Contemporary Mathematics
Volume 47, 1985

Rank Deficient Least Squares and the Numerical Solution of Linear Singular Implicit Systems of Differential Equations

Stephen L. Campbell[1]

ABSTRACT. An approach for the numerical solution of linear systems of differential equations of the form $A(t)x'(t) + B(t)x(t) = f(t)$ with $A(t)$ singular is discussed. The key step in this approach is the solution of a rank deficient least squares problem. The solution of this least squares problem is investigated. These results are then applied to the system of differential equations.

1. INTRODUCTION. Many models involving electrical, mechanical, economic, or control systems are most naturally initially formulated as an implicit system of differential equations [1], [2], [3], [8], [10], [17], [18], [21]

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

Systems such as (1.1) also arise when solving partial differential equations by the method of lines. It is often desirable to work with (1.1) directly rather than to try to compute an equivalent state space form

$$z' = G(t,z). \tag{1.2}$$

This may be done to preserve sparsity, because the variables are of physical interest, or because (1.2) does not even exist [3].

We are specifically interested in the case when $F_{x'} = \frac{\partial F}{\partial x'}$, is always singular. Such systems are referred to as singular, degenerate, differential-algebraic, descriptor, generalized state space, non-canonic, constrained, or semi-state systems. See [2], [3], [8] for additional applications, references, and results.

The theory for the linear time-invariant problem

$$Ax' + Bx = f \tag{1.3}$$

is reasonably complete. However, until the last few years the linear time varying case

1980 Mathematics Subject Classification. 34A08, 65L05, 65F20

[1]Research supported in part by the National Science Foundation under Grant No. DMS-8318026 and the Air Force Office of Scientific Research under Grant No. 84-0240.

$$A(t)x'(t) + B(t)x(t) = f(t) \tag{1.4}$$

was largely unexamined. The system (1.4) has recently been examined more carefully [5], [6], [11]. One discovery has been that implicit backward difference formulas (IBDF) need not work on (1.4) [14]. Recently [4] an approach for numerically solving (1.4) has been proposed that works in all cases where (1.4) is known to be solvable (defined in the next section). The key step in this approach is the solution of a rank deficient least squares problem.

This paper has three purposes. One is to bring these somewhat unusual linear algebra problems to the attention of those interested in the interaction of linear algebra, numerical analysis, and systems theory. The second is to investigate some of the numerical linear algebra problems in implementing this approach. The third is to develop a better way to apply this approach of [4] to (1.4) than the Taylor methods given in [4]. Our intention is to address general questions rather than derive detailed error estimates. The actual implementation of many of the ideas in this paper would necessitate such a finer error analysis.

2. PROBLEM NOTATION AND TERMS. The system (1.4) is <u>solvable</u> on an interval I if for every sufficiently smooth f(t) there exists a smooth solution to (1.4) on I and the solutions for a given f are uniquely determined by their value at any $t_0 \in I$. Some algorithms that, in principle, determine solvability are found in [5], [6], [11]. We assume that (1.4) is solvable unless we state otherwise and that A(t), B(t) are $n \times n$.

An initial condition x^0 at t_0 is <u>consistent</u> (for a given f) if there exists a solution to (1.4) such that $x(t_0) = x^0$.

We also assume that A(t) is singular for all $t \in I$ and that the matrix pencil $\lambda A(t) + B(t)$ for each t is either not regular ($\det(\lambda A(t) + B(t)) = 0$ for all λ) or has index ≥ 2 (is <u>higher index</u>). That is, $(\lambda A(t) + B(t))^{-1}A(t)$ contains a nonzero nilpotent Jordan block in its Jordan cannonical form of size $k(t) \geq 2$. The function k(t) is called the index of (1.4). These problems have also been referred to as algebraically incomplete [20], or in optimal control theory as higher order.

It is known that, in general, the solution of (1.4) can involve not only derivatives of f but derivatives of the coefficients A(t), B(t) [5]. Thus some differentiation is necessary. Our goal is to do the differentiation on those terms which are explicitly known, in this case the original A(t), B(t), f(t), rather than some transformed version of them.

