ITERATIVE SOLUTION OF LINEAR EQUATIONS IN ODE CODES

by

C. W. Gear
Y. Saad

MASTER

January 1981

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

UIUCDCS-R-81-1054

ITERATIVE SOLUTION OF LINEAR EQUATIONS IN ODE CODES

by

C. W. Gear

Y. Saad

January 1981

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

Supported in part by the U.S. Department of Energy, Grant DE-AS02-76ER02383.A003.

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

fcy

1. Introduction

The solution of the large sparse nonlinear equations arising at each step of stiff ODE methods constitutes an important part of the cost of integrators. The use of Newton's method transforms each nonlinear system into a series of linear systems, $Ax = b$, of equations where A is a shift of the Jacobian matrix. These systems are very often sparse nonsymmetric and they are usually solved by direct methods. Little has been done so far on treating these linear systems by iterative methods. The symmetric case has been recently considered by Miranker and Chern [6] using conjugate gradient methods. Iterative techniques are advantageous in many cases, more particularly when the number of equations is very large and when the Jacobian is not easily computable. Although the use of iterative techniques might seem costly at first, there are several reasons why it is not so. The first reason is that the solutions of the linear systems need not be very accurate. In fact, the aim of the corrector step is to attempt to gain stability by eliminating the components of the error vector, corresponding to the large eigenvalues of the Jacobian, that has been amplified by the predictor.

This last observation raises the problem of choosing an iterative technique more suitable for stiff ODE methods. More precisely, the question is: which iterative techniques possess the property of being more accurate in the eigenspace corresponding to the large eigenvalues?

The class of the projection methods on the Krylov subspaces $K_m = \text{span}[r_0, Ar_0, \dots, A^{m-1}r_0]$ possesses the above property. The best example of a projection method on the Krylov subspace is the conjugate gradient algorithm for symmetric positive definite matrices. The method can be extended to unsymmetric matrices by requiring that the residual vectors form an orthogonal sequence, a property which is known to be true in the symmetric case.

In the context of stiff ODEs, the projection method must be used at each step. In each projection process a basis of the Krylov subspace is generated along with a representation of (the restriction of) the matrix A in this subspace K_m . Here we obtain further simplifications in regard to the integration of stiff ODEs:

- (i) The basis of the Krylov subspace and the $m \times m$ representation of A need not be generated at each integration step; once they are computed, they can be saved and used in several subsequent steps.
- (ii) When the stepsize is changed by the integrator the shift of the Jacobian matrix changes, but this does not change the Krylov subspace so it need not be recomputed, only the factorization of the $m \times m$ representation of A in the subspace must be recomputed and, as will be seen, this is not costly.
- (iii) There is no need to store the Jacobian in any fashion, all is needed is to be able to compute $y = Jx$ for any given x and this

may be easily realized (see Section 2.2).

2. The Krylov Subspace Method for Solving Linear Systems

2.1. The Krylov subspace projection method

This subsection summarizes what has been fully developed in the earlier paper [7]. Consider the system

$$Ax = b \quad (2.1)$$

where A is $N \times N$ (nonsymmetric). Suppose that $V_m = [v_1, \dots, v_m]$ is an orthonormal system in $\mathbb{C}^N$ and let x_0 be an initial guess of the solution. By definition, a projection method on the subspace $K_m = \text{span}[V_m]$ is a method that obtains a solution of the form $x_m = x_0 + V_m y_m$ such that

$$V_m^T (b - Ax_m) = 0 \quad (2.2)$$

therefore

$$x_m = x_0 + V_m (V_m^T A V_m)^{-1} V_m^T r_0 \quad (2.3)$$

where $r_0 = b - Ax_0$ is the initial residual. Note that the $m \times m$ matrix $H_m = V_m^T A V_m$ is the representation of the linear application $\Pi_m A|_{K_m}$ where Π_m is the orthogonal projector onto K_m .

Of particular interest is the Krylov subspace projection method where K_m is the Krylov subspace $K_m = \text{span}[r_0, Ar_0, \dots, A^{m-1}r_0]$. The following algorithm, described first by Arnoldi for the solution of eigenvalue problems, provides a simple way of building an orthogonal

basis $v_1, \dots, v_m$ of K_m along with the representation $H_m = V_m^T A V_m$:

Algorithm

1. Compute $r_0 = b - Ax_0$ and take $v_1 = r_0 / \|r_0\|$
2. For $j = 1, 2, \dots, m$ do

$$\hat{v}_{j+1} = Av_j - \sum_{i=1}^j h_{ij} v_i \quad (2.4)$$

where

$$h_{ij} = (Av_j, v_i) \quad (2.5)$$

and

$$v_{j+1} = \hat{v}_{j+1} / h_{j+1,j}$$

where

$$h_{j+1,j} = \|\hat{v}_{j+1}\| \quad (2.6)$$

An important fact is that the matrix $V_m^T A V_m$ is the $m \times m$ upper Hessenberg matrix whose nonzero elements are given by (2.5) and (2.6). Furthermore the vector $V_m^T r_0$ in (2.3) is equal to $\|r_0\| e_1$ where $e_1 = (1, 0, 0, \dots, 0)^T$.

The approximate solution x_m is therefore easy to compute and (2.3) becomes

$$x_m = x_0 + \|r_0\| V_m H_m^{-1} e_1 \quad (2.7)$$

The j -th step of stage 2 in the above algorithm represents a Schmidt orthogonalization of Av_j against all the previous vectors $v_1, v_2, \dots, v_j$. When j increases, this becomes intolerably time consuming and above all requires too much memory. A natural idea to escape this difficulty is to orthogonalize Av_j against the p previous vectors where p is some small integer. The algorithm which is derived from this, produces a sequence of vectors $v_1, v_2, \dots, v_m$ which is no longer orthonormal but satisfies the incomplete orthogonality property

$$(v_i, v_j) = 0 \text{ for } |i-j| < p \quad (2.8)$$

The first algorithm will be referred to as Arnoldi's algorithm and the second as the Incomplete Orthogonalization Method. The only difference between the two algorithms lies in the equation (2.4) which for IOM becomes

$$\hat{v}_{j+1} = Av_j - \sum_{i=1}^{i_j} h_{ij} v_i \quad (2.9)$$

where $i_j = \max\{1, j-p\}$

It is important to mention that although all vectors v_i are needed for forming the approximate solution x_m by formula (2.7), only $p+1$ of them are necessary at step j to form v_{j+1} . This means that one may store the vectors v_j in secondary storage as soon as they are computed, and read them back at the final stage when the combination $x_m = x_0 + V_m y_m$ is formed (with $y_m = \|r_0\| H_m^{-1} e_1$).

