxation parameter of unity. Figure 5 shows the result for a range of relaxation parameters. The quantity $(d/dn)(\log \Delta\lambda_1)$ is the asymptotic slope of $\Delta\lambda_1$ on a log plot.

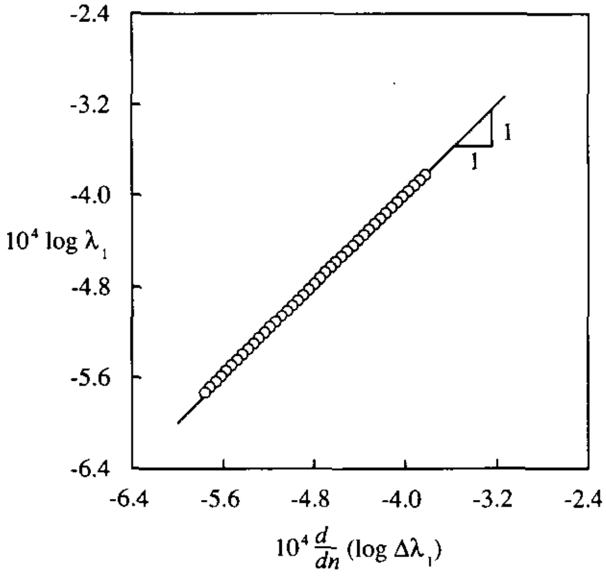

FIG. 5. Residual slope versus eigenvalue residual slope of the nonlinear Burger's equation for a range of ω's.

The eigenvalue itself was found from a first-order MPE calculation. The slope was found from linear least squares of the change in λ_1 for the last 500 iterations before machine accuracy was reached. A slope of unity in a plot like Fig. 5 indicates first-order nonlinearity from (37). The problem appears to be first-order nonlinear throughout the range of the dominant eigenvalues considered. The implication is that extrapolations of greater than first-order would not yield significantly better results.

To verify that this results from the nonlinearity of the problem, a purely linear version of the same problem was solved numerically. In this constructed linear problem, the differenced form of the derivative on the right-hand side of (34) was replaced by the *exact* value found from differentiating (33). Eigenvalues were again varied by applying the relaxation formula, (35). Since the linear and nonlinear schemes have the same converged value, it was expected that the asymptotic slopes (log λ_1) would match. However, since the residual of λ_1 should go like $(\lambda_2/\lambda_1)^n$, the eigenvalue residual slope of the purely linear problems should be steeper than that of the nonlinear one. (This will be true even though in the limit the solutions are identical!) It is clear from Fig. 6 that this is so for a relaxation parameter of unity. Indeed, Fig. 7 shows that this is the case for the entire range of eigenvalues considered. These results appear to verify that the theory of nonlinear systems outlined above is correct.

RESULTS FROM NONLINEAR FLUIDS PROBLEMS

Examples from the Literature

If this theory is correct, attempts to apply vector sequence extrapolation methods should show roughly 50% savings in run times when applied to problems of fluid mechanics. The DMR results of Lee and Dulikravich [9] agree with this prediction when the problem considered was strongly nonlinear. For the weakly nonlinear case, the savings were more on the order of 60%. The results of Sidi and Celestina [13] show strong agreement with this prediction as well. Figure 9 is a residual plot obtained by them for a 10th-order extrapolation of a solution of the Navier–Stokes equations. The straight line is added to show the slope predicted by the λ_1^2 limitation where λ_1 was found by graphical analysis

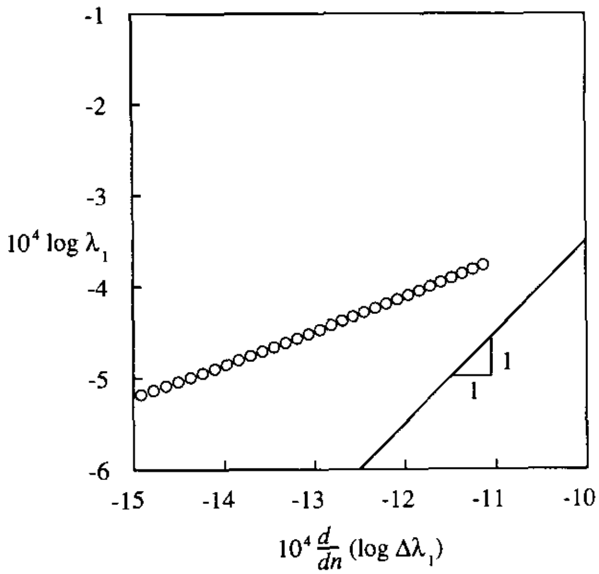

FIG. 7. Residual slope versus eigenvalue residual slope of the linear form of Burger's equation for a range of ω's.