Proceeding as in [4] assume (1.4) is solvable and we are trying to find the solution x(t). Assume also that at time $\hat{t}$ we have computed $x(\hat{t})$. We wish to estimate $x(\hat{t} + h)$. Let

$$x(t) = \sum_{i=0} x_i\delta^i, \quad A(t) = \sum_{i=0} A_i\delta^i, \quad B(t) = \sum_{i=0} B_i\delta^i$$
$$f = \sum_{i=0} f_i\delta^i \tag{2.1}$$

where $\delta = t - \hat{t}$, so that $c_i = \frac{c^{(i)}(\hat{t})}{i!}$ where $c = x,A,B,f$ are the usual Taylor coefficients at $\hat{t}$. The series will be considered infinite but only the first few terms are needed and hence A,B,f,x need only be sufficiently differentiable. By substituting the expansions (2.1) into (1.4) and equating like terms we see that any solution of (1.4) must satisfy

$$\begin{bmatrix} A_0 & 0 & \cdot\ \cdot & 0 \\ A_1 + B_0 & 2A_0 & \cdot\ \cdot\ \cdot & \cdot \\ \cdot & 2A_1 + B_0 & \cdot\ \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot\ \cdot & 0 \\ A_{j-1} + B_{j-2} & \cdot & \cdot\ \cdot & jA_0 \end{bmatrix} \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_j \end{bmatrix} = \begin{bmatrix} f_0 - B_0x_0 \\ \cdot \\ \cdot \\ \cdot \\ f_{j-1} - B_{j-1}x_0 \end{bmatrix} \tag{2.2}$$

for any $j \geq 1$. System (2.2) will be denoted by

$$\mathcal{A}_j\underline{x}_j = \underline{f}_j, \quad j \geq 1. \tag{2.3}$$

Notice that this equation is to hold exactly and that $\mathcal{A}_j$ is a square $jn \times jn$ singular matrix since $A_0 = A(t)$ is singular so that (2.3) does not uniquely determine $\underline{x}_j$.

The first z components of a consistent system of equations $Eu = b$ are uniquely determined if and only if there is a nonsingular matrix R such that

$$RE = \begin{bmatrix} I_{z\times z} & 0 \\ 0 & H \end{bmatrix}. \tag{2.4}$$

Equivalently, the first z columns of E are linearly independent, and the first z columns are linearly independent from the remaining columns.

We shall say $\mathcal{A}_j$ is <u>s-full</u> if (2.4) holds with $z = sn$. Thus (2.3) uniquely determines $x_1, \ldots, x_s$ if and only if it is r-full with $r \geq s$. If the entries of $\mathcal{A}_j$ are smooth functions of t and R can also be taken to be a smooth function of t, we shall call $\mathcal{A}_j$ <u>smoothly</u> <u>s-full</u>.

There are many ways to solve (2.3) since solutions are nonunique. We shall solve it in the least squares sense so that the solution may be denoted by $\mathcal{A}_j^{\dagger}\underline{f}_j$ where $\mathcal{A}_j^{\dagger}$ is the Moore-Penrose inverse [10]. Thus numerous packages for its computation exist. Additional reasons for this choice will be examined later in this paper.

To complete our background development we quote two results from [4].

THEROEM 2.1. If A_j is 1-full for $t \in I$ and there exists a continuous invertible $R(t)$ so that

$$R(t)A_j(t) = \begin{bmatrix} I_{n\times n} & 0 \\ 0 & H(t) \end{bmatrix},$$

let

$$R(t) = \begin{bmatrix} R_{11} & \cdots & R_{1j} \\ R_{21} & \cdots & R_{2j} \end{bmatrix} \text{ where each } R_{1i} \text{ is } n \times n.$$

Then every solution of (1.4) is a solution of

$$x' = Q(t)x + q(t) \tag{2.5}$$

where

$$Q(t) = -\sum_{i=0}^{j-1} R_{1,i+1}(t)B_i(t),\ q(t) = \sum_{i=0}^{j-1} R_{1,i+1}(t)f_i(t). \tag{2.6}$$

PROPOSITION 2.1. Suppose that A_j is 1-full for $t \in I$ and that rank (A_j) is constant on I. Then the first n rows of $A_j^{+}(t)$, may be used for $[R_{11}\ldots R_{ij}]$.

In [4], it is shown that for all known classes of solvable systems A_j is smoothly 1-full and rank A_j is constant provided $j \geq k + 1$ for all t.

At this point [4] proceeded to develop a family of Taylor methods. However, as first noted in [7], this is not the best way to proceed.