Thus far nothing has been said about the way to choose the two parameters m and p . The dimension m of K_m must be, theoretically, equal to N in which case the approximate solution x_m is exactly $A^{-1}b$. However a good accuracy is achieved for m for less than N . In practice an upper bound $m_{\max}$ is fixed a priori and the residual norms are estimated periodically by using the formula (see [7]):

$$\|b - Ax_m\| = h_{m+1,m} |y_m^T e_m| \quad (2.10)$$

As soon as the residual norm estimate is small enough or the dimension $m_{\max}$ is reached, the algorithm stops. After that one can restart by taking x_0 equal to the last approximate solution provided. In stiff ODEs, however, we shall use the algorithm without restarting. The choice of the parameter p , the number of vectors against which the vector Av_j is orthogonalized at each step, is more difficult. The optimal p for a given problem is related, in a nonobvious way, to the degree of symmetry of A . When A is symmetric then with $p = 2$, the algorithm is equivalent to the conjugate gradient method. Also, when A is almost symmetric, the choice $p = 2$ will be the best possible. In any case, p should not exceed 10 or 15 and should be taken smaller if A is nearly symmetric and larger if not. Note that there is also a possibility of reducing p during the integration by examining the matrix H_m : p should be such that the h_{ij} 's, $i < j-p$ can be considered small in comparison with the h_{ij} 's for $j-p < i < j$.

2.2. Application to stiff ODEs

Let $y' = f(t,y)$ be the system of differential equations to be solved, and let J denote the Jacobian of f relative to y . In all the methods for integrating stiff ODEs one encounters at every step an implicit equation of the form

$$y_{n+1} - \beta_0 h f(t, y_{n+1}) = \psi_n \quad (2.11)$$

The use of Newton's method for solving (2.11) gives rise to a sequence of linear problems

$$(I - h\beta_0 J)x_{n+1}^{(i)} = b_{n+1}^{(i)}, \quad i = 1, 2, \dots \quad (2.12)$$

To simplify the notations we shall denote by b the right hand side and by x the solution.

The matrix J appearing in (2.12) might be difficult to compute and above all to store. However, a look at the algorithm of Section 2.1 shows that the matrix $A = I - h\beta_0 J$ is not really needed explicitly. All is needed instead is to be able to compute the vector $u = Av$ for any given v , and this can be achieved through the formula

$$Jv \approx \frac{1}{\epsilon} [f(t, y + \epsilon v) - f(t, y)]$$

which will necessitate only one function evaluation if $f(t,y)$ is already available from the previous computations.

Suppose that at a certain integration step t_n we use the Krylov subspace method for the system $A\delta = b$ where $A = I - \beta_0 h J$. This means that we generate a sequence of vectors $V_m = [v_1, v_2, \dots, v_m]$ and a Hessenberg matrix H_m , representing A in the subspace $K_m = \text{span}[V_m]$, and that we solve the problem $Ax = b$ at the step t_n by formula (2.7). The important question is whether we can use the same subspace K_m and the representation H_m to solve the following problems arising at steps $t_{n+1}, t_{n+2}, \dots$. Assume for simplicity that the Jacobian is not time dependent and let the new system have the form

$$A'x = b' \quad (2.13)$$

where

$$A' = I - h'\beta'_0 J \quad (2.14)$$

Note that the stepsize h' and the coefficient β'_0 of the method can be very different from h and β_0 . According to (2.3), if we apply Arnoldi's algorithm on K_m to solve the new system (2.13), we obtain the approximate solution

$$x_m = x_0 + V_m (V_m^T A' V_m)^{-1} V_m^T r_0 \quad (2.15)$$

where $r_0 = b' - A'x_0$ is the initial residual (x_0 is a initial guess to the solution of (2.13)). The matrix $V_m^T A' V_m$ may, at first, look difficult to form, but it turns out that it can be easily expressed in terms of the Hessenberg matrix H_m generated at step t . Let us consider $(\beta_0 h / \beta'_0 h') V_m^T A' V_m$ instead of $V_m^T A' V_m$. If we set $\alpha = \beta_0 h / \beta'_0 h'$ we have

$$\begin{aligned}
 \alpha V_m^T A' V_m &= \alpha V_m^T [I - \beta_0' h' J] V_m \\
 &= \alpha I - \beta_0 h V_m^T J V_m J V_m = \alpha I - \beta_0 h V_m^T J V_m \\
 &= V_m^T (I - \beta_0 h J) V_m + (\alpha - 1) I
 \end{aligned}$$

since $V_m^T V_m = I$. Finally,

$$\alpha V_m^T A' V_m = H_m + (\alpha - 1) I \quad (2.16)$$

and (2.15) becomes

$$x_m = x_0 + \alpha V_m (H_m + (\alpha - 1) I)^{-1} \tilde{q}_m \quad (2.17)$$

where

$$\tilde{q}_m = V_m^T r_0 \quad (2.18)$$

A few remarks are in order. First, notice that no function evaluation is required to build the approximate solution x_m . In fact, all that is needed is to compute $\alpha V_m^T r_0$ to solve an $m \times m$ system having a Hessenberg form, and to accumulate $V_m y_m$ with $y_m = \alpha (H_m + (\alpha - 1) I)^{-1} \tilde{q}_m$. Secondly, although the derivation of (2.17) assumes that V_m is an orthonormal system, this will not be true in the case of the IOM method. However, it can be shown that in the general case the formula (2.17) is still valid if we replace the vector $\tilde{q}_m$ by the m -dimensional vector $\tilde{q}_m$ containing the components of the orthogonal projection of r_0 onto K_m , that is, $\tilde{q}_m = (V_m^T V_m)^{-1} V_m^T r_0$, and replace H_m by $(V_m^T V_m)^{-1} V_m^T (I - \beta_0 h J) V_m$. However, it would be needlessly expensive to compute $\tilde{q}_m$ since we only need a rough solution to (2.13) and we cannot ensure that the exact $\tilde{q}_m$

would provide a better solution anyway. This raises another question: can we improve the efficiency by replacing $\tilde{q}_m$ by something simpler to compute? Practically, $V_m^T r_0$ is not the best possible coefficient vector because the system is not orthogonal. However, if we notice that at the earlier step t , when V_m has been generated, the residual r_0 is, apart from a multiplicative constant, equal to v_1 , we may think to replace $V_m^T r_0$ by $v_1^T r_0 e_1$, or even better, to take $\tilde{q}_m = \sum_{i=1}^{\ell} (r_0 v_i) e_i$ where ℓ is some integer of the same order as p , e.g. $\ell = \max(5, p+1)$.

