SIAM J. NUMER. ANAL.
Vol. 21, No. 4, August 1984

006

ODE METHODS FOR THE SOLUTION OF DIFFERENTIAL/ALGEBRAIC SYSTEMS

C. W. GEAR† AND L. R. PETZOLD‡

Abstract. In this paper we study the numerical solution of the differential/algebraic systems $F(t, y, y') = 0$. Many of these systems can be solved conveniently and economically using a range of ODE methods. Others can be solved only by a small subset of ODE methods, and still others present insurmountable difficulty for all current ODE methods. We examine the first two groups of problems and indicate which methods we believe to be best for them. Then we explore the properties of the third group which cause the methods to fail. We describe a reduction technique which allows systems to be reduced to ones that can be solved. It also provides a tool for the analytical study of the structure of systems.

Key words. nilpotency, singularity, matrix pencils, numerical methods

1. Introduction. We are interested in initial value problems for the differential/algebraic equation (DAE)

$$F(t, y, y') = 0, \tag{1}$$

where F, y, and y' are s-dimensional vectors. F will be assumed to be suitably differentiable. Many of these problems can be solved conveniently and economically using numerical ODE methods. Other problems cause serious difficulties for these methods. Our purpose in this paper is first to examine those classes of problems that are solvable by ODE methods, and to indicate which methods are most advantageous for this purpose. Secondly, we want to describe the problems which are not solvable by ODE methods, and the properties of these problems which cause the methods to fail. Finally, we want to discuss some analytical techniques for rewriting systems in a form which can be solved by numerical methods.

The idea of using ODE methods for solving DAE systems directly was introduced in [Gear71], and is best illustrated by considering the simplest possible algorithm, based on the backward Euler method. In this method the derivative $y'(t_{n+1})$ at time t_{n+1} is approximated by a backward difference of $y(t)$, and the resulting system of nonlinear equations is solved for y_{n+1},

$$F(t_{n+1}, y_{n+1}, (y_{n+1} - y_n)/(t_{n+1} - t_n)) = 0. \tag{2}$$

In this way the solution is advanced from time t_n to time t_{n+1}. Higher order techniques such as backward differentiation formulas (BDF), Runge–Kutta methods, and extrapolation methods are generalizations of this simple idea.

One of the main advantages in using ODE methods directly for solving DAE systems is that these methods preserve the sparsity of the system. For example, one set of DAE systems which is particularly simple to solve consists of systems which are really ODEs in disguise. If, in (1), $\partial F/\partial y'$ is nonsingular, then the system can, in principle, be inverted to obtain an explicit system of ODEs

$$y' = f(t, y). \tag{3}$$

* Received by the editors September 14, 1982, and in revised form September 30, 1983. Supported in part by the U.S. Department of Energy, Grant DEAC0276ERO2383 and by the U.S. Department of Energy Office of Basic Energy Sciences.

† Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801.
‡ Sandia National Laboratories, Livermore, California 94550.

However, if $\partial F/\partial y'$ is a sparse matrix, its inverse may not be sparse. Thus it is preferable to solve the system directly in its original form. Similarly, it is possible to reduce some more complex DAE systems to a standard form which, though not as simple as (3), may be handled via well known techniques. This approach also tends to destroy the natural sparsity of the system.

The most challenging difficulties for solving DAE systems occur when $\partial F/\partial y'$ is singular. These are the systems with which we are concerned here. In some sense the simplest, or at least the best understood, class of DAE systems is that which is linear with constant coefficients. These systems,

$$Ay'(t)+By(t)=g(t), \tag{4}$$

can be completely understood via the Kronecker canonical form of the matrix pencil (A, B). The important characteristic of equation (4) that determines the behavior of the system and numerical methods is the index of nilpotency of the matrix pencil (A, B). Numerical methods such as (2) can be used to solve linear and nonlinear systems of index no greater than one with no great difficulty. Algorithms based on these methods experience problems when the index is greater than one. We will introduce a scheme for determining if a system has index greater than one. This scheme can be used in a code to warn the user of probable difficulty. With some care, techniques based on higher order methods such as extrapolation can be constructed for solving systems of the form (4), even if the index exceeds one. We consider these issues in § 2.

One might hope that the study of (4) could be used as a guide for understanding more complicated DAE systems. In general this fails to be true. The structure of the local constant-coefficient system may not describe the behavior of solutions to the DAE, for nonlinear or even linear, nonconstant-coefficient systems whose index is greater than one. Numerical methods which work for (4) break down when the matrices are time-dependent and the index is greater than one. In fact, we are not aware of any numerical methods (based on ODE techniques or otherwise) for solving general linear DAE systems, let alone nonlinear systems. In § 3 we examine the structure of time-dependent problems and show where the difficulties with conventional methods arise. In the last section we describe some analytical techniques for rewriting systems in a form which can be solved by numerical methods. These techniques are useful not only for simplifying systems in practice, but also as theoretical tools for exposing the underlying structure of high index systems.

2. Problems which can be solved by ODE methods. In this section we study problems whose index is no greater than one, and linear constant coefficient systems of arbitrary index. All of these problems are solvable by ODE methods.