Solving the linear system (2.3) for a given x_0, t_0 is, in fact, computing $x'(t_0)$ or equivalently $Q(t_0)x(t_0) + q(t_0)$ in (2.5). Thus it is possible to apply almost any of the standard one step or multistep methods to (1.4) by applying them to (2.5) and solving (2.3) to evaluate the right hand side of (2.5). The error in solving (2.3) becomes the function evaluation error when analyzing the ordinary differential equation solver.

Section 3 will examine the numerical solution of (2.3). Section 4 discusses some of the considerations in choosing an ordinary differential equation solver.

3. SOLUTION OF (2.3). The algebraic problem (2.3) is unusual in several respects. First, the coefficient matrix is square and singular. Secondly, we want only x_1, the first n components of $\underline{x}_j$. Finally, there is a definite structure to (2.3).

For any of the numerical procedures for solving (1.4) developed in the next section we will know x_0 to $O(h^r)$ accuracy and will need to compute x_1 to

$O(h^r)$ accuracy so that our updated value will also be $O(h^r)$.

We shall assume that A_j has constant rank on I and thus there is a number $\sigma_0 > 0$ such that the largest and smallest singular values of A_j, $\sigma_m(t)$, $\sigma_s(t)$ satisfy $\sigma_m(t) \geq \sigma_s(t) \geq \sigma_0 > 0$ for all $t \in I$. As noted earlier this assumption on the rank of A_j does not appear to be restrictive. For convenience, we assume $\sigma_0 < 1$; $h < 1$.

The solution of (2.3) is a major portion of the computational effort in using the method of this paper. If the A_i, B_i are needed explicitly then that limits the size of the systems and the indexes that can be practically considered (since $j \geq k + 1$). Thus it is desirable to sometimes be able to estimate the A_i, B_i. Also there will be errors present in any numerical calculations. To examine the effect of these estimation and computational errors suppose that we are actually solving exactly

$$\tilde{A}_j \tilde{\underline{x}}_j = \tilde{\underline{f}}_j \tag{3.1}$$

where $\tilde{A}_j = A_j + \varepsilon_1$, $\tilde{\underline{f}}_j = \underline{f}_j + \varepsilon_2$.

Here ε_1, ε_2 are a matrix and vector perturbation respectively. In the calculations that follow $||\cdot||$ denotes a matrix norm and we assume $||\varepsilon_1||$, $||\varepsilon_2|| \leq 1$. In order to emphasize the role of σ_s we do not incorporate it into the order estimates. If rank $\tilde{A}_j$ = rank A_j, then

$$\tilde{\underline{x}}_j = \tilde{A}_j^{+} \tilde{\underline{f}}_j = (A_j^{+} + \frac{1}{\sigma_s^2} O(||\varepsilon_1||))(\underline{f}_j + \varepsilon_2)$$

$$= A_j^{+} \underline{f}_j + \frac{1}{\sigma_s^2} O(||\varepsilon_1||)\underline{f}_j + \frac{1}{\sigma_s} O(||\varepsilon_2||) + \frac{1}{\sigma_s^2} O(||\varepsilon_1||)\, O(||\varepsilon||)$$

Thus to have $\tilde{x}_1 - x_1 = O(h^r)$ it suffices to have $\varepsilon_1 = \sigma_s^2\, O(h^r)$, $\varepsilon_2 = \sigma_s O(h^r)$. Of course $\sigma_s\, O(h^r) = O(h^r)$ but we wish to stress that if σ_s is "small" then ε_1 may, in fact, need to be taken smaller than ε_2 for a given stepsize.

This qualitative argument suggests that if $\sigma_M/\sigma_0 < h^r$ (σ_M = machine or rounding error), the A_i are estimated to $\sigma_0^2\; O(h^r)$ accuracy, the B_i, f_i are estimated to $\sigma_0\; O(h^r)$ accuracy, then as long as the theoretical rank of A_j does not change, then the estimate of x_1 will also be $O(h^r)$ accurate.

Unfortunately it can be shown that only a few of the higher A_i, B_i in the lower left hand corner of A_j can be estimated and not alter the rank of A_j, [9]. In general, as examples in [9] show, any estimation of A_1, A_2, B_1, will alter the rank. Normally this would present major numerical problems. However, there is a way around this difficulty. The theoretical rank of $A_j(t)$ is being assumed constant. Thus a careful determination of the rank using the singular value decomposition (SVD) need be done only initially at time t_0. Call this rank r_0. At the same time $\sigma_s(t_0)$ can be found as an initial estimate of σ_0.