3. The Problem of Stability

3.1. Accuracy of the solution in the dominant subspace

3.1.1. Basic property

An important property of the Krylov subspace method is that it provides an approximate solution $\tilde{x}$ to $Ax = b$, which is more accurate along the eigenvectors of A corresponding to the eigenvalues lying in the outermost part of the spectrum

Figure 1. The shaded region indicates the part of the spectrum where the solution is accurate.

In the case of stiff equations we are particularly interested in the dominant subspace which corresponds to the right part of the shaded spectrum in Figure 1 since $A = I - h\beta_0 J$.

Suppose that the basic algorithm of Arnoldi is used to solve the system $Ax = b$ and let x_m be the approximate solution obtained at the m -th step. Let us denote by Π_m the orthogonal projector onto the subspace K_m , and by v_i , an eigenvector of A associated with λ_i . It is known that the distance $\|(I - \Pi_m)u_i\|$ between u_i and the subspace K_m decreases rapidly to zero with m , for the eigenvectors u_i associated with the eigenvalues situated in the outermost part of the spectrum (see [8]). The following lemma gives an explanation of the above mentioned property.

Lemma 1. The residual $r_m = b - Ax_m$ is such that

$$|(r_m, u_i)| \leq \|r_m\| \|(I - \Pi_m)u_i\| \quad (3.1)$$

Proof. $(r_m, u_i) = (r_m, u_i - \Pi_m u_i)$ because $\Pi_m u_i \in K_m$ and the residual r_m is orthogonal to K_m . The result (3.1) follows immediately. Q.E.D.

3.1.2. Symmetric case

We shall first consider the consequence of Lemma 1 when A is a symmetric matrix.

Proposition 1. Let $\lambda_1 > \lambda_2 > \dots > \lambda_N$ be the eigenvalues of A , and let θ_1 be the acute angle between the initial vector v_1 and the eigenvector u_1 . Let

$$\gamma_1 = 1 + 2 \frac{\lambda_1 - \lambda_{i+1}}{\lambda_{i+1} - \lambda_N} ; \quad K_1 = \prod_{j=1}^{i-1} \frac{\lambda_j - \lambda_N}{\lambda_j - \lambda_1} \quad (K_1 = 1)$$

Then

$$|(r_m, u_1)| \leq \|r_m\| \frac{K_1}{T_{m-1}(\gamma_1)} \tan \theta_1 \quad (3.2)$$

where T_k represents the k -th degree Chebyshev polynomial of the first kind.

Proof. The result follows from (3.1) and an inequality for $\|(I - \Pi_m)u_1\|$ established in [8]. Q.E.D.

It can be seen that in general $K_1/T_{m-1}(\gamma_1)$ is very small when λ_1 belongs to either extremity of the spectrum and becomes larger as λ_1 is in the middle of the spectrum.

Since we have $\|r_m\| = \left| \sum_{i=1}^N (r_m, u_i)^2 \right|^{1/2}$, the above proposition means that for λ_1 belonging to the outermost part of the spectrum, the component (r_m, u_1) of the residual along the eigenvector u_1 is much smaller than the average $\|r_m\|$.

It is natural to ask whether a similar result holds for the error vector $(x^* - x_m)$ where x^* is the exact solution of $Ax = b$. The answer

is given by the following corollary.

Corollary 1. The error vector $x^* - x_m$ is such that

$$|(x^* - x_m, u_i)| \leq \frac{\|A\|}{|\lambda_i|} \|x^* - x_m\| \frac{K_i}{T_{m-1}(\lambda_i)} \tan \theta_i \quad (3.3)$$

Proof. Noticing that $r_m = A(x^* - x_m)$ gives with (3.2)

$$|(A(x^* - x_m), u_i)| \leq \|A(x^* - x_m)\| K_i \tan \theta_i / T_{m-1}(\gamma_i)$$

$$|(x^* - x_m, Au_i)| \leq \|A\| \|x^* - x_m\| K_i \tan \theta_i / T_{m-1}(\gamma_i)$$

But $Au_i = \lambda_i u_i$, which gives the desired inequality. Q.E.D.

Note that when A is positive definite then the coefficient $\|A\|/|\lambda_i|$ is just λ_1/λ_i .

The above result can be interpreted in the same way as for the residual vector.

3.1.3. Unsymmetric case

When A is nonsymmetric, the exploitation of Lemma 1 is more complicated because the distance $\|(I - \Pi_m)u_i\|$ is more difficult to estimate. A result as simple as (3.2) is not available in the general case where A admits complex eigenvalues. However, it can still be shown, in a way similar to that used in [8], that $\|(I - \Pi_m)u_i\|$ decreases rapidly to zero in general, thus showing that we have essentially the same phenomenon as above, namely that the components of the residual corresponding to the outermost part of the spectrum will be damped out

more rapidly than the others.

In the particular case where all the eigenvalues of A are real and simple, a result similar to that of the symmetric case can still be shown. Suppose that $\lambda_1 > \lambda_2, \dots, > \lambda_N$ are the eigenvalues of A and let

$v_1 = \sum_{i=1}^N \alpha_i u_i$. Then we can formulate the following.

Proposition 2. Suppose that $\alpha_i \neq 0$ and let $\beta_1 = \sum_{j=i+1}^N |\alpha_j / \alpha_i|$. Let γ_i and K_i be defined as in Proposition 1. Then the residual vector r_m satisfies

$$|(r_m, u_1)| \leq \|r_m\| \frac{K_1}{T_{m-1}(\gamma_1)} \beta_1 \quad (3.4)$$

Proof. Inequality (3.4) is a consequence of Lemma 1 and the following inequality that can be shown by arguments very similar to those used in [8]

$$\|(I - \Pi_m)u_1\| \frac{K_1}{T_{m-1}(\gamma_1)} \beta_1$$

Q.E.D.