The properties of linear constant-coefficient systems (4) are easily understood by transforming the system to Kronecker canonical form (KCF). For details see [SiEY81]. We give only an overview. The main idea is that there exist nonsingular matrices P and Q which reduce (A, B) to canonical form. When P and Q are applied to the constant-coefficient problem (4), we obtain

$$PAQQ^{-1}y'+PBQQ^{-1}y=Pg(t), \tag{5}$$

where (PAQ, PBQ) is the canonical form. When $A+\lambda B$ is singular for all values of λ, no solutions exist, or infinitely many solutions exist. It is not even reasonable to try to solve these systems numerically in the absence of any additional information about the solutions to the system. Fortunately, numerical ODE methods reject these problems almost automatically because they have to solve a linear system involving the matrix

$A+h\beta B$ (where h is the stepsize and β is a scalar which depends on the method and recent stepsize history); this matrix is singular for all values of h. When $\det(A+\lambda B)$ is not identically zero, the system is "solvable" by the following definition, which was introduced in [SiEY81]. Here we give it for the time varying linear problem.

DEFINITION. A linear system $A(t)y'+B(t)y=g(t)$ is *solvable* iff for any sufficiently smooth input function $g(t)$, solutions to the differential/algebraic equation exist, and solutions which have the same initial values are identical.

In the following we will deal only with solvable systems.

For solvable systems the KCF form (5) of a constant-coefficient problem can be written as

$$y_1'(t)+Cy_1(t)=g_1(t), \tag{6a}$$

$$Ey_2'(t)+y_2(t)=g_2(t), \tag{6b}$$

where

$$Q^{-1}y(t)=\begin{vmatrix} y_1(t) \\ y_2(t) \end{vmatrix}, \qquad Pg(t)=\begin{vmatrix} g_1(t) \\ g_2(t) \end{vmatrix},$$

and E has the property that either there exists an integer m such that $E^m=0$, $E^{m-1}\neq 0$ or E is the "empty" (or zero by zero) matrix (where we have assumed for completeness that $O^0=I$). In the latter case, m is defined as 0. The value of m is defined to be the index of nilpotency of the system. The matrix E is composed of Jordan blocks of the form

$$\begin{vmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{vmatrix},$$

and m is the size of the largest of these blocks.

The behavior of numerical methods for solving standard ODE systems (6a) is well understood and will not be discussed here. Since the systems (6a) and (6b) are completely uncoupled and the methods we are interested in are linear, it suffices for understanding (4) to study the action of numerical methods on subsystems of the form (6b), where E is a single Jordan block. When E is a single Jordan block of size m, the system is referred to as a canonical (index $=m$) subsystem.

2.1. Index one problems. In the nonlinear case we associate the matrices A and B with $\partial F/\partial y'$ and $\partial F/\partial y$, respectively. The pencil (A, B) and its index depend on y, y', and t, but in many cases of practical importance, the structure of the pencil is fixed. For example, the system

$$y'=f(y, z, t), \qquad 0=g(y, z, t),$$

has index 1 for all y, z and t such that $[\partial g/\partial z]^{-1}$ exists and is bounded.

We will say that a system has *uniform* index m if the index of the pencil $(A, B)=(\partial F/\partial y', \partial F/\partial y)$ is independent of the points of evaluation of *each* of the elements of the pencil. If, in addition, the transformations to canonical form are sufficiently smooth, index one problems can be handled by the BDF methods. To state the theorem easily, we need a little notation. Note that (A, B) has $2s^2$ elements because F, y and y' are s-dimensional vectors, and each element is a separate function of values of y and y' and t so $(A, B)\in R^{2s^2(2s+1)}\to R^{2s^2}$. Let $w\in R^{2s^2(2s+1)}$. The P and Q which transform (A, B) to canonical form are also functions of w.

THEOREM 2.1. *If*

(i) *System* (1) *has uniform index* 1,

(ii) $Q(w)$ *and* $Q^{-1}(w)$ *exist and are bounded for all* w,

(iii) $Q^{-1}(w_1)Q(w_2) = I + O(\|w_1 - w_2\|)$ *for all* w_1, w_2,

then the solution of system (1) *by the k-step* BDF *method with fixed stepsize h converges to order* $O(h^k)$ *if all initial values are correct and* $k < 7$.

The proof of this theorem is given in [GePe82b]. In fact, the conditions need only be satisfied in a neighborhood of the solution, as is stated in the referenced proof. It should also be pointed out that if the index is uniformly zero, it is trivial to prove a similar result.

While the ODE methods behave basically as expected for the index = 1 problems, there are still some practical difficulties involved in implementing these methods for this class of problems. Some of these problems are discussed in [Petz81], [Petz82]; we will not discuss most of these difficulties here.

Most automatic codes for solving DAE systems [Petz82] are designed to handle nonlinear systems of index $\leqq 1$. These codes cannot handle systems of higher index, and it would be desirable in such codes to detect higher index problems and stop.