We are led then to consider (2.3) where the rank is known. Note that we are not assuming $A(t) = A_0(t)$ has constant rank. Also ε_1 has a definite pattern and thus is what Stewart refers to as a "highly structured error" [22].

We see then that if the rank is considered known then one should be able to estimate the coefficients A_i, B_i and get an $O(h^r)$ estimate of x_1 provided the estimation error ε_1 satisfies

$$||\varepsilon_1|| \leq m_1 = \min(\sigma_0^2\, O(h^r),\ \sigma_0) \tag{3.2}$$

and the error ε_2 satisfies

$$||\varepsilon_2|| \leq m_2 = \min(\sigma_0\, O(h^r),\ O(h^r)) \tag{3.3}$$

and $m_1, m_2 \geq \sigma_M$. If σ_0 is a small number, say $\sigma_0 \approx h$ for a step size h being considered, then (3.2), (3.3) suggest that the B_i, f_i need to be found to less accuracy than the A_i. Of course (3.2), (3.3) are really order estimates so that the estimation error ε_1 should be significantly less than σ_0 (in order to not alter the numerical rank) and ε_i should be less than m_i for $i = 1, 2$.

When the A_i, B_i are known analytically, the approach of this paper has two advantages over implicit backward differences (IBDF) [13], [14]. One, the IBDF need not work on all solvable systems. Secondly, the matrices in the IBDF become more ill conditioned as the step size decreases, while the conditioning of (2.3) is independent of step size. Thus a larger index puts more severe restrictions on the smallness of the step size from IBDF than for the approach of [4]. The discussion of the preceeding section, however, suggests that in order to estimate the A_i, B_i they must be estimated to higher accuracy than σ_0 and at least the accuracy of the differential equation solver. Suppose, for simplicity that (1.4) has index k and that $j = k + 1$ in (2.3). Then we require a $k^{\underline{th}}$ derivative of A in A_j and a $k^{\underline{th}}$ derivative of B, f in $\underline{f}_j$. If A_{k+1} is computed using a simple difference scheme with step size ρ, then the error will be approximately

$$O(\rho) + \frac{\varepsilon_M}{\rho^k}\,. \tag{3.4}$$

Now from (3.2) we need (approximately)

$$O(\rho) + \frac{\varepsilon_M}{\rho^k} < \sigma_0$$

$$O(\rho) + \frac{\varepsilon_M}{\rho^k} < \sigma_0^2\, O(h^r)$$

or

$$\varepsilon_M < \rho^k(\sigma_o - O(\rho)) \tag{3.5}$$

and

$$\varepsilon_M < \rho^k(\sigma_o^2\, O(h^r) - O(\rho)) \tag{3.6}$$

Taking $\rho < \sigma_o$, $\rho < h^r\sigma_o^2$ and assuming $h^r < \sigma_o \leq 1$ we get (3.5), (3.6) imply the order inequalities $\sigma_M < \rho^k\sigma_o$,

$$\varepsilon_M < \rho^k h^r \sigma_o^2 < (\sigma_o^2 h^r)^k h^r \sigma_o^2 \leq h^{r(k+1)} \tag{3.7}$$

which is putting a restriction the step size h, and order r of the ODE solver. This restriction is more severe than the one for an IBDF which is $\varepsilon_M < h^{k+r}$. However, if say Richardson extrapolation is used in finding the A_i, B_i we can get, in principle, $O(\rho^s) + \frac{\varepsilon_M}{\rho^k}$ if $\rho^s > \varepsilon_M/\rho^k$. Then (3.5), (3.6) become

$$\varepsilon_M < \rho^k(\sigma_o - O(\rho^s)) \tag{3.8}$$

$$\varepsilon_M < \rho^k(\sigma_o^2 O(h^r) - O(\rho^s)). \tag{3.9}$$

Taking $\rho^s < h^r\sigma_o^2$, $\rho^s < \sigma_o$ we get

$$\varepsilon_M < \rho^k h^r \sigma_o^2 < h^{\frac{kr}{s}} h^r = h^{r+rk/s}. \tag{3.10}$$

Now consider r, k fixed. Then by taking ρ's closer to one we may increase s and still satisfy the constraints so that asymptotically we have $\frac{k}{s} \approx 0$ or

$$\varepsilon_M < h^r \tag{3.11}$$

which is the restriction on h, r for an ordinary explicit ODE solver. To summarize, it may be possible to attain higher accuracy than an IBDF and still estimate the coefficients of A_j. But it will require utilizing, for example a number of Richardson extrapolations at larger step sizes, or some other procedure that provides high order accuracy in the estimates.