The above inequality does not show, however, that the component of the residual or of the error is small. Suppose $u_1, u_2, \dots, u_\ell$ are eigenvectors corresponding to the largest eigenvalues and let

$r_m = \sum_{i=1}^{\ell} \gamma_i u_i$ be the orthogonal projection of r_m onto the invariant subspace $\text{span}[u_1, u_2, \dots, u_\ell]$, that is the γ_i 's, $i = 1, 2, \dots, m$,

minimize the norm

$$\|r_m - \sum_{i=1}^{\ell} \gamma_i u_i\|$$

If we set $U_{\ell} = [u_1, u_2, \dots, u_{\ell}]$, $g = (\gamma_1, \gamma_2, \dots, \gamma_{\ell})^H$ then we have $g = (U_{\ell}^H U_{\ell})^{-1} U_{\ell}^H r_m$ and hence

$$\|g\| \leq \tau \|U_{\ell}^H r_m\| \quad (3.5)$$

where τ is the norm of the inverse of the Gramian of the system U_{ℓ} . Therefore, the components γ_i , $i=1, \ell$ of the residual in the subspace $\text{span}[U_{\ell}]$ will be small because the inner products $U_{\ell}^H r_m$ are small in general as inequality (3.4) shows. This obviously depends on the norm of $[U_{\ell}^H U_{\ell}]^{-1}$ which is also equal to the inverse of the smallest singular value of the system U_{ℓ} .

3.2. The problem of stability in presence of an approximate inverse of the Jacobian

3.2.1. The quasi-Newton corrector

A P(EC) predictor-quasi-Newton-corrector method can be written as follows

$$P: y_n^{(0)} = \sum_{i=1}^k \bar{\alpha}_i y_{n-1} + \bar{\beta}_i d_{n-1}$$

C: For $j = 1, 2, \dots, M-1$ do

$$y_{n,j+1} = y_{n,j} + B^{-1} [h\beta_0 f(y_{n,j}) + \sum_{i=1}^k (\alpha_i y_{n-1} + \beta_i d_{n-1}) - y_{n,j}] \quad (3.6)$$

$$y_n = y_{n,M} ; d_n = \frac{1}{\beta_0} [y_n - \sum_{i=1}^k (\alpha_i y_{n-i} + \beta_i d_{n-i})]$$

where B^{-1} is an approximation to the inverse of $(I - h\beta_0 J)$ and J is the Jacobian of f relative to the variable y . (Note that d_n is the approximation to hy'_n that satisfies the corrector equation. It is not $hf(y_n)$ unless the corrector has been solved exactly by the iteration. This is important for stability.)

We shall simplify the notation by setting

$$\sum_c = \sum_{i=1}^k (\alpha_i y_{n-i} + \beta_i d_{n-i}) \quad (3.7)$$

The corrector iteration (3.6) corresponds to M steps of the Newton method for solving the equation

$$y_n = h\beta_0 f(y_n) + \sum_c \quad (3.8)$$

starting with the predicted value $y_n^{(0)}$ and using the approximation B^{-1} for the inverse of $I - \beta_0 hJ$.

3.2.2. Convergence of the corrector iteration

The first question that arises is under what conditions on B would the corrector iteration converge if it were iterated for an infinite number of steps. Let us assume that equation (3.8) has an exact solution $y_{n,*}$.

Consider the error $y_{n,j+1} - y_{n,*}$

$$y_{n,j+1} - y_{n,*} = y_{n,j} - y_{n,*} + B^{-1} [h\beta_0 f/y_{n,j}] + \sum_c - y_{n,j} \quad (3.9)$$

$f(y_{n,j})$ can be expressed as

$$f(y_{n,j}) = f(y_{n,*}) + J_j(y_{n,j} - y_{n,*}) \quad (3.10)$$

where the (i,k) entry of the matrix J_j is the partial derivative of f_i relative to the k -th variable $y_{(k)}$, evaluated at a point ξ between $(y_{n,*})_{(k)}$ and $(y_{n,j})_{(k)}$. Substituting (3.10) into (3.9) and using the fact that $y_{n,*} = h\beta_0 f(y_{n,*}) + \sum_c$ we get

$$y_{n,j+1} - y_{n,*} = y_{n,j} - y_{n,*} + B^{-1} [h\beta_0 J_j(y_{n,j} - y_{n,*}) + y_{n,*} - y_{n,j}]$$

or

$$y_{n,j+1} - y_{n,*} = [I - B^{-1}(I - h\beta_0 J_j)] (y_{n,*} - y_{n,j}) \quad (3.11)$$

As a result of (3.11) we can derive sufficient conditions for convergence of the corrector iteration such as $\|I - B^{-1}(I - h\beta_0 J_j)\| < \alpha < 1 \forall j$, for a certain norm. This means, however, infinitely many assumptions. If we make the simplifying assumption that the Jacobian is constant, then the above condition becomes

$$\|I - B^{-1}(I - h\beta_0 J)\| < 1 \quad (3.12)$$

Furthermore, we have a choice for the norm used in (3.12). Since we would like to interpret the result (3.12) in terms of the eigenvectors of J , it seems natural to consider the following norm*

*Note that this norm is just the classical norm $\|\cdot\|_1$ relative to the eigenbasis.

$$\|A\|_1 = \max_{i=1, \dots, N} \|Au_i\|_1$$

Where $\{u_i\}_{i=1, \dots, N}$ is a basis of $\mathbb{C}^N$ consisting of unit eigenvectors of J and where we set $\|x\|_1 = \sum_{i=1}^N |\alpha_i|$ for $x = \sum_{i=1}^N \alpha_i u_i$, $\|A\|_1$ is the corresponding matrix norm. Equation (3.12) then reads

$$\max_{i=1, \dots, N} \|(I - (1 - h\beta_0 \lambda_i)B^{-1})u_i\|_1 < 1$$

or

$$\| [B^{-1} - \frac{1}{1-h\beta_0 \lambda_i} I] u_i \|_1 < |1 - h\beta_0 \lambda_i|^{-1}, \quad i=1, \dots, N \quad (3.13)$$

The left side of (3.13) is just the residual norm of the eigenpair $u_i, (1 - h\beta_0 \lambda_i)^{-1}$ by B^{-1} .

