Carnegie Mellon University
Research Showcase @ CMU

Department of Chemical Engineering | Carnegie Institute of Technology

1997

A successive linear programming approach for initialization and reinitialization after discontinuities of differential algebraic equations

Vipin Gopal
Carnegie Mellon University

Lorenz T. Biegler

Follow this and additional works at: http://repository.cmu.edu/cheme

Recommended Citation
.

This Technical Report is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase @ CMU. It has been accepted for inclusion in Department of Chemical Engineering by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

A Successive Linear Programming Approach for Initialization and ReMtiaNntaAto Discontinuities of DiflerentialAlgebraicEquabons

V. GopaiandLT.Biegler

EDRC 06-235-97

A SUCCESSIVE LINEAR PROGRAMMING APPROACH FOR INITIALIZATION AND REINITIALIZATION AFTER DISCONTINUITIES OF DIFFERENTIAL ALGEBRAIC EQUATIONS

VIPIN GOPAL'AND LORENZ T. BIEGLER*
Department of Chemical Engineering and Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213, USA

July 17, 1996

Abstract. Determination of consistent initial conditions is an important aspect of the solution of differential algebraic equations (DAEs). Specification of inconsistent initial conditions, even if they are slightly inconsistent, often leads to a failure in the initialization problem. In this paper, we present a Successive Linear Programming (SLP) approach for the solution of the DAB derivative array equations for the initialization problem. The SLP formulation handles roundoff errors and inconsistent user specifications among others and allows for reliable convergence strategies that incorporate variable bounds and trust region concepts. A new consistent set of initial conditions is obtained by minimizing the deviation of the variable values from the specified ones. For problems with discontinuities caused by a step change in the input functions, a new criterion is presented for identifying the subset of variables which are continuous across the discontinuity. The LP formulation is then applied to determine a consistent set of initial conditions for further solution of the problem in the domain after the discontinuity. Numerous example problems are solved to illustrate these concepts.

Key words, differential-algebraic equations, consistent initialization, discontinuity, successive linear programming

AMS subject classification. 65L05, 90C30

*vipinQandrew.cmu.edu
*To whom all correspondence should be addressed, *biegler0cmu.edu*

1 Introduction

Differential Algebraic Equations (DAEs) occur frequently in a variety of engineering problems. Here we consider the form:

$$F\{x\backslash x_{y}y,t) \quad = \quad 0 \tag{1}$$

where F € R^n,x,x' € R^n*,y € $R^{n\sim nd}$ and x' is the time derivative of x. DAEs are characterized by their *index* which is defined (Brenan *et al.*, 1995) as the minimum number of times F, or the equations derived from it, should be differentiated to get a system of the form

$$v' \quad = \quad /(\mathrm{iM}) \tag{2}$$

where

$$v = \{x,y\},v£R^n. \tag{3}$$

Determination of consistent initial conditions for the DAE is a problem which lias been of considerable interest among researchers dealing with systems of the form (1). Differentiation of (1) could generate new algebraic equations. These *hidden* algebraic equations in (1) impose constraints on the initial conditions of variables which can be specified independently. Thus the consistent initialization problem requires identification of these constraints, implicitly or explicitly, and their treatment along with the original set of equations.

Pantelides (1988) utilizes the property that the hidden or *latent* equations are generated from those subsets of equations in whidi the number of variables (x\y) present in the subset is less than the number of equations themselves. He uses a graph-theoretical algorithm to locate the *minimally structurally singular* subsets and differentiates them. However, the algorithm is structural and subsets of equations which are numerically singular might escape detection.

Unger *et al.* (1995) also present an algorithm based on structural analysis. They determine the structural rank and the structural index of the problem by using a structural version of the symbolic algorithm for index reduction by Gear (1988). Gear's algorithm is an extension of the algorithm by Bachmann *et al.* (1990) for linear systems. Here the structural rank and index of the DAE (1) is solely determined by the patterns of dF/dv^1 and dF/dv. However the structurally determined index and degrees of freedom will only provide a lower bound and upper bound respectively for their corresponding differential quantities. They also proposed a combination of structural methods with numerical/symbolic methods to overcome these difficulties as a direction for future research.

Campbell and Moore (1994) proposed a method for solving nonlinear, high index problems for which BDF and Implicit RK methods might be inappropriate. They define the derivative array equations

$$F''(a:,z,z',\ldots,*V) = 0 \tag{4}$$

as the set of equations derived by differentiating the original DAE (1) *v* number of times where *v* is the index of (1) and

$$* = \{*',y\} \tag{5}$$

They solve $F^{¥} = 0$ by a least squares iterative solution. The method was found to be computationally expensive especially for computing the singular value decomposition of *JF»* (the Jacobian of the derivative array equations) used for the least squares solution.

Leimkuhler *et al.* (1991) characterize the consistency requirement by posing a set of equations which comprise the derivative array equations (4) (defined at time to) and the set of user specified information on initial conditions (6).

$$£(*<>,z_0,y_0) = 0 \tag{6}$$

The higher derivatives are approximated by forward finite differences. The resulting approximate system, however, might not have a solution and hence is solved in a least squares sense. This is complicated as well because of the rank deficiency of the Jacobian. The method is illustrated for index one and semi explicit index 2 problems in the triangular form.

Chung and Westerberg (1990) and Chung (1991) proposed a numerical algorithm for consistent initialization by identifying singular subsets of equations from the Jacobian and symbolically differentiating them. Majer *et al.* (1995) considers the problem of reinitialization of DAEs after discontinuities which we also discuss in detail in section 3.

Leimkuhler *et al.* (1991) defines the consistent initialization problem as: *given specified information about the initial state of the problem Uiat is sufficient to specify a unique solution to a DAE, determine the complete vector* (t/(t<>), *v'(to))*. However, for a particular problem, the user may not know which variables to specify to sufficiently determine an exact solution for the initial condition vector. In other words, the user does not know the degrees of freedom a priori. On the other hand, the user may know a certain set of specifications and we would like our initial conditions to be as dose to those given by the user (whether it is over or exactly specified). Many approaches analyze the set of equations and ask the user for the initial conditions for a particular set of variables, which he might not have. Hence, it would make sense to work the other way. Given the

known set of initial conditions we would like to find a consistent initial vector (x',x,y) closest to the known set. In many practical applications, the exact numerical value of the initial conditions may not be known for satisfying the relevant set of equations within the specified error criteria. The minimization formulation which we present in section 2 has its significance in this context. The method is also relevant when there are roundoff errors.

In die next section we develop a successive linear programming (SLP) formulation for the consistent initialization of DAE systems. The motivation for this approach is that the derivative array equations require solution of an underdetermined system, but roundoff errors and incomplete user specifications may render this system inconsistent. An SLP formulation handles this underdetermined nature efficiently and allows for reliable convergence strategies that incorporate variable bounds and trust region concepts. Section 3 deals with the related problem of reinitialization after input discontinuities are encountered for the DAE system. Here an analysis on the continuity of the state profiles is presented and compatible SLP formulations are derived for this problem. The SLP formulations in both sections are illustrated with the successful treatment of numerous high index examples. Section 4 concludes the paper and outlines directions for future work.

2 Formulation for Consistent Initialization

2.1 The problem and its properties

To derive the SLP formulation we first consider properties of the general DAE system (1) and the associated derivative array equations (4). Let A, B and C be the Jacobian of F in (1) with respect to x,y and x' respectively. Thus

$$J_F = [\overset{x}{A} \quad \overset{y}{B} \quad \overset{x_*}{C}] \qquad m \tag{7}$$

On differentiating, F' will have terms of x,y,x',y' and x". Now the Jacobian of F' with respect to these variables is related to the arrays A, B and C in JF and the Jacobian with respect to x and y will be A' and B' respectively. With respect to x\ the terms involving x' in F' come from terms involving x or x' in F. Thus, the derivative of F' with respect to x' will be A + C. By similar arguments it is not difficult to see that the derivative of F' with respect to y' and x" will be B and C respectively. If we continue the same pattern, the

Jacobian of the derivative array equations will look like:

$$J_{F^v} = \begin{array}{c} \begin{array}{cccccccccc} \mathrm{I} & y & x' & v' & x'' & y'' & x''' & y''' & x^{(4)} & .. \end{array} \\ \left[\begin{array}{cccccccccc} A & B & C & & & & & & & \\ A & B' & A+C & B & C & & & & & \\ A'' & B'' & 2A+C'' & Iff & A+2C & B & C & & & \\ A''' & B''' & ZA''+C'' & ZB'' & 3A'+3C'' & ZB' & A+ZC & B & C & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right] \end{array} \begin{array}{c} \mathrm{F} \\ \mathrm{F}' \\ \mathrm{F}'' \\ F''' \\ \vdots \end{array} \qquad (8)$$

Generalizing, the Jacobian of $F^\wedge$ with respect to x^{*m*} (or $y^{*m>}$) is the sum of the element in the Jacobian of $F^{\wedge''1*}$ with respect to $x^{\{m\sim l)}$ (or $y^{(m\sim l)}$) and the derivative of $F^{(\dot{*}''!)}$ with respect to $x^{(m)}$ (or $y^{(m)}$). Note also that for a constant coefficient linear DAE, (8) simplifies to

$$J_{F^v,linear} = \begin{array}{c} \begin{array}{cccccccccc} \mathrm{x} & \mathrm{y} & \mathrm{x}' & »' & \mathrm{x}'' & y'' & x^{1''} & y''' & x^{(4)} & .. \end{array} \\ \left[\begin{array}{cccccccccc} A & B & C & & & & & & & \\ & & & A & B & C & & & & \\ & & & & & & A & B & C & \\ & & & & & & & & A & B & C \\ & & & & & & & & & \vdots & \vdots \end{array}\right] \end{array} \begin{array}{c} F \\ F' \\ F'' \\ pm \\ \vdots \end{array} \qquad (9)$$

The derivative array equations consist of an underdetermined system of (1/ + l)n 4- *rid* variables and *(v + l)n* equations. The Jacobian of the derivative array equations is rank deficient because of the algebraic equations which are implicit within them.

We still need an efficient method for the solution of this nonlinear underdetermined system which is rank deficient. Additional problems include inconsistent linearizations, roundoff errors and poor starting points and user specifications. To deal with this problem we propose the following optimization formulation which we solve with SLP.

2.2 SLP Formulation

Consider the solution of a nonlinear system of equations subject to inequality constraints and variable bounds (CSP).

$$\begin{array}{l} h(x)=0 \\ 9(*)\leq 0 \\ x^L \leq x < x^u \end{array} \qquad \text{(CSP)}$$

To solve (CSP), a merit function is defined as

$$\mu(x^i) = \sum_j |h_j(x^i)| + \sum_k g_k(x^i)_+ \tag{10}$$

where jte(x')+ = maxfOj^x1')]. The function /x(x) is nondifferentiable and attains its minimum value of zero if and only if (CSP) is solved. Using SLP we minimize jx(x) directly subject to $x^L \leq x \leq x^u$. By adding auxiliary variables, this problem can be written as :

$$\begin{aligned} & Min\ ]C\ to + {}^{n}i) + Yl^{Sk} \\ & \quad\ \ i \qquad\qquad\qquad\quad k \\ S.t. \quad & kj(x) = Pj - Tlj \\ & 9k\{x) \leq s_k \\ & x^L \leq x \leq x^u \end{aligned} \tag{11}$$

It can be shown that *(pj + tij)* represents |/ij(x)| and 5^ represents *gk(x)+*. Linearizing the constraints about x* leads to the following constrained simulation linear program, CSLP.

$$\begin{aligned} & Min \sum_j (p_j + n_j) + \sum_k s_k \\ s.t. \quad & h_j(x^i) + \nabla h_j(x^i)^T d^i = p_j - n_j \\ & g_k(x^i) + \nabla g_k(x'fcf \leq 5_{fc} \\ & x^L \leq x^i + d^i \leq x^u \\ & p_j, n_j, s_k \geq 0 \end{aligned} \tag{CSLP}$$

Hie solution of (CSLP) generates a search direction d\ Through the addition of artificial variables for the constraints, the LP is always feasible and generates a descent direction for every nonzero solution of the LP. A steplength algorithm combining both line search and trust region methods has been shown to enhance the computational efficiency and increase the robustness of the algorithm. A complete description of the SLP strategy is presented in Gopal and Biegler (1995). Zhang *et al.* (1985) and Duff *et al.* (1987) have also described trust region based Successive Linear Programming approaches in the context of nonlinear programming and nonlinear equation solving, respectively.

We can also extend problem (CSP) and the SLP approach by minimizing some distance from the specified conditions for the initialization problem. Let the specified initial conditions be $X\{ = A'^{*p}_i$ for t = l,...ni$_C$ where A' $\subseteq$ {x,x\y}.

An optimization formulation for consistent initialization is

$$
\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x, z, z', \ldots, z^{\nu}, t) = 0 \\
& -Si \leq Xi - Xf_t^p < \delta_i \qquad i = 1, \ldots, n_t > \\
& < J_t \geq 0 \qquad t = 1, \ldots, n < c \\
& A'/'' \leq Xi \leq X? \qquad (12)
\end{aligned}
$$

where *Ui* are weights that scale the deviations from the specifications. The NLP (12) is solved using Successive Linear Programming (SLP). The method generates a search direction from a linear program obtained from the linearization of (12). If the given specification is exact, the solution of the problem will correspond to a consistent set of initial conditions and a zero objective function. In case of an inconsistent overspecification, the optimum solution of (12) will be a positive value of the objective function, with a consistent set of initial conditions for the problem, closest to the specified set.

Note that the formulation (12) involves minimization of the weighted 1-norm of the violation of initial condition specifications. Hence the number of initial conditions set equal to the specified values at the solution need not be equal to the degrees of freedom for the specification of initial conditions (r) for the problem. Nonetheless it should also be noted that the objective value at the solution of the LP will be less than or equal to what it would have been if exactly r of the variables were set equal to their specified values. This is because (12) is not constrained to fix r initial conditions to their specified values but to find a consistent set of initial conditions closest to their specified values.