It should be noted in (3.7) that the k is really the number of numerical differentiations and not the index. Thus if we could compute $A_o,\dots,A_\ell$, $B_o,\dots,B_\ell$, $f_o,\dots,f_\ell$ analytically, and then obtain the rest numerically by differencing we would replace $\varepsilon_M\rho^{-k}$ by $\varepsilon_M\rho^{-k+\ell}$ in all of the calculations. Thus the restriction in (3.7) becomes

$$\varepsilon_M < (h^{k+1-\ell})^r. \tag{3.12}$$

Similarly (3.10) is

$$\epsilon_M < h^{r+r(k-\ell)/s}. \tag{3.13}$$

Suppose now that we have arrived at (2.3) and wish to solve it at time $\hat{t}$. Assume that the SVD has been used earlier to determine the rank r_0. Assume also that $\sigma_0 >> \sigma_M$ and $A_j(\hat{t})$ has been computed to sufficient accuracy as discussed earlier. We wish to discuss the solution of (2.3). In [16], essentially two methods are proposed; QR with column pivoting and the SVD. As noted in [16] the QR algorithm is not as reliable as the SVD. However, it frequently works well in practice. Sparse algorithms are given in [15].

We shall discuss the QR method first. One option is to just apply the QR algorithm with full column pivoting [16, Algorithm 6.4-1]. However, because of the special nature of the problem to be solved here it seems natural to suggest several modifications. It should be stressed these are suggestions. Their actual efficiency and stability remain to be examined and tested.

Since we are only interested in x_1 and by assumption A_j is 1-full we can modify the pivoting strategy as follows. For the first n steps only consider the first n columns when interchanging columns. After that only consider the last $(j - 1)n$ columns. There is sometimes no need to save the last $(j - 1)n$ pivoting strategies. We arrive then at

$$\begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ 0 \end{bmatrix} \tag{3.14}$$

where R_{11}, R_{22} are nonsingular and upper triangular and the lower row of zeros is computed on the basis of our knowledge of r_0.

Now there are several ways to proceed. One is to apply Householder transformations on the right to

$$\begin{bmatrix} R_{12} & R_{13} \\ R_{22} & R_{23} \end{bmatrix}$$

to yield

$$\begin{bmatrix} R_{11} & \tilde{R}_{12} & 0 \\ 0 & \tilde{R}_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{z}_1 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \\ 0 \end{bmatrix} \tag{3.15}$$

and $\tilde{R}_{22}$, R_{11} are upper triangular. $\tilde{R}_{23} = 0$ by the choice of Householder transformation. $\tilde{R}_{13} = 0$ by the assumption of 1-fullness for A_j. Then $\tilde{z}_1$ and $\tilde{x}_1$ may be solved for by back substitution. The entries of $\tilde{x}_1$ are then permuted to reflect the pivoting to give x_1.

Since the entries of A_j are assumed to be smooth functions of t, a moderate savings of effort can be made by not updating pivoting strategies at every time step.

Return now to (3.14). The basic solution [12] $\tilde{x}_{1B}$, $\tilde{z}_{1B}$ to (3.14) is the solution of

$$R_{11}\tilde{x}_{1B} = \hat{f}_1 - R_{12}\tilde{z}_{1B} \qquad (3.16)$$

$$R_{22}\tilde{z}_{1B} = \hat{f}_2.$$

If all our calculations had been exact, then since A_j is 1-full, we have that once the entries of $\tilde{x}_{1B}$ are unpivoted that $x_1 = x_{1B}$ also. However, at this time we have not had a chance to investigate this idea fully. Since setting $z_2 = 0$ is no longer a unitary operation, there is a potential for increasing the error in x_1. However, the basic solution is less effort to compute and is often sufficiently accurate. Theoretically $\tilde{x}_1 = \tilde{x}_{1B}$ if A_j is 1-full. To get an intuitive understanding of how $\tilde{x}_1$ and $\tilde{x}_{1B}$ could differ numerically, note from (3.16), (3.17) that

$$\tilde{x}_1 - \tilde{x}_{1B} = R_{11}^{-1}(R_{12}R_{22}^{-1} - \tilde{R}_{12}\tilde{R}_{22}^{-1})\hat{f}_2. \qquad (3.17)$$

