COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 75 (1989) 77–89
NORTH-HOLLAND

RECENT DEVELOPMENTS IN THE NUMERICAL SOLUTION OF DIFFERENTIAL/ALGEBRAIC SYSTEMS

Linda R. PETZOLD
Computing and Mathematics Research Division, L-316, Lawrence Livermore National Laboratory, Livermore, CA 94550, U.S.A.

In this paper we survey some recent developments in the numerical solution of nonlinear differential/algebraic equation (DAE) systems of the form $0 = F(t, y, y')$, where $\partial F/\partial y'$ may be singular. Initial value problems in DAEs arise in a wide variety of applications, including circuit and control theory, chemical kinetics, modeling of constrained mechanical systems, fluid dynamics and robotics. DAE systems include standard form ODEs as a special case, but they also include problems which are in many ways quite different from ODEs. We explore some classes of initial value problems which can be solved by backward differentiation formulas, and discuss some results on the order of convergence of implicit Runge–Kutta methods applied to DAE systems. Finding a consistent set of initial conditions is often a problem for DAE systems arising in applications. We outline some recent work on a general algorithm for finding consistent initial conditions. Finally, we discuss some new developments in the numerical solution of DAE boundary value problems.

1. Introduction

In this paper we survey some recent results on the solution by numerical ODE methods of linear and nonlinear differential/algebraic (DAE) systems of the form

$$0 = F(t, y, y') , \tag{1.1}$$

where $\partial F/\partial y'$ may be singular. Differential/algebraic systems include standard form ODEs as a special case, but they also include problems which are in many ways quite different from ODEs. Initial value problems in DAEs arise in a wide variety of applications, including circuit and control theory, chemical kinetics, modeling of constrained mechanical systems, fluid dynamics and robotics. We explore some classes of initial value problems which can be solved by backward differentiation formulas, and discuss some results on the order of convergence of implicit Runge–Kutta methods applied to DAE systems. Finding a consistent set of initial conditions is often a problem for DAE initial value systems arising in applications. We outline some recent work on a general algorithm for finding consistent initial conditions. Finally, we discuss some new developments in the numerical solution of DAE boundary value problems. These problems arise in the modeling of semiconductor devices and in control theory.

The basic idea of using a numerical method for solving DAE systems was introduced by Gear [13], and consists of replacing y and/or y' in (1.1) by a difference approximation, and then solving the resulting equation for an approximation to y. In this paper we consider both backward differentiation formulas (BDF) and implicit Runge–Kutta methods (IRK) for

0045-7825/89/$3.50 © 1989, Elsevier Science Publishers B.V. (North-Holland)

defining the difference approximation. The simplest example of a numerical ODE method for (1.1) is the backward Euler method. Using this approach, the derivative $y'(t_n)$ at time t_n is approximated by a backward difference of $y(t)$ and the resulting system of nonlinear equations is solved for y_n,

$$0 = F\left(t_n, y_n, \frac{y_n - y_{n-1}}{h}\right), \tag{1.2}$$

where $h = t_n - t_{n-1}$. In this way the solution is advanced from time t_{n-1} to t_n.

This scheme of approximating the solution and/or the derivative in (1.1) and solving the resulting nonlinear equations is quite convenient for the solution of many different problems arising from practical applications. Systems in applications are often written naturally in the form (1.1). If we consider numerical methods for solving these very general systems, there will be no need to rewrite the system to fit into some special form. In addition, even though some DAE systems can be rewritten as standard form ODEs ($y' = f(t, y)$) or in other very special forms, often this is very inconvenient and it sometimes leads to a loss of sparsity in the matrices which are involved in the solution process. Finally, it is quite easy to explain to users of codes how to pose a problem in the form (1.1).

The disadvantage of this quite general formulation is that there are problems which can be written in the form (1.1) which are not solvable by ODE methods. Also there are problems which are solvable by some numerical ODE methods, but we must take very special care in choosing an appropriate method and also in implementing it. To distinguish these classes of problems requires a means of investigating the underlying structure of the DAEs.