It can be done by the following algorithm, described in [Luen77] amongst other places. This algorithm is an application of a more general one discussed in § 4 (Algorithm 4.1) for higher-index problems. We state it as a theorem (whose straightforward proof is given in [GePe82b]).

THEOREM 2.2.

(i) *If* A *is nonsingular, the index of* (A, B) *is zero.*

(ii) *If* A *is singular and* R *is nonsingular such that*

$$RA = \begin{vmatrix} A_1 \\ 0 \end{vmatrix},$$

where the $q \times s$ *matrix* A_1 *has full rank* q, *then, if the matrix*

$$\begin{vmatrix} A_1 \\ B_2 \end{vmatrix}$$

is nonsingular, the index is one, where

$$RB = \begin{vmatrix} B_1 \\ B_2 \end{vmatrix}$$

and B_1 *is* $q \times s$.

2.2. Linear, constant-coefficient systems. Systems of index greater than one have several properties which are not shared by the lower index systems. The properties of these high index constant-coefficient systems which cause codes to fail are discussed in much greater detail in [Petz81]; we give only a brief outline here. We can understand many of the properties of (4) and of numerical methods by studying the simplest index 3 problem,

$$z_1 = g(t), \quad z_1' - z_2 = 0, \quad z_2' - z_3 = 0.$$

The solution to this problem is $z_1 = g(t)$, $z_2 = g'(t)$, $z_3 = g''(t)$. If initial values are specified for the z_i, the solution has a discontinuity unless these initial values are compatible with the solution. If the driving term $g(t)$ is not twice differentiable everywhere, the solution will not exist everywhere. For example, if $g(t)$ has a simple jump discontinuity at some point, z_2 includes a Dirac delta function, and z_3 includes the derivative of a Dirac delta.

What happens when a numerical method is applied to one of these problems? It is surprising that some of the numerical ODE methods work so well on these problems which are so unlike ODEs. We can best explain how the methods work by example. When the backward Euler method is used to solve the simple index 3 problem above, we find that the solution at time t_n is given in terms of the solution at time t_{n-1} by

$$z_{1,n} = g_n, \quad z_{2,n} = (z_{1,n} - z_{1,n-1})/h, \quad z_{3,n} = (z_{2,n} - z_{2,n-1})/h. \tag{7}$$

The values of z_1 will be correct at all steps (if roundoff error is ignored), although the initial value $z_{1,0}$ may be incorrect. If the initial values (which need not be specified for the original problem but must be specified for the numerical procedure) are inconsistent, the values of $z_{2,1}$ and $z_{3,1}$ are incorrect. In fact, as $h \to 0$ they diverge. However, after two steps we obtain an $O(h)$ correct value of $z_{2,2}$ because it is obtained by the divided difference of $g(t)$. Finally, after the third step we obtain a good approximation to z_3 which is given by the second divided difference of $g(t)$. After the third step all the components will be $O(h)$ accurate.

The behavior of a general BDF method is very similar to that of backward Euler for fixed stepsize as shown in the following theorem, proved in [SiEY81].

THEOREM 2.3. *If the k-step, constant-stepsize* BDF *method is applied to the constant-coefficient linear problem* (4) *with* $k < 7$, *the solution is* $O(h^k)$ *accurate globally after a maximum of* $(m-1)k+1$ *steps.*

Unfortunately, these results for BDF break down when the stepsize is not constant, as shown in the next theorem, proved in [GeHP81].

THEOREM 2.4. *If the k-step* BDF *method is applied to* (4) *with* $k < 7$ *and the ratio of adjacent stepsizes is bounded, then the global error is* $O(h_{\max}^q)$, *where* $q = \min(k, k-m+2)$.

The difficulty can be seen by considering (7) for variable stepsizes. In that case we get

$$z_{1,n} = g_n, \quad z_{2,n} = (z_{1,n} - z_{1,n-1})/h_n, \quad z_{3,n} = (z_{2,n} - z_{2,n-1})/h_n.$$

Even after the initial errors have disappeared we find that

$$z_{3,n} = \frac{(g_n - g_{n-1})/h_n - (g_{n-1} - g_{n-2})/h_{n-1}}{h_n}.$$

If this were to be an $O(h)$ correct approximation to g_n'' the denominator should be $(h_n + h_{n-1})/2$. Hence the error is

$$\frac{1}{2}\left(\frac{h_{n-1}}{h_n} - 1\right) g_n''$$

which is $O(1)$ if $h_n = O(h_{n-1})$ but $h_n \neq h_{n-1}$.

Although, in principle, a problem of index no greater than 7 could be solved by the six-step BDF method with variable stepsize, the hypothesis in Theorem 2.4, that the ratio of adjacent steps is bounded, is not a reasonable model in practice. When a code is attempting to take the next step, all previous stepsizes are now fixed, and the next step must be chosen to achieve the desired error. In this model the error of a BDF formula used for numerical differentiation is $O(h)$, where h is the current stepsize. Consequently, if the index exceeds 2, the error of one step does not converge as that stepsize goes to zero, and diverges if the index exceeds 3. This can be seen in the above example in which the error in $z_{3,n}$, namely $(h_{n-1}/h_n - 1)g_n''$, behaves like $O(h_n^{-1})$.