However, the user might want to have r variables to be set equal to their specified values, if the solution of (12) fixes less than r specifications. In other words, he may wish to pick the best set of r variables so that the specification violation of the rest of the variables is minimized. The degrees of freedom could be determined from the rank analysis of the Jacobian of the derivative array equations (Campbell and Moore, 1994). Once r is known, the following Mixed Integer NonLinear Programming (MINLP) formulation (13) could be used for consistent initialization. A;s are binary variables and M;s correspond to the big-M constraints.

$$
\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x, z, z', \ldots, z^{(\nu)}, t) = 0
\end{aligned}
$$

$$-6_{\{} \leq Xi-Xf^{p} \leq 6i \qquad i = 1,\ldots, n_{<c} \qquad (13a)$$

$$0 \leq Si \leq M,(1 - X_t) \qquad t = 1 \ldots n_{ie} \qquad (136)$$

$$\sum^{n_{ie}} \lambda_i = r \qquad (13c)$$

$$A<=O,1 \qquad t = l,\ldots,flfc$$

$$A'/' \leq X_{\{} \leq X? \qquad (13)$$

A value of A» = 1 =>·6, = 0 which implies that variable *Xi* is set equal to its specified value at the solution of the problem. The constraint *Yl'£i* A* = ^{r} serves of purpose of setting at least r variables to their specified values. However, (13) is an MINLP and could be expensive for the solution of the above problem. We would like to simplify the above formulation to a form where it could be solved using SLP.

If we rewrite (13a) as

$$Si + Xi - A'f^{p} \geq 0 \qquad (14ai)$$

$$*«-A', + A'F \geq 0 \qquad (Ma*)$$

The following inequalities are introduced in place of (13b).

$$Xi\,[\{Si + Xi - X?) + (Si - Xi + X?)] \leq 0 \qquad (1460$$

$$0 \leq Xi \leq 1 \qquad (14fr_2)$$

Note that A; in (14b) are continuous variables bounded between zero and one as opposed to the binary A; used in (13b). (14b) could be simplified to:

$$\lambda_i \delta_i \leq 0 \qquad (14c_1)$$

$$0 \leq Xi \leq 1 \qquad (14c_2)$$

Consider the case A; > 0. (14cO will be satisfied *iff Si* = 0 because of the nonnegativity constraint on *S{. ie. Xi* = X_i^{p}. A nonzero A* corresponds to the variable X,- set equal to its specified value. Similarly a nonzero *Si (Xi* ^ X_t^{*p}) implies A; = 0.

The summation equation (13c) is retained. (14ci) smd the nonnegativity conditions At ≥ 0 and *Si* ≥ 0 reduces to the complementarity condition Aj<5» = 0. The simplified formulation using continuous variables is given by (14).

$$Min \sum_{i=i}^{n_{ie}} \omega_i \delta_i$$

$$
\begin{aligned}
\text{s.t.}\quad & F^{\nu}(x, z, z', \ldots, z^{(\nu)}, t) = 0 \\
& -\delta_i \le X_i - X_i^{sp} \le \delta_i \qquad i = 1, \ldots, n_{ic} \\
& \lambda_i \delta_i = 0 \qquad i = 1, \ldots, n_{ic} \\
& \sum_{i=1}^{n_{ic}} \lambda_i = r \\
& 0 \le A_{f'} \le 1 \qquad i = 1, \ldots, n_{fc} \\
& \langle J_i \ge 0 \qquad i = 1, \ldots, n_{\langle c} \\
& A_{f'}^{L} \le A'_{f} \le A_{f}
\end{aligned}
\tag{14}
$$

The reformulation of the original MINLP problem to an NLP (14) constrained by complementarity conditions presents us with an opportunity to use a variety of techniques suggested to solve complementarity problems (Billups *et* a/., 1995). Chen and Mangasarian (1995) proposed a class of smoothing functions to solve the complementarity problem approximately. A and *6* satisfy the complementarity condition if and only if A = (A - <$)+ where (A - <$)+ = max(0, A - *S*). The max function is approximated using the smoothing function

$$\max(O,q) \ll 17 + 0 \ \log(l + e - \bar{w}) \tag{15}$$

As the parameter *0* approaches zero the smooth approximation approadies the nonsmooth max function. Using the smooth approximation (15), (14) can be reformulated as

$$
\begin{aligned}
& A/in \sum_{1=1}^{n_{ic}} \omega_i \delta_i \\
\text{s.t.}\quad & F''(x, z_y; z', \ldots, z^{(\nu)}, t) = 0 \\
& -\delta_i \le X_i - X_i^{sp} \le \delta_i \qquad i = 1, \ldots, n_{ic} \\
& Si = 0 \ \log(l + e^{-(\lambda_i - \delta_i)/\beta}) \qquad i = 1, \ldots, n_{ic} \\
& \sum_{i=1}^{n_{ic}} \lambda_i = r \\
& 0 \le A_{f'} \le 1 \qquad i = 1, \ldots, n_{fc} \\
& Si \ge 0 \qquad i = 1, \ldots, n_{\langle c} \\
& A'/' \le A'_{f} \le A'/
\end{aligned}
\tag{16}
$$

Formulation (16) is an NLP and is solved using SLP since a vertex solution is expected. (16) is less cumbersome to solve when compared to (13). We also show in the Appendix that if the degrees of freedom (r) are specified exactly, under certain conditions, a nonzero A, will take its upper bound value of 1,

which is a solution of the MINLP formulation (13).

2.3 Examples

Below, we consider three example problems to illustrate the application of formulations (12) and (16).

Example 1: Chemical Reactor Problem

Consider the problem described by (17) (Campbell and Moore, 1994; Pantelides, 1988).

$$\begin{aligned} C' + C + R &= 4 + t + i^1 \\ T^1 + 2T + R + Tc &= 14 - eT^l \\ T\sim^l + ln\{R/C) &= 0 \\ C &= cosh\{t - 1) \end{aligned} \tag{17}$$

The problem is index 3 and has a unique solution. There are no degrees of freedom for specification of the initial conditions. However since there are two differential equations, the user may be tempted to specify initial conditions corresponding to the differential variables. The formulation (12) should be able to take care of any such inconsistent specifications and give us a consistent set of initial conditions closest to the specified ones (in this case the unique initial vector). In table 1 we consider two inconsistent specifications for the initial conditions. In both cases the SLP method solved for the correct set of initial conditions with a nonzero objective function at the solution. Note that the second case corresponds to the commonly used steady state initial condition specifications. D

Example 2: Pendulum Problem

This is the popular pendulum problem considered by various authors (Pante lides, 1988; Gear, 1988; Unger *et* a/., 1995). The formulation is given by (18).

$$\begin{aligned} x^1 &= t; \\ y' &= w \\ v^1 &= -Tx \\ nf &= -Ty - g \end{aligned} \tag{18}$$

$$X^2 \text{"I } 1/^2 \text{ TM" } M.^i$$

The problem is of index 3 and has 2 degrees of freedom for specification of initial

Table 1: Initial Conditions for the Chemical Reactor Problem

Specified ICs	Actual ICs (SLP Solution)	Number of SLP Iterations
C = l 7b = 0	C = 1.5431 R = 3.6321 T =-1.1682 T_c = -0.5726	12
C' = 0 T' = 0	C = -1.1752 T = 1.2768	6

conditions. The two underlying algebraic equations (19) are derived from the successive differentiation of (18).

$$\begin{aligned} xv + y\,w &= 0 \\ -T(x^2 - y^2) + tr^2 + ur^2 - gy &= 0 \end{aligned} \tag{19}$$

Table 2 shows how the proposed method solves for a consistent set of initial conditions for three different specifications.

The number of specifications in the first two sets is equal to the degrees of freedom (2) and the sets *(x,v)* and (z,T) form a feasible pair (The Jacobian of the original set of equations with respect to all the variables other than the specified ones is nonsingular). As seen, applying SLP to (12) solves for the consistent initial vector with the solution for the specified variables being equal to their specified values. However, the third set is an overspecification. In this case SLP finds the initial condition vector which is closest, in the 1-norm sense, to the specified ones.

Now consider the application of (16) to this problem. As seen from the specifications in table 3, even though there are two degrees of freedom for this problem, solution with (12) sets only one variable to its specified value. On the other hand, the problem when solved with (16) and r = 2, sets x and w to their specified values. The corresponding objective function is 0.3660 which is the deviation in v. This objective is greater than the value of 0.3128 from (12). D

Table 2: Initial Conditions for the pendulum problem

Specified ICs	Solution from (12)	Number of LPs solved
i = 0.5 **v = 1.0**	**i = 0.5** **i> = 1.0**	11
x = 0.5 *T* = 0.5	**i = 0.5** **r = o.5**	12
i = 0.5 w=1.0 T = 0.5	1 = 0.5 *v* = 1.012 T = 0.5	12

Table 3: Solution for the pendulum problem from inconsistent specifications using (12) and (16)

Specified ICs	Solution from (12)	Solution from (16)
i = -0.5 « = 0.5 w = 0.5	i =-0.5463 y = 0.8376 *v* = 0.7665 u; = 0.5000 *T* = 0.0000 £<S, = 0.3128	*x* = -0.5000 y = 0.8660 v = 0.8660 w = 0.5000 *T* = 0.1340 **Efc = 0.3660**

Example 3: Trajectory Prescribed Path Control (TPPC) Problem
Here we consider a nonlinear example used in the simulation of space vehicles

when the shape of the trajectory is prescribed. TPPC problems are treated extensively in Brenan *et al.* (1995). .(20) is an index 3 problem where the first six equations are the equations of motion in relative coordinates and the last equation is the algebraic path constraint.

$$
\begin{aligned}
&= V^*\sin 7 \\
\xi &= \frac{V_R \cos\gamma \sin A}{r\cos A} \\
A' &= \frac{VR}{r}\text{COS7COS/1} \\
V_R' &= \frac{D}{m}0\sin 7\text{-ft£}^2 r\cos A(\sin A\cos 4\cos 7\text{-}\cos A\text{siii}7) \qquad (20) \\
\gamma' &= \frac{L\cos\beta}{mV_R} + \frac{\cos\gamma}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega_E \cos\lambda \sin A + \\
&\quad \frac{\Omega_E^2 r\cos\lambda}{V_R}(\sin A\cos A\sin 7 + \cos A\cos 7) \\
A' &= \frac{L\sin\beta}{mV_R\cos\gamma} + \frac{V_R}{r}\cos\gamma\sin A\tan\lambda - 2\Omega_E(\cos\lambda\cos A\tan\gamma - \sin\lambda) + \\
&\quad \frac{Q_E r\cos A\sin A\sin X}{V_R\cos\gamma} \\
\frac{D}{m} &- \left[C_0 + C_1(V_R - V_0) + C_2(V_R - V_0)^2 + C_3(V_R - V_0)^3\right] = 0
\end{aligned}
$$

Computations by Campbell and Moore (1994) using the initial values of the state variables obtained from Brenan *et al.* (1995) found that their values were not exact. Using these values as initial guesses, they generated new initial conditions using a least squares method. There are two points to be noted in this problem:

(1) Arbitrary specification of all 7 state variables will lead to problems even if they are slightly inconsistent (as observed by Campbell and Moore (1995)) since the degrees of freedom is 4.

(2) A different undesirable control history of *0* < 0 is possible for this problem. *0* must be positive in the given set of initial conditions.

The initialization problem when formulated using (12) and initialized using the values from Brenan *et al* (1995), returned consistent initial conditions within an accuracy of 10^{-12} in just 2 iterations. The SLP method has the added advantage of an easy imposition of a lower bound of zero on *0* to avoid the undesirable control history. •

hi summary, when the failure of the solver is due to inconsistent specification of initial conditions by the user, it is advantageous that the user be presented

with a consistent set of initial conditions which is closest to what had been specified before and is numerically consistent. The two formulations, (12) and (16), serves this purpose and give the user flexibility to use the available information to the best advantage. Campbell and Moore (1994) point out that in practical applications, one would expect that some initial values are known and there are reasonable estimates for others. The method has relevance in this context too, where the weights <JK in (12) and (16) could be chosen to help the user determine a consistent initial vector reflecting his confidence level in the initial value estimates.

3 Reinitialization after discontinuities

In this section, we look at the problem of reinitialization of DAEs after discontinuities, some underlying principles for this problem and the application of the minimization formulation.

3.1 Problem description

The task of finding a consistent set of initial conditions for the reinitialization of a DAE is often more complicated than the *pure* initialization problem. The reason is that the reinitialization problem involves an additional exercise of determining the variables which are continuous across a discontinuity. The numerical values of the continuous variables at the point of discontinuity contribute to the specifications for the reinitialization problem.

Majer *et al.* (1995) address the problem of reinitialization of DAEs after discontinuities. They specifically look at index one linear-implicit DAEs of the form (21), where there are discontinuous steps in the forcing function *u.*

$$T\{v,u,t)v' = h(v,u,t) \tag{21}$$

u is the vector corresponding to the input variables and an algebraic transformation of the form (22) exists

$$\begin{aligned} T_{11}(x,y,u)x' + T_{12}(x,y,u)y' &= h_1(x,y,u) \\ 0 &= \mathrm{Mx,y,u)} \end{aligned} \tag{22}$$

They show that the continuity assumptions of state variables can be safely assumed by default only when TV; = 0. For the case T_n £ 0, conditions are laid out for the solvability of consistency equations.

Brüll and Paliaske (1992) considered the same index one problem and derived the necessary and sufficient condition for the above system to have a *genuine*

initial value for the state variable vector *v* after a jump in the input variable u.

However, no work has been done for determining the validity of the continuity assumptions of state variables for higher index nonlinear DAEs. In this section, we consider a class of such problems in whidi the discontinuity is caused by a step change in the input function *u.*

An assumption about this class of problems is that the index of the problem remains the same at all times. Also, if we were to look at the discrete models at any time as independent problems, the degrees of freedom for the specification of initial conditions remain the same. Often it is assumed that the differential variables are continuous if the composition of the differential variables appearing in the describing equations remain unchanged across the discontinuity. This is not always true as was shown by Majer *et* a/. (1995) and Brüll and Pallaske (1992) for hilly implicit index one systems. When considering problems of arbitrary index, what is missing are rigorous criteria to determine whidi variables are continuous and whidi are not, across a discontinuity caused by a step diange in one or more input functions.

3.2 Continuity assumptions for problems of arbitrary index

Consider a DAE of the form

$$F(x',x,_{Vl}ti,t) \quad = \quad 0 \tag{23}$$

where u € R^{n}". *The continuity assumptions hold for tliat subset of variables* v_0, *whose underlying ODEs do not depend on any of tlie derivatives of Hie function u causing the discontinuity.*

Let the system be of index *v.* Differentiation of (23) *v* times will yidd an ODE of the form (24) whidi is termed as the underlying ODE for the DAE (23). *v is* defined as in (3).

$$t/ \quad = \quad /(ti,ti,u'. . . . .iiM.t) \tag{24}$$

Denote tT = $v(t^9$ - 0) and v+ = $v(t^*$ + 0), where t^m is the time of discontinuity. Let $v_{\{}$ be a variable which has an underlying ODE of the form (25) where $u^{(k)}$ is tlie highest derivative of *u* explicitly present in the underlying ODE.

$$v'i \quad = \quad fi\{v, ti_f \ldots_f u<«_f t) \tag{25}$$

Let *ust* be a smoothed step for *u* defined for_*V* $\leq t<V+6t$ and v_{st} be the corresponding smoothed profiles of the variables. As in Majer *et al.* (1995),

$$v_i^+ = lim_{\delta t \to 0}\{v_i^- \quad {}^{\cdot t+tt} \quad fi\{v6t,u_{6t},...,u^{(k}_{6\ t}\backslash t)dT\} \tag{26}$$

The derivatives of u are not bounded for step changes in u. The integral will vanish only when the derivatives of u do not form part of the integral, explicitly or implicitly through the variables appearing in the underlying ODE. Thus, only those variables whose underlying ODEs could be shown to be independent of any derivative of the input function causing the discontinuity, are continuous across it.