The zeroing of R_{23} by Householder transformations causes an increase in the absolute value of the diagonal entries of $\tilde{R}_{22}$. The same operations increase (in the appropriate norm), R_{12}. Thus a possible counter indication for the use of the basic solution is an R_{23} which is large relative to the diagonal of R_{22} and a large R_{13} since under these conditions $\tilde{R}_{12}$, $\tilde{R}_{22}$ will differ "greatly" from R_{12}, R_{22}.

If σ_0 is small enough that it is at all close to the estimation error or the desired accuracy then it is probably essential to use the SVD.

The preceeding comments are applicable if we not only want x_1 but also x_2 or x_3, provided j is also increased.

4. CHOICE OF ODE SOLVER. There exist many numerical methods that may be applied to (2.5). However, there are several important considerations. Among them are;

> If estimation of the A_j is being done the computation of A_j for a given $\hat{t}$ can be expensive. (4.1)

> The calculation of x_1 for a given x_0, t is probably the most expensive part of most of these methods. (4.2)

Suppose that we were to want to attempt to construct a Runge-Kutta (RK) method. Then it is necessary to find x_1 for two x_0 at the same time. If we know both x_0 at the same time this would not be much of a difficulty. However in a Runge-Kutta method the second x_0 value depends on the first. Thus it will be necessary to save the QR (or SVD) factorization of A_j so that it can be reused the second time. This increases the storage requirements. In what follows let us agree that compute $Q(t)$, $q(t)$, form $Q(t)x(t) + q(t)$, etc., means to solve (2.3) for x_1 and save the information needed if we need to solve for x_1 again given a different x_0 at the same $\hat{t}$.

One could then define a fourth order Runge-Kutta method as follows. Suppose at time t_i we have already available $Q(t_i)$, $q(t_i)$. Let $t_{i+1} = t_i + h$ and compute

$$F_1 = x_1^i = Q(t_i)x(t_i) + q(t_i).$$

Then compute $Q(t_i + \frac{h}{2})$, $q(t_i + \frac{h}{2})$ and set

$$F_2 = Q(t_i + \tfrac{h}{2})(x(t_i) + \tfrac{h}{2} F_1) + q(t_i + \tfrac{h}{2})$$

$$F_3 = Q(t_i + \tfrac{h}{2})(x(t_i) + \tfrac{h}{2} F_2) + q(t_i + \tfrac{h}{2}).$$

Next compute $Q(t_i + h)$, $q(t_i + h)$ and set

$$F_4 = Q(t_i + h)(x(t_i) + hF_3) + q(t_i + h).$$

Finally let

$$x(t_i + h) = x(t_i) + \frac{h}{6}(F_1 + 2F_2 + 2F_3 + F_4).$$

Less computational effort however would be to define an Adams-Bashforth Method. Using a fixed step size, compute starting values with a one step method (Taylor or RK). Then let $x_1^i = Q(t_i)x(t_i) + q(t_i)$, $t_{i+1} = t_i + h$, be the value of x_1 found from (2.3) where (2.3) is 1-full. Then use one of

$$x(t_{i+1}) = x(t_i) + \frac{h}{2}(3x_1^i - x_1^{i-1}) \quad (2^{\underline{nd}} \text{ order}) \tag{4.3}$$

$$x(t_{i+1}) = x(t_i) + \frac{h}{12}(23x_1^i - 16x_1^{i-1} + 5x_1^{i-2}) \quad (3^{\underline{rd}} \text{ order}) \tag{4.4}$$

$$x(t_{i+1}) = x(t_i) + \frac{h}{24}(55x_1^i - 59x_1^{i-1} + 37x_1^{i-2} + 9x_1^{i-3}) \quad (4^{\underline{th}} \text{ order}) \tag{4.5}$$