The above results suggest that variable-stepsize BDF is not a suitable method for solving constant-coefficient DAEs with arbitrary index.

Fortunately, fixed stepsize BDF methods have an asymptotic expansion of the global error of the form

$$y(t,h) = y(t) + \sum_{i=k}^{N} \tau_i(t) h^i + O(h^{N+1}) \tag{8}$$

where $y(t,h)$ is the numerical solution by a k-step BDF method for $k<7$ and the stepsize $h=(t-t_0)/n$ for integer n. This indicates that constant-coefficient linear DAEs could be solved by extrapolation methods applied to fixed stepsize BDF methods.

Extrapolation is based on computing $y(t,h)$ for $N-k+2$ different values of $h=(t-t_0)/n_i$, $i=1,2,\cdots,N-k+2$ using (8) to compute $y(t)$ under the assumption that the $O(h^{N+1})$ terms can be ignored. Techniques such as those discussed in [Deuf80] can be used to vary the effective stepsize and order, but must be modified to ensure that all n_i exceed the value given in Theorem 2.3, namely $(m-1)k+1$. For this reason the backward Euler ($k=1$) is to be preferred. In the extrapolation tableau (see [Deuf80]) the diagonal and sufficient subdiagonal values must be discarded to avoid small n_i. It is possible that one could obtain an estimate of the index by observing how many terms must be discarded, but no experiments have been done. In practice, the use of this technique is complicated somewhat by the possibility of discontinuities in the function g, and also by the fact that, for higher index systems, the matrices needed for solving for the solution of the backward Euler formula are likely to be severely ill-conditioned. This technique is the best approach that we know of for solving linear constant-coefficient DAE systems.

3. Linear nonconstant-coefficient systems. In this section we study the nonconstant-coefficient linear problem,

$$A(t)y'(t) + B(t)y(t) = g(t). \tag{9}$$

We explore the underlying structure of these systems, and examine the reasons why they have proven to be so difficult to solve.

When the coefficients are not constant, as in (9), there are several possible ways to define the index of the system. We can clearly define the *local index*, $l(t)=$ index $(A(t), B(t))$, whenever the pencil $(A(t), B(t))$ is nonsingular. We can also define the *global index*, when it exists, in terms of possible reductions of the DAE to a semi-canonical form. By making a change of variables $y=H(t)z$ and scaling the system by $G(t)$, where $G(t)$ and $H(t)$ are nonsingular, we obtain from (9)

$$G(t)A(t)H(t)z' + (G(t)B(t)H(t) + G(t)A(t)H'(t))z = G(t)g(t). \tag{10}$$

Now, if there exist $G(t)$ and $H(t)$ so that

$$\begin{aligned} G(t)A(t)H(t) &= \begin{vmatrix} I_1 & 0 \\ 0 & E \end{vmatrix}, \\ G(t)B(t)H(t) + G(t)A(t)H'(t) &= \begin{vmatrix} C(t) & 0 \\ 0 & I_2 \end{vmatrix}, \end{aligned} \tag{11}$$

and the index of E is m, we will say that the system has global index of m. Note that the global index is the local index of this semi-canonical form.

Clearly, it is the global index that determines the behavior of the solution. If the global index is a constant m, we know that n_1 independent initial values can be chosen,

where n_1 is the dimension of the "differential" part of the system, and that the driving term can be subject to differentiation $m-1$ times. (Changes in the index or the structure of the system are called turning points. Problems with turning points are of importance in electrical network analysis. See Sastry, et al. [SaDV80] for a discussion in that context, and Campbell [Camp81] for a discussion of types of turning points.)

The local index in some sense governs the behavior of the numerical method. For example, if the matrix pencil is singular, then numerical ODE methods cannot solve the problem because they will be faced with the solution of singular linear equations. In understanding why numerical ODE methods break down, it is natural to ask how the local index and global index are related. The next theorem answers this question.

THEOREM 3.1. *If the local index is not greater than one, then it is not changed by a smooth transformation. If the local index is greater than one, then a smooth, nonconstant transformation of variables in* (9) *will yield a system whose local index is two unless additional constraints are satisfied by the transformation. A restricted set of transformations will cause the index to be greater than two, or the pencil to be singular. When the transformation to semi-canonical form* (11) *is used, this shows the relationship between the local and global indices.*

The proof of this result and some examples can be found in [GePe82a], and also in [Camp82, Chap. 5].

Whenever the global index exists, we have a good understanding of the behavior of the solutions to the system. Thus it is important to know if this index exists. That is, when does there exist a nonsingular scaling and change of variables transforming (9) to the semi-canonical form (11)?