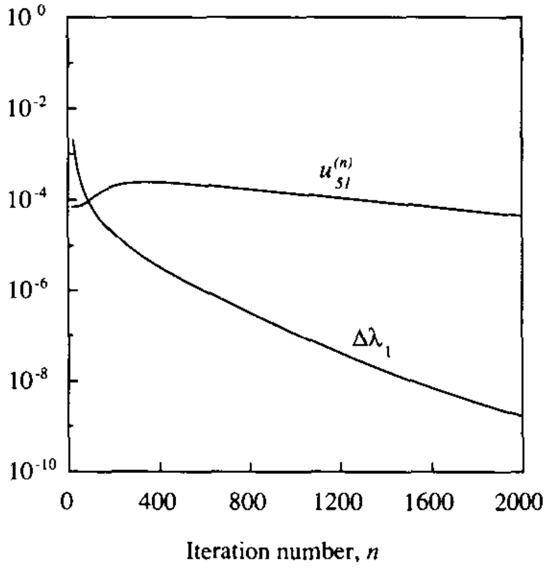

FIG. 6. Comparison of residual and eigenvalue residual of the linear form of Burger's equation with $\omega = 1$.

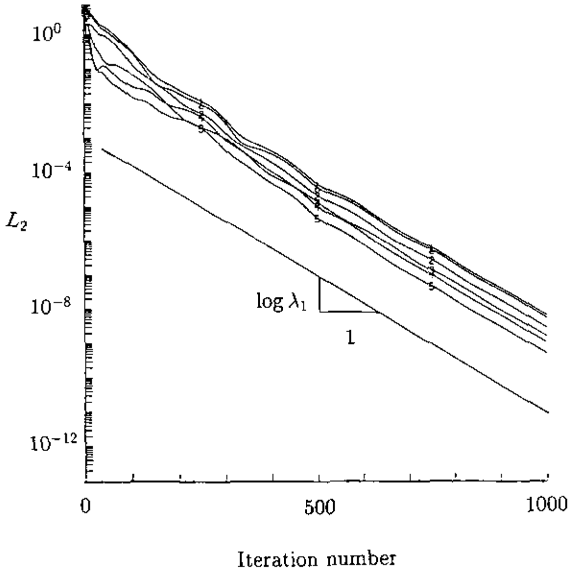

FIG. 8. Unextrapolated residuals of the Navier–Stokes solution of Sidi and Celestina.

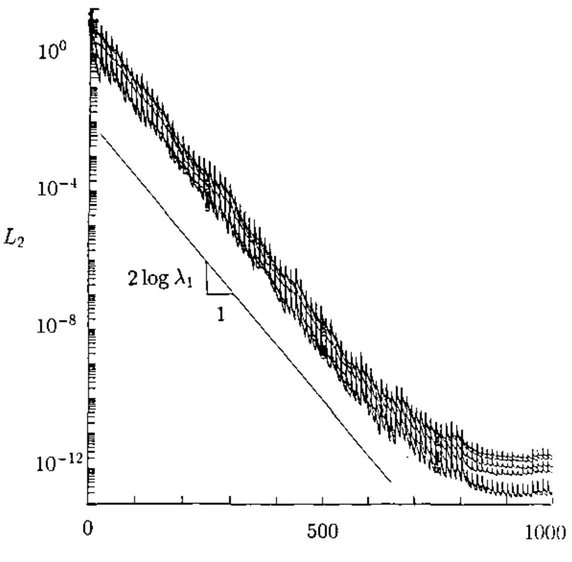

FIG. 9. Extrapolated residuals of the Navier–Stokes solution of Sidi and Celestina.

of the unaccelerated results (Fig. 8). The accelerated convergence rate agrees quite favorably with the predicted mean rate.

Interacting Boundary Layer Results

To further demonstrate the effect of nonlinearity on the acceleration of nonlinear fluid flow calculations, the results from an interacting boundary layer code (described by Black and Rothmayer [2]) were studied. Minimal polynomial extrapolation was applied to study the effect of nonlinearity.

The boundary layer calculations proceed by solving the viscous equations subject to outer boundary conditions that allow for strong viscous–inviscid interaction. The boundary layer equations are written in a baseline coordinate system that would ideally coincide with the displacement surface of the boundary layer. The similarity transformation of Görtler [7] is applied:

$$\xi = \int_0^s u_{e0}(s)\, ds, \qquad \eta = \frac{u_{e0}}{\sqrt{2\xi}} N. \tag{38}$$

In these coordinates, F and V are the velocities in the tangential and normal directions, u_{e0} is the inviscid surface speed past the baseline, and β_0 is the baseline pressure gradient parameter defined as

$$\beta_0 = \frac{2\xi}{u_{e0}} \frac{du_{e0}}{d\xi}. \tag{39}$$

In addition, the true pressure gradient parameter is defined as

$$\beta_1 = \frac{2\xi}{u_{e0}^2} u_e \frac{du_e}{d\xi}, \tag{40}$$

where u_e is the actual surface speed at the edge of the boundary layer. The continuity, tangential momentum, and normal momentum equations are (as shown by Davis and Werle [5])

$$V_\eta + F + 2\xi F_\xi = 0, \tag{41}$$

$$F_{\eta\eta} - VF_\eta + FV_\eta + (1 - \beta_0) F^2 + \beta_1 = 0, \tag{42}$$

and

$$\beta_{1\eta} = 0. \tag{43}$$

The equations are subject to inner boundary conditions of no-slip past surfaces and symmetry in wakes. The outer boundary conditions are obtained by taking the limit $\eta \to \infty$ of the continuity and tangential momentum equations and the V-matching condition (see Van Dyke [16]).