to estimate succeeding values of $x(t)$. In terms of computational effort this is substantially better than the Taylor methods of [4] since a $4^{\underline{th}}$ order method is being based on the solution of (2.3) of size $jn \times jn$ instead of $(j + 3)n \times (j + 3)n$. As long as a fixed step size is being used, the Adams method (4.5) is clearly less work than the Runge-Kutta method given. The RK however is self starting and could be used to generate initial values of (4.5) after a step size change. We now show that unless the index is at least four, the RK method is probably better than the Taylor method in terms of amount of effort. Counting only function evaluations, for each time step the RK performs the equivalent of c evaluations where $2 \le c \le 4$. The exact value of c remains to be determined. Since each evaluation is the least squares solution of a $jn \times jn$ matrix we get our computational estimate as

$$c(jn)^3 L \tag{4.6}$$

for a constant L [16]. A fourth order Taylor method does one evaluation on a $(j + 3)n \times (j + 3)n$ matrix to give an estimate of

$$[(j + 3)n]^3 L. \tag{4.7}$$

Using $2 \le c \le 4$ we see that the RK method is definitely less work until $j \ge 5$ and may be less effort for j as high as 11. This suggests that all other considerations aside, the RK would often be preferable to the Taylor method in practice.

Since the Adams-Moulton methods have larger stability regions they are sometimes preferable to the Adams-Bashforth at a cost of somewhat more computation. For example one could proceed as follows to define Adams-Moulton methods.

Use an Adams-Bashforth to predict $x(t_i + h) = \hat{x}(t_{i+1})$. Using $\hat{x}(t_{i+1})$ compute $\hat{x}_1^{i+1} = Q(t_{i+1})\hat{x}(t_{i+1}) + q(t_{i+1})$ from (2.3). Save the values of $Q(t_{i+1})$, $q(t_{i+1})$ for the next prediction step. Then use a corrector to get the final estimate of $x(t_{i+1})$. Two common correctors are

$$x(t_{i+1}) = x(t_i) + \frac{h}{2}(\hat{x}_1^{i+1} + x_1^i) \quad (2^{\underline{nd}} \text{ order}) \tag{4.8}$$

and

$$x(t_{i+1}) = x(t_i) + \frac{h}{24}(9\hat{x}_1^{i+1} + 19x_1^i - 5x_1^{i-1} + x_1^{i-2}) \quad (4^{\underline{th}} \text{ order}). \tag{4.9}$$

The Adams-Moulton is more work since it involves computing $\hat{x}_1^i$, x_1^i for each step i. However, as noted earlier this is not twice as much work.

If the equation (2.5) is stiff, then it might be desirable to use implicit methods on (2.5). Note that this might require actually computing Q(t) and not just Q(t)x(t).

5. REMAINING QUESTIONS. There remains, to our way of thinking, several major unresolved problems of varying difficulty concerning the approach of this paper and singular systems.

PROBLEM 1. Step size control strategies need to be developed. By increasing j, A_j can be made 2-full and then solved to yield not only x_1, but $x_2 = \frac{x''}{2}(\hat{t})$. This and other alternatives need to be examined.

PROBLEM 2. Reducing the effort in solving (2.3) is important. This might be done by exploiting the zero and block structure of the matrix A_j. The best way to solve (2.3) for two x_0 and the same $\hat{t}$ needs to be looked at more carefully if Adams-Moulton or Runge-Kutta methods are to be used.

PROBLEM 3. The choice of method is affected by the structure of Q(t). It would be nice if the important properties of Q(t), such as location of eigenvalues, could be related to properties of A_j in (2.3). This is probably the hardest of the problems.

PROBLEM 4. Good general methods for the calculation of consistent initial conditions need to be developed.

PROBLEM 5. The problem of determining a priori from the structure of A(t), B(t) whether (1.4) is solvable and whether a IBDF will work is not fully resolved.

Many other options besides those given in Section 4 are possible, for example, variable step size methods. The effect of the evaluation costs imposed by (2.3) is far from being fully delineated.

BIBLIOGRAPHY

1. K. E. Brenan, Stability and Convergence of Difference Approximations for Higher Index Differential-Algebraic Systems with Applications in Trajectory Control, Ph.D. Thesis, University of California, Los Angeles, 1984.

2. S. L. Campbell, Singular Systems of Differential Equations, Pitman, Marshfield, MA, 1980.

3. S. L. Campbell, Singular Systems of Differential Equations II, Pitman, Marshfield, MA, 1982.

4. S. L. Campbell, "The numerical solution of higher index linear time varying singular systems of differential equations", SIAM J. Sci. Stat. Comp. (to appear).