In [CaPe82] examples are given to show that it is not in general possible to get the semi-canonical form everywhere with constant E, but that if A and B are analytic, there do exist analytic G and H for a reduction to (11) with time varying strictly lower triangular $E(t)$. In this form the index can be seen to change as $E(t)$ changes, although the dimension of the manifold of solutions (size of $C(t)$) does not change. In [GePe82b] it is shown constructively in a misstated theorem that a reduction to form (11) exists. The construction fails at isolated points but in any closed interval not containing such points, G and H exist. For many practical problems such a canonical form exists due to the structure of the matrices.

Since solvable systems are so closely related to systems of the form (11) (where the singular part of the system has constant coefficients), we might hope that some of the same techniques which work for solving constant-coefficient problems numerically might also be effective for general linear problems. Unfortunately, this turns out not to be the case.

We have seen that the constant stepsize BDF method can be used for constant-coefficient problems. What happens when it is applied to nonconstant-coefficient problems? If the local index is two we may have a stability problem depending on the rate of change of the coefficients. If the local index is greater than two, we almost always have a stability problem. We want to stress that this is a stability problem and not an accuracy question, so it does not appear that higher order methods will help. Also note that it depends on the local index while the behavior of the underlying equation depends on the global index.

We start by examining the application of the backward Euler method (BEM) to a linear problem which can be transformed locally to a canonical local index m problem. The general problem of this form can be written as

$$(PEQ)z' + (PQ)z = Pq,$$

where the $m \times m$ matrix E is given by

$$E = \begin{vmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{vmatrix}$$

and the matrices P and Q are possibly time-dependent transformations. The BEM gives the recurrence relation

$$[P_n E Q_n + h P_n Q_n] z_n = P_n E Q_n z_{n-1} + h P_n q_n \tag{12}$$

or

$$z_n = S_n z_{n-1} + u_n, \tag{13}$$

where

$$S_n = Q_n^{-1}[E + hI]^{-1} E Q_n,$$

$$u_n = h Q_n^{-1}[E + hI]^{-1} q_n.$$

The true solution satisfies

$$z(t_n) = S_n z(t_{n-1}) + u_n - \frac{h^2}{2} S_n z_n'', \tag{14}$$

where z_n'' is evaluated somewhere in the interval separately for each component.

Defining the global error $e_n = z_n - z(t_n)$ we get the usual error equation

$$e_n = S_n e_{n-1} + \frac{h^2}{2} S_n z_n'' \tag{15}$$

so the solution is

$$e_N = \frac{h^2}{2} \sum_{i=1}^{N} \left(\prod_{j=i}^{N} S_j \right) z_i'' + \prod_{j=0}^{N} S_j e_0. \tag{16}$$

The usual ODE argument says that if $S_i^N = \prod_{j=i}^{N} S_j$ is bounded, z_i'' is bounded, and e_0 goes to zero, then we have convergence. However, in this problem we have several difficulties. The solution may have jump discontinuities so z'' may not be bounded, e_0 will not usually go to zero because we do not know how to compute the initial conditions, and S_j may not be bounded. However, the first two of these difficulties can be overcome in the constant-coefficient case because the nilpotency of E is reflected in the nilpotency of $S = S_j$. By direct calculation it can be verified that $S^m = 0$ if Q is independent of n. Consequently, in this case we find that for $N > m$

$$e_N = \frac{h^2}{2} \sum_{i=0}^{m-2} S^{i+1} z_{N-i}'' = \frac{h^2}{2} \sum_{i=0}^{m-2} S^{i+1} R^i z_N'' \tag{17}$$

where $R z_n = z_{n-1}$ is the "backward" operator. The elements of S (assuming that $Q = I$ without loss of generality) are

$$S = \begin{vmatrix} 0 & & & & \\ h^{-1} & & & & \\ -h^2 & \ddots & & & \\ h^{-3} & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & h^{-3} & -h^{-2} & h^{-1} & 0 \end{vmatrix}$$

from which it follows that

$$\sum_{i=0}^{m-2} S^{i+1}R^i = -\begin{vmatrix} 0 & & & & \\ \phi & & & & \\ \phi^2 & \ddots & \ddots & & \\ \vdots & \ddots & & \ddots & \\ \phi^{m-1} & & \phi^2 & \phi & 0 \end{vmatrix} \tag{18}$$

where $-\phi = (1-R)/h$ is the backward Euler approximation to the derivative operator.

In the nonconstant-coefficient case this proof breaks down because S_n is time-dependent, so perhaps $S_n^N \neq 0$ for all $N > n$. We will examine the $m=2$ and $m=3$ cases to see why this happens. For the 2×2 case we find that

$$S_n = Q_n^{-1}\begin{vmatrix} 0 & 0 \\ 1/h & 0 \end{vmatrix} Q_n. \tag{19}$$

Assuming that Q' exists, we can write

$$\begin{aligned} Q_n &= (Q_nQ_{n-1}^{-1})Q_{n-1} = [Q_n(Q_n^{-1} - h(Q_n^{-1})' + O(h^2))]Q_{n-1} \\ &= [I + hQ_n'Q_n^{-1} + O(h^2)]Q_{n-1} \\ &= [I + hD_n + O(h^2)]Q_{n-1}, \end{aligned}$$