2. Structure of DAE systems

In some sense the simplest class of DAEs is the class of linear constant-coefficient systems of the form

$$Ey' + Fy = g(t). \tag{2.1}$$

When $E + \lambda F$ is singular for all values of λ, then no solutions exist, or infinitely many solutions exist. We are not interested in the solution of these systems. The remaining systems, where $E + \lambda F$ is nonsingular, can be completely understood by transforming the system to Kronecker canonical form [29, 12]. These systems are solvable in the sense that solutions exist for all sufficiently smooth input functions $g(t)$, and two solutions which have the same initial value are identical. For solvable systems, there exist nonsingular matrices P and Q which decouple the system,

$$y_1'(t) + Cy_1(t) = g_1(t), \qquad Ny_2'(t) + y_2(t) = g_2(t), \tag{2.2}$$

where

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

into a ‘differential’ part and a ‘singular’ part. The matrix E is a block diagonal matrix whose blocks have the form

$$\begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & 0 & \\ & & & 1 & 0 \end{bmatrix}.$$

The most important concept in classifying these systems is the index. The index is the dimension of the largest block of E. The index is zero if there is no 'singular' part to the system (i.e., if the system is a standard form ODE or a system which can be rewritten as a standard form ODE by multiplying by E^{-1}). In general, the higher the index the more difficulties we are likely to encounter in trying to solve the system by a numerical ODE method.

To put these definitions into perspective, consider a canonical index two problem

$$y_1' = g(t)\,, \qquad y_2 = y_1'$$

and note that for this system, in contrast to an ODE, the initial values must be consistent with the input function $g(t)$, and the solution is not as smooth as the input.

For nonlinear systems (1.1), we can define the local index as the index of the local constant-coefficient system at any given time, or the global index, which we will be primarily concerned with here, in terms of the number of differentiations of the system which are necessary to be able to solve for y' uniquely in terms of y and t [8].

DEFINITION 2.1. The global index of a solvable DAE (1.1) is the smallest nonnegative integer m such that F has m continuous derivatives and the nonlinear system

$$\begin{aligned} &F(t, y, y') = 0\,, \\ &\frac{\mathrm{d}F}{\mathrm{d}t}(t, y, y', y'') = \frac{\partial F}{\partial y}\, y' + \frac{\partial F}{\partial y'}\, y'' + \frac{\partial F}{\partial t} = 0\,, \\ &\vdots \\ &\frac{\mathrm{d}^m F}{\mathrm{d}t^m}(t, y, y', y'', \ldots, y^{(m+1)}) = 0\,, \end{aligned} \tag{2.3}$$

can be solved for y' uniquely in terms of y and t: $y' = \varphi(y, t)$.

There are two important classes of index one systems. The first is the general fully implicit index one system (1.1). If we assume that the rank of $\partial F/\partial y'$ is constant, and that the index is identically equal to one in a neighborhood of the solution, then we will refer to these systems as uniform index one. The second class of index one systems is that of semi-explicit index one systems. These are systems which are written in the special form

$$0 = F_1(x, x', y, t)\,, \qquad 0 = F_2(x, y, t)\,, \tag{2.4}$$

where $\partial F_1/\partial x$ is nonsingular. The system (2.4) is index one if and only if $\partial F_2/\partial y$ is nonsingular. These semi-explicit index one systems arise frequently in applications. It is important to distinguish them from the general index one system (1.1) because there are numerical methods which perform well for (2.4) but poorly for (1.1).

For high index systems in the general form (1.1), numerical ODE methods such as the backward Euler method can be unstable for small stepsizes [16]. It is often possible to reduce the index of a semi-explicit system by analytically differentiating the constraints [16]. Campbell [8] has devised an algorithm for solving higher index systems which repeatedly differentiates the entire system and solves the resulting overdetermined system for y' in terms of y and t in the spirit of Definition 2.1. However, in its present form this algorithm is not useful for the solution of most problems in applications because it requires knowledge of the analytic partial derivative matrices of F and time derivatives of these matrices and requires a large degree of smoothness which may not be present in all applications.

Despite the discouraging results on the instability of numerical ODE methods for general high index linear systems, we can identify important subsets of high index nonlinear problems for which we can obtain positive results. These systems arise in the modeling of electrical networks and constrained mechanical systems and in the solution of the equations of fluid flow. For index two, these are semi-explicit DAE systems of the form (2.4). The system is index two if $(\partial F_2/\partial x)(\partial F_1/\partial y)$ is nonsingular. For index three systems, these are semi-explicit systems which can be written in a triangular form which includes the class of index three constrained mechanical systems described in the next section.

Gear [14] has recently noted a simple relationship between the fully implicit index one system and the semi-explicit index two system with which it is often possible to transfer methods and convergence results from one class of problems to the other.

3. Backward differentiation formulas

In this section we explore some results on the solution by BDF methods of nonlinear index one systems and some special index two and three nonlinear systems which occur frequently in practical applications. We find that BDF methods converge with the expected order of accuracy for several large classes of DAEs of practical interest.