The inner and outer flows are coupled in the manner of Davis and Werle [5] using the bluff body formulation of Rothmayer [12] to model the outer flow, which for symmetric cases reduces to

$$\frac{u_e}{u_{e0}} = 1 + \frac{Re^{-1/2}}{\pi} \int_0^\infty \frac{(d/d\xi)[u_e(\delta^* - t)]\, du}{\xi - u}, \tag{44}$$

where δ^* is the scaled boundary layer displacement thickness and t is the distance from the baseline curve to the body of interest measured inward. This integral is integrated by parts, discretized, and coupled with the V-matching condition to provide for strong interaction (see Black and Rothmayer [2]).

The governing equations are linearized, differenced in second-order accurate form, and solved by repeated sweeping in the ξ-direction. At each streamwise grid line, the equations form block-tridiagonal systems that are solved by the Thomas algorithm. Separated regions are stabilized by applying the FLARE approximation.

The main case considered in the present study was the laminar flow past a 6% thickness to chord ratio NACA00xx symmetric airfoil, using the x-axis as the baseline curve. The grid contained 120 points in the streamwise direction (80 on the airfoil) and 200 in the normal direction. Reynolds number was 10^6 based on chord length. Figure 10 shows the converged, unextrapolated solution in terms of wall shear on the body and wake centerline velocity in the wake. This configuration results in a separation bubble of approximately 35% chord in extent, centered at the trailing edge.

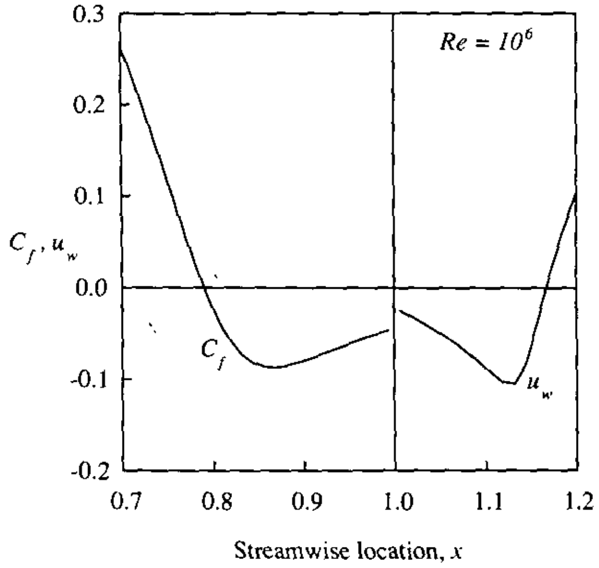

FIG. 10. Converged wall shear and centerline wake velocity for the NACA0006 airfoil.

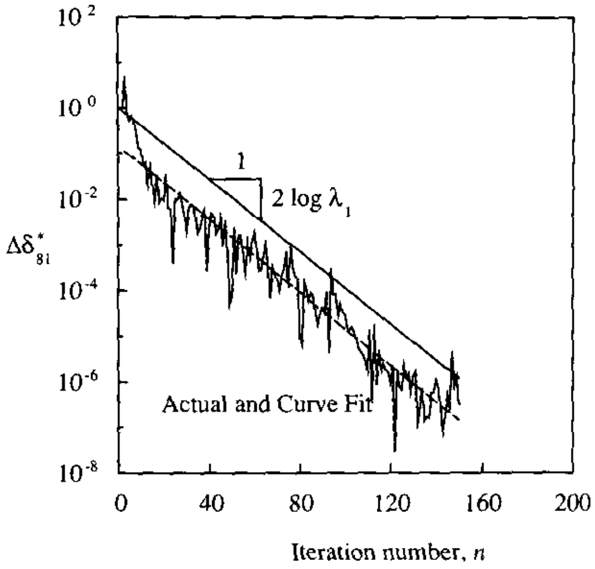

FIG. 12. Extrapolated residuals with curve fit for the first-order extrapolation.

