Constraint Partitioning for Structure in Path-Constrained Dynamic Optimization Problems [*]

Soumyendu Raha [†] Linda R. Petzold [‡]

May 16, 2000

Abstract

In this paper an algorithm for identifying an index 1 or 2 differential-algebraic subsystem from a possibly higher index path-constrained dynamical system is proposed. The algorithm is useful for diagnostic purposes in model development, and in the formulation of dynamic optimization problems to be solved by shooting or multiple shooting type methods.

1 Introduction

In this paper we study the problem of automatically detecting a high-index differential-algebraic equation (DAE) and extracting the largest lower index (index 1 or index 2) subsystem which can be solved by existing DAE software [5], [19].

There are several important applications of such a partitioning algorithm. The first is in the development of models for simulation. If a proposed model is found to be high-index and the variables and constraints associated with the high-index structure identified, then the model developer can focus on revising the model and/or applying index-reduction techniques [14] to yield a system that can be reliably simulated.

The second application arises when the shooting or multiple shooting method is used in dynamic optimization of the DAE system. Then there is a choice of how to handle the constraints and the associated algebraic variables: should they be included with the DAE and thus enforced by the DAE solver at all times and on every optimization iteration, or are they better handled directly by the optimizer (hence for an infeasible optimization method, the constraints may not be satisfied until convergence). Partitioning of the DAE can affect both the *structure* and the *stability* of the resulting subsystem. In this paper we present an algorithm to partition the DAE to yield the largest possible subsystem with lower index (index 1 or mixed index 1/ Hessenberg index 2). In a companion paper [17] we study the problem of partitioning for stability.

Several algorithms for partitioning have been proposed in the literature [7], [13]. In [7] a graph based algorithm is given for determination of index 1 structure, and in [20] and in [8] a method is described for determination of DAE structure based on non-singularity of the index 1 Jacobian. The algorithm we describe here differs from the previous work in that it is designed to partition for index 1 and index 2, and in that it examines the index of the *locally linearized system* by a direct numerical

[*]This research was supported by NSF Multidisciplinary Challenge grant CCR-98-9896198, by NSF KDI grant ATM-9873133, by the NSF/DARPA Virtual Integrated Processing program, and by DOE DE-FG03-98ER25354.

[†]Scientific Computation Program, Department of Computer Science, University of Minnesota, Minneapolis, MN 55455

[‡]*All communications should be addressed to this author.* Department of Mechanical and Environmental Engineering and Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA 93106

approach rather than the global system by a structural approach. The latter is both an advantage and a disadvantage of the proposed algorithm. On one hand as pointed out in [18], structural algorithms can confuse certain types of index 1 systems with higher index systems. On the other hand, the local index may not be equal to the global index for DAE systems of index 2 or greater [5]. Hence it should be emphasized that the algorithm described here is designed to determine whether the *local index* is 0,1,2 or greater, and to partition the system so that the partitioned DAE is index 0, 1 or Hessenberg index 2.

2 Partitioning for Structure

Many physical systems are naturally described by DAEs. A general DAE system can be written as

$$F(\dot{y}, y, t) = 0. \tag{1}$$

The class of problems (1) includes standard form ODEs as well as other systems which may have a *high index* [5], [11] structure. The current generation of DAE software [5], [11], [2] is designed to handle DAEs of index 0, index 1 and index 2 systems with Hessenberg structure, and combinations of these systems. Hence it is important in formulating a problem for simulation to identify the index and the source of any high index behavior (i.e., the variables and constraints contributing to the higher index structure) for possible reformulation of the problem.

A general dynamic optimization problem can be written as

$$\begin{aligned}
\text{Minimize} \qquad J = \int_0^{t_{\text{final}}} \chi(y, u, t)\, dt & \qquad \text{(2a)} \\
\text{subject to} & \\
F(\dot{y}, y, u, t) = 0 & \qquad \text{(2b)} \\
g(y, u, t) = 0 & \qquad \text{(2c)} \\
h(y, u, t) \leq 0 & \qquad \text{(2d)}
\end{aligned}$$

where (2c) are additional equality constraints and (2d) are additional inequality constraints. In shooting or multiple shooting type methods for solving the dynamic optimization problem, any constraints included in the DAE (2b) will be enforced at all times on all optimization iterations by the DAE solver, whereas the additional constraints will be enforced by the optimizer.