To solve (1.1) by a BDF method, we replace $y'(t_n)$ by a k-step backward differentiation formula,

$$\rho y_n = \sum_{i=0}^{k} \alpha_i y_{n-i} , \tag{3.1}$$

to obtain the system of nonlinear equations

$$F\left(t_n, y_n, \frac{\rho y_n}{h}\right) = 0 . \tag{3.2}$$

Then we have the following result for BDF applied to index one systems [16].

THEOREM 3.1. If F is uniform index one and is differentiable with respect to y and y', the solution of (1.1) *by the k-step* BDF *method with fixed stepsize h for $k < 7$ converges to order* $O(h^k)$ *if all initial values are correct to order* $O(h^k)$.

An extension of this theorem to variable stepsizes is given in [15].

While the BDF methods converge as expected for index one problems, there are still practical difficulties in implementing these methods for this class of problems [25, 27]. One of the difficulties is to obtain a consistent set of initial conditions. We will discuss this problem in Section 5. It should be noted that systems whose index is less than or equal to one are the problems that general-purpose codes [25] are designed to handle.

For higher index systems, we have noted earlier that it is not possible to obtain convergence even of backward Euler, in general. However, we can obtain some encouraging results [22] if we restrict our attention to semi-explicit higher index systems (2.4). Systems of this form arise for example in the simulation of electrical networks, the solution of constrained mechanical systems of the form

$$M(q)q'' = f(q, q', t) + G(q)\lambda , \qquad \varphi(q) = 0 , \qquad \frac{\partial \varphi}{\partial q} = G^{t} , \tag{3.3}$$

where q are the positions and λ the Lagrange multipliers, and the solution of the finite difference or finite element approximation to the equations of fluid flow,

$$M\dot{U} + (K + N(U))U + CP = f(U, P) , \qquad C^{t}U = 0 , \tag{3.4}$$

where U is the approximation to the velocity and P approximates the pressure. The index of the mechanical systems (3.3) is three and that of the fluid flow equations (3.4) is two.

A simple example of a system of form (3.3) is given by the equations describing a pendulum in cartesian coordinates. Let L denote the length of the bar, λ the force in the bar, and x and y the coordinates of the infinitesimal ball of mass one located at the free end of the bar. Then x, y and λ solve the DAE system

$$x'' = \lambda x , \qquad y'' = \lambda y - \mathrm{g} , \qquad 0 = \frac{1}{2}(x^2 + y^2 - L^2) ,$$

where g is the gravity constant. The index of this system is three.

For semi-explicit index two systems and for index three constrained mechanical systems of the form (3.3), the k-step constant-stepsize BDF method converges to order of accuracy $O(h^k)$ if the initial values are sufficiently accurate [22, 3]. Gear et al. [15] generalized this result to show that variable stepsize BDF methods converge for semi-explicit index two systems.

It is possible to use a general purpose code based on backward differentiation formulas to solve these special high index nonlinear systems [27]. However, there are some practical difficulties which must be dealt with. It can be shown that for an index m system, the iteration matrix which the code uses in the Newton iteration for solving the nonlinear equation (3.2) has a condition number which is $O(1/h^m)$. This difficulty can be remedied by scaling the equations and the variables. The convergence test and error test must also be modified to allow a variable-stepsize BDF code to solve these types of problems.

4. Implicit Runge–Kutta methods

In this section we describe some results on the properties of implicit Runge–Kutta methods applied to index one DAE systems. These methods are potentially advantageous over multistep methods for some systems (e.g., for large systems with frequent discontinuities). However, some care must be taken in choosing an implicit Runge–Kutta method which is appropriate for DAEs, as these methods do not in general attain the same order of accuracy for DAEs as they do for ODEs.

An M-stage implicit Runge–Kutta (IRK) method applied to the system of DAEs (1.1) is written as

$$F\Big(t_{n-1} + c_i h,\ y_{n-1} + h \sum_{j=1}^{M} a_{ij} Y'_j,\ Y'_i\Big) = 0\,, \qquad y_n = y_{n-1} + h \sum_{i=1}^{m} b_i Y'_i\,. \tag{4.1}$$

The method can be written in the shorthand notation which displays the matrix of coefficients,

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1M} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_M & a_{M1} & a_{M2} & \cdots & a_{MM} \\ \hline & b_1 & b_2 & \cdots & b_M \end{array}$$

We will assume that the matrix $\mathscr{A} = (a_{ij})$ is nonsingular.

We will first give some results on the properties of implicit Runge–Kutta methods applied to different classes of index one systems, and then we will discuss the properties of these methods applied to nonlinear semi-explicit index two systems. First we need to define some terminology.