Let the set of continuous variables be v_0. From the above analysis, it is clear that there is no guarantee that all differential variables will fall in the set *VQ*. Hence, the assumption of continuity of differential variables across the discontinuity, often used in dynamic simulators, is not always true.

An underlying ODE can be determined using a symbolic algorithm, for example by Gear (1988). However, the form of the ODE found depends on the algebraic manipulation one carries out while deriving it. The above analysis provides a framework for determining the variables which are continuous across a discontinuity, but from a computational standpoint the issue of uniqueness of the underlying ODE of a given DAE (1) is to be addressed.

Example 4
Consider the DAE described by (27).

$$\begin{aligned} y' + x &= u\backslash \\ y &= t i_2 \end{aligned} \tag{27}$$

Consider the following two forms of the underlying ODE, (28) and (29).

$$\begin{aligned} x' &= u\backslash - u\$ \\ y' &= u!_2 \end{aligned} \tag{28}$$

$$\begin{aligned} x' &= u\backslash - u\$ \\ y' &= u_x - x \end{aligned} \tag{29}$$

In the event of a step change in u>, (28) gives a clear picture of the behavior of y, whereas from (29), it is not clear because of the implicit dependence on x. O

However, in practical applications, one would not prefer to explicitly determine an underlying ODE and then validate the continuity assumptions. Given all the constraining algebraic equations for a DAE and any particular underlying ODE, one could derive other forms of underlying ODEs. The derivative array equations, in principle, contain the differential and all the constraining algebraic equations for a given DAE. Hence, it would be easier to work with the derivative array equations and determine the dependencies of the differential

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

A Successive Linear Programming Approach for Initialization and ReMtiaNntaAto Discontinuities of DiflerentialAlgebraicEquabons

V. GopaiandLT.Biegler

EDRC 06-235-97

A SUCCESSIVE LINEAR PROGRAMMING APPROACH FOR INITIALIZATION AND REINITIALIZATION AFTER DISCONTINUITIES OF DIFFERENTIAL ALGEBRAIC EQUATIONS

VIPIN GOPAL'AND LORENZ T. BIEGLER*
Department of Chemical Engineering and Engineering Design Research Center
Carnegie Mellon University
Pittsburgh, PA 15213, USA

July 17, 1996

Abstract. Determination of consistent initial conditions is an important aspect of the solution of differential algebraic equations (DAEs). Specification of inconsistent initial conditions, even if they are slightly inconsistent, often leads to a failure in the initialization problem. In this paper, we present a Successive Linear Programming (SLP) approach for the solution of the DAB derivative array equations for the initialization problem. The SLP formulation handles roundoff errors and inconsistent user specifications among others and allows for reliable convergence strategies that incorporate variable bounds and trust region concepts. A new consistent set of initial conditions is obtained by minimizing the deviation of the variable values from the specified ones. For problems with discontinuities caused by a step change in the input functions, a new criterion is presented for identifying the subset of variables which are continuous across the discontinuity. The LP formulation is then applied to determine a consistent set of initial conditions for further solution of the problem in the domain after the discontinuity. Numerous example problems are solved to illustrate these concepts.

Key words, differential-algebraic equations, consistent initialization, discontinuity, successive linear programming

AMS subject classification. 65L05, 90C30

*vipinQandrew.cmu.edu
*To whom all correspondence should be addressed, *biegler0cmu.edu*

1 Introduction

Differential Algebraic Equations (DAEs) occur frequently in a variety of engineering problems. Here we consider the form:

$$F\{x\backslash x_{y}y,t) \quad = \quad 0 \tag{1}$$

where F € R^n,x,x' € R^n*,y € $R^{n\sim nd}$ and x' is the time derivative of x. DAEs are characterized by their *index* which is defined (Brenan *et al.*, 1995) as the minimum number of times F, or the equations derived from it, should be differentiated to get a system of the form

$$v' \quad = \quad /(\mathrm{iM}) \tag{2}$$

where

$$v = \{x,y\},v£R^n. \tag{3}$$

Determination of consistent initial conditions for the DAE is a problem which lias been of considerable interest among researchers dealing with systems of the form (1). Differentiation of (1) could generate new algebraic equations. These *hidden* algebraic equations in (1) impose constraints on the initial conditions of variables which can be specified independently. Thus the consistent initialization problem requires identification of these constraints, implicitly or explicitly, and their treatment along with the original set of equations.

Pantelides (1988) utilizes the property that the hidden or *latent* equations are generated from those subsets of equations in whidi the number of variables (x\y) present in the subset is less than the number of equations themselves. He uses a graph-theoretical algorithm to locate the *minimally structurally singular* subsets and differentiates them. However, the algorithm is structural and subsets of equations which are numerically singular might escape detection.

Unger *et al.* (1995) also present an algorithm based on structural analysis. They determine the structural rank and the structural index of the problem by using a structural version of the symbolic algorithm for index reduction by Gear (1988). Gear's algorithm is an extension of the algorithm by Bachmann *et al.* (1990) for linear systems. Here the structural rank and index of the DAE (1) is solely determined by the patterns of dF/dv^1 and dF/dv. However the structurally determined index and degrees of freedom will only provide a lower bound and upper bound respectively for their corresponding differential quantities. They also proposed a combination of structural methods with numerical/symbolic methods to overcome these difficulties as a direction for future research.

Campbell and Moore (1994) proposed a method for solving nonlinear, high index problems for which BDF and Implicit RK methods might be inappropriate. They define the derivative array equations

$$F''(a:,z,z',\ldots,*V) = 0 \tag{4}$$

as the set of equations derived by differentiating the original DAE (1) *v* number of times where *v* is the index of (1) and

$$* = \{*',y\} \tag{5}$$

They solve $F^{¥} = 0$ by a least squares iterative solution. The method was found to be computationally expensive especially for computing the singular value decomposition of *JF»* (the Jacobian of the derivative array equations) used for the least squares solution.

Leimkuhler *et al.* (1991) characterize the consistency requirement by posing a set of equations which comprise the derivative array equations (4) (defined at time to) and the set of user specified information on initial conditions (6).

$$£(*<>,z_0,y_0) = 0 \tag{6}$$

The higher derivatives are approximated by forward finite differences. The resulting approximate system, however, might not have a solution and hence is solved in a least squares sense. This is complicated as well because of the rank deficiency of the Jacobian. The method is illustrated for index one and semi explicit index 2 problems in the triangular form.

Chung and Westerberg (1990) and Chung (1991) proposed a numerical algorithm for consistent initialization by identifying singular subsets of equations from the Jacobian and symbolically differentiating them. Majer *et al.* (1995) considers the problem of reinitialization of DAEs after discontinuities which we also discuss in detail in section 3.

Leimkuhler *et al.* (1991) defines the consistent initialization problem as: *given specified information about the initial state of the problem Uiat is sufficient to specify a unique solution to a DAE, determine the complete vector* (t/(t<>), *v'(to))*. However, for a particular problem, the user may not know which variables to specify to sufficiently determine an exact solution for the initial condition vector. In other words, the user does not know the degrees of freedom a priori. On the other hand, the user may know a certain set of specifications and we would like our initial conditions to be as dose to those given by the user (whether it is over or exactly specified). Many approaches analyze the set of equations and ask the user for the initial conditions for a particular set of variables, which he might not have. Hence, it would make sense to work the other way. Given the

known set of initial conditions we would like to find a consistent initial vector (x',x,y) closest to the known set. In many practical applications, the exact numerical value of the initial conditions may not be known for satisfying the relevant set of equations within the specified error criteria. The minimization formulation which we present in section 2 has its significance in this context. The method is also relevant when there are roundoff errors.

In die next section we develop a successive linear programming (SLP) formulation for the consistent initialization of DAE systems. The motivation for this approach is that the derivative array equations require solution of an underdetermined system, but roundoff errors and incomplete user specifications may render this system inconsistent. An SLP formulation handles this underdetermined nature efficiently and allows for reliable convergence strategies that incorporate variable bounds and trust region concepts. Section 3 deals with the related problem of reinitialization after input discontinuities are encountered for the DAE system. Here an analysis on the continuity of the state profiles is presented and compatible SLP formulations are derived for this problem. The SLP formulations in both sections are illustrated with the successful treatment of numerous high index examples. Section 4 concludes the paper and outlines directions for future work.

2 Formulation for Consistent Initialization

2.1 The problem and its properties

To derive the SLP formulation we first consider properties of the general DAE system (1) and the associated derivative array equations (4). Let A, B and C be the Jacobian of F in (1) with respect to x,y and x' respectively. Thus

$$J_F = [\overset{x}{A} \quad \overset{y}{B} \quad \overset{x_*}{C}] \qquad m \tag{7}$$

On differentiating, F' will have terms of x,y,x',y' and x". Now the Jacobian of F' with respect to these variables is related to the arrays A, B and C in JF and the Jacobian with respect to x and y will be A' and B' respectively. With respect to x\ the terms involving x' in F' come from terms involving x or x' in F. Thus, the derivative of F' with respect to x' will be A + C. By similar arguments it is not difficult to see that the derivative of F' with respect to y' and x" will be B and C respectively. If we continue the same pattern, the

Jacobian of the derivative array equations will look like:

$$J_{F^v} = \begin{array}{c} \begin{array}{cccccccccc} \mathrm{I} & y & x' & v' & x'' & y'' & x''' & y''' & x^{(4)} & .. \end{array} \\ \left[\begin{array}{cccccccccc} A & B & C & & & & & & & \\ A & B' & A+C & B & C & & & & & \\ A'' & B'' & 2A+C'' & Iff & A+2C & B & C & & & \\ A''' & B''' & ZA''+C'' & ZB'' & 3A'+3C'' & ZB' & A+ZC & B & C & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right] \end{array} \begin{array}{c} \mathrm{F} \\ \mathrm{F}' \\ \mathrm{F}'' \\ F''' \\ \vdots \end{array} \qquad (8)$$

Generalizing, the Jacobian of $F^\wedge$ with respect to x^{*m*} (or $y^{*m>}$) is the sum of the element in the Jacobian of $F^{\wedge''1*}$ with respect to $x^{\{m\sim l)}$ (or $y^{(m\sim l)}$) and the derivative of $F^{(\dot{*}''!)}$ with respect to $x^{(m)}$ (or $y^{(m)}$). Note also that for a constant coefficient linear DAE, (8) simplifies to

$$J_{F^v,linear} = \begin{array}{c} \begin{array}{cccccccccc} \mathrm{x} & \mathrm{y} & \mathrm{x}' & »' & \mathrm{x}'' & y'' & x^{1''} & y''' & x^{(4)} & .. \end{array} \\ \left[\begin{array}{cccccccccc} A & B & C & & & & & & & \\ & & & A & B & C & & & & \\ & & & & & & A & B & C & \\ & & & & & & & & A & B & C \\ & & & & & & & & & \vdots & \vdots \end{array}\right] \end{array} \begin{array}{c} F \\ F' \\ F'' \\ pm \\ \vdots \end{array} \qquad (9)$$

The derivative array equations consist of an underdetermined system of (1/ + l)n 4- *rid* variables and *(v + l)n* equations. The Jacobian of the derivative array equations is rank deficient because of the algebraic equations which are implicit within them.

We still need an efficient method for the solution of this nonlinear underdetermined system which is rank deficient. Additional problems include inconsistent linearizations, roundoff errors and poor starting points and user specifications. To deal with this problem we propose the following optimization formulation which we solve with SLP.

2.2 SLP Formulation

Consider the solution of a nonlinear system of equations subject to inequality constraints and variable bounds (CSP).

$$\begin{array}{l} h(x)=0 \\ 9(*)\leq 0 \\ x^L \leq x < x^u \end{array} \qquad \text{(CSP)}$$

To solve (CSP), a merit function is defined as

$$\mu(x^i) = \sum_j |h_j(x^i)| + \sum_k g_k(x^i)_+ \tag{10}$$

where jte(x')+ = maxfOj^x1')]. The function /x(x) is nondifferentiable and attains its minimum value of zero if and only if (CSP) is solved. Using SLP we minimize jx(x) directly subject to $x^L \leq x \leq x^u$. By adding auxiliary variables, this problem can be written as :

$$\begin{aligned} & Min\ ]C\ to + {}^{n}i) + Yl^{Sk} \\ & \quad\ \ i \qquad\qquad\qquad\quad k \\ S.t. \quad & kj(x) = Pj - Tlj \\ & 9k\{x) \leq s_k \\ & x^L \leq x \leq x^u \end{aligned} \tag{11}$$

It can be shown that *(pj + tij)* represents |/ij(x)| and 5^ represents *gk(x)+*. Linearizing the constraints about x* leads to the following constrained simulation linear program, CSLP.

$$\begin{aligned} & Min \sum_j (p_j + n_j) + \sum_k s_k \\ s.t. \quad & h_j(x^i) + \nabla h_j(x^i)^T d^i = p_j - n_j \\ & g_k(x^i) + \nabla g_k(x'fcf \leq 5_{fc} \\ & x^L \leq x^i + d^i \leq x^u \\ & p_j, n_j, s_k \geq 0 \end{aligned} \tag{CSLP}$$

Hie solution of (CSLP) generates a search direction d\ Through the addition of artificial variables for the constraints, the LP is always feasible and generates a descent direction for every nonzero solution of the LP. A steplength algorithm combining both line search and trust region methods has been shown to enhance the computational efficiency and increase the robustness of the algorithm. A complete description of the SLP strategy is presented in Gopal and Biegler (1995). Zhang *et al.* (1985) and Duff *et al.* (1987) have also described trust region based Successive Linear Programming approaches in the context of nonlinear programming and nonlinear equation solving, respectively.

We can also extend problem (CSP) and the SLP approach by minimizing some distance from the specified conditions for the initialization problem. Let the specified initial conditions be $X\{ = A'^{*p}_i$ for t = l,...ni$_C$ where A' $\subseteq$ {x,x\y}.