5. S. L. Campbell, "Index two linear time-varying singular systems of differential equations", SIAM J. Alg. & Discrete Methods, 4 (1983), 237-243.

6. S. L. Campbell, "One canonical form for higher-index linear time-varying singular systems", Circuits, Systems & Signal Processing, 2 (1983), 311-325.

7. S. L. Campbell, "Explicit methods for solving singular differential equation systems", Proc. Midwest Symposium on Circuits & Systems, Univ. of W. Virginia, June, 1984 (to appear).

8. S. L. Campbell, "Nonlinear time-varying generalized state-space systems: an overview", Proc. 23 IEEE-SIAM Conference on Decision & Control, Dec., 1984. (to appear).

9. S. L. Campbell, "Non-BDF methods for the solution of linear time varying implicit differential equations", Proc. 1984 Amer. Control Conf., June, 1984, 1315-1318.

10. S. L. Campbell and C. D. Meyer, Jr., Generalized Inverses of Linear Transformations, Pitman, 1979.

11. S. L. Campbell and L. R. Petzold, "Canonical forms and solvable singular systems of differential equations", SIAM J. Alg. & Discrete Methods, 4(1983), 517-521.

12. R. E. Cline and R. J. Plemmons, "ℓ_2-solutions to underdetermined linear systems", SIAM Review, 18 (1976), 92-106.

13. C. W. Gear, "Simultaneous numerical solution of differential algebraic equations", IEEE Trans. Circuit Theory, CT-18 (1971) 95-95.

14. C. W. Gear and L. R. Petzold, "ODE methods for the solution of differential/algebraic systems", SANDIA Report SAND 82-8051, 1982.

15. A. George, M. T. Heath, and E. Ng, "Solution of sparse underdetermined systems of linear equations", SIAM J. Sci. Stat. Comp, (to appear).

16. G. H. Golub and C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, 1983.

17. P. Lötstedt, "Numerical simulation of time-dependent contact and friction problems in rigid body mechanics", SIAM J. Sci. Stat. Comp. 5 (1984), 370-393.

18. P. Lötsted and L. Petzold, "Numerical solution of nonlinear differential equations with algebraic constraints", SANDIA Report, SAND 83-8877, 1983.

19. L. R. Petzold, "Differential/algebraic equations are not ODE's", SIAM J. Sci. & Stat. Comp. 3 (1982), 367-384.

20. W. C. Rheinboldt, "Differential-algebraic systems as differential equations on mainfolds", Institute for Computational Mathematics and Applications, Technical Report ICMA-83-55, University of Pittsburgh, 1983.

21. R. F. Sincovec, A. M. Erisman, E. L. Yip, and M. A. Epton, "Analysis of descriptor systems using numerical algorithms", IEEE Trans. Aut. Control, AC-26 (1981), 139-147.

22. G. W. Stewart, "Rank degeneracy", SIAM J. Sci. Stat. Comp. 5 (1984), 403-413.

DEPARTMENT OF MATHEMATICS
BOX 8205
NORTH CAROLINA STATE UNIVERSITY
RALEIGH, NORTH CAROLINA 27695-8205

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR- 86-0420	2. GOVT ACCESSION NO.: ADA 170163
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Rank Deficient Least Squares and the Numerical Solution of Linear Singular Implicit Systems of Differential Equations	5. TYPE OF REPORT & PERIOD COVERED: Reprint
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Dr. Stephen L. Campbell	8. CONTRACT OR GRANT NUMBER(s): AFOSR-84-0240
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Matnematics, North Carolina State University, Raleigh, North Carolina 27695-8205	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102, 2304, A5
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research, Air Force Systems Command	12. REPORT DATE: May 8, 1986
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report)
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Linear Algebra and Its Role in Systems Theory, AMS Contemporary Mathematics Series, Vol. 47, 1985, 51-63.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Implicit differential equation, singular systems, rank deficient least squares, numerical analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An approach for the numerical solution of linear systems of differential equations of the form $A(t)x'(t) + B(t)x(t) = f(t)$ with $A(t)$ singular is discussed. The key step in this approach is the solution of a rank deficient least squares problem. The solution of this least squares problem is investigated. These results are then applied to the system of differential equations.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)