The Runge–Kutta method will be called strictly stable if the difference between a perturbed Runge–Kutta step

$$\begin{aligned} &F\Big(t_{n-1} + c_i h,\ z_{n-1} + h \sum_{j=1}^{M} a_{ij} Z'_j + \delta_n^{(i)},\ Z'_i\Big) = 0\,, \quad i = 1, 2, \ldots, M\,, \\ &z_n = z_{n-1} + h \sum_{i=1}^{M} b_i Z'_i + \delta_n^{(M+1)}\,, \end{aligned} \tag{4.2}$$

where $z_0 = y_0 + \delta_0^{(M+1)}$, $\|\delta_n^{(i)}\| \leqslant \Delta$, and an unperturbed Runge–Kutta step (4.1) satisfies $\|y_n - z_n\| \leqslant K_0 \Delta$, where $0 < h \leqslant h_0$ and K_0, h_0 are constants depending only on the method and the DAE.

Defining the stability constant $r = 1 - b^{\mathrm{t}} \mathscr{A}^{-1} \varepsilon_M$, where $\varepsilon_M = (1, 1, \ldots, 1)^{\mathrm{t}}$, it is easy to show for index one DAEs that the IRK method (4.1) is stable if and only if the method coefficients satisfy the strict stability condition $|r| < 1$. This stability condition for DAEs is related to the stability criterion $|R(z)| \leq 1$, $R(z) = 1 + zb^t(I - z\mathcal{A})^{-1}\varepsilon_M$, where $z = h\lambda$, for the stiff model problem $y' = \lambda y$ because $\lim_{|z|\to\infty} R(z) = 1 - b^t\mathcal{A}^{-1}\varepsilon_M$.

Applying the IRK method (4.1) to a canonical constant-coefficient index one system $u = g(t)$, we find that the method is locally accurate to order $O(h^{k_a+1})$ if and only if the method coefficients satisfy $A_1(k_a)$, where $A_1(w)$ is defined

$$A_1(w): \quad b^t\mathcal{A}^{-1}c^j = 1\,, \quad j = 1, 2, \ldots, w\,, \tag{4.3}$$

where $c^j = (c_1^j, c_2^j, \ldots, c_M^j)^t$. If the method satisfies the strict stability condition and the order of the IRK method for purely differential systems is k_d, then it is easy to show that the method is globally accurate to order $O(h^{\min(k_d, k_a+1)})$ for index one constant-coefficient DAE systems.

One class of IRK methods which appears to be particularly promising for the solution of DAEs is that of stiffly accurate methods. These are methods whose coefficient matrices satisfy $c_M = 1$, $a_{Mj} = b_j$, $j = 1, \ldots, M$ and $\mathcal{A}$ is nonsingular. For semi-explicit index one systems, these methods have the property which is analogous to BDF that the constraint equations are satisfied exactly at the end of each time step. Griepentrog [17] and Deuflhard et al. [11] have shown that there is no order reduction for stiffly accurate methods applied to semi-explicit index one systems.

For more general implicit Runge–Kutta methods, Roche [28] has recently derived a set of order conditions which include (4.3) using the theory of Butcher series and rooted trees, for semi-explicit index one systems.

The IRK methods are, in general, even less accurate for fully implicit nonlinear index one systems than for semi-explicit systems. The additional loss of accuracy comes about because of mixing which can occur between the errors in the differential and singular parts of the system. To state the results, we must first define some terminology.

Define

$$\begin{aligned} C(w)&: \quad \sum_{j=1}^{M} a_{ij}c_j^{k-1} = \frac{c_i^k}{k}\,, \quad i = 1, 2, \ldots, M\,, \quad k = 1, 2, \ldots, w\,, \\ B(w)&: \quad \sum_{j=1}^{M} b_j c_j^{k-1} = \frac{1}{k}\,, \quad k = 1, 2, \ldots, w\,. \end{aligned} \tag{4.4}$$

Then we have the following result, which is shown in [6].

THEOREM 4.1. Suppose that (1.1) *is uniform index one and linear in* y', *the Runge–Kutta method satisfies the stability condition* $|r| \leq 1$, *the errors in the initial conditions are* $O(h^G)$ *and the errors in terminating the Newton iterations are* $O(h^{G+\delta})$, *where* $\delta = 1$ *if* $|r| = 1$ *and* $\delta = 0$ *otherwise, and* $G \geq 2$. *Then the global errors satisfy* $\|e_n\| = O(h^G)$ *where*

$$G = \begin{cases} q\,, & \text{if } C(q) \text{ and } B(q)\,, \\ q+1\,, & \text{if } C(q),\ B(q+1) \text{ and } -1 \leq r < 1\,, \\ q+1\,, & \text{if } C(q),\ B(q+1),\ A_1(q+1) \text{ and } r = 1\,. \end{cases}$$