where $D = Q'Q^{-1}$, to get

$$S_nS_{n-1} = Q_n^{-1}\begin{vmatrix} 0 & 0 \\ 0 & d_{n12}+O(h) \end{vmatrix}\begin{vmatrix} 0 & 0 \\ 1/h & 0 \end{vmatrix} Q_{n-1} \tag{20}$$

where d_{n12} is the $(1,2)$ element of D_n. Hence

$$S_j^N = S_{j+1}^N S_j = Q_N^{-1}\begin{vmatrix} 0 & 0 \\ x & \prod_{j+1}^{N}(d_{n12}+O(h)) \end{vmatrix}\begin{vmatrix} 0 & 0 \\ 1/h & 0 \end{vmatrix} Q_j \tag{21}$$

where x is an unimportant element. From this follows.

THEOREM 3.2. *If $|d_{j12}| \leqq \alpha < 1$ and $\|Q_j\|$, $\|Q'\|$, and $\|Q_j^{-1}\|$ are bounded for all j, the* BEM *converges for the 2 by 2 problem provided the initial error is $o(h)$ and z'' is bounded.*

Proof. This follows by observing that (21) permits a bound of the form

$$\|S_j^N\| \leqq K\|Q_N^{-1}\|h^{-1}\|Q_j\|\eta^{N-j} \quad \text{where } |\eta| < 1, \tag{22}$$

which, with (16) implies

$$\|e_N\| \leqq \tilde{K}h\|e_0\|\eta^N + \sum_{j=0}^{N-1} \eta^{N-j}\tilde{K}hT \tag{23}$$

where $T \geqq \|z''\|$.

Under the hypotheses given, $\|e_N\| \to 0$ as $h \to 0$. Q.E.D.

On the other hand, if $|d_{j12}| > 1$, S_j^N diverges and we fail to get convergence. If $d_{j12} = 1$, a more careful analysis is necessary to determine stability. [GeHP81] gives the example

$$\begin{vmatrix} 0 & 0 \\ 1 & \eta t \end{vmatrix} y' + \begin{vmatrix} 1 & \eta t \\ 0 & 1+\eta \end{vmatrix} y = g$$

for which $d_{12} = \eta/(1+\eta)$ and we get a recurrence relation for z, the second component of y, of the form

$$z_{n+1} = \frac{\eta}{1+\eta} z_n + u_n$$

which is obviously unstable if $\eta < -\frac{1}{2}$.

For the 3×3 system with local nilpotency 3,

$$S_n = Q_n^{-1} \begin{vmatrix} h & 0 & 0 \\ 1 & h & 0 \\ 0 & 1 & h \end{vmatrix}^{-1} EQ_n$$

or

$$S_n = Q_n^{-1} \begin{vmatrix} 0 & 0 & 0 \\ 1/h & 0 & 0 \\ -1/h^2 & 1/h & 0 \end{vmatrix} Q_n. \tag{24}$$

The development from this point depends on which of the d_{ij} are nonzero. In general, all are nonzero so we can write $S_n S_{n-1}$ as

$$\begin{aligned} S_n S_{n-1} &= Q_n^{-1} \begin{vmatrix} 0 & 0 & 0 \\ 1/h & 0 & 0 \\ -1/h^2 & 1/h & 0 \end{vmatrix} (I + hD_n + O(h^2)) \begin{vmatrix} 0 & 0 & 0 \\ 1/h & 0 & 0 \\ -1/h^2 & 1/h & 0 \end{vmatrix} Q_{n-1} \\ &= Q_n^{-1} \begin{vmatrix} 0 & 0 \\ 1/h & d_{12} \\ -1/h^2 & (1-d_{12})/h \end{vmatrix} \begin{vmatrix} 0 & 0 & 0 & 0 \\ d_{13} & 1/h & 0 & 0 \\ 1/h & -1/h^2 & 1/h & 0 \end{vmatrix} Q_{n-1}(1+O(h)). \end{aligned} \tag{25}$$

This and (16) imply nonconvergence as h decreases.

An "explanation" of what is happening is as follows. In the constant-coefficient case for index m we have $m-1$ principal vectors and one eigenvector for the operator $S = [E - hI]^{-1}E$. One application of this operator maps each principal vector into the next with an amplification of up to h^{1-m}. The last principal vector gets mapped into an eigenvector while the eigenvector is annihilated. However, in the nonconstant-coefficient case there is a transformation between each step, giving rise to the $Q_n Q_{n-1}^{-1}$ term which is $I + hD_n + O(h^2)$. This can add an $O(h)$ multiple of one principal vector or eigenvector to each of the others. In the $m = 2$ case, the application of S multiplies the part of the error in the principal vector direction by $O(1/h)$ and moves it to the eigenvector. The rotation multiplies this by $O(h)$ and moves it back to the principal vector again. Thus, it is multiplied by $O(1)$ in each step, so the stability of the process depends on the magnitude of these mappings. If $m \geqq 3$, one step can amplify an error by $O(h^{-1})$.