An optimization formulation for consistent initialization is

$$
\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x, z, z', \ldots, z^{\nu}, t) = 0 \\
& -Si \leq Xi - Xf_t^p < \delta_i \qquad i = 1, \ldots, n_t > \\
& < J_t \geq 0 \qquad t = 1, \ldots, n < c \\
& A'/'' \leq Xi \leq X? \qquad (12)
\end{aligned}
$$

where *Ui* are weights that scale the deviations from the specifications. The NLP (12) is solved using Successive Linear Programming (SLP). The method generates a search direction from a linear program obtained from the linearization of (12). If the given specification is exact, the solution of the problem will correspond to a consistent set of initial conditions and a zero objective function. In case of an inconsistent overspecification, the optimum solution of (12) will be a positive value of the objective function, with a consistent set of initial conditions for the problem, closest to the specified set.

Note that the formulation (12) involves minimization of the weighted 1-norm of the violation of initial condition specifications. Hence the number of initial conditions set equal to the specified values at the solution need not be equal to the degrees of freedom for the specification of initial conditions (r) for the problem. Nonetheless it should also be noted that the objective value at the solution of the LP will be less than or equal to what it would have been if exactly r of the variables were set equal to their specified values. This is because (12) is not constrained to fix r initial conditions to their specified values but to find a consistent set of initial conditions closest to their specified values.

However, the user might want to have r variables to be set equal to their specified values, if the solution of (12) fixes less than r specifications. In other words, he may wish to pick the best set of r variables so that the specification violation of the rest of the variables is minimized. The degrees of freedom could be determined from the rank analysis of the Jacobian of the derivative array equations (Campbell and Moore, 1994). Once r is known, the following Mixed Integer NonLinear Programming (MINLP) formulation (13) could be used for consistent initialization. A;s are binary variables and M;s correspond to the big-M constraints.

$$
\begin{aligned}
& Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x, z, z', \ldots, z^{(\nu)}, t) = 0
\end{aligned}
$$

$$-6_{\{} \leq Xi-Xf^{p} \leq 6i \qquad i = 1,\ldots, n_{<c} \qquad (13a)$$

$$0 \leq Si \leq M,(1 - X_t) \qquad t = 1 \ldots n_{ie} \qquad (136)$$

$$\sum^{n_{ie}} \lambda_i = r \qquad (13c)$$

$$A<=O,1 \qquad t = l,\ldots,flfc$$

$$A'/' \leq X_{\{} \leq X? \qquad (13)$$

A value of A» = 1 =>·6, = 0 which implies that variable *Xi* is set equal to its specified value at the solution of the problem. The constraint *Yl'£i* A* = ^{r} serves of purpose of setting at least r variables to their specified values. However, (13) is an MINLP and could be expensive for the solution of the above problem. We would like to simplify the above formulation to a form where it could be solved using SLP.

If we rewrite (13a) as

$$Si + Xi - A'f^{p} \geq 0 \qquad (14ai)$$

$$*«-A', + A'F \geq 0 \qquad (Ma*)$$

The following inequalities are introduced in place of (13b).

$$Xi\,[\{Si + Xi - X?) + (Si - Xi + X?)] \leq 0 \qquad (1460$$

$$0 \leq Xi \leq 1 \qquad (14fr_2)$$

Note that A; in (14b) are continuous variables bounded between zero and one as opposed to the binary A; used in (13b). (14b) could be simplified to:

$$\lambda_i \delta_i \leq 0 \qquad (14c_1)$$

$$0 \leq Xi \leq 1 \qquad (14c_2)$$

Consider the case A; > 0. (14cO will be satisfied *iff Si* = 0 because of the nonnegativity constraint on *S{. ie. Xi* = X_i^{p}. A nonzero A* corresponds to the variable X,- set equal to its specified value. Similarly a nonzero *Si (Xi* ^ X_t^{*p}) implies A; = 0.

The summation equation (13c) is retained. (14ci) smd the nonnegativity conditions At ≥ 0 and *Si* ≥ 0 reduces to the complementarity condition Aj<5» = 0. The simplified formulation using continuous variables is given by (14).

$$Min \sum_{i=i}^{n_{ie}} \omega_i \delta_i$$

$$
\begin{aligned}
\text{s.t.}\quad & F^{\nu}(x, z, z', \ldots, z^{(\nu)}, t) = 0 \\
& -\delta_i \le X_i - X_i^{sp} \le \delta_i \qquad i = 1, \ldots, n_{ic} \\
& \lambda_i \delta_i = 0 \qquad i = 1, \ldots, n_{ic} \\
& \sum_{i=1}^{n_{ic}} \lambda_i = r \\
& 0 \le A_{f'} \le 1 \qquad i = 1, \ldots, n_{fc} \\
& \langle J_i \ge 0 \qquad i = 1, \ldots, n_{\langle c} \\
& A_{f'}^{L} \le A'_{f} \le A_{f}
\end{aligned}
\tag{14}
$$

The reformulation of the original MINLP problem to an NLP (14) constrained by complementarity conditions presents us with an opportunity to use a variety of techniques suggested to solve complementarity problems (Billups *et* a/., 1995). Chen and Mangasarian (1995) proposed a class of smoothing functions to solve the complementarity problem approximately. A and *6* satisfy the complementarity condition if and only if A = (A - <$)+ where (A - <$)+ = max(0, A - *S*). The max function is approximated using the smoothing function

$$\max(O,q) \ll 17 + 0 \ \log(l + e - \bar{w}) \tag{15}$$

As the parameter *0* approaches zero the smooth approximation approadies the nonsmooth max function. Using the smooth approximation (15), (14) can be reformulated as

$$
\begin{aligned}
& A/in \sum_{1=1}^{n_{ic}} \omega_i \delta_i \\
\text{s.t.}\quad & F''(x, z_y; z', \ldots, z^{(\nu)}, t) = 0 \\
& -\delta_i \le X_i - X_i^{sp} \le \delta_i \qquad i = 1, \ldots, n_{ic} \\
& Si = 0 \ \log(l + e^{-(\lambda_i - \delta_i)/\beta}) \qquad i = 1, \ldots, n_{ic} \\
& \sum_{i=1}^{n_{ic}} \lambda_i = r \\
& 0 \le A_{f'} \le 1 \qquad i = 1, \ldots, n_{fc} \\
& Si \ge 0 \qquad i = 1, \ldots, n_{\langle c} \\
& A'/' \le A'_{f} \le A'/
\end{aligned}
\tag{16}
$$

Formulation (16) is an NLP and is solved using SLP since a vertex solution is expected. (16) is less cumbersome to solve when compared to (13). We also show in the Appendix that if the degrees of freedom (r) are specified exactly, under certain conditions, a nonzero A, will take its upper bound value of 1,

which is a solution of the MINLP formulation (13).

2.3 Examples

Below, we consider three example problems to illustrate the application of formulations (12) and (16).

Example 1: Chemical Reactor Problem

Consider the problem described by (17) (Campbell and Moore, 1994; Pantelides, 1988).

$$\begin{aligned} C' + C + R &= 4 + t + i^1 \\ T^1 + 2T + R + Tc &= 14 - eT^l \\ T\sim^l + ln\{R/C) &= 0 \\ C &= cosh\{t - 1) \end{aligned} \tag{17}$$

The problem is index 3 and has a unique solution. There are no degrees of freedom for specification of the initial conditions. However since there are two differential equations, the user may be tempted to specify initial conditions corresponding to the differential variables. The formulation (12) should be able to take care of any such inconsistent specifications and give us a consistent set of initial conditions closest to the specified ones (in this case the unique initial vector). In table 1 we consider two inconsistent specifications for the initial conditions. In both cases the SLP method solved for the correct set of initial conditions with a nonzero objective function at the solution. Note that the second case corresponds to the commonly used steady state initial condition specifications. D

Example 2: Pendulum Problem

This is the popular pendulum problem considered by various authors (Pante lides, 1988; Gear, 1988; Unger *et* a/., 1995). The formulation is given by (18).

$$\begin{aligned} x^1 &= t; \\ y' &= w \\ v^1 &= -Tx \\ nf &= -Ty - g \end{aligned} \tag{18}$$

$$X^2 \text{"I } 1/^2 \text{ TM" } M.^i$$

The problem is of index 3 and has 2 degrees of freedom for specification of initial

Table 1: Initial Conditions for the Chemical Reactor Problem

Specified ICs	Actual ICs (SLP Solution)	Number of SLP Iterations
C = l 7b = 0	C = 1.5431 R = 3.6321 T =-1.1682 T_c = -0.5726	12
C' = 0 T' = 0	C = -1.1752 T = 1.2768	6

conditions. The two underlying algebraic equations (19) are derived from the successive differentiation of (18).

$$\begin{aligned} xv + y\,w &= 0 \\ -T(x^2 - y^2) + tr^2 + ur^2 - gy &= 0 \end{aligned} \tag{19}$$

Table 2 shows how the proposed method solves for a consistent set of initial conditions for three different specifications.

The number of specifications in the first two sets is equal to the degrees of freedom (2) and the sets *(x,v)* and (z,T) form a feasible pair (The Jacobian of the original set of equations with respect to all the variables other than the specified ones is nonsingular). As seen, applying SLP to (12) solves for the consistent initial vector with the solution for the specified variables being equal to their specified values. However, the third set is an overspecification. In this case SLP finds the initial condition vector which is closest, in the 1-norm sense, to the specified ones.

Now consider the application of (16) to this problem. As seen from the specifications in table 3, even though there are two degrees of freedom for this problem, solution with (12) sets only one variable to its specified value. On the other hand, the problem when solved with (16) and r = 2, sets x and w to their specified values. The corresponding objective function is 0.3660 which is the deviation in v. This objective is greater than the value of 0.3128 from (12). D

Table 2: Initial Conditions for the pendulum problem

Specified ICs	Solution from (12)	Number of LPs solved
i = 0.5 **v = 1.0**	**i = 0.5** **i> = 1.0**	11
x = 0.5 *T* = 0.5	**i = 0.5** **r = o.5**	12
i = 0.5 w=1.0 T = 0.5	1 = 0.5 *v* = 1.012 T = 0.5	12

Table 3: Solution for the pendulum problem from inconsistent specifications using (12) and (16)

Specified ICs	Solution from (12)	Solution from (16)
i = -0.5 « = 0.5 w = 0.5	i =-0.5463 y = 0.8376 *v* = 0.7665 u; = 0.5000 *T* = 0.0000 £<S, = 0.3128	*x* = -0.5000 y = 0.8660 v = 0.8660 w = 0.5000 *T* = 0.1340 **Efc = 0.3660**

Example 3: Trajectory Prescribed Path Control (TPPC) Problem
Here we consider a nonlinear example used in the simulation of space vehicles

when the shape of the trajectory is prescribed. TPPC problems are treated extensively in Brenan *et al.* (1995). .(20) is an index 3 problem where the first six equations are the equations of motion in relative coordinates and the last equation is the algebraic path constraint.

$$
\begin{aligned}
&= V^*\sin 7 \\
\xi &= \frac{V_R \cos\gamma \sin A}{r\cos A} \\
A' &= \frac{VR}{r}\text{COS7COS/1} \\
V_R' &= \frac{D}{m}0\sin 7\text{-ft£}^2 r\cos A(\sin A\cos 4\cos 7\text{-}\cos A\text{siii}7) \qquad (20) \\
\gamma' &= \frac{L\cos\beta}{mV_R} + \frac{\cos\gamma}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega_E \cos\lambda \sin A + \\
&\quad \frac{\Omega_E^2 r\cos\lambda}{V_R}(\sin A\cos A\sin 7 + \cos A\cos 7) \\
A' &= \frac{L\sin\beta}{mV_R\cos\gamma} + \frac{V_R}{r}\cos\gamma\sin A\tan\lambda - 2\Omega_E(\cos\lambda\cos A\tan\gamma - \sin\lambda) + \\
&\quad \frac{Q_E r\cos A\sin A\sin X}{V_R\cos\gamma} \\
\frac{D}{m} &- \left[C_0 + C_1(V_R - V_0) + C_2(V_R - V_0)^2 + C_3(V_R - V_0)^3\right] = 0
\end{aligned}
$$

Computations by Campbell and Moore (1994) using the initial values of the state variables obtained from Brenan *et al.* (1995) found that their values were not exact. Using these values as initial guesses, they generated new initial conditions using a least squares method. There are two points to be noted in this problem:

(1) Arbitrary specification of all 7 state variables will lead to problems even if they are slightly inconsistent (as observed by Campbell and Moore (1995)) since the degrees of freedom is 4.

(2) A different undesirable control history of *0* < 0 is possible for this problem. *0* must be positive in the given set of initial conditions.

The initialization problem when formulated using (12) and initialized using the values from Brenan *et al* (1995), returned consistent initial conditions within an accuracy of 10^{-12} in just 2 iterations. The SLP method has the added advantage of an easy imposition of a lower bound of zero on *0* to avoid the undesirable control history. •

hi summary, when the failure of the solver is due to inconsistent specification of initial conditions by the user, it is advantageous that the user be presented

with a consistent set of initial conditions which is closest to what had been specified before and is numerically consistent. The two formulations, (12) and (16), serves this purpose and give the user flexibility to use the available information to the best advantage. Campbell and Moore (1994) point out that in practical applications, one would expect that some initial values are known and there are reasonable estimates for others. The method has relevance in this context too, where the weights <JK in (12) and (16) could be chosen to help the user determine a consistent initial vector reflecting his confidence level in the initial value estimates.

3 Reinitialization after discontinuities

In this section, we look at the problem of reinitialization of DAEs after discontinuities, some underlying principles for this problem and the application of the minimization formulation.

3.1 Problem description

The task of finding a consistent set of initial conditions for the reinitialization of a DAE is often more complicated than the *pure* initialization problem. The reason is that the reinitialization problem involves an additional exercise of determining the variables which are continuous across a discontinuity. The numerical values of the continuous variables at the point of discontinuity contribute to the specifications for the reinitialization problem.

Majer *et al.* (1995) address the problem of reinitialization of DAEs after discontinuities. They specifically look at index one linear-implicit DAEs of the form (21), where there are discontinuous steps in the forcing function *u.*

$$T\{v,u,t)v' = h(v,u,t) \tag{21}$$

u is the vector corresponding to the input variables and an algebraic transformation of the form (22) exists

$$\begin{aligned} T_{11}(x,y,u)x' + T_{12}(x,y,u)y' &= h_1(x,y,u) \\ 0 &= \mathrm{Mx,y,u)} \end{aligned} \tag{22}$$

They show that the continuity assumptions of state variables can be safely assumed by default only when TV; = 0. For the case T_n £ 0, conditions are laid out for the solvability of consistency equations.