It should be noted that implicit Runge–Kutta methods with $|r| = 1$ are in some sense unstable. For example März [24] observes that if the implicit midpoint method is used to solve the algebraic equation $y = 0$ with initial value $y_0 = 0$, and a perturbation of size $(-1)^n\delta$ is made in each step, we obtain

$$y_{n+1} = -y_n + (-1)^n\delta .$$

Thus $|y_{n+1}| = (n+1)|\delta|$. There is a linear instability in these methods. This is why, in the theorem, the Newton iteration must be solved more precisely for $|r| = 1$ than for $|r| < 1$. Based on our experience and numerical experiments, we believe that for most machines and for most problems, this instability is not so severe that it would prevent us from considering these methods.

It should also be noted that Theorem 4.1 gives only a lower bound on the order. Thus it is possible that the observed order for some methods and some problems may exceed this lower bound. In numerical experiments described in [26, 6], we found that the lower bounds were actually observed for some methods in the stiff ODE literature. A few methods appear to perform better than the lower bounds would indicate.

These results for index one DAEs are in agreement with the order reduction results for stiff ODEs given in [7] in the limit as the stiffness becomes infinite [6].

For nonlinear semi-explicit index two systems, because of the close relationship between fully implicit index one problems and semi-explicit index two systems, we have that the global error in the 'differential' variable x is given by Theorem 4.1. The global errors in the y variable are $O(h^{G_y})$ [4], where G_y is given by

$$G_y = \begin{cases} q , & \text{if } C(q), B(q), A_1(q) , \\ q+1 , & \text{if } C(q), B(q+1), A_1(q+1), A_2(q+1) . \end{cases}$$

where A_2 are the order conditions for Runge–Kutta methods applied to constant-coefficient index two systems and are given by

$$\begin{aligned} A_2(w)\colon \quad & b^t\mathcal{A}^{-1}\varepsilon_M = b^t\mathcal{A}^{-2}c , \\ & b^t\mathcal{A}^{-2}c^i = i , \quad i = 2, 3, \ldots, w . \end{aligned}$$

Numerical experiments given in [4] show that the additional order reduction does occur for index two systems, including problems which are of interest in applications. It is difficult to find higher order IRK methods which maintain the same order of accuracy in all of the variables for an index two system.

5. Determining consistent initial conditions

Given the available software for solving DAEs with BDF methods [25, 18], often the most difficult part of solving a problem in applications is to determine a consistent set of initial conditions with which to start the computation. More precisely, we formulate this problem as follows. Given information about the initial state of the system which is sufficient (in a mathematical sense) to specify a unique solution to the DAE, determine the complete initial state $(x(t_0), x'(t_0))$ corresponding to this unique solution. For example, the user may specify information about the solution and/or its derivatives at the initial time, and the problem is to determine the remaining values of the solution and its derivatives. This work is described in more detail by Leimkuhler et al. [21].

For a simple algorithm, one might consider substituting $y = y_0$, $y' = y_0'$ and solving the resulting system, together with the user-defined information, for y_0 and y_0'. For example, for an ODE $y' = Ay$ together with user-defined information $Cy_0' + Dy_0 = q_0$ at the initial point, we have

$$\begin{bmatrix} -A & I \\ D & C \end{bmatrix} \begin{bmatrix} y_0 \\ y_0' \end{bmatrix} = \begin{bmatrix} 0 \\ q_0 \end{bmatrix}, \tag{5.1}$$

which we can solve uniquely for y_0 and y_0' provided the left-hand matrix is of full rank and q_0 is in the range of the matrix $CA + D$. In this case, these are the same conditions which the matrices D and C must satisfy to ensure that the user has given enough information about the inital state to specify a unique solution to the mathematical problem.

On the other hand, consider the index one DAE given by

$$Y_1' + Y_2' + Y_1 = g_1(t), \qquad Y_2 = g_2(t). \tag{5.2}$$

To specify a unique solution to the DAE, it is sufficient to give the value of either Y_1 or Y_1' at t_0, because Y_2 and Y_2' must satisfy the constraint and its derivative. However, evaluating Y_1, Y_1', Y_2 and Y_2' at time t_0 in (5.2), it is apparent that it is not possible to obtain $Y_1(t_0)$ uniquely if only $Y_1'(t_0)$ is specified. Thus this simple algorithm fails to give the solution, although the user has given enough information to specify a unique solution to the original problem.