The interpretation of (3.13) is as follows. Since B^{-1} is an approximation to $(I - h\beta_0 J)^{-1}$, we can consider that the eigenvectors u_i of $(I - h\beta_0 J)^{-1}$ associated with $(1 - h\beta_0 \lambda_i)^{-1}$ are approximate eigenvectors to B^{-1} . Inequality (3.13) tells us that a sufficient condition for convergence of the corrector is that the residuals of the eigenpairs $u_i, (1 - h\beta_0 \lambda_i)^{-1}$ by B^{-1} have a norm smaller than $|1 - h\beta_0 \lambda_i|^{-1}$. Such a condition will be more difficult to satisfy for the most negative λ_i for which $|1 - h\beta_0 \lambda_i|^{-1}$ is small, while it will not cause any difficulty for the small eigenvalues for which $|1 - h\beta_0 \lambda_i|^{-1} \approx 1$. In other words, B^{-1} should be a good approximation to $(I - h\beta_0 J)^{-1}$ in the dominant space.

3.2.3. The error equation

We now assume that the sufficient condition (3.12) is fulfilled and we will analyze the stability of the method for a linear equation of the form $y' = Jy$. Since there are difficulties in estimating the global error $y(t_n) - y_{n,M}$, we proceed in a different way by estimating $\tilde{e}_n = y_n - y_{n,M}$ where y_n is the approximation at t_n that would be obtained if the corrector were solved exactly at each integration step.

We will distinguish between the two sequences by denoting by $\tilde{y}_n$ the sequence obtained as described in 3.2.1 when an approximate inverse B^{-1} to $A = (I - h\beta_0 J)$ is used with M iterations, that is, $\tilde{y}_n = y_{n,M}$. These two sequences satisfy:

$$y_n = h\beta_0 J y_n + \sum_c \quad (3.14)$$

$$\tilde{y}_n = h\beta_0 J \tilde{y}_n + \sum_c - r_n \quad (3.15)$$

where

$$\sum_c = \sum_{i=1}^k (\alpha_i y_{n-i} + \beta_i d_{n-i}), \quad \tilde{\sum}_c = \sum_{i=1}^k (\alpha_i \tilde{y}_{n-i} + \beta_i \tilde{d}_{n-i})$$

and r_n is a residual term which takes into account the fact that the corrector is not solved exactly.

Subtracting (3.14) from (3.15) we get

$$\tilde{e}_n = h\beta_0 J \tilde{e}_n + \sum_{i=1}^k (\alpha_i \tilde{e}_{n-i} + \beta_i \tilde{e}'_{n-i}) - r_n \quad (3.16)$$

Let $\beta_i = 0$, $i = 1, \dots, k$ (BDF formulas) then

$$\tilde{e}_n = A^{-1} \sum_{i=1}^k \alpha_i \tilde{e}_{n-i} - A^{-1} r_n \quad (3.17)$$

If we expand $\tilde{e}_j$ and r_n in the eigenbasis $\{u_i\}_{i=1, N}$ as

$$\tilde{e}_j = \sum_{i=1}^n \delta_j^{(i)} u_i, \quad r_n = \sum \epsilon_n^{(i)} u_i$$

we obtain for each component the equation

$$\delta_n^{(i)} = \left(\frac{1}{1 - \beta_0 h \lambda_i} \right) \sum_{i=1}^k \alpha_i \delta_{n-i}^{(i)} - \frac{\epsilon_n^{(i)}}{1 - \beta_0 h \lambda_i} \quad (3.18)$$

Let us now introduce the sequence of vectors $w_n = (\delta_n, \delta_{n-1}, \dots, \delta_{n-k})^T$, $n = 0, 1, \dots$, where the subscript i is dropped for simplicity and δ_j is taken to be zero for a negative j . Then (3.18) becomes

$$w_n = M_\lambda w_{n-1} - \epsilon'_n e_1 \quad (3.19)$$

where $e_1 = (1, 0, \dots, 0)^T$, $\epsilon'_n = (1 - \beta_0 h \lambda)^{-1} \epsilon_n$ and

$$M_\lambda = \begin{vmatrix} \alpha_1^1 & \alpha_2^1 & \dots & \dots & \alpha_k^1 \\ 1 & 0 & \dots & \dots & \cdot \\ 0 & 1 & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \cdot \\ 0 & 0 & \dots & 1 & 0 \end{vmatrix}$$

with $\alpha_j' = (1 - \beta_0 h \lambda)^{-1} \alpha_j$. Solving (3.19) we get

$$w_n = \sum_{i=0}^n \varepsilon'_{n-i} M_{\lambda}^i e_1 \quad (3.20)$$

Hence

$$\|w_n\| \leq \sum_{i=0}^n |\varepsilon'_{n-i}| \|M_{\lambda}^i e_1\|$$

or

$$\|w_n\| \leq \max_{i=0,n} |\varepsilon'_i| \sum_{i=0}^n \|M_{\lambda}^i e_1\| \quad (3.21)$$

The boundedness of w_n is related to the spectral radius of the matrix M_{λ} whose characteristic polynomial is

$$\xi^k - \sum_{i=1}^{k-1} \alpha'_i \xi^{k-i} = \xi^k - \sum_{i=1}^k \frac{1}{1 - \beta_0 h \lambda} \alpha_i \xi^{k-i} \quad (3.22)$$

The roots $\xi_{h\lambda}$ of the polynomial (3.22) are the same as those of the characteristic polynomial $\zeta(\xi) - h\lambda\sigma(\xi)$ associated with the corrector. Therefore, $\zeta_{h\lambda} < 1$ iff $h\lambda$ belongs to the region of absolute stability of the corrector.* In that case we can distinguish two cases:

- either $\zeta_{h\lambda} < 1$ in which case there exists a constant K such that

$$\|w_n\| \leq K \max_{i=0,n} |\varepsilon'_i| \quad (3.23)$$

*A problem could have small positive eigenvalues which are outside of the region of absolute stability, but these lead to slowly growing solutions so the interval of integration will have to be restricted to bound errors. We will ignore this case.

• or $\zeta_{h\lambda} = 1$ and it can only be guaranteed that

$$\|w_n\| < n^{\nu+1} K \max_{i=0,n} |\varepsilon'_1| \quad (3.24)$$

where ν is the difference between the (algebraic) multiplicity and the geometric multiplicity of the dominant eigenvalue of M_λ .

The relations (3.23) and (3.24) can be established by using the Jordan Canonical form of M_λ .