Brüll and Paliaske (1992) considered the same index one problem and derived the necessary and sufficient condition for the above system to have a *genuine*

initial value for the state variable vector *v* after a jump in the input variable u.

However, no work has been done for determining the validity of the continuity assumptions of state variables for higher index nonlinear DAEs. In this section, we consider a class of such problems in whidi the discontinuity is caused by a step change in the input function *u.*

An assumption about this class of problems is that the index of the problem remains the same at all times. Also, if we were to look at the discrete models at any time as independent problems, the degrees of freedom for the specification of initial conditions remain the same. Often it is assumed that the differential variables are continuous if the composition of the differential variables appearing in the describing equations remain unchanged across the discontinuity. This is not always true as was shown by Majer *et* a/. (1995) and Brüll and Pallaske (1992) for hilly implicit index one systems. When considering problems of arbitrary index, what is missing are rigorous criteria to determine whidi variables are continuous and whidi are not, across a discontinuity caused by a step diange in one or more input functions.

3.2 Continuity assumptions for problems of arbitrary index

Consider a DAE of the form

$$F(x',x,_{Vl}ti,t) \quad = \quad 0 \tag{23}$$

where u € R^{n}". *The continuity assumptions hold for tliat subset of variables* v_0, *whose underlying ODEs do not depend on any of tlie derivatives of Hie function u causing the discontinuity.*

Let the system be of index *v.* Differentiation of (23) *v* times will yidd an ODE of the form (24) whidi is termed as the underlying ODE for the DAE (23). *v is* defined as in (3).

$$t/ \quad = \quad /(ti,ti,u'. . . . .iiM.t) \tag{24}$$

Denote tT = $v(t^9$ - 0) and v+ = $v(t^*$ + 0), where t^m is the time of discontinuity. Let $v_{\{}$ be a variable which has an underlying ODE of the form (25) where $u^{(k)}$ is tlie highest derivative of *u* explicitly present in the underlying ODE.

$$v'i \quad = \quad fi\{v, ti_f \ldots_f u<«_f t) \tag{25}$$

Let *ust* be a smoothed step for *u* defined for_*V* $\leq t<V+6t$ and v_{st} be the corresponding smoothed profiles of the variables. As in Majer *et al.* (1995),

$$v_i^+ = lim_{\delta t \to 0}\{v_i^- \quad {}^{\cdot t+tt} \quad fi\{v6t,u_{6t},...,u^{(k}_{6\ t}\backslash t)dT\} \tag{26}$$

The derivatives of u are not bounded for step changes in u. The integral will vanish only when the derivatives of u do not form part of the integral, explicitly or implicitly through the variables appearing in the underlying ODE. Thus, only those variables whose underlying ODEs could be shown to be independent of any derivative of the input function causing the discontinuity, are continuous across it.

Let the set of continuous variables be v_0. From the above analysis, it is clear that there is no guarantee that all differential variables will fall in the set *VQ*. Hence, the assumption of continuity of differential variables across the discontinuity, often used in dynamic simulators, is not always true.

An underlying ODE can be determined using a symbolic algorithm, for example by Gear (1988). However, the form of the ODE found depends on the algebraic manipulation one carries out while deriving it. The above analysis provides a framework for determining the variables which are continuous across a discontinuity, but from a computational standpoint the issue of uniqueness of the underlying ODE of a given DAE (1) is to be addressed.

Example 4
Consider the DAE described by (27).

$$\begin{aligned} y' + x &= u\backslash \\ y &= t i_2 \end{aligned} \tag{27}$$

Consider the following two forms of the underlying ODE, (28) and (29).

$$\begin{aligned} x' &= u\backslash - u\$ \\ y' &= u!_2 \end{aligned} \tag{28}$$

$$\begin{aligned} x' &= u\backslash - u\$ \\ y' &= u_x - x \end{aligned} \tag{29}$$

In the event of a step change in u>, (28) gives a clear picture of the behavior of y, whereas from (29), it is not clear because of the implicit dependence on x. O

However, in practical applications, one would not prefer to explicitly determine an underlying ODE and then validate the continuity assumptions. Given all the constraining algebraic equations for a DAE and any particular underlying ODE, one could derive other forms of underlying ODEs. The derivative array equations, in principle, contain the differential and all the constraining algebraic equations for a given DAE. Hence, it would be easier to work with the derivative array equations and determine the dependencies of the differential

variables on the derivatives of u rather than deriving the required form of the underlying ODE explicitly.

In the following section we describe two approaches for finding these dependencies for linear DAEs.

3.3 Identification of the continuous set

Consider the derivative array equations (30) of the DAE (23). Let us consider the linear case.

$$F^{\nu}(x^{(\nu+1)}, \ldots x, y^{(\nu)}, \ldots, y, u^{(\nu)}, \ldots u, t) = 0 \tag{30}$$

/. Analysis of the Jacobian of the derivative army equations

An analysis of the Jacobian (coefficient matrix in the linear case) of the derivative array equations can be used to determine the set of variables for which the continuity assumption is valid. Construct the Jacobian of (30). Note that the higher derivatives of u are treated as independent variables here. Consider the example where a Gauss-Jordan elimination gives us the following sub-matrix (31) within the Jacobian where &, &, (3,... are the incidences in the columns corresponding to the derivatives of u.

$$\bar{J}_{ode} = \begin{array}{c} \begin{array}{cccccc} & & v' & & tt' \text{«"} & \ldots \quad u<^{\nu)} \end{array} \\ \left[\begin{array}{cccccc|cc} 1 & & & & & & & \xi_1 \\ & 1 & & & & & \xi_2 & \\ & & \cdot & & & & & \\ & & & \cdot & & & & \\ & & & & 1 & & & \\ & & & & & 1 & \xi_3 & \end{array}\right] \end{array} \tag{31}$$

The continuous set, t*>, could be easily identified from (31). The state variables corresponding to the rows whidi do not contain incidences of u', ..., $u^{(i/)}$ are continuous across the discontinuity caused by the forcing function *u.* However we need to safeguard against implicit dependencies. This could be done by choosing an appropriate pivoting strategy while doing symbolic elimination on the Jacobian of the derivative array equations. A verification of this can be carried out from the structure of the Jacobian after performing the elimination. If a particular variable t/* is found to be continuous, we should not be able to trace a path from an incidence in the column under v'_k to an incidence in the derivatives of u. That is, the set of reachable nodes from v_k should not contain any of the derivatives of u, if v* is continuous. If such a path is found, further

pivoting can give us the desired structure. Pivoting is straightforward in the case of linear systems. Below we illustrate how this approach solves the example of (27).

We could determine the implicit dependehdes from a form like (29) by path tracing on the corresponding Jacobian of the derivative array equations (32).

$$\begin{aligned} y' + x &= u_1 \\ y'' + x' &= u_1' \\ y &= u_2 \\ y' &= u_2' \\ y'' &= u_2'' \end{aligned} \tag{32}$$

Consider the Jacobian (33) of F(32).

$$\begin{array}{cccccccccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \\ \end{array}$$
$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c} & 1 & & & 1 & -1 & & & & \\ 1 & & 1 & & & 1 & -1 & & & \\ & & & 1 & & 1 & & -1 & & \\ & 1 & & & & 1 & & & -1 & \\ 1 & & & & & 1 & & & & -1 \end{array}\right] \tag{33}$$

Manipulations on the Jacobian (33) will give the following incidence matrix for the form (29).

$$\begin{array}{cccccccccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \\ \end{array}$$
$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c} & 1 & & & 1 & -1 & & & & \\ & & 1 & & & & -1 & & & 1 \\ & & & 1 & & & & -1 & & \\ & & & & -1 & 1 & & & -1 & \\ 1 & 1 & & & & & & & & -1 \end{array}\right] \tag{34}$$

We can trace a path from an incidence in the column y' to that in u^ showing that there is dependence on u^ which is implicit in (29). Further pivoting in the column under x will give the desired form (35) which corresponds to (28).

$$\begin{array}{cccccccccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \\ \end{array}$$
$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c} & 1 & & & & & & & -1 & \\ & & 1 & & & & -1 & & & 1 \\ & & & 1 & & & & -1 & & \\ & & & & -1 & 1 & & & -1 & \\ 1 & 1 & & & & & & & & -1 \end{array}\right] \tag{35}$$

t Sensitivity analysis of the derivative array equations

An alternate approach suggested to find the dependencies of the variable derivatives on the derivatives of the input function is to do a sensitivity analysis on the derivative array equations. Caracotsios and Stewart (1985) and Maly and Petzold (1995) describe sensitivity analysis for DAEs. However, since we are not interested in finding the sensitivity of the entire profile, but just on the dependencies at the point of discontinuity, we consider the following SLP formulation:

$$
\begin{aligned}
& Min \sum_{i=1}^{n} (p_j + n_j) \\
s.t. \quad & F/(x<^{"+1}>,...x,y^{(1/)},...,y,ul»K...u,t) = {}_{Pj} - nj \quad j = 1,...,(*+1)n \\
& {}_{u}(*)_{=ti}(*) - \qquad ik = 0,...,i/ \\
& Xt \leq Xi \leq X? \qquad (36)
\end{aligned}
$$

where $u^{w}\sim = u\wedge(t^{m} - 0)$. The numerical value of the variables at the point of discontinuity is the optimal solution to (36). Sensitivity analysis is carried out with u and its higher derivatives being treated as independent variables and by linearizing (36) at the variables values before the discontinuity. If a particular derivative $x'j$ is seen to be insensitive to the parameters $u',....,u^{(1"H}\backslash$ it is because it has no local dependence on any of the derivatives of the input variables ie. it is continuous across a discontinuity caused by a step change in u. However, one should be aware tiiat the test is local and the analysis provides a framework which is necessary but not sufficient for determining the continuities. The sensitivity analysis is performed using LP sensitivity, which is computationally cheap.

Nonlinear DAEs

The current approach we use for determining the dependencies for nonlinear DAEs is the sensitivity analysis described above. Since (36) is solved using SLP, it could be used for nonlinear systems as well. The sensitivity analysis would effectively be performed on the linearized form of the equations at $(t^{m} - 0)$. The method has been applied to the example problems solved in section 3.5.

Application of pivoting strategies on the Jacobian of the derivative array equations for the nonlinear case is limited as one can only determine, in graph theory notation, a *maximal* set of reachable nodes. If this maximal set does not contain any nodes corresponding to the implicit dependencies, the sets determined thus far are correct. Hence, analysis on the Jacobian may not give the desired result for nonlinear DAEs.

3.4 Applications of the LP Method for Reinitialization

A certain number of specifications for the initial conditions is required for restarting the problem after the discontinuity. Some of these are obtained from the criterion laid out for the validity of continuity assumptions of the variables across the discontinuity. However, the number of variables which are continuous across the discontinuity could be less than, equal to or greater than the degrees of freedom for the specification of initial conditions. Let us analyze the three cases and look at how the problem could be tackled in each of these cases.

/. Number of variables continuous across the discontinuity is greater than the number of degrees of freedom.

Here the number of specifications is greater than the number of degrees of freedom, the specifications being the numerical values of the variables which are found to be continuous across the discontinuity. This is an overspecified problem. It is quite likely that if we fix the values of these variables and solve the equations for consistent initialization, we might not be able to converge it within the required error criterion. One of the reasons for this is the roundoff errors arising from integration in the previous time domain. It is well known that if the initial conditions are not sufficiently accurate, further solution of the problem could be difficult and inaccurate. For this reason we would like to determine the initial conditions such that the equations are solved within a desired error criterion. The following formulation, which is similar to (12), is introduced for this purpose.

$$
\begin{aligned}
& Min \sum_{i \in v_0} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x^{(\nu+1)}, \ldots x, y^{(\nu)}, \ldots, y, u^{(\nu)}{}_{f} \ldots u.t) = 0 \\
& -\delta_i \leq v_i - vf^{itc} \leq \delta_i \qquad i \in v_0 \\
& 6i \geq 0 \qquad i \in VQ \\
& Xt \leq Xi \leq X? \qquad (37)
\end{aligned}
$$

VQ is the set of variables which are found to be continuous across the discontinuity and vf^{itc} are their numerical values at the point of discontinuity.

2. *The number of continuous variables is equal to the degrees of freedom.*

This is the easiest case where we have the exact number of specifications. (37) could be applied to determine the values of the remaining variables and the solution will correspond to a zero objective function.

3. The number of continuity conditions is less tJian the degrees of freedom.

This is an underspecified problem. However, in process engineering problems there may be discontinuity constraints which are known along with a step change in the input functions (For example, correlations for computing impulse forces in the case of collision of nonpenetrating rigid bodies). Let *ff(v)* = 0 be the discontinuity relations for the profiles. Assuming that the discontinuity constraints provide at least the minimum required information to cover the remaining degrees of freedom, the following formulation could be used for consistent reinitialization.

$$
\begin{aligned}
& \text{A/in} \quad 5>\text{M?} + \sum_{i=1}^{n_f} \omega_i^2 \delta_i^f \\
& \quad t\in v_0 \\
\text{s.t.} \quad & F\ {}^{n_f}x^{(d,u+1)}, \ldots x, y^{(\nu)}, \ldots, y, u^{(\nu)}, \ldots u, t) = 0 \\
& -\delta_i^c \le v_i - v_i^{disc} \le \delta_i^c \qquad i \in v_0 \\
& -\delta_i^f \le f_i^d \le \delta_i^f \qquad i = 1, \ldots n_f \\
& \delta \qquad \ge 0 \\
& Xt \le X_{\{} \le \text{Af}
\end{aligned}
\tag{38}
$$

The same formulation could be applied to cases 1 and 2 as well. One of the major difficulties associated with reinitialization problems is to determine whether a particular discontinuity presents an under, exact or over specified problem and then the associated clumsy process of the selection of that subset of equations which has to be solved. (38) presents a single consistent formulation to handle all of these cases.

One question still unanswered is the presence of the input function *u* and its higher derivatives in the derivative array equations used in (38). While state profiles cannot be calculated that capture the exact discontinuities, profiles can be determined to reinitialize the problem in the next time domain. Here we let

$$
u = \begin{cases} f_a(t) & \\ f_b(t) & t>\underline{t} \end{cases} \tag{39}
$$

Using the Heaviside function (40), u is written as (41)

$$
u(t) = f_a(t)(1 - Y(t^*)) + f_b(t)(Y(t^*)) \tag{41}
$$

ťth derivative of (41) could be written as

$$
u^{(i)}(t) = \sum_{j=0}^{i} \binom{i}{j} Y^{(j)}(t^*)(f_b - f_a)^{(i-j)} + f_a^{(i)} \tag{42}
$$

FVom the theory of distributions (Banros-Neto, 1973), the first derivative of the Heaviside function corresponds to the Dirac delta function and its higher derivatives in (42) are the higher derivatives of this function. These functions are defined only at f.