The solution was then rerun using MPE to predict the converged value of displacement thickness. Extrapolated displacement thickness distributions were calculated after each iteration. The order of these extrapolations was varied from one to five. Figure 11 is an example of the results. In this plot, the change in the extrapolated value of δ^* at the trailing edge from one iteration to the next is shown for the first-order fit. The noise in the result makes it difficult to see the trends. Figure 12 compares a least squares exponential fit with the predicted mean line. Note the close match between the slope of the curve fit and the λ_1^2 line, implying that this case is first-order nonlinear. If this is true, the convergence rates of higher order extrapolations should be the same as that of the first-order fit. Figure 13 shows the fitted curve results for a second-order extrapolation, while Fig. 14 shows the result for a fifth-order extrapolation. Recall that in all cases extrapolated values were calculated after every iteration as soon as enough data was available to conduct the calculation.

As predicted, the mean convergence rate of these higher order methods is the same as that of the first-order method. In this case, there would be no point in using a higher order

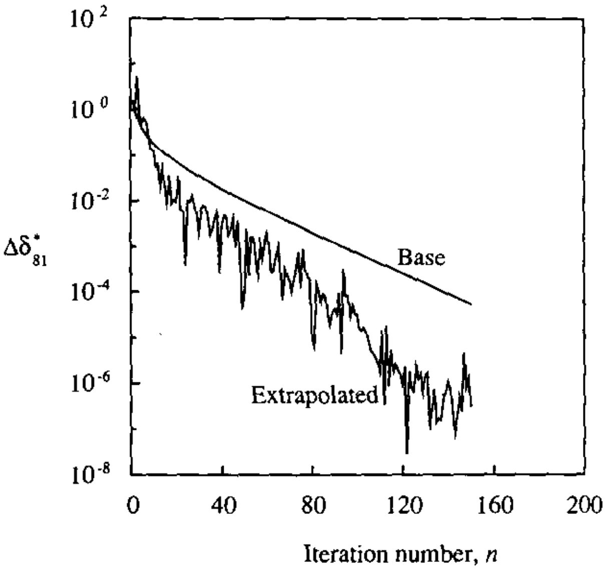

FIG. 11. Base and extrapolated residuals for the NACA0006 airfoil using first-order extrapolation.

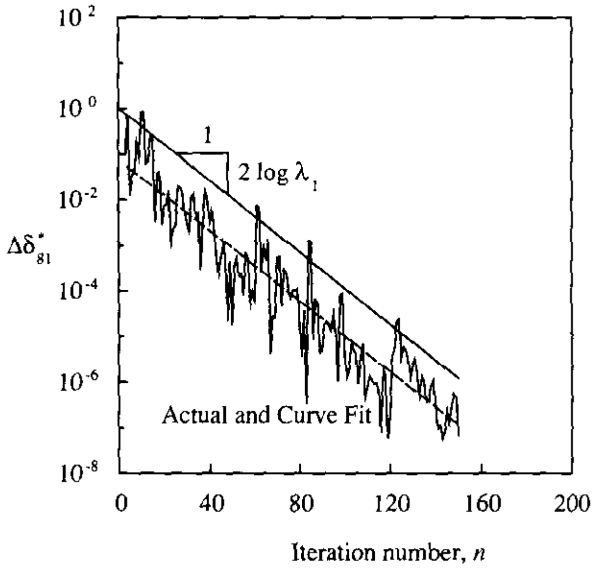

FIG. 13. Extrapolated residuals with curve fit for the second-order extrapolation.

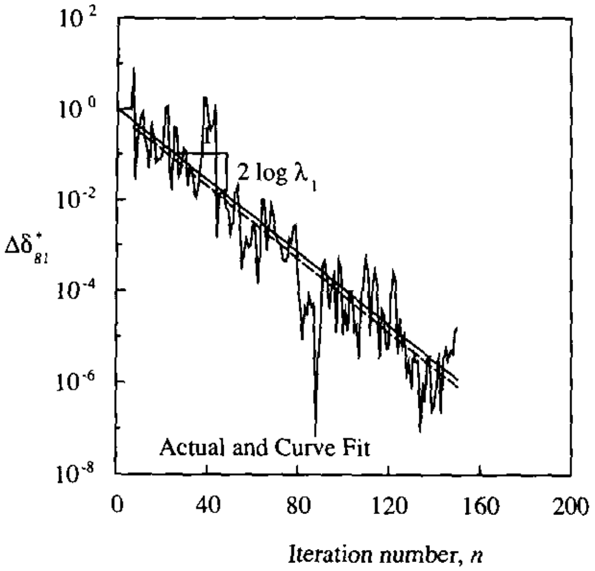

FIG. 14. Extrapolated residuals with curve fit for the fifth-order extrapolation.

method that takes a toll in computation time and storage with no improvement in convergence rate.