The meaning of (3.23) and (3.24) is that if $h\lambda$ is (strictly) inside the region of absolute stability, then the distance $\tilde{e}_n$ between y_n , the approximate solution obtained when the corrector equation is solved exactly at each step, and $\tilde{y}_n$ the approximate solution obtained by using an approximate Jacobian will not grow with n provided that the errors

$|\varepsilon'_1| = \frac{|\varepsilon_1|}{|1 - \beta_0 h \lambda|}$ remain bounded. An important fact is that the error terms $|\varepsilon_1|$ are nothing but the components of the residual of the last linear system encountered when solving the corrector along the eigenvector u associated with λ . This residual can be made as small as desired according to the results of Section 3.2.2, provided that B^{-1} is a fairly good representation of $(I - \beta_0 h J)^{-1}$ in the dominant subspace. Further, if $h\lambda$ is strictly inside the region of absolute stability, we know that the distance between y_n and $y(t_n)$ is bounded for the constant coefficient linear problem $y' = Jy$. Hence, for this case the iterative process is stable.

3.2.4. Practical considerations

There are a number of difficult-to-analyze techniques that can be used to get the greatest advantage from iterative methods. Some of these will be discussed in this section.

Reducing h

As the stepsize is reduced, fewer eigenvalues are part of the dominant subspace, so the dimension m needed for V_m can be reduced. In some cases it will be worth restricting h to be smaller than that dictated by integration accuracy in order to reduce m and reduce the work per step. For example, if halving h reduces m by more than a factor of two, the integrator will probably be faster with the smaller h .

Recomputing V_m

If the problem has a few isolated large eigenvalues, then it is likely that V_m can be kept constant over several steps. However, if there are several clusters of large eigenvalues, it is probably more efficient to recompute V_m frequently. This is because the dimension of V_m will have to be order of the number of large eigenvalues if V_m is to correct errors anywhere in the dominant subspace (and this is necessary for stability) whereas if there are a few clusters, reasonable accuracy may be obtained with m equal to the number of clusters if V_m is recomputed anew for each step starting with the corresponding residual.

Finally, it is worth mentioning that in practice the need for recomputing a new V_m can be detected in a way much similar to the way that the need for a new Jacobian is detected in the codes which implicitly use the Jacobian matrix--that is, by examining the convergence (or otherwise) of the iterative steps in the Newton solution of the corrector.

Representing the dominant space

We have seen that B should be a good representation of $A = (I - h\beta_0 J)$ in the dominant subspace. Because B is based on the Krylov subspace $\{A^q r_0, q = 0, 1, \dots, m-1\}$ it is a good representation for the extreme eigenvalues including the small ones. If, instead, we use the space $\{A^q r_0, q = s, s+1, \dots, s+m-1\}$, we will emphasize the large eigenvalues. This effect can be increased by using J instead of A since the unimportant eigenvalues of J are those near zero. The use of $A^q r_0$ as the starting vector for the Krylov space reduces the significance of the component corresponding to the smallest eigenvalue λ_N by $|(1 - h\beta\lambda_N)/(1 - h\beta_0\lambda_1)|^q$ compared to the component corresponding to the largest. For small $|h\beta_0\lambda_N|$ and large $|h\beta_0\lambda_1|$ this is approximately $|h\beta_0\lambda_1|^{-q}$. Use of $J^q r_0$ results in a reduction of $|\lambda_N/\lambda_1|^q$. The ratio of these is $|h\beta_0\lambda_N|^q$, and for the nonstiff eigenvalue λ_N , $|h\beta_0\lambda_N|$ is less than one.

If we use $J^q r_0$ as a starting vector, the initial approximation x_0 must be chosen carefully. In a typical code we first approximate y_n

with an explicit predictor p_n , and then solve (3.8) for the correction by

$$\tilde{y}_n = p_n + x$$

where

$$Ax = \sum_c + h\beta_0 f(p_n) - p_n = r_0$$

If we solve this only in the dominant space, then the component of $\tilde{y}_n$ in the subdominant space is simply the predicted value. Therefore, we should set x_0 to be r_0 , which is equivalent to treating the nonstiff components with a simple PEC method.

Starting with $J^q r_0$ suffers from a serious drawback, namely that we then need to compute $[V_m^T V_m]^{-1} V_m^T r_0$ and cannot use the approximations suggested in Section 2.2. However, we will see that an outcome of another modification suggested below is to effectively use $q = 1$ without additional difficulty.

The method used above is, of course, a method for stiff equations such as BDF. This is not the best choice for nonstiff problems, so it is natural to ask if it is possible to use a nonstiff method such as Adams for the nonstiff components. The answer is shown to be affirmative below. Suppose that method (A) (e.g., Adams) is a nonstiff method and method (B) (e.g., BDF) is a stiff method. Superscripts will be used to distinguish the methods. We get a PEC method (A) by executing

$$\tilde{y}_n^{(A)} = p_n + x_1$$

with

$$x_1 = r_0 = \int_c^{(A)} + h\beta_0^{(A)} f(p_n) - p_n \quad (3.25)$$

Now we consider solving (3.8) for method (B) setting

$$\begin{aligned} \tilde{y}_n &= \tilde{y}_n^{(A)} + x_2 \\ (I - h\beta_0^{(B)} J)x_2 &= A^{(B)} x_2 = r_1 = \int_c^{(B)} + h\beta_0^{(B)} f(\tilde{y}_n^{(A)}) - \tilde{y}_n^{(A)} \end{aligned} \quad (3.26)$$

If this is solved only in the dominant subspace, this causes components in that subspace to be solved by method (B). We can prove the following.

Proposition 3. Suppose that J has a complete set of eigenvectors and that the left and right invariant subspaces corresponding to the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_m$ of J are spanned by the rows of U^T and columns of V , respectively, where U^T and V are N by m matrices. Then, if the operator $(A^{(B)})^{-1}$ needed to solve (3.26) is represented by $V[U^T A^{(B)} V]^{-1} U^T$, the solution to $y' = Jy$ by (3.25) and (3.26) is that obtained by treating the components corresponding to $\lambda_1, \dots, \lambda_m$ by method (B) and the remainder by method (A).

Proof. The proof follows by diagonalizing the system. square

The significance of this is that if V_m and V_m^T approximate V and U^T , we can expect to approximate this behavior which is desirable if λ_1 to λ_m are the only dominant eigenvalues.