Our problem is to find initial conditions for the derivatives of *u* for the solution of the problem in the time domain *after* the discontinuity. This is different from tracing the solution from one side of the discontinuity to the other where the derivatives of *u* are unbounded at the point of discontinuity. Here the idea is to find the initial conditions for the derivatives of u which will enable us to correctly trace the solution trajectory after the discontinuity. This is obtained by finding the limit of (42) when t -+t^m from the rhs. The limit is given by

$$\lim_{\epsilon \to 0+} v^{(i)}(t^m + €) = \operatorname*{Um}_{\epsilon \to 0+} V(«\bullet)(/* - f*)^{(i)}\{t^m + *) + f_a^{(i)}(t^{\bullet} + \epsilon)$$

$$= An \qquad (43)$$

All the terms containing the higher derivatives of *Y* vanish as they are zero at any point $(t^m$ + c) however small *e* may be. Thus for the reinitialization problem, the higher derivatives of it present in the derivative array equations take the fonn of the higher derivatives of their functional fonns defined in the region after the discontinuity.

Let us consider a simple example to illustrate the point. Consider the index three DAE of the form

$$x\backslash = y\text{-}xi$$
$$x'2 = X\backslash - X2$$
$$x_2 = «(t) \qquad (44)$$

where

«◈ - {? JI?

The problem is of zero degrees of freedom and the constraining algebraic equations are

$$X'l = U(t)$$
$$xx = u'(t)+ti(t)$$
$$y = u''(t) + 2u'(t)+u(t) \qquad (45)$$

The profiles can be plotted as in Figure 1.

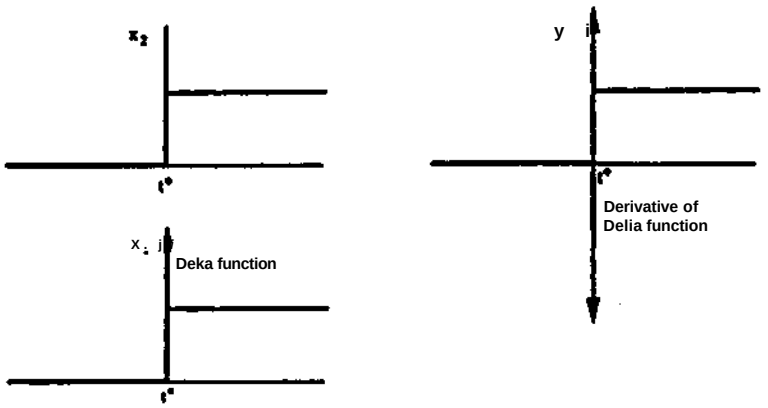

Figure 1: *Variable profiles*

As seen from Figure 1, $x\backslash$ and y are described by unbounded functions at t". Our problem is not to solve the problem at t^m_y but to solve for the initial conditions of xi, x-2 and y for the further solution of the problem after the discontinuity, u = 1 (and hence u^* = 0 and u" = 0) will give us values of xi = x_{-2} = y = 1 which are the correct initial conditions for the further solution of the problem in the region after the discontinuity.

Below we consider some example problems to illustrate the ideas described in this section.

3.5 Examples

We us consider four examples, two where the continuity assumption for the differential variables is valid and the other two where it is not. .The first three are nonlinear.

Example 5

This example looks at an implicit index-1 problem (46) considered by Brüll and Pallaske(1992). This is a case where the continuity assumptions are not necessarily valid after the discontinuity.

$$\begin{aligned} x_1' + y_1 x_2' &= 0 \\ x_2 &= U_X(t) \\ Vi &= u_2(t) \end{aligned} \tag{46}$$

After making reductions and simplifications, it is clear that the underlying ODE for (46) is

$$x_1' = -u_2(t)u_1'(t)$$

$$x_2' = u_1'(t) \qquad (47)$$
$$yi = txf_2(t)$$

As seen from (47), none of the variables are continuous after a step change in the input functions tti and u_2- However, if u\ is the only function causing the discontinuity, t/i is found to be continuous across the discontinuity. Similarly, xi and x-i are continuous across a discontinuity caused by a step change in the input function u≥. D

Example 6: Pendulum problem revisited

Consider the pendulum problem from (18) in a slightly different context. Let the pendulum be subject to time dependent input forces with the g in (18) replaced by an input function u. Below, we examine the continuity of the variables when there is a step change in the forcing function u. The underlying ODE is

$$\begin{aligned} z' &= t; \\ y' &= w \\ v' &= (yu\text{-}tr^{2}\text{-}tir^{3})x \qquad (48) \\ vf &= (yu - v^{1} - ti/^{2})y - it \\ T &= Zwu + yu^{1} \end{aligned}$$

FVom (48), it is clear that V depends on the first derivative of u whereas the derivatives of the other variables depend only on u.

$$«o = \{x,y,v,ti;\} \qquad (49)$$

The position and velocity variables are continuous but the string tension exhibits a step change in the event of a step change in the forcing function. Similar result is obtained from the sensitivity analysis. O

Example 7

Consider the DAE (50) for which the continuity assumption is valid. This problem was considered previously by Majer *et* a/.(1995)

$$\begin{aligned} *' &= yi - 1 \\ 0 &= x\text{-}yt + yi + u \qquad (50) \\ 0 &= 2y\{/^{3} + » \wedge - 4 \end{aligned}$$

Consider the solution of the reinitialization problem of (50). The problem has a steady state solution of yi = 1, $y2$ = 4 and x = 3 for u = 0. At time t = f, a step change is applied to u from 0 to 5.

Table 4: Reinitialization of Example 7

Initial Guesses	Solution	Number of iterations
Vi = 1 »2=4 i = 3	x = 3.0000 yi = 0.1975 yi = 8.0390 x' = -0.8025	10
Vi=0 yj=o i' = 0		12

From the sensitivity analysis, we find that x is continuous across the discontinuity caused by a step change in the input variable u. Setting v_0 = {z}, the initialization problem was solved using (38). Table 4 reports the solution of the problem for two sets of initial guesses.

The first set of initial guesses correspond to the values of the variables immediately before a step change in u was applied. This is probably the most logical initial guess to be diosen when carrying out dynamic simulation of the above system.

The second set is the initial guess used by Majer *tt* a/. (1995) for the solution of the problem. They point out that (50) cannot be solved by Newton's method from this set of initial guesses because of the limited region of convergence. They use a more expensive continuation method for the solution of the problem. However, it should be noted that the LP based formulation (38) encountered no difficulty in solving the problem.

Apart from being an efficient way for dealing with difficult nonlinear reinitialization problems, the method has been successfully applied to high index problems as well. Below we consider a high index example problem.

Example 8: The CSTR problem

Bachmann *tt al.* (1989) describe a problem of A^r Continuous Stirred Tank Re-

actors (CSTRs) in series without reaction which constitutes a system of index $N + 1$. The equations are

$$c_i' = c_{i-1} - c_i \quad i = 1, \ldots, N$$
$$CN = u(t) \tag{51}$$

Consider the case of 10 CSTRs. The problem is of index 11. Define u as

$$u(t) = \begin{cases} 10 + t & t \leq 10 \\ t & t \geq 10 \end{cases} \tag{52}$$

First the continuity assumptions were checked from the analysis of the Jacobian of the derivative array equations and from sensitivity analysis. It was found that none of the variables are continuous across a discontinuity caused by a step change in the u. The problem was then solved using the formulation (38). No variables were found to satisfy the continuity assumptions and there arc no explicitly specified discontinuity constraints. Note that this problem is exactly specified as the degrees of freedom and the number of continuous variables is zero.

The problem, which has 164 variables, 132 equalities and 22 inequalities was solved in 2 iterations. It was initialized at the solution of the problem before the discontinuity. The initial value for any c_i was found to be

$$Anitial \_\_ on \quad \rightarrow \quad \rightarrow \_\_ n \quad 10 \tag{53}$$

4 Conclusions

Overspecification of initial conditions, which occurs mainly due to the lack of knowledge of the degrees of freedom for the specification of initial conditions, could lead to convergence failures in the initialization problem. Most existing methods expect the user to supply the exact number of initial conditions. Instead a minimization formulation is presented here for initialization as well as for reintialization in the case of problems with discontinuity, which takes care of the overspecifications. This ensures that the initial conditions are determined within a required accuracy; failure to do so may result in convergence to poor or inaccurate solutions on further integration.

In section 3 we presented a criterion for checking the validity of the assumption of continuity of differential variables across a discontinuity caused by a step change in the input variables. Such criteria have not been reported in the literature for problems of general index. The popular assumption of continuity of all differential variables across a discontinuity (Pantelides and Barton, 1993) is shown to be not necessarily valid.

The applications of the ideas outlined in these sections have been demonstrated on small problems, of low and high index. The concepts apply irrespective of the size and index of the problem and application to larger problems is expected to be straightforward. Also, the proposed LP based approach is easily interfacable to DAE solvers like DASSL. This will make the treatment of the initialization and reinitialization problems easier and more user friendly within these solvers.

Acknowledgements
The authors are thankful to Prof. Steven L. Campbell for his valuable comments and for suggesting the example in Sections 3.2 and 3.3.

References

[I] R. BACHMANN, L. BRULL, T. MRZIGLOD AND U. PALLASKE, *On Mctliods for Reducing the Index of Differential - Algebraic Equations,* Computers chein. Engng., 14(1990), pp. 1271-1273.

[2] J. BARROS-NBTO, *An Introduction to the Theory of Distributions,* Marcel Dckker, 1973.

[3] S. C. BILLUPS, S. P. DIRSKE, AND M.C. FERRIS, *A Comparison of Algorithms for Large Scale Mixed Complementarity Problems,* Technical Report, Computer Sciences Department, University of Wisconsin, 1995.

[4] K. E. BRENAN, S. L. CAMPBELL, AND L.R. PETZOLD, *Numerical Solution of Initial'Value Problems in Differential'Algebraic Equations,* SIAM, Philadelphia, 1995.

[5] L. BRULL AND U. PALLASKE, *On Consistent Initialization of Differential-Algebraic Equations with Discontinuities,* in Proceedings of the Fourth European Conference on Mathematics in Industry, Tcubner, Stuttgart, 1992, pp. 213-217.

[6] S. L. CAMPBELL AND E. MOORE, *Progress on a General Numerical Method for Nonlinear Higher Index DAEs II,* Circuits Systems Signal Process, 13(1994), pp. 123-138.

[7] M. CARACOTSIOS AND W. E. STEWART, *Sensitivity Analysis of Initial Value Problems with Mixed ODEs and Algebraic Equations,* Computers diem. Engng., 9(1985), pp. 359-365.

[8] C. CHEN AND O.L.MANGASARIAN, *A Class of Smoothing Functions for Nonlinear and Mixed Complementarity Problems,* Technical Report, Computer Sciences Department, University of Wisconsin, 1995.

[9] Y. CHUNG AND A.W.WESTERBERG, *A Proposed Numerical Method for Solving Nonlinear Index Problems,* Ind. Eng. Chem. Res., 29(1990), pp. 1234-1239.

[10] Y. CHUNG, *Solving Index and Near Index Problems in Dynamic Simulation,* Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA 15213, 1991.

[II] I. S. DUFF, J. NOCEDAL, AND J.K. REID, *The Use of Linear Programming for the Solution of Sparse Sets of Nonlinear Equations,* SIAM J. Sci. Stat. Comput., 8(1987), pp. 99-108.

[12] C.W. GEAR, *Differential-Algebraic Equation Index Transformation,* SIAM J. Sci. Stat. Comput., 9(1988), pp. 39-47.

[13] V. GOPAL AND L. T. BIBGLER, *Nonsmooth Dynamic Simulation with Linear Programming Based Methods,* To appear, Computers chera. Bngng. (1996).

[14] B. LBIMKUHLER, L. R. PETZOLD AND C. W. GEAR, *Approximation Methods for the Consistent Initialization of Differential - Algebraic Equations,* SIAM J. of Numer. Anal., 28(1991), pp. 205-226.

[15] C. MAJER, W. MARQUARDT AND E.D. GILLES, *Reinitialization of DAEs After Discontinuities,* Computers chem. Engng. 19(1995), pp. S507-S512.

[16] T. MALY AND L. R. PETZOLD, *Numerical Methods and Software for Sensitivity Analysis of Differential* Algebraic Systems,* To appear, Applied Numerical Mathematics (1996).

[17] C. C. PANTELIDES AND P. I. BARTON, *Equation-Oriented Dynamic Simulation: Current Status and Future Perspectives,* Computers chem. Engng. 17(1993), pp. S263-S285.

[18] C. C. PANTELIDES, *The Consistent Initialization of Differential - Algebraic Systems,* SIAM J. Sci. Stat. Comput., 9(1988), pp. 213-231.

[19] J. UNGER, A. KRONER, AND W. MARQUARDT, *Structural Analysis of Differential - Algebraic Equation Systems - Theory and Applications,* Computers chem. Engng., 19(1995), pp. 867-882.

[20] J. ZHANG, N. KIM AND L. LASDON, *An Improved Successive Linear Programming Algorithm,* Mgrat. Sci., 31(1985), pp. 1312-1331.