Questions may arise as to whether the VSE methods and limitations given here are applicable for highly clustered grids. The same calculations as above were done on clustered grids where the normal to tangential grid spacing ratios were 10:1, 500:1, and 5000:1. Normal grid spacings increased geometrically away from the wall. Figure 15 shows the results for all three grids using first-order extrapolation. All three have reached the λ_1^2 limit. It is safe to conclude that these methods will work for *some* stretched grids. However, not too much should be made of this test. It is well known that the convergence rate (and λ_1) for interacting boundary layer methods are primarily functions of how the outer flow solution is discretized. The convergence rates are practically insensitive to the grid spacing in the viscous layer. Thus, we should not be too surprised that the extrapolated convergence rates also show this insensitivity.

A NONLINEAR EXTRAPOLATION METHOD

The essential problem with current vector sequence extrapolation methods is that a linear model is used to compute the accelerated values. Such models do not correctly account for nonlinear behaviors, as seen above. Is this limitation insuperable? In this section, we hope to demonstrate otherwise. It appears that a relatively simple extension of the methods described above yields a means to circumvent the limitations seen in the linear form. The following method is presented to demonstrate that nonlinear vector sequences can be accelerated by more than 50% if the nonlinear nature of the sequence is retained. The method is not being presented as a final solution to the problem. Many questions concerning numerical sensitivity and methodology remain.

To begin, we examine the convergence behavior of a scalar, nonlinear system: the logistic map. This system is defined to be

$$x^{(n+1)} = \lambda x^{(n)}(1 - x^{(n)}). \tag{45}$$

In this case, we choose $\lambda = 0.999$. This particular form where λ is less than unity is convergent to zero. The asymptotic convergence rate is easily seen to be just λ. Figure 16 illustrates the convergence behavior of this system. After a

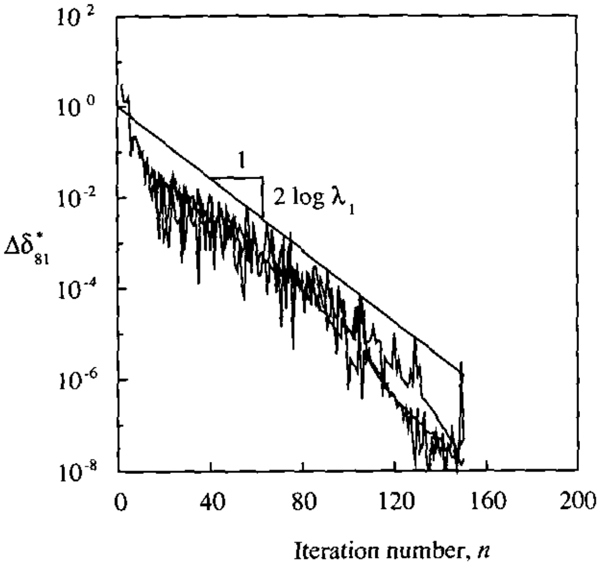

FIG. 15. Effect of severe grid stretching on extrapolating residuals.

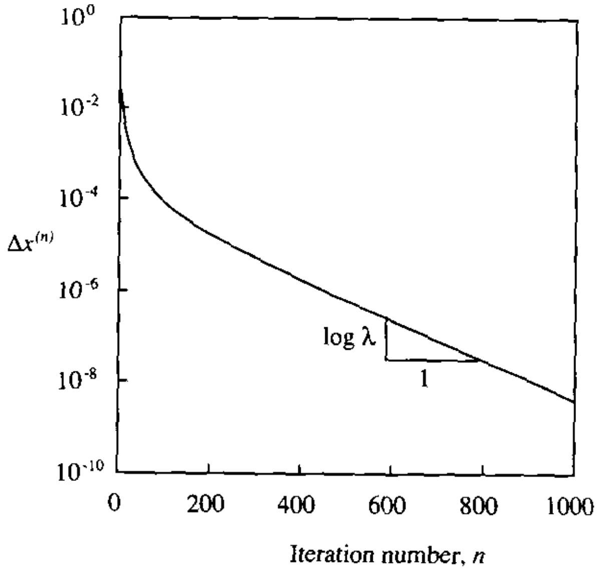

FIG. 16. Residuals of the logistic map.

smooth transient, the mapping residual takes on a log-linear form similar to that seen in Fig. 1. An analogous linear scalar mapping would not display this transient behavior. An understanding of why these systems resemble their high dimension linear counterparts in convergence mode leads to an alternative method of extrapolation that eliminates the λ_1^2 limit seen so far.