Clearly the difficulty with the above procedure for equations (5.2) is that the simple algorithm has no way of obtaining the information about the derivative of the constraint which is inherent in the system. In the fully implicit index one problem it is not in general possible to isolate the constraints and differentiate them. Thus we are led to consider the following algorithm, motivated by Definition 2.1.

Solve

$$\begin{aligned} &F(t_0, y_0, y_0') = 0, \\ &\frac{\mathrm{d}F}{\mathrm{d}t}(t_0, y_0, y_0', y_0'') = 0, \\ &\vdots \\ &\frac{\mathrm{d}^m F}{\mathrm{d}t^m}(t_0, y_0, y_0', \ldots, y_0^{(m+1)}) = 0, \end{aligned} \tag{5.3}$$

coupled with the user-defined information

$$B(y_0, y_0', t_0) = 0. \tag{5.4}$$

It is easy to show that if the user-defined information (5.4) is sufficient to determine a unique solution to the DAE, then (5.3), (5.4) have a solution $(y_0, y_0', \ldots, y_0^{(m+1)})$ and the first two components y_0, y_0' are the solution to the DAE at t_0. Note that the higher derivatives of y are not determined uniquely by this algorithm.

Since it is obviously not practical to obtain the derivatives of F analytically, we are led to consider approximating the derivatives of F. Because F is possibly only defined for $t \geqslant t_0$, it is natural to consider using one-sided differences. For example, the simplest one-sided difference is given by

$$\mathrm{D}_h F = \frac{F(y_0 + hy_0', y_0' + hy_0'', t_0 + h) - F(y_0, y_0', t_0)}{h} .$$

We can define higher-order difference approximations by

$$\mathrm{D}_h F = \frac{1}{h}\left[\sum_{i=1}^{s} \alpha_i F(y_0 + hc_i y_0', y_0' + hc_i y_0'', t_0 + hc_i) - \left(\sum_{i=1}^{s} \alpha_i\right) F(y_0, y_0', t_0)\right] \tag{5.5}$$

by choosing the constants α_i, c_i appropriately. The higher derivatives of F can be approximated similarly.

The algorithm obtained by replacing the consistency equations (5.3) by their approximations (5.5), coupled with the user-defined information (5.4) produces a rank-deficient overdetermined nonlinear system. Unlike the analytic consistency equations, the approximate system may not have an exact solution because some of the derivative approximations to F may be approximating user-defined information. Thus the approximate system is solved in a least squares sense. However, the solution of this system is complicated by the rank-deficiency of the Jacobian. Using the structure of the system, it is possible to show that the minimum norm solution to this nonlinear least squares problem converges to the correct solution as the approximations D_h become more accurate, and to formulate a scheme for replacing the system with a full-rank system which has the same solution for y_0, y_0'. Numerical results for this technique appear to be promising.

6. Boundary value problems

We consider linear boundary value problems in DAEs of the form

$$Ly(t) \equiv E(t)y'(t) + F(t)y(t) = f(t) , \tag{6.1}$$

$$By(t) \equiv B_a y(a) + B_b y(b) = \beta , \quad B_a, B_b \in \mathcal{R}^{r \times n}, \ \beta \in \mathcal{R}^r , \tag{6.2}$$

where E, F, f are sufficiently smooth, and r is the dimension of the solution manifold of the homogeneous DAE $Ly = 0$. $E(t)$ is singular on $[a, b]$, possibly with variable rank, and the DAE may have index greater than one. DAE BVPs arise in the modeling of semiconductor devices [1] and in control theory [2, 9].

The literature on DAE BVPs consists mainly of the paper by März [24] and its extensions summarized in the recent text [23]. The focus of this work is on difference and shooting methods for nonlinear fully-implicit systems (1.1), coupled with the boundary conditions

$G(y(a), y(b)) = 0$, under a transferability hypothesis which guarantees that (1.1) is a uniform index one system and the nullspace of $F_{y'}$ is independent of y, y' and has constant dimension. The numerical approach in [23] requires knowledge of some projection onto $\ker(F_{y'})$ and its derivative at each mesh point. There are some theoretical results for the subclass of index two systems which are tractable, but it is implied that a successful numerical approach involves regularizing the DAE to a nonsingular or index one system and then numerically solving the regularization.

Clark and Petzold [10] investigate numerical methods for DAE BVPs which do not require the use of regularization or the computation of projection matrices at each mesh point. The results and details of the theory and methods are straightforward extensions of the methods and results for ODE BVPs in [19, 20]. Knowledge of the solution manifold is required only at the initial time point $t_0 = a$, or in the case of parallel shooting at each parallel node τ_j, and not at every mesh point t_n.