Appendix - Properties of Smooth Formulation

Conditions for which *Si* = 0 =» A* = 1

The smooth formulation presented in Section 2 is:

$$Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i$$

s.t. $F^*\{x,z,z\ldots_{\%}z^{i¥}\backslash t)=Q$ **(16a)**

$-\delta_i \leq X_i - X_i^{sp} \leq Si \quad \dot{t} = 1,\ldots,n_{fc}$ **(166)**

$\delta_i - \beta \log(1 + e^{-(A\cdot\,\wedge)\wedge}) = 0 \quad \dot{t} = 1,\ldots,n_{t\cdot c}$ **(16c)**

$\sum_{i=1}^{n_{ic}} \lambda_i - r = 0$ **(16d)**

$0 \leq Ai \leq 1 \quad \dot{t} = 1,\ldots,nfc$ **(16c)**

$Si \geq 0 \quad \dot{t} = 1,\ldots,n_{\dot{t}c}$ **(16/)**

$Xt \leq Xi \leq X?$ **(16$)**

The Karush-Kuhn-TVicker conditions of (16) arc

$$\frac{\partial L}{\partial \lambda_i} = \phi_i \frac{e^{-(\lambda_i - \delta_i)/\beta}}{1 + e^{-(\lambda_i - \delta_i)/\beta}} + \gamma - \xi_i^1 + \xi_i^2 \quad = \quad 0 \tag{54}$$

$$\xi_i^1 \lambda_i \quad = \quad 0 \tag{55}$$

$$\xi_i^2 (\lambda_i - 1) \quad = \quad 0 \tag{56}$$

where &, 7 and & are the multipliers of (16c), (16d) and (16e) respectively. FVom (16c),

$$1 + e^{-(\lambda_i - \delta_i)/\beta} \quad = \quad e^{\delta_i/\beta} \tag{57}$$

(54) could be simplified as

$$_{A\ e} - \wedge + 7 - 6^1 + €? \quad = \quad 0 \tag{58}$$

The smoothing constraint (16c) could be written as

$$_{e} - */* \ _{+\ c} - V * - \_i \tag{59}$$

When <J = 0, € - * " = 1. FVom (59),

$$e^{-\lambda_i/\beta} = 0 \tag{60}$$

For any /? > 0, A* -* 00. Choosing *0* small will allow A* < 00, in order to satisfy $_{e}$-*Xi/0* $\leq$ e_t a nonzero tolerance. FVom (60) and (55),

$$tf \quad = \quad 0 \tag{61}$$

FVom (58), (60) and (61), we have when *6*< = 0,

$$7 + \ tf \quad = \quad 0 \tag{62}$$

To show that A» goes to its upper bound (ie. £*j* £ 0) we need to show that 7 is nonzero. The following conditions are considered based on the application of (16).

- Consider initial condition specifications where the number of specifications is greater than or equal to r and any subset of these specifications with number of elements greater than r is inconsistent.

  (16a) has r degrees of freedom. Hence, for the above case, the number of initial condition specifications which could be enforced at the solution of the problem will be at most r. *ie.* the number of <5jS which can be zeros at the solution will be less than or equal to r. Let r_0 be the cardinality of the set of <5,s which are zero at the solution of the problem. Then

$$r_0 \leq r \tag{63}$$

If we assume 7 = 0 , from (62), *£}* = 0. The complementarity condition (56) gives A, < 1. Summing up all the A;s, we have

$$\sum_{1}^{ro} A_j < ro \leq r \Rightarrow \sum_{1}^{rp} 2 A_i < r \tag{64}$$

This is clearly a contradiction to (16d).

For (16d) to be satisfied, the A;s corresponding to all nonzero <5;s will have to go to their upper bounds and the cardinality of the corresponding set of *SiS,* r*o = r. ie. exactly r of the initial condition specifications are set equal to their specified values at the solution of the problem.

- Consider the case when the number of specifications is greater than r, there exists a subset of greater than r elements, or the entire set of specifications, which when specified simultaneously is consistent.

  In these cases, the number of specifications which could be enforced at the solutions could be greater than r because there exists a set of cardinality greater than r which is consistent. A could take fractional values in this case and the corresponding solution with (16) could fix greater than *r* variables at their specified values. The formulation, nevertheless, serves the purpose of choosing the best set of r or greater than r specifications. The form of the solution will be different from (13) in this case.

pivoting can give us the desired structure. Pivoting is straightforward in the case of linear systems. Below we illustrate how this approach solves the example of (27).

We could determine the implicit dependehdes from a form like (29) by path tracing on the corresponding Jacobian of the derivative array equations (32).

$$\begin{aligned} y' + x &= u_1 \\ y'' + x' &= u_1' \\ y &= u_2 \\ y' &= u_2' \\ y'' &= u_2'' \end{aligned} \tag{32}$$

Consider the Jacobian (33) of F(32).

$$\begin{array}{cccccccccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \\ \end{array}$$
$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c} & 1 & & & 1 & -1 & & & & \\ 1 & & 1 & & & 1 & -1 & & & \\ & & & 1 & & 1 & & -1 & & \\ & 1 & & & & 1 & & & -1 & \\ 1 & & & & & 1 & & & & -1 \end{array}\right] \tag{33}$$

Manipulations on the Jacobian (33) will give the following incidence matrix for the form (29).

$$\begin{array}{cccccccccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \\ \end{array}$$
$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c} & 1 & & & 1 & -1 & & & & \\ & & 1 & & & & -1 & & & 1 \\ & & & 1 & & & & -1 & & \\ & & & & -1 & 1 & & & -1 & \\ 1 & 1 & & & & & & & & -1 \end{array}\right] \tag{34}$$

We can trace a path from an incidence in the column y' to that in u^ showing that there is dependence on u^ which is implicit in (29). Further pivoting in the column under x will give the desired form (35) which corresponds to (28).

$$\begin{array}{cccccccccc} y'' & y' & x' & y & x & u_1 & u_1' & u_2 & u_2' & u_2'' \\ \end{array}$$
$$\left[\begin{array}{c|c|c|c|c|c|c|c|c|c} & 1 & & & & & & & -1 & \\ & & 1 & & & & -1 & & & 1 \\ & & & 1 & & & & -1 & & \\ & & & & -1 & 1 & & & -1 & \\ 1 & 1 & & & & & & & & -1 \end{array}\right] \tag{35}$$

t Sensitivity analysis of the derivative array equations

An alternate approach suggested to find the dependencies of the variable derivatives on the derivatives of the input function is to do a sensitivity analysis on the derivative array equations. Caracotsios and Stewart (1985) and Maly and Petzold (1995) describe sensitivity analysis for DAEs. However, since we are not interested in finding the sensitivity of the entire profile, but just on the dependencies at the point of discontinuity, we consider the following SLP formulation:

$$
\begin{aligned}
& Min \sum_{i=1}^{n} (p_j + n_j) \\
s.t. \quad & F/(x<^{"+1}>,...x,y^{(1/)},...,y,ul»K...u,t) = {}_{Pj} - nj \quad j = 1,...,(*+1)n \\
& {}_{u}(*)_{=ti}(*) - \qquad ik = 0,...,i/ \\
& Xt \leq Xi \leq X? \qquad (36)
\end{aligned}
$$

where $u^{w}\sim = u\wedge(t^{m} - 0)$. The numerical value of the variables at the point of discontinuity is the optimal solution to (36). Sensitivity analysis is carried out with u and its higher derivatives being treated as independent variables and by linearizing (36) at the variables values before the discontinuity. If a particular derivative $x'j$ is seen to be insensitive to the parameters $u',....,u^{(1"H}\backslash$ it is because it has no local dependence on any of the derivatives of the input variables ie. it is continuous across a discontinuity caused by a step change in u. However, one should be aware tiiat the test is local and the analysis provides a framework which is necessary but not sufficient for determining the continuities. The sensitivity analysis is performed using LP sensitivity, which is computationally cheap.

Nonlinear DAEs

The current approach we use for determining the dependencies for nonlinear DAEs is the sensitivity analysis described above. Since (36) is solved using SLP, it could be used for nonlinear systems as well. The sensitivity analysis would effectively be performed on the linearized form of the equations at $(t^{m} - 0)$. The method has been applied to the example problems solved in section 3.5.

Application of pivoting strategies on the Jacobian of the derivative array equations for the nonlinear case is limited as one can only determine, in graph theory notation, a *maximal* set of reachable nodes. If this maximal set does not contain any nodes corresponding to the implicit dependencies, the sets determined thus far are correct. Hence, analysis on the Jacobian may not give the desired result for nonlinear DAEs.

3.4 Applications of the LP Method for Reinitialization

A certain number of specifications for the initial conditions is required for restarting the problem after the discontinuity. Some of these are obtained from the criterion laid out for the validity of continuity assumptions of the variables across the discontinuity. However, the number of variables which are continuous across the discontinuity could be less than, equal to or greater than the degrees of freedom for the specification of initial conditions. Let us analyze the three cases and look at how the problem could be tackled in each of these cases.

/. Number of variables continuous across the discontinuity is greater than the number of degrees of freedom.

Here the number of specifications is greater than the number of degrees of freedom, the specifications being the numerical values of the variables which are found to be continuous across the discontinuity. This is an overspecified problem. It is quite likely that if we fix the values of these variables and solve the equations for consistent initialization, we might not be able to converge it within the required error criterion. One of the reasons for this is the roundoff errors arising from integration in the previous time domain. It is well known that if the initial conditions are not sufficiently accurate, further solution of the problem could be difficult and inaccurate. For this reason we would like to determine the initial conditions such that the equations are solved within a desired error criterion. The following formulation, which is similar to (12), is introduced for this purpose.

$$
\begin{aligned}
& Min \sum_{i \in v_0} \omega_i \delta_i \\
s.t. \quad & F^{\nu}(x^{(\nu+1)}, \ldots x, y^{(\nu)}, \ldots, y, u^{(\nu)}{}_{f} \ldots u.t) = 0 \\
& -\delta_i \leq v_i - vf^{itc} \leq \delta_i \qquad i \in v_0 \\
& 6i \geq 0 \qquad i \in VQ \\
& Xt \leq Xi \leq X? \qquad (37)
\end{aligned}
$$

VQ is the set of variables which are found to be continuous across the discontinuity and vf^{itc} are their numerical values at the point of discontinuity.

2. *The number of continuous variables is equal to the degrees of freedom.*

This is the easiest case where we have the exact number of specifications. (37) could be applied to determine the values of the remaining variables and the solution will correspond to a zero objective function.

3. The number of continuity conditions is less tJian the degrees of freedom.

This is an underspecified problem. However, in process engineering problems there may be discontinuity constraints which are known along with a step change in the input functions (For example, correlations for computing impulse forces in the case of collision of nonpenetrating rigid bodies). Let *ff(v)* = 0 be the discontinuity relations for the profiles. Assuming that the discontinuity constraints provide at least the minimum required information to cover the remaining degrees of freedom, the following formulation could be used for consistent reinitialization.

$$
\begin{aligned}
& \text{A/in} \quad 5>\text{M?} + \sum_{i=1}^{n_f} \omega_i^2 \delta_i^f \\
& \quad t\in v_0 \\
\text{s.t.} \quad & F\ {}^{n_f}x^{(d,u+1)}, \ldots x, y^{(\nu)}, \ldots, y, u^{(\nu)}, \ldots u, t) = 0 \\
& -\delta_i^c \le v_i - v_i^{disc} \le \delta_i^c \qquad i \in v_0 \\
& -\delta_i^f \le f_i^d \le \delta_i^f \qquad i = 1, \ldots n_f \\
& \delta \qquad \ge 0 \\
& Xt \le X_{\{} \le \text{Af}
\end{aligned}
\tag{38}
$$

The same formulation could be applied to cases 1 and 2 as well. One of the major difficulties associated with reinitialization problems is to determine whether a particular discontinuity presents an under, exact or over specified problem and then the associated clumsy process of the selection of that subset of equations which has to be solved. (38) presents a single consistent formulation to handle all of these cases.

One question still unanswered is the presence of the input function *u* and its higher derivatives in the derivative array equations used in (38). While state profiles cannot be calculated that capture the exact discontinuities, profiles can be determined to reinitialize the problem in the next time domain. Here we let

$$
u = \begin{cases} f_a(t) & \\ f_b(t) & t>\underline{t} \end{cases} \tag{39}
$$

Using the Heaviside function (40), u is written as (41)

$$
u(t) = f_a(t)(1 - Y(t^*)) + f_b(t)(Y(t^*)) \tag{41}
$$

ťth derivative of (41) could be written as

$$
u^{(i)}(t) = \sum_{j=0}^{i} \binom{i}{j} Y^{(j)}(t^*)(f_b - f_a)^{(i-j)} + f_a^{(i)} \tag{42}
$$

FVom the theory of distributions (Banros-Neto, 1973), the first derivative of the Heaviside function corresponds to the Dirac delta function and its higher derivatives in (42) are the higher derivatives of this function. These functions are defined only at f.

Our problem is to find initial conditions for the derivatives of *u* for the solution of the problem in the time domain *after* the discontinuity. This is different from tracing the solution from one side of the discontinuity to the other where the derivatives of *u* are unbounded at the point of discontinuity. Here the idea is to find the initial conditions for the derivatives of u which will enable us to correctly trace the solution trajectory after the discontinuity. This is obtained by finding the limit of (42) when t -+t^m from the rhs. The limit is given by

$$\lim_{\epsilon \to 0+} v^{(i)}(t^m + €) = \operatorname*{Um}_{\epsilon \to 0+} V(«\bullet)(/* - f*)^{(i)}\{t^m + *) + f_a^{(i)}(t^{\bullet} + \epsilon)$$

$$= An \qquad (43)$$

All the terms containing the higher derivatives of *Y* vanish as they are zero at any point $(t^m$ + c) however small *e* may be. Thus for the reinitialization problem, the higher derivatives of it present in the derivative array equations take the fonn of the higher derivatives of their functional fonns defined in the region after the discontinuity.

Let us consider a simple example to illustrate the point. Consider the index three DAE of the form

$$x\backslash = y\text{-}xi$$
$$x'2 = X\backslash - X2$$
$$x_2 = «(t) \qquad (44)$$

where

«◈ - {? JI?

The problem is of zero degrees of freedom and the constraining algebraic equations are

$$X'l = U(t)$$
$$xx = u'(t)+ti(t)$$
$$y = u''(t) + 2u'(t)+u(t) \qquad (45)$$

The profiles can be plotted as in Figure 1.

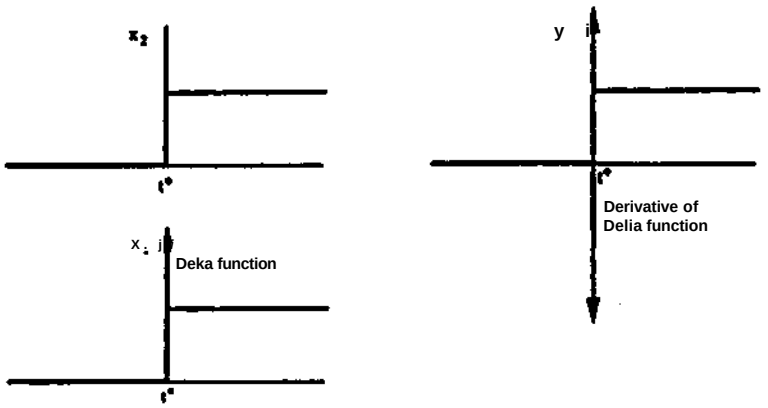

Figure 1: *Variable profiles*

As seen from Figure 1, $x\backslash$ and y are described by unbounded functions at t". Our problem is not to solve the problem at t^m_y but to solve for the initial conditions of xi, x-2 and y for the further solution of the problem after the discontinuity, u = 1 (and hence u^* = 0 and u" = 0) will give us values of xi = x_{-2} = y = 1 which are the correct initial conditions for the further solution of the problem in the region after the discontinuity.