To see how this behavior can be explained, we consider the Taylor series expansion of a general nonlinear mapping of the form given in (2):

$$x^{(n+1)} = b_1 + a_{11}x^{(n)} + a_{12}(x^{(n)})^2 + \cdots. \tag{46}$$

This mapping then can be considered linear in terms of $x, x^2, x^3, \ldots$. If the solution variable at iteration level $n+1$ is augmented by the algebraic process of raising to integral powers, the mapping can be represented as

$$\begin{aligned}(x^{(n+1)})^2 &= b_2 + a_{21}x^{(n)} + a_{22}(x^{(n)})^2 + \cdots\\ (x^{(n+1)})^3 &= b_3 + a_{31}x^{(n)} + a_{32}(x^{(n)})^2 + \cdots\\ &\vdots\\ (x^{(n+1)})^m &= b_m + a_{m1}x^{(n)} + a_{m2}(x^{(n)})^2 + \cdots\\ &\vdots\end{aligned} \tag{47}$$

In general, we can say

$$y_i^{(n+1)} = b_i + A_{ij}y_j^{(n)}, \tag{48}$$

where $y_i = x^i$ (*Note.* x^i is the ith power of x) and the indices are countably infinite. A similar expansion can be done for nonlinear mappings of finite dimension.

This treatment underscores the basic difference between linear and nonlinear systems of finite dimension. Whereas linear systems of dimension N can have at most N distinct eigenvalues, nonlinear systems of the same dimension potentially have a countably infinite set of eigenvalues. The existence of these additional eigenvalues explains the transient behavior seen in Fig. 16.

With only one solution variable, only first-order extrapolation is allowable if we confine ourselves to linear terms only. This extrapolation will be mathematically equivalent to Aitken's method [1] regardless of the algorithm used. The theory thus far developed predicts that the convergence of this residual will go like λ_1^2. Figure 17 shows that this is indeed the case.

The arguments used to explain the transient behavior in the base sequence can now be used to develop an extrapolation technique that takes better account of the nonlinearity. In applying a first-order extrapolation, we are assuming that the sequence generator is approximated by the mapping

$$x^{(n+1)} = K_0 + K_1 x^{(n)}. \tag{49}$$

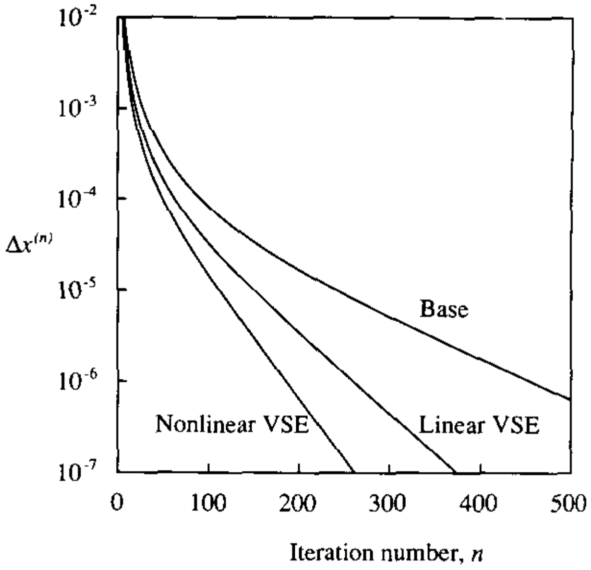

FIG. 17. Residuals of the logistic map from linear and nonlinear extrapolation.

We could also develop a kth order extrapolation that assumes

$$y_i^{(n+1)} = b_i + A_{ij}y_j^{(n)} \tag{50}$$

with $i, j = 1, k$, by assuming that the residuals, $w_i = (x^{(n+1)})^i - (x^{(n)})^i$, of the y vector converge in the same way as their linear counterparts. We can predict the limit of this sequence in much the same way as for linear cases. In particular, we will assume that if we write the residual sequence in the form

$$c_0 w_i^{(n)} + c_1 w_i^{(n+1)} + \cdots + c_k w_i^{(n+k)} = -w^{(n+k+1)}, \tag{51}$$

we can predict the limit of the first component of the y vector (x itself) by

$$x^{(\infty)} = \gamma_i x^{(n+i)}, \tag{52}$$

where the weighting coefficients are defined as

$$\gamma_i = \frac{c_i}{\sum_{j=0}^{k+1} c_j} \tag{53}$$

with $c_{k+1} = 1$. This is essentially the form assumed for MPE. The main difference is that the number of components of the residuals of y_i can be selected to be the same as the order of the fit, resulting in a system that is not overdetermined, as is the case for MPE. Thus, simple matrix inversion is sufficient to find the weighting coefficients for the extrapolated value of x.

Using a second-order version of this method, where the residuals of x and x^2 are used to find the weighting coefficients, we see that the accelerated sequence converges at a much faster rate than the previous λ^2 limitation would allow. In fact, for this scalar case, the asymptotic convergence rate appears to go like λ^3 (Fig. 17). The implication is that wile using this method will result in significant savings over linear-based methods, it may not be desirable to implement much higher order extrapolations. If the accelerated convergence rate goes like λ^k, we would quickly reach a point where incremental increases in k would not result in very substantial increases in convergence rate.