We will say that the DAE BVP (6.1), (6.2) is solvable if and only if (6.1) is a solvable DAE and for every $\beta \in \mathcal{R}^r$ there exists a unique solution y to (6.1), (6.2). Here a solvable DAE is defined as in Section 2, with some additional smoothness conditions which are needed to characterize the manifold of consistent initial conditions in the higher index case.

Let $M_f(a)$ be the set of consistent initial conditions for (6.1) at $t = a$. The initial conditions are characterized by

$$M_a y(a) = g(a) ,$$

where $\dim(\ker(M_a)) = r$ and M_a, $g(a)$ depend on E, F, f and, for higher index systems, several of their derivatives at $t = a$. For index one systems, the equation

$$[\mathrm{I} - EE^{\dagger}]Fy = [\mathrm{I} - EE^{\dagger}]f ,$$

evaluated at $t = a$, completely specifies the set of consistent initial conditions for (6.1). $E^{\dagger}$ is the Moore–Penrose inverse of E. The additional consistency requirements for the higher index case are obtained from differentiations as in Section 5. We can formulate a shooting method for the DAE BVP as follows. Let $y_p^0 \in M_f(a)$ and assume $\{\varphi_i^0\}_1^r$ is any basis for $M_0 = \ker(M_a)$. Let $y_p(t)$ and the fundamental matrix $Y(t) = [\varphi_1(t), \ldots, \varphi_r(t)]$ be the solutions to the $r + 1$ initial value problems

$$Ly_p(t) = f(t) , \qquad y_p(a) = y_p^0 \in M_f(a) , \tag{6.3}$$

$$LY(t) = 0 , \qquad Y(a) = Y_0 = [\varphi_1^0, \ldots, \varphi_r^0] . \tag{6.4}$$

$Y(t)$ has full column rank for all $t \in I$ since (6.1) is a solvable system. Using the fact that

$$y(t) = y_p(t) + \sum_{i=1}^{r} \psi_i \varphi_i(t) \tag{6.5}$$

and imposing the boundary condition (6.2), we find that y is a solution of (6.1), (6.2) if and only if the vector $\psi = (\psi_1, \ldots, \psi_r)^{\mathrm{t}}$ satisfies

$$[B_aY_0 + B_bY(b)]\psi = \beta - (B_ay_p(a) + B_by_p(b)) . \tag{6.6}$$

As in the ODE case, the $r \times r$ matrix

$$S = B_aY_0 + B_bY(b) \tag{6.7}$$

is the shooting matrix for (6.1), (6.2) and is unique up to a change of basis for M_0. Thus we have the BVP is solvable if and only if S is invertible. The desired solution is given by (6.5), where y_p^0, Y satisfy (6.3), (6.4) and ψ is the solution to (6.6). For the situation of partially separated end conditions, if either B_a or B_b are rank-deficient, it is possible to formulate the algorithm so that the number of initial value problems to be solved can be reduced to $q + 1$, where $q = \min(\text{rank}(B_a), \text{rank}(B_b))$.

It is possible to formulate variations of parallel shooting which yield full-rank linear systems, and to show that if the underlying initial value method using in the shooting technique is globally accurate to $O(h^k)$, then the numerical solution to the BVP is accurate to $O(h^k)$. These results are valid even if the index is larger than one.

For finite difference methods applied to fully implicit index one BVP systems, we can formulate a BVP method by appending the consistency condition $\hat{M}_ay(a) = \hat{g}(a)$, where $\hat{M}_a$ is a set of $n - r$ linearly independent rows of M_a, and $\hat{g}(a)$ are the corresponding rows of $g(a)$, to the boundary conditions. Then it is possible to show, analogously to the ODE case [20] that the resulting DAE BVP methods are accurate to order $O(h^k)$ if and only if the underlying IVP methods are globally accurate to $O(h^k)$. The linear systems to be solved in the method are square and nonsingular. The Runge–Kutta methods of Section 4 with $|r| = 1$ are important in this context. We are currently studying a similar method for dealing with semi-explicit index two systems directly.

References

[1] U. Ascher, On numerical differential algebraic problems with application to semiconductor device simulation, SIAM J. Numer. Anal., to appear.

[2] J. Betts, et al., Solving the optimal control problem using a nonlinear programming technique, part I: general formulation, AIAA-84-2037, Proc. 1984 AIAA/AAS Astrodynamics Conference, Seattle, WA, 1984.

[3] K.E. Brenan and B. Engquist, Backward difference approximations of nonlinear differential-algebraic equations, ATR-85 (9990)-5, The Aerospace Corporation, El Segundo, California, July 1985.