Below we consider some example problems to illustrate the ideas described in this section.

3.5 Examples

We us consider four examples, two where the continuity assumption for the differential variables is valid and the other two where it is not. .The first three are nonlinear.

Example 5

This example looks at an implicit index-1 problem (46) considered by Brüll and Pallaske(1992). This is a case where the continuity assumptions are not necessarily valid after the discontinuity.

$$\begin{aligned} x_1' + y_1 x_2' &= 0 \\ x_2 &= U_X(t) \\ Vi &= u_2(t) \end{aligned} \tag{46}$$

After making reductions and simplifications, it is clear that the underlying ODE for (46) is

$$x_1' = -u_2(t)u_1'(t)$$

$$x_2' = u_1'(t) \qquad (47)$$
$$yi = txf_2(t)$$

As seen from (47), none of the variables are continuous after a step change in the input functions tti and u_2- However, if u\ is the only function causing the discontinuity, t/i is found to be continuous across the discontinuity. Similarly, xi and x-i are continuous across a discontinuity caused by a step change in the input function u≥. D

Example 6: Pendulum problem revisited

Consider the pendulum problem from (18) in a slightly different context. Let the pendulum be subject to time dependent input forces with the g in (18) replaced by an input function u. Below, we examine the continuity of the variables when there is a step change in the forcing function u. The underlying ODE is

$$\begin{aligned} z' &= t; \\ y' &= w \\ v' &= (yu\text{-}tr^{2}\text{-}tir^{3})x \qquad (48) \\ vf &= (yu - v^{1} - ti/^{2})y - it \\ T &= Zwu + yu^{1} \end{aligned}$$

FVom (48), it is clear that V depends on the first derivative of u whereas the derivatives of the other variables depend only on u.

$$«o = \{x,y,v,ti;\} \qquad (49)$$

The position and velocity variables are continuous but the string tension exhibits a step change in the event of a step change in the forcing function. Similar result is obtained from the sensitivity analysis. O

Example 7

Consider the DAE (50) for which the continuity assumption is valid. This problem was considered previously by Majer *et* a/.(1995)

$$\begin{aligned} *' &= yi - 1 \\ 0 &= x\text{-}yt + yi + u \qquad (50) \\ 0 &= 2y\{/^{3} + » \wedge - 4 \end{aligned}$$

Consider the solution of the reinitialization problem of (50). The problem has a steady state solution of yi = 1, $y2$ = 4 and x = 3 for u = 0. At time t = f, a step change is applied to u from 0 to 5.

Table 4: Reinitialization of Example 7

Initial Guesses	Solution	Number of iterations
Vi = 1 »2=4 i = 3	x = 3.0000 yi = 0.1975 yi = 8.0390 x' = -0.8025	10
Vi=0 yj=o i' = 0		12

From the sensitivity analysis, we find that x is continuous across the discontinuity caused by a step change in the input variable u. Setting v_0 = {z}, the initialization problem was solved using (38). Table 4 reports the solution of the problem for two sets of initial guesses.

The first set of initial guesses correspond to the values of the variables immediately before a step change in u was applied. This is probably the most logical initial guess to be diosen when carrying out dynamic simulation of the above system.

The second set is the initial guess used by Majer *tt* a/. (1995) for the solution of the problem. They point out that (50) cannot be solved by Newton's method from this set of initial guesses because of the limited region of convergence. They use a more expensive continuation method for the solution of the problem. However, it should be noted that the LP based formulation (38) encountered no difficulty in solving the problem.

Apart from being an efficient way for dealing with difficult nonlinear reinitialization problems, the method has been successfully applied to high index problems as well. Below we consider a high index example problem.

Example 8: The CSTR problem

Bachmann *tt al.* (1989) describe a problem of A^r Continuous Stirred Tank Re-

actors (CSTRs) in series without reaction which constitutes a system of index $N + 1$. The equations are

$$c_i' = c_{i-1} - c_i \quad i = 1, \ldots, N$$
$$CN = u(t) \tag{51}$$

Consider the case of 10 CSTRs. The problem is of index 11. Define u as

$$u(t) = \begin{cases} 10 + t & t \leq 10 \\ t & t \geq 10 \end{cases} \tag{52}$$

First the continuity assumptions were checked from the analysis of the Jacobian of the derivative array equations and from sensitivity analysis. It was found that none of the variables are continuous across a discontinuity caused by a step change in the u. The problem was then solved using the formulation (38). No variables were found to satisfy the continuity assumptions and there arc no explicitly specified discontinuity constraints. Note that this problem is exactly specified as the degrees of freedom and the number of continuous variables is zero.

The problem, which has 164 variables, 132 equalities and 22 inequalities was solved in 2 iterations. It was initialized at the solution of the problem before the discontinuity. The initial value for any c_i was found to be

$$Anitial \_\_ on \quad \rightarrow \quad \rightarrow \_\_ n \quad 10 \tag{53}$$

4 Conclusions

Overspecification of initial conditions, which occurs mainly due to the lack of knowledge of the degrees of freedom for the specification of initial conditions, could lead to convergence failures in the initialization problem. Most existing methods expect the user to supply the exact number of initial conditions. Instead a minimization formulation is presented here for initialization as well as for reintialization in the case of problems with discontinuity, which takes care of the overspecifications. This ensures that the initial conditions are determined within a required accuracy; failure to do so may result in convergence to poor or inaccurate solutions on further integration.

In section 3 we presented a criterion for checking the validity of the assumption of continuity of differential variables across a discontinuity caused by a step change in the input variables. Such criteria have not been reported in the literature for problems of general index. The popular assumption of continuity of all differential variables across a discontinuity (Pantelides and Barton, 1993) is shown to be not necessarily valid.

The applications of the ideas outlined in these sections have been demonstrated on small problems, of low and high index. The concepts apply irrespective of the size and index of the problem and application to larger problems is expected to be straightforward. Also, the proposed LP based approach is easily interfacable to DAE solvers like DASSL. This will make the treatment of the initialization and reinitialization problems easier and more user friendly within these solvers.

Acknowledgements
The authors are thankful to Prof. Steven L. Campbell for his valuable comments and for suggesting the example in Sections 3.2 and 3.3.

References

[I] R. BACHMANN, L. BRULL, T. MRZIGLOD AND U. PALLASKE, *On Mctliods for Reducing the Index of Differential - Algebraic Equations,* Computers chein. Engng., 14(1990), pp. 1271-1273.

[2] J. BARROS-NBTO, *An Introduction to the Theory of Distributions,* Marcel Dckker, 1973.

[3] S. C. BILLUPS, S. P. DIRSKE, AND M.C. FERRIS, *A Comparison of Algorithms for Large Scale Mixed Complementarity Problems,* Technical Report, Computer Sciences Department, University of Wisconsin, 1995.

[4] K. E. BRENAN, S. L. CAMPBELL, AND L.R. PETZOLD, *Numerical Solution of Initial'Value Problems in Differential'Algebraic Equations,* SIAM, Philadelphia, 1995.

[5] L. BRULL AND U. PALLASKE, *On Consistent Initialization of Differential-Algebraic Equations with Discontinuities,* in Proceedings of the Fourth European Conference on Mathematics in Industry, Tcubner, Stuttgart, 1992, pp. 213-217.

[6] S. L. CAMPBELL AND E. MOORE, *Progress on a General Numerical Method for Nonlinear Higher Index DAEs II,* Circuits Systems Signal Process, 13(1994), pp. 123-138.

[7] M. CARACOTSIOS AND W. E. STEWART, *Sensitivity Analysis of Initial Value Problems with Mixed ODEs and Algebraic Equations,* Computers diem. Engng., 9(1985), pp. 359-365.

[8] C. CHEN AND O.L.MANGASARIAN, *A Class of Smoothing Functions for Nonlinear and Mixed Complementarity Problems,* Technical Report, Computer Sciences Department, University of Wisconsin, 1995.

[9] Y. CHUNG AND A.W.WESTERBERG, *A Proposed Numerical Method for Solving Nonlinear Index Problems,* Ind. Eng. Chem. Res., 29(1990), pp. 1234-1239.

[10] Y. CHUNG, *Solving Index and Near Index Problems in Dynamic Simulation,* Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA 15213, 1991.

[II] I. S. DUFF, J. NOCEDAL, AND J.K. REID, *The Use of Linear Programming for the Solution of Sparse Sets of Nonlinear Equations,* SIAM J. Sci. Stat. Comput., 8(1987), pp. 99-108.

[12] C.W. GEAR, *Differential-Algebraic Equation Index Transformation,* SIAM J. Sci. Stat. Comput., 9(1988), pp. 39-47.

[13] V. GOPAL AND L. T. BIBGLER, *Nonsmooth Dynamic Simulation with Linear Programming Based Methods,* To appear, Computers chera. Bngng. (1996).

[14] B. LBIMKUHLER, L. R. PETZOLD AND C. W. GEAR, *Approximation Methods for the Consistent Initialization of Differential - Algebraic Equations,* SIAM J. of Numer. Anal., 28(1991), pp. 205-226.

[15] C. MAJER, W. MARQUARDT AND E.D. GILLES, *Reinitialization of DAEs After Discontinuities,* Computers chem. Engng. 19(1995), pp. S507-S512.

[16] T. MALY AND L. R. PETZOLD, *Numerical Methods and Software for Sensitivity Analysis of Differential* Algebraic Systems,* To appear, Applied Numerical Mathematics (1996).

[17] C. C. PANTELIDES AND P. I. BARTON, *Equation-Oriented Dynamic Simulation: Current Status and Future Perspectives,* Computers chem. Engng. 17(1993), pp. S263-S285.

[18] C. C. PANTELIDES, *The Consistent Initialization of Differential - Algebraic Systems,* SIAM J. Sci. Stat. Comput., 9(1988), pp. 213-231.

[19] J. UNGER, A. KRONER, AND W. MARQUARDT, *Structural Analysis of Differential - Algebraic Equation Systems - Theory and Applications,* Computers chem. Engng., 19(1995), pp. 867-882.

[20] J. ZHANG, N. KIM AND L. LASDON, *An Improved Successive Linear Programming Algorithm,* Mgrat. Sci., 31(1985), pp. 1312-1331.

Appendix - Properties of Smooth Formulation

Conditions for which *Si* = 0 =» A* = 1

The smooth formulation presented in Section 2 is:

$$Min \sum_{i=1}^{n_{ic}} \omega_i \delta_i$$

s.t. $F^*\{x,z,z\ldots_{\%}z^{i¥}\backslash t)=Q$ **(16a)**

$-\delta_i \leq X_i - X_i^{sp} \leq Si \quad \dot{t} = 1,\ldots,n_{fc}$ **(166)**

$\delta_i - \beta \log(1 + e^{-(A\cdot\,\wedge)\wedge}) = 0 \quad \dot{t} = 1,\ldots,n_{t\cdot c}$ **(16c)**

$\sum_{i=1}^{n_{ic}} \lambda_i - r = 0$ **(16d)**

$0 \leq Ai \leq 1 \quad \dot{t} = 1,\ldots,nfc$ **(16c)**

$Si \geq 0 \quad \dot{t} = 1,\ldots,n_{\dot{t}c}$ **(16/)**

$Xt \leq Xi \leq X?$ **(16$)**

The Karush-Kuhn-TVicker conditions of (16) arc

$$\frac{\partial L}{\partial \lambda_i} = \phi_i \frac{e^{-(\lambda_i - \delta_i)/\beta}}{1 + e^{-(\lambda_i - \delta_i)/\beta}} + \gamma - \xi_i^1 + \xi_i^2 \quad = \quad 0 \tag{54}$$

$$\xi_i^1 \lambda_i \quad = \quad 0 \tag{55}$$

$$\xi_i^2 (\lambda_i - 1) \quad = \quad 0 \tag{56}$$

where &, 7 and & are the multipliers of (16c), (16d) and (16e) respectively. FVom (16c),

$$1 + e^{-(\lambda_i - \delta_i)/\beta} \quad = \quad e^{\delta_i/\beta} \tag{57}$$

(54) could be simplified as

$$_{A\ e} - \wedge + 7 - 6^1 + €? \quad = \quad 0 \tag{58}$$

The smoothing constraint (16c) could be written as

$$_{e} - */* \ _{+\ c} - V * - \_i \tag{59}$$

When <J = 0, € - * " = 1. FVom (59),

$$e^{-\lambda_i/\beta} = 0 \tag{60}$$

For any /? > 0, A* -* 00. Choosing *0* small will allow A* < 00, in order to satisfy $_{e}$-*Xi/0* $\leq$ e_t a nonzero tolerance. FVom (60) and (55),

$$tf \quad = \quad 0 \tag{61}$$

FVom (58), (60) and (61), we have when *6*< = 0,

$$7 + \ tf \quad = \quad 0 \tag{62}$$

To show that A» goes to its upper bound (ie. £*j* £ 0) we need to show that 7 is nonzero. The following conditions are considered based on the application of (16).

- Consider initial condition specifications where the number of specifications is greater than or equal to r and any subset of these specifications with number of elements greater than r is inconsistent.

  (16a) has r degrees of freedom. Hence, for the above case, the number of initial condition specifications which could be enforced at the solution of the problem will be at most r. *ie.* the number of <5jS which can be zeros at the solution will be less than or equal to r. Let r_0 be the cardinality of the set of <5,s which are zero at the solution of the problem. Then

$$r_0 \leq r \tag{63}$$

If we assume 7 = 0 , from (62), *£}* = 0. The complementarity condition (56) gives A, < 1. Summing up all the A;s, we have

$$\sum_{1}^{ro} A_j < ro \leq r \Rightarrow \sum_{1}^{rp} 2 A_i < r \tag{64}$$

This is clearly a contradiction to (16d).

For (16d) to be satisfied, the A;s corresponding to all nonzero <5;s will have to go to their upper bounds and the cardinality of the corresponding set of *SiS,* r*o = r. ie. exactly r of the initial condition specifications are set equal to their specified values at the solution of the problem.

- Consider the case when the number of specifications is greater than r, there exists a subset of greater than r elements, or the entire set of specifications, which when specified simultaneously is consistent.

  In these cases, the number of specifications which could be enforced at the solutions could be greater than r because there exists a set of cardinality greater than r which is consistent. A could take fractional values in this case and the corresponding solution with (16) could fix greater than *r* variables at their specified values. The formulation, nevertheless, serves the purpose of choosing the best set of r or greater than r specifications. The form of the solution will be different from (13) in this case.