Because of this stability problem, we do not know of any numerical techniques for the general linear problem (9), let alone the nonlinear problem (1) although some problems of high index can be solved with constant stepsize BDF methods. The latter situation could arise if suitable elements of D_n in (25) are zero so that $S_n^N = 0$ for large enough $N - n$. A nontrivial example of this appears to be given by the system of five equations

$$x' = u, \quad y' = v, \quad u' = Tx, \quad v' = -Ty + 1, \quad x^2 + y^2 = 1.$$

These describe a simple pendulum of length and mass 1 with unit gravity. The dependent variables are the distances x and y from the pivot, and the string tension T. The techniques of the next section can be used to show that this system has a local and global index of 3. However, after three steps of the BEM at constant stepsize, initial errors have been experimentally observed to be damped out so the solution is first order accurate. (This is an example of a problem described by Euler–Lagrange equations with holonomic constraints for which the local and global index can be shown to be always at least 3.)

4. Reduction techniques. It is sometimes possible to use analytical techniques to rewrite the system in a form with lower index which can be solved numerically. In this section we discuss two reduction techniques. The first is useful for reducing the index of systems (and also determining their index). The second is actually the idea behind the technique for constructing the transformation matrices to bring a system into semi-canonical form.

The first technique is described below for linear systems (9), but it applies directly to nonlinear problems (1) when F is linear in y'. It has been introduced for solving problems in optimal control in [Luen77] and [Silv69].

ALGORITHM 4.1.

(1) If A in (9) is nonsingular, then we are done.

(2) Otherwise premultiply (9) by a nonsingular matrix $P(t)$ to zero out a maximal number of rows of A and permute the zero rows to the bottom to obtain:

$$\begin{vmatrix} A_{11} \\ 0 \end{vmatrix} y' + \begin{vmatrix} B_{11} \\ B_{12} \end{vmatrix} y = g(t).$$

(3) Differentiate the bottom half of the system to obtain the new system

$$\begin{vmatrix} A_{11} \\ B_{12} \end{vmatrix} y' + \begin{vmatrix} B_{11} \\ B'_{12} \end{vmatrix} y = \hat{g}(t).$$

Now apply the process to this new system.

Intuitively, the idea behind this algorithm is that by differentiating the "algebraic" constraints of the system we can reduce its index without changing the solution to the system. If this is repeated, as in Algorithm 4.1, eventually we should produce a system of ODEs which can be solved by numerical methods. That this intuition is correct is stated in Theorem 4.2 below.

Of course, by differentiating we have introduced a number of constants of integration, which means that we must determine the correct initial conditions. This can be done by satisfying the initial system and each of its differentiated forms at the initial point.

THEOREM 4.2. *For solvable linear systems* (9) *with no turning points, Algorithm* 4.1 *terminates in m iterations iff the global index is m. Algorithm* 4.1 *does not terminate for systems which are not solvable. Proofs of this theorem can be found in* [Silv69] *and* [GePe82b].

We also note that Algorithm 4.1 can be used to find the local index of a system by considering the matrices A and B at some time t to be constant, and then applying the algorithm to the resulting system. In this case, the algorithm terminates in m steps iff the local index is m. Since by Theorem 3.1 the local index is equal to the global index if the index is one, the algorithm terminates after one iteration iff the index is one. This provides a proof of Theorem 2.2.

Another possible approach to simplifying the system (9) is based on using the "algebraic" constraints to solve for some variables in terms of the remaining variables, and thus reducing the size of the system. We can use this idea to construct a scheme for reducing solvable linear systems (18) to semi-canonical form. When the algorithm can be carried to completion, it produces nonsingular time-dependent matrices which reduce the system to semi-canonical form. When it cannot be completed, the system is not solvable. We outline the algorithm here for general linear systems. It has been described in [Camp80] and [SiDe78] for constant-coefficient systems.

ALGORITHM 4.3.

(1) If A in (9) is nonsingular, then we are done.

(2) Otherwise, premultiply (9) by a nonsingular matrix $P(t)$ to zero out a maximal number of rows of A and permute the zero rows to the bottom to obtain:

$$\begin{vmatrix} A_{11} & A_{12} \\ 0 & 0 \end{vmatrix} y' + \begin{vmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{vmatrix} y = g(t).$$

(3) Permute the columns of A and B so that B_{22} is a nonsingular matrix. (It can be shown that for solvable systems, $[B_{21}, B_{22}]$ has full rank.)

(4) Solve for $y_2 = B_{22}^{-1}(g_2 - B_{21}y_1)$, and differentiate this expression to solve for y_2' in terms of y_1 and y_1'.

(5) Substitute the expressions for y_2 and y_2' into the top half of the system, and rewrite to obtain a system of form (9) for y_1 (which is always smaller than the original system).

(6) Go back to step (1).

Note that while one would probably not want to use Algorithms 4.1 and 4.3 to solve a system, they are powerful tools both for rewriting a system in a form in which it can more easily be solved, and for discovering the underlying analytical structure of a system.

5. Conclusions. This paper has described a number of theoretical results which depend on the index of a system. In general, the index of a system, like the rank of a matrix, is not something one should attempt to compute numerically, so what does the ordinary user with the DAE (1) do?