[4] K.E. Brenan and L.R. Petzold, The numerical solution of higher index differential/algebraic equations by implicit Runge-Kutta methods, SIAM J. Numer. Anal., to appear.

[5] K. Burrage, Efficiently implementable algebraically stable Runge-Kutta methods, SIAM J. Numer. Anal. 19 (1982) 245–258.

[6] K. Burrage and L.R. Petzold, On order reduction for Runge-Kutta methods applied to differential/algebraic systems and to stiff systems of ODEs, SIAM J. Numer. Anal., to appear.

[7] K. Burrage, W.H. Hundsdorfer and J.A. Verwer, A study of B-convergence of Runge-Kutta methods applied to stiff systems, Computing 36 (1986) 17–34.

[8] S.L. Campbell, A general form for solvable linear time varying singular systems of differential equations, SIAM J. Math. Anal. 18 (1987) 1101–1115.

[9] K. Clark, A structural form for higher index semistate equations I: Theory and applications to circuit and control theory, Linear Algebra Appl. 98 (1988) 169–197.

[10] K. Clark and L.R. Petzold, Numerical solution of boundary value problems in differential/algebraic systems, SIAM J. Sci. Stat. Comp., to appear.

[11] P. Deuflhard, E. Hairer and J. Zugck, One-step and extrapolation methods for differential-algebraic systems, Preprint Nr. 318, Universität Heidelberg, Institut für Angewandte Mathematik, 1985.

[12] F.R. Gantmacher, The Theory of Matrices, Vol. 2 (Chelsea, New York, 1964).

[13] C.W. Gear, Simultaneous numerical solution of differential/algebraic equations, IEEE Trans. Circuit Theory 18 (1971) 89–95.

[14] C.W. Gear, Differential-algebraic equation index transformations, SIAM J. Sci. Stat. Comput. 9 (1988) 39–47.

[15] C.W. Gear, B. Leimkuhler and G.K. Gupta, Automatic integration of Euler-Lagrange equations with constraints, J. Comput. Appl. Math. 12 and 13 (1985) 77–90.

[16] C.W. Gear and L.R. Petzold, ODE methods for the solution of differential/algebraic systems, SIAM J. Numer. Anal. 21 (1984) 367–384.

[17] E. Griepentrog, Initial value problems for implicit differential equation systems, Humboldt-Universität zu Berlin, reprint, 1984.

[18] A.C. Hindmarsh, ODE solvers for use with the method of lines, in: R. Vichnevetsky and R.S. Stepleman, eds., Advances in Computer Methods for Partial Differential Equations IV (IMACS, New Brunswick, NJ, 1981) 312–316.

[19] H.B. Keller, Approximation methods for nonlinear problems with application to two-point boundary value problems, Math. Comp. 29 (1975) 464–474.

[20] H.B. Keller, Numerical solution of two-point boundary value problems, in: Regional Conference Series in Applied Mathematics, Vol. 24 (SIAM, Philadelphia, 1976).

[21] B.J. Leimkuhler, L.R. Petzold and C.W. Gear, On the consistent initialization of differential-algebraic equations, SIAM J. Numer. Anal., to appear.

[22] P. Lötstedt and L. Petzold, Numerical solution of nonlinear differential equations with algebraic constraints I: Convergence results for backward differentiation formulas, Math. Comp. 46 (1986) 491–516.

[23] E. Griepentrog and R. März, Differential-Algebraic Equations and their Numerical Treatment (Teubner-Texte, Leipzig, 1986).

[24] R. März, On difference and shooting methods for boundary value problems in differential-algebraic equations, preprint No. 24, Humboldt-Universität zu Berlin, Berlin, 1982.

[25] L.R. Petzold, A description of DASSL: A differential/algebraic system solver, in: R.S. Stepleman, ed., IMACS Trans. on Scientific Computation, Vol. 1 (North-Holland, Amsterdam, 1982).

[26] L.R. Petzold, Order results for implicit Runge-Kutta methods applied to differential/algebraic systems, SIAM J. Numer. Anal. 23 (1986) 837–852.

[27] L. Petzold and P. Lötstedt, Numerical solution of nonlinear differential equations with algebraic constraints II: Practical implications, SIAM J. Sci. Statist. Comput. 7 (1986) 720–733.

[28] M. Roche, Implicit Runge-Kutta methods for differential algebraic equations, SIAM J. Numer. Anal., to appear.

[29] R.F. Sincovec, B. Dembart, M.A. Epton, A.M. Erisman, J.W. Manke and E.L. Yip, Solvability of Large-Scale Descriptor Systems (Boeing Computer Services Company, Seattle, Washington, 1979).