When (3.26) is solved iteratively, we should build the Krylov space $K_m = \text{span}[r_1, Jr_1, \dots, J^{m-1}r_1]$ and a basis V_m to solve $Ax_2 = r_1$. Let us examine r_1 in terms of r_0 . Combine (3.25) and (3.26) and assume $f(y) = Jy$ to get

$$\begin{aligned} r_1 &= \sum_c^{(B)} + h\beta_0^{(B)} f(p_n + r_0) - p_n - r_0 \\ &= \sum_c^{(B)} + h\beta_0^{(B)} f(p_n) + h\beta_0^{(B)} Jr_0 - p_n - \sum_c^{(A)} - h\beta_0^{(A)} f(p_n) + p_n \\ &= [\sum_c^{(B)} + h\beta_0^{(B)} f(p_n) - \sum_c^{(A)} - h\beta_0^{(A)} f(p_n)] + h\beta_0^{(B)} Jr_0 \end{aligned}$$

The first term is the difference between the solutions by a PEC method using methods (B) and (A), respectively. For nonstiff component it is $O(h^{p+1})$ where p is the minimum of the orders of the two methods and one plus the order of the predictor. Hence, it is small for nonstiff components. Hence, r_1 has properties very similar to that of $h\beta_0^{(B)} Jr_0$: smaller in the nonstiff components by a factor proportional to $1/|\lambda_1|$. Thus, the Krylov subspace $\{J^q r_1, q = 0, \dots, m-1\}$ has similar properties to $\{J^q r_0, q = 1, \dots, m\}$.

4. Numerical Experiments

The numerical experiments described below have been performed on a Prime 650. The first series of tests treated has been adapted from Krogh's example given in [2, p. 218].

First n functions $z_i, i = 1, 2, \dots, n$ are defined by

$$\frac{dz_i}{dt} = \beta_i z_i + z_i^2 \quad i = 1, 2, \dots, n \quad (4.1)$$

where the β_i are negative constants. Then a vector y is set equal to

$$y = Uz \quad (4.2)$$

where

$$U = I - \frac{2}{v^T u} uv^T \quad (4.3)$$

Obviously y satisfies a differential equation of the form

$$y' = Ay + g(y) \quad (4.4)$$

where

$$A = U \text{diag}(\beta_i) U$$

$$[g(y)]_i = \sum_j U_{ij} \left(\sum_k U_{jk} y_k \right)^2$$

(Notice that $U^{-1} = U$).

The solution of each equation (4.1) is

$$z_i(t) = \frac{-\beta_i}{1 + c_i e^{-\beta_i t}} \quad (4.5)$$

where c_i is adjusted according to the initial value $z_i(0)$. The exact solution of (4.4) is therefore immediately obtainable by use of (4.5) and (4.2).

Another interesting feature of this class of examples is that one can easily choose u and v such that the Jacobian is more or less close

to a symmetric (or a skew symmetric) matrix. Also, the degree of nonlinearity can be controlled by adding a multiplicative constant γ before the term z_i^2 in (4.1):

$$dz_i/dt = \beta_i z_i + \gamma z_i^2$$

[In that case the solution (4.5) must be replaced by:
 $-\beta_i/(\gamma + c_i e^{-\beta_i t})$.] The following expression of the Jacobian of (4.4)

$$J(t) = U[\text{diag}(\beta_i + 2z_i)]U \quad (4.6)$$

can be used for comparisons with methods using explicit Jacobians.

The dimension N has been taken relatively small so as to allow comparisons with other methods. [Note that the storage of the Jacobian increases as N^2]. The following distribution has been taken for the elements β_i : $\beta_1 = -1000$, $\beta_2 = -800$, $\beta_3 = -500$; $\beta_4 = -300$, $\beta_i = -100 (N-i+1)/(N-5)$ for $i > 5$. The vectors u and v have been chosen as follows: $u = (0, 1, 1, \dots, 1, 1)^T$, $v = (1, 1, 1, \dots, 1)^T$ and the initial values are such that $y^{(0)} = Uz^{(0)}$ with

$$z^{(0)} = (-1, -1, \dots, -1)^T$$

Three different methods have been run on this example. They all use the BDF schemes with automatic determination of stepsize and order as implemented by Hindmarsh in the code LSODE. LSODE was modified to include the IOM method and projection process described in Section 2 as an additional option. The first method uses the (full) Gaussian

elimination to solve the linear systems and the Jacobian is directly provided by the user. The corresponding method flag in LSODE is $MF = 21$. The second method, as the first, uses the Gaussian algorithm but the Jacobian is internally generated rather than provided by the user. The Method Flag is $MF = 22$. The third method is a combination of the IOM method and the projection process. The Method Flag is $MF = 26$ for this new method. For the last case, the parameter p (the number of vectors against which the v_1 is orthogonalized) has been tried with two values $p = 4$ and $p = 9$, while $m_{\max}$ (the maximum number of vectors v_1 generated) is set to $3 \times (p+1)$. Table 1 compares, for $N = 50$, the results obtained when the problem (4.4) is integrated from $t_0 = 0$ to $t = 1$. The tolerance EPS was set to 10^{-6} .

TABLE 1				
	MF=21	MF=22	MF=26	MF=26
			P=4	P=9
Total number of function evaluations	182	1382	720	643
Total number of steps	151	151	184	180
Core memory required	2972	2972	1035	1750
Run times (minutes on PRIME 650)	1'08	1'09	1'19	1'05

The run times are not very different from those of the first two methods ($MF = 21$ and $MF = 22$) because the dimension is relatively small. When $N = 80$, the results shown in Table 2 are obtained:

TABLE 2				
	Mf=21	Mf=22	Mf=26	Mf=26
			P=4	P=9
Total number of function evaluations	175	1935	733	640
Total number of steps	149	149	186	172
Core memory required	7142	7142	1545	2410
Run times (minutes on PRIME 650)	2'57	3'03	1'55	1'38

Another class of examples for test purposes is provided by the heat equation on a domain Ω of $\mathbb{R}^2$.

$$\begin{aligned} \Delta u &= \frac{\partial u}{\partial t} \quad \forall x \in \Omega, \quad \forall t \in [t_0, t_1] \\ u(t, x) &= 0, \quad \forall x \in \partial \Omega, \quad \forall t \in [t_0, t_1] \\ u(t, x) &= u, \text{ given in } \Omega. \end{aligned}$$

Suppose Ω is the square $]0, 1[\times]0, 1[$ and that the five point discretization is employed for the x variable. Then the above problem yields the linear system of ODEs

$$\begin{aligned} y'(t) &= Ay(t), \quad t \in [t_0, t_1] \\ y_i(t_0) &= u(t_0, x_i), \quad i = 1, 2, \dots, N \end{aligned}$$

where $x_1, x_2, \dots, x_N$ are the N interior discretization points of the square with $N = n^2$ if n interior points on each side are taken, and where A is the $N \times N$ block tridiagonal matrix

$$A = \begin{bmatrix} B & \dots & I & \dots \\ I & \dots & \dots & \dots \\ \dots & \dots & \dots & I \\ \dots & \dots & I & B \end{bmatrix}$$

where B is a symmetric in-diagonal matrix with diagonal elements -4 and off-diagonal elements 1.