Given a DAE (1) for simulation, or (2b, 2c) in the dynamic optimization problem where constraints (2d) are candidates for inclusion in the partitioned DAE along with corresponding components of u, our goal is to determine whether the index of the DAE is 0, 1, 2 or higher and, to the extent possible, to *partition* the DAE as follows

$$\begin{aligned}
F_d(\dot{x}, x, v, w, u(t), t) = 0 & \qquad \text{(3a)} \\
\text{constraints to be included in the partitioned DAE:} & \\
\Psi_1(x, v, t) = 0 \quad \text{(index 1 constraints)} & \qquad \text{(3b)} \\
\Psi_2(x, t) = 0 \quad \text{(index 2 constraints)} & \qquad \text{(3c)} \\
\text{additional constraints:} & \\
p(y, u(t), t) = 0 & \qquad \text{(3d)}
\end{aligned}$$

where $y = (\, x, v, w\,)^T$, and $\frac{\partial F_d}{\partial \dot{x}}$, $\frac{\partial \Psi_1}{\partial v}$, and $\frac{\partial \Psi_2}{\partial x}\frac{\partial F_d}{\partial w}$ are non-singular. In the partitioned problem, y has been partitioned into three types of variables: x are the differential variables, v the index 1 algebraic variables, and w the index 2 algebraic variables. The variables $u(t)$ generally correspond to higher-index variables of the original problem and enter into the partitioned problem as time-dependent input functions.

We remark here that for DAEs where the differential and algebraic variables are strongly coupled it may not be possible to partition the system in terms of the original variables. For these problems we will make a determination of the index and then stop. We proceed by first identifying the differential part (3a) of the DAE (1). Then we consider each of the algebraic constraints of the DAE (1) and decide whether appending the constraints yields an index 1 or index 2 partitioned DAE (pDAE). Thus we need to be able to determine whether any given constraint leads to index 1 or index 2 structure. In the remainder of this section we derive conditions for identifying index 1 and index 2 constraints.

A dynamic optimization problem for the pDAE is then given by

$$\text{Minimize} \qquad J = \int_{t_0}^{t_{\text{final}}} \chi(x, v, w, u, t)\, dt \tag{4}$$

subject to the partitioned DAE (3a, 3b, 3c) and to the additional equality constraints (3d) and additional inequality constraints (2d).

2.1 Index 1 Structure

A set of constraints of the form (3b) when appended to an implicit ODE of the form (3a) yields an index 1 system if $\Psi_{1,v}$ is invertible. If the DAE system has been linearized at $t = t_k$, then it can be rewritten as

$$F_d := \dot{x} - f(x, v, w, u_2(t), t) := \dot{x} - (Hx + Qv + Tw + r(t)) = 0 \tag{5a}$$

$$\Psi_1 := C_v v + C_1 x + r_1(t) = 0 \tag{5b}$$

$$\Psi_2 := C_2 x + r_2(t) = 0 \tag{5c}$$

where $H \in \mathcal{R}^{n_x \times n_x}$, $Q \in \mathcal{R}^{n_x \times n_v}$, $T \in \mathcal{R}^{n_x \times n_w}$, $C_v \in \mathcal{R}^{n_v \times n_v}$, $C_1 \in \mathcal{R}^{n_v \times n_x}$, and $C_2 \in \mathcal{R}^{n_w \times n_x}$. The vectors $x \in \mathcal{R}^{n_x}$, $v \in \mathcal{R}^{n_v}$, $w \in \mathcal{R}^{n_w}$, $r(t) : \mathcal{R} \to \mathcal{R}^{n_x}$, $r_1(t) : \mathcal{R} \to \mathcal{R}^{n_v}$, and $r_2(t) : \mathcal{R} \to \mathcal{R}^{n_w}$.

Lemma 1 (Index 1 Structure.) *For the pDAE subsystem in the form (5a - 5b) to be index 1, C_v must be invertible. In general, the matrix $\begin{pmatrix} \frac{\partial F_d}{\partial \dot{x}} & Q \\ 0 & C_v \end{pmatrix}$ must be of full rank.*

Proof. If C_v is invertible, Ψ_1 can be differentiated once with respect to t and $\dot{v}$ and v can be determined explicitly and substituted in the ODE part to obtain the underlying ODE (UODE) for the index 1 subsystem. The UODE corresponding to the index 1 subsystem is

$$\dot{x} = (H - QC_v^{-1}C_1)x + Tw(t) + r(t) \tag{6}$$

and requires C_v to be invertible. For systems where there is no clear distinction between a differential variable and an index 1 algebraic variable, the pDAE may appear in the form

$$F_d := M_x \dot{x} + M_v \dot{v} - f(x, v, w(t), u_2(t), t) = 0 \tag{7a}$$

$$C_1 x + C_v v + r_1(t) = 0. \tag{7b}$$

Then the matrix $\begin{pmatrix} \frac{\partial F_d}{\partial \dot{x}} & Q \\ 0 & C_v \end{pmatrix}$ is full rank if both $\frac{\partial F_d}{\partial \dot{x}}$ and C_v are full rank invertible systems. □

Corollary. When $n_v = 1$, i.e., a single constraint equation is being considered, C_v is a scalar and it is sufficient that $|C_v| > 0$ for the constraint to be index 1.