We have seen that the method described above can accelerate a nonlinear scalar sequence at more than twice the convergence rate of the base sequence. We will now show that the method, with suitable modification, can accelerate a nonlinear vector sequence with similar savings. We again consider the nonlinear mapping to be a linear mapping of infinite dimension (Eq. (48)). However, now the vector y includes not only terms like x_i^m but also products of powers of the terms in the solution vector. If it is necessary to include all such terms in a nonlinear vector sequence extrapolation method, the size of the problem will almost immediately become prohibitive. To circumvent this growth in size, we will assume that we can extrapolate an individual component of the solution vector using only the residuals of that component and its powers. This is similar to using Aitken's method for a linear system (treating each solution vector component as a scalar variable), even though it is known that the components are generated in a coupled fashion. It is also akin to Jespersen and Bunning's [8] use of projections of the full vector to find dominant eigenvalues for linear systems.

With these assumptions, it is possible to apply the same method as used for the scalar problem separately for each component of a vector sequence. This was done for the Burgers' equation problem discussed earlier. Due to the very slow convergence of the original problem, the equation with the same boundary conditions as before was solved on a 26-point grid. Figure 18 shows the residual of the base sequence at the center point of the grid. Figure 19 shows the residuals again at the center point using standard MPE and the nonlinear method described above. Nonlinear extrapolations were second order (residuals of v and v^2 were used to find weighting coefficients for v). Again, we see that the linear extrapolation shows a 50% savings in convergence rate, while the nonlinear method shows substantially better savings. Numerical sensitivity problems ensued rather early in the nonlinear calculation but not before the limit was essentially reached.

No great effort was made to improve the numerical properties of this method, the objective being primarily to demonstrate that the limitation of linear-based methods can be overcome through the consideration of the true nonlinear nature of the vector sequence generator. It is worth noting that both the nonlinear and linear extrapolations show savings in CPU time as compared to the base algorithm. (See Fig. 20.) For this model problem, this is not surprising since the extrapolations in both cases represent less than 10% of the total computational effort. In realistic problems, it is possible that the acceleration techniques may take as much time per iteration as the base algorithm. In this case, no real savings (and sometimes even penalties) in CPU time will be realized. The solution in such situations is to apply the acceleration infrequently. The mean

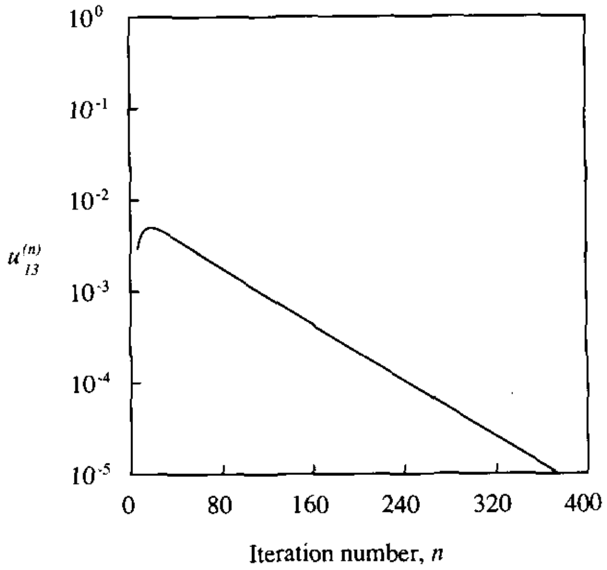

FIG. 18. Residuals of Burgers' equation solution on a 26-point grid.

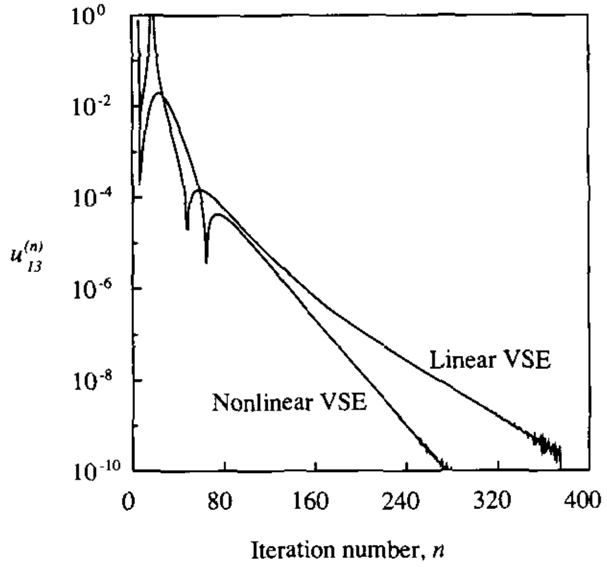

FIG. 19. Residuals of the 26-point grid Burgers' equation using linear and nonlinear extrapolation.