Since A is symmetric, the parameter p must be taken equal to 2 which corresponds to the conjugate gradient method. The initial distribution of temperature is taken equal to 1 (uniform distribution) in the interior. When $n = 10$, and $t_1 = 10^2$, the results obtained are the following:

total number of steps: 193

total number of function evaluations: 851

total number of reevaluations of V_m and H_m : 105

Work space required: 1660

LSODE can also treat this problem by considering A as a banded matrix of half bandwidth 11. Let us compare the results obtained above with those obtained by LSODE with MF = 25:

total number of steps: 155

total number of function evaluations: 691

total number of reevaluations of the Jacobian: 24

work space required: 4322

Since the matrix has a banded form, the second method (MF = 25) is obviously to be preferred as it is faster. Note, however, that the work space required can become very large when the dimension and the bandwidth increase.

LIST OF REFERENCES

- [1] Alfeld, P. and J. D. Lambert, Corrections in the dominant space: a numerical technique for a certain class of stiff initial value problems, Math. Comp. 31, 1977, 922-938.
- [2] Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall: New Jersey, 1971.
- [3] Gear, C. W., Future developments in stiff integration techniques: stability methods that do not use an exact Jacobian, UIUCDCS-R-76-839, Dept. Computer Sci., Univ. Illinois, 1976.
- [4] Gear, C. W. and L. F. Shampine, A user's view of solving stiff ordinary differential equations, SIAM Review 21, 1974, 1-17.
- [5] Klopfenstein, R. W., Numerical differentiation formulas for stiff systems of ordinary differential equations, RCA Review 32, 1971, 447-462.
- [6] Miranker, W. L. and I. L. Chern, Didiotomy and conjugate gradients in the stiff initial value problem, IBM Research Report 8032-34917, January 1980.
- [7] Saad, Y., Krylov subspace methods for solving large unsymmetric linear systems, to appear Math. Comp.
- [8] Saad, Y., Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, Lin. Alg. Appl. 34, 269-295, 1980.
- [9] Shampine, L. F., Implementation of implicit formulas for the solution of ODEs, SIAM J. Sci. and Stat. Computing 1 103-118, 1980.

U.S. DEPARTMENT OF ENERGY

UNIVERSITY-TYPE CONTRACTOR AND GRANTEE RECOMMENDATIONS
FOR DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

See Instructions on Reverse Side

1. DOE Report No. COO-2383-0078		3. Title ITERATIVE SOLUTION OF LINEAR EQUATIONS IN ODE CODES
2. Contract No. DE-AC02-76ER02383.A003		
4. Type of Document ("X" one) ☒ a. Scientific and technical report ☐ b. Conference paper: Title of conference _____ Date of conference _____ Exact location of conference _____ Sponsoring organization _____ ☐ c. Other (Specify Thesis, Translations, etc.) _____		
5. Recommended Announcement and Distribution ("X" one) ☒ a. DOE's normal announcement and distribution procedures may be followed. ☐ b. Make available only within DOE and to DOE contractors and other U.S. Government agencies and their contractors.		
6. Reason for Recommended Restrictions		
7. Patent Information Does this information product disclose any new equipment, process or material? ☐ Yes ☐ No Has an invention disclosure been submitted to DOE covering any aspect of this information product? If so, identify the DOE (or other) disclosure number and to whom the disclosure was submitted. ☐ Yes ☐ No Are there any patent related objections to the release of this information product? If so, state these objections.		
8. Submitted by Name and Position (Please print or type) C. W. Gear, Professor and Principal Investigator Organization Department of Computer Science, University of Illinois at Urbana-Champaign 61801 Signature <i>Charles W. Gear</i> Date January 1981		

FOR DOE USE ONLY

9. Patent Clearance ("x" one)
- ☐ a. DOE patent clearance has been granted by responsible DOE patent group.
- ☐ b. Report has been sent to responsible DOE patent group for clearance.
- ☐ c. Patent clearance not required.

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-81-1054	2.	3. Recipient's Accession No.
4. Title and Subtitle ITERATIVE SOLUTION OF LINEAR EQUATIONS IN ODE CODES		5. Report Date January 1981	
		6.	
7. Author(s) C. W. Gear and Y. Saad		8. Performing Organization Rept. No. R-81-1054	
9. Performing Organization Name and Address Department of Computer Science University of Illinois U-C Urbana, IL 61801		10. Project/Task/Work Unit No.	
		11. Contract/Grant No. DE-AC02-76ER02383.A003	
12. Sponsoring Organization Name and Address U.S. Department of Energy Chicago Operations Office Argonne, IL 60439		13. Type of Report & Period Covered technical	
		14.	
15. Supplementary Notes			
16. Abstracts Each integration step of a stiff equation involves the solution of a nonlinear equation, usually by a quasi-Newton method which leads to a set of linear problems. Iterative methods for these linear equations are studied. Of particular interest are methods which do not require an explicit Jacobian but can work directly with differences of function values using $J \approx f(x+\delta) - f(x)$. Some numerical experiments using a modification of LSODE are reported.			
17. Key Words and Document Analysis. 17a. Descriptors Newton nonlinear equations stiff equations initial value problems ordinary differential equations			
17b. Identifiers/Open-Ended Terms			
17c. COSATI Field/Group			
18. Availability Statement unlimited		19. Security Class (This Report) UNCLASSIFIED	21. No. of Pages 33
		20. Security Class (This Page) UNCLASSIFIED	22. Price