If the index does not exceed 1, automatic codes such as [Petz82] can solve them with no trouble. (Theorem 2.1 required constant stepsizes, but nonconstant stepsizes do not cause difficulty as long as the stability of the BDF method is not disturbed. This is a potential problem in the differential part of the system rather than the algebraic part. So, Theorem 2.1 was not extended to nonconstant stepsizes to avoid unnecessary detail, although it will apply under reasonable restrictions on the rate of change of stepsize—see [GeTu74], for example.)

If the problem has index greater than one, an automatic code will usually fail—the stepsize is reduced repeatedly but it cannot satisfy its error tolerance criterion. In that case it would be desirable to apply the technique of Theorem 2.2 to determine if the failure was due to a high index. An integrator for (1) will have computed approximations to $A = \partial F/\partial y'$ and $B = \partial F/\partial y$. Theorem 2.2 can be applied to these approximations. It requires a rank determination which we know is not reasonable. However, if the problem is "near" to a high index problem, it will cause numerical difficulties. Hence, in determining the "rank" we should treat values below appropriately scaled error tolerances as zero. (We have not investigated ways to scale appropriately since we do

not yet fully understand how to scale the differential equations.) If Theorem 2.2 suggests that the index is greater than one, the user should be encouraged to reduce it.

The reduction described in Algorithm 4.1 can be applied in many cases because the index is determined by the nonzero structure of the matrices rather than the actual values of their entries as in the pendulum example at the end of § 3. If we differentiate the last equation three times, substituting for the derivatives of x, y, u and v from the earlier equations each time, we arrive at a differential equation for T, so that we have an explicit ODE system.

REFERENCES

[Camp80] S. L. CAMPBELL, *Singular Systems of Differential Equations*, Pitman, London, 1980.

[Camp82] ———, *Singular Systems of Differential Equations* II, Pitman, London, 1982.

[CaPe82] S. L. CAMPBELL AND L. PETZOLD, *Canonical forms and solvable singular systems of differential equations*, SIAM J. Alg. Disc. Meth., 4 (1983), 517–521.

[Deuf80] P. DEUFLHARD, *Order and stepsize control in extrapolation methods*, Preprint No. 93, Univ. Heidelberg, 1980.

[Gear71] C. W. GEAR, *The simultaneous numerical solution of differential-algebraic equations*, IEEE Trans. Circuit Theory, TC-18 (1971), pp. 89–95.

[GeHP81] C. W. GEAR, H. H. HSU AND L. PETZOLD, *Differential-algebraic equations revisited*, Proc. Numerical Methods for Solving Stiff Initial Value Problems, Oberwolfach, W. Germany, June 28–July 4, 1981.

[GePe82a] C. W. GEAR AND L. R. PETZOLD, *Differential/algebraic systems and matrix pencils*, Proc. Conference on Matrix Pencils, Pitea, Sweden, March 1982; also, Dept. Rpt. UIUCDCS-R-82-1086, 1982.

[GePe82b] ———, ODE *methods for the solution of differential/algebraic systems*, Dept. Rpt. UIUCDCS-R-82-1103, 1982.

[GeTu74] C. W. GEAR AND K. TU, *The effect of variable mesh size on the stability of multistep methods*, this Journal, 11 (1974), pp. 1025–1043.

[Luen77] D. LUENBERGER, *Dynamic equations in descriptor form*, IEEE Trans. Automat. Control, 22 (1977), pp. 312–321.

[Pain81] J. F. PAINTER, *Solving the Navier-Stokes equations with* LSODI *and the method of lines*, Lawrence Livermore Laboratory Rpt. UCID-19262, 1981.

[Petz81] L. R. PETZOLD, *Differential/algebraic equations are not* ODEs, SIAM J. Sci. Stat. Comp., 3 (1982), pp. 367–384.

[Petz82] ———, *A description of* DASSL: *A differential/algebraic system solver*, Proc. IMACS World Congress, Montreal, Canada, August 1982, to appear.

[Silv69] L. SILVERMAN, *Inversion of multivariable linear systems*, IEEE Trans. Automat. Control, AC-14 (1969), pp. 270–276.

[Star76] J. W. STARNER, *A numerical algorithm for the solution of implicit algebraic-differential systems of equations*, Tech. Rpt. 318, Dept. Mathematics and Statistics, Univ. New Mexico, Albuquerque, May 1976.

[SaDV80] S. S. SASTRY, C. A. DESOER AND P. P. VARAIYA, *Jump behavior of circuits and systems*, Memorandum No. UCB/ERL M80/44, Electronics Research Laboratory, University of California-Berkeley, CA, October 1980; also, IEEE Intl. Symposium on Circuits and Systems Proc., Vol. 2, 1981.

[SiEY81] R. F. SINCOVEC, A. M. ERISMAN, E. L. YIP AND M. A. EPTON, *Analysis of descriptor systems using numerical algorithms*, IEEE Trans. Automat. Control, AC-26 (1981), pp. 139–147.