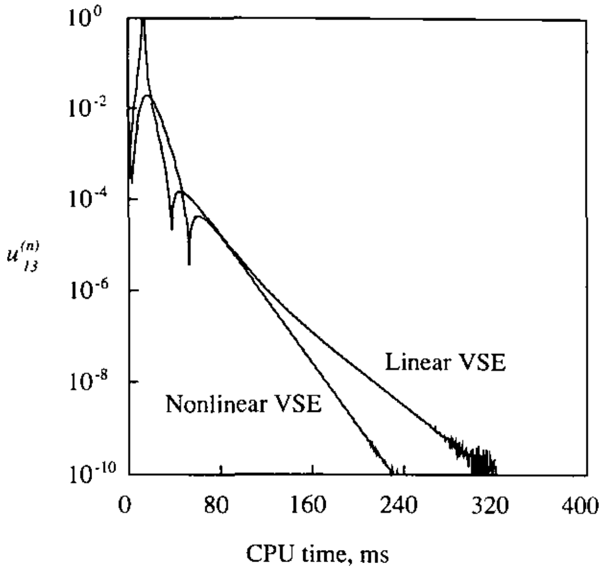

FIG. 20. CPU time histories for the linear and nonlinear extrapolations of the 26-point Burgers' equation solution.

convergence rate is relatively unaffected by applying the acceleration scheme every 10–20 iterations as opposed to applying at the end of each global iteration.

CONCLUSION

We have seen that fundamental limitations exist in the use of linear-based extrapolation methods applied to nonlinear systems. In particular, it seems that while convergence rates of the extrapolated sequences are limited only by numerical accuracy for linear systems, they will be limited to twice the base convergence rate for most nonlinear systems of interest. In itself, this does not make such methods useless; 50% savings in run times are, after all, quite significant.

The dependence of the extrapolated convergence rate on the convergence rate of the base algorithms highlights the necessity of continuing to improve the base numerical methods used in the solution of nonlinear problems (i.e., decrease λ_1). This dependence on λ_1 is present in both the linear models used thus far and in the suggested nonlinear alternative method. At this time, it would appear that only through improved understanding of nonlinear systems can the performance of such systems be improved by orders of magnitude.

Continued development of the nonlinear extrapolation technique will hinge on questions of the importance of coupling of variables in nonlinear systems. If it appears that for some nonlinear systems, individual variables can be treated as scalars, a method similar to that described above can be applied without modification. Still, many problems concerning the best numerical way to solve such systems must be resolved. If, on the other hand, the entire solution vector must be considered, the rapid growth in the size of the problem would probably make such methods impractical.

ACKNOWLEDGMENTS

This work was supported in part by grants from the National Science Foundation and United Technologies Research Center. The authors would like to thank Dr. M. Hafez for his helpful discussion and suggestions. We would also like to thank the referees for bringing the work of Jespersen and Bunning as another example of linear-based eigenvalue extrapolation techniques to our attention.

REFERENCES

1. A. C. Aitken, *Proc. R. Soc. Edinburgh* **46**, 289 (1926).
2. D. W. Black and A. P. Rothmayer, "Computation of Trailing-Edge Separation Using an Accelerated, Updated Bluff-Body Interacting Boundary Layer Algorithm," in *Separated Flows and Jets* (Springer-Verlag, Berlin, 1991).
3. S. Cabay and L. W. Jackson, *SIAM J. Numer. Anal.* **13**, 734 (1976).
4. S. Cheung, A. Cheer, M. Hafez, and J. Flores, *AIAA J.* **29**, 1214 (1990).
5. R. T. Davis and M. J. Werle, "Progress on Interacting Boundary Layer Computations at High Reynolds Numbers," in *Numerical and Physical Aspects of Aerodynamic Flows* (Springer-Verlag, Berlin, 1982).
6. R. P. Eddy, "Extrapolating to the Limit of a Vector Sequence," in *Information Linkage between Applied Mathematics and Industry* (Academic Press, New York, 1979).
7. H. Görtler, *J. Math. Mech.* **6**, 1 (1957).
8. D. C. Jespersen and P. G. Bunning, *SIAM J. Sci. Statist. Comput.* **6**, 639 (1985).
9. S. Lee and G. Dulikravich, *AIAA J.* **28**, 939 (1990).
10. L. A. Lyusternik, *Trudy Mat. Inst. Steklov.* **20**, 49 (1947).
11. M. Mešina, *Comput. Methods Appl. Mech. Eng.* **10**, 165 (1977).
12. A. P. Rothmayer, "A new Interacting Boundary Layer Formulation for Flow Past Bluff Bodies," in *Boundary Layer Separation* (Springer-Verlag, Berlin, 1987).
13. A. Sidi and M. L. Celestina, NASA Technical Memorandum 101327, August 1988 (unpublished).
14. A. Sidi, TECHNION Technical Report No. 620, April 1990 (unpublished).
15. D. A. Smith, W. F. Ford, and A. Sidi, *SIAM Rev.* **29**, 199 (1987).
16. M. Van Dyke, *Perturbation Methods in Fluid Mechanics* (Parabolic Press, Stanford, 1975).