Lemma 2 *When appending a single constraint of the form $c_v v_1 + \tilde{C}_1 x + c_r v + r_{11}(t) = 0$ to an ODE (3a) and a group of index 1 constraints of the form $C_v v + C_1 x + c_c v_1 + r_1(t) = 0$, the constraint under consideration leads to an index 1 pDAE if $|c_v - c_r C_v^{-1} c_c| > 0$.*

Proof. The updated C_v-matrix $\begin{pmatrix} C_v & c_c \\ c_r & c_v \end{pmatrix}$ must be invertible for the constraints (including the new constraint) to be index 1 (Lemma 1). The updated matrix is invertible when $|c_v - c_r C_v^{-1} c_c| > 0$. The condition is obtained if the updated matrix is inverted as a partitioned matrix. □

The rank r of the Jacobian matrix $\begin{pmatrix} \frac{\partial F_d}{\partial x} & Q \\ 0 & C_v \end{pmatrix}$ at the given time indicates the number of equations/variables (both differential and algebraic) that can be included in an index 1 pDAE [20]. Any of the constraints, along with the ODEs, which can make a sub-matrix of full column rank r along with corresponding algebraic variables v qualifies for this structurally partitioned index 1 pDAE.

Once the differential equations and variables have been identified, our algorithm examines one constraint at a time, checking the condition in Lemma (2) to detect the index 1 structure. Associated with deciding on an index 1 constraint equation is the problem of choosing the corresponding index 1 algebraic variable if more than one variable qualifies for this decision. This issue is resolved based on the likely effect on the stability of the pDAE. The choice of variable is discussed later.

2.2 Index 2 Structure

The requirement for index 2 structure can be deduced by constructing an UODE. For a pDAE of the form

$$F_d := \dot{x} - f(x, v, w, u_2(t), t) := \dot{x} - (Hx + Qv + Tw + r(t)) = 0$$
$$\Psi_1 := C_v v + C_1 x + r_1(t) = 0$$
$$\Psi_2 := C_2 x + r_2(t) = 0,$$

if the appended constraint (5c) is index 2, differentiating it once with respect to t and appending the differentiated constraint to (5a - 5b) will yield an index 1 system if the condition in Lemma (1) is satisfied. The differentiated form of (5c)

$$\dot{C}_2 x + C_2 \dot{x} + \dot{r}_2 \equiv \dot{C}_2 x + C_2(Hx + Qv + Tw + r(t)) + \dot{r}_2 = 0 \tag{8}$$

is index 1 if $(C_2 T)$ is invertible.

Lemma 3 *A constraint of the form (5c) is index 2 if (C_2T) is invertible.*

Proof. The equation (8) is obtained after one differentiation of the constraint (5c) with respect to t. Following Lemma (2), the equation (8) is index 1 if (C_2T) is invertible. Thus (5c) is index 2. □

Lemma 4 *The pDAE has an index 2 structure if each new algebraic constraint of the form $\psi \equiv \tilde{C}_2 x + \tilde{r}_2 = 0$ (along with a corresponding algebraic variable w_2), being appended to an original system of equations of the form*

$$\begin{aligned} F_d &\equiv \dot{x} - (Hx + \tilde{T} w_2 + Tw + r(t)) = 0 \\ \Psi_2 &\equiv C_2 x + r_2 = 0 \end{aligned}$$

is individually checked for index 2, i. e., whether for the algebraic variable w_2

$$|\tilde{C}_2 \tilde{T} - \tilde{C}_2 T (C_2 T)^{-1} C_2 \tilde{T}| > 0.$$

Proof. For a single constraint $\tilde{C}_2$ is a row vector and $\tilde{T}$ is a column vector. The matrix $\begin{pmatrix} C_2 \\ \tilde{C}_2 \end{pmatrix} \begin{pmatrix} T & \tilde{T} \end{pmatrix}$ must be invertible for the pDAE to have an index-2 structure. The condition claimed here ensures the invertibility of this matrix (Lemma (2)). □

In our algorithm, the condition in Lemma (4) is checked to detect index 2 structure for each constraint under consideration.

3 Incorporation of Stability Estimate in Partitioning

Partitioning can affect not only the *structure* but also the *stability* of the pDAE. Addition or deletion of constraints and corresponding algebraic variables can either stabilize or destabilize the pDAE.

Consider the following simple example

$$\begin{aligned} \dot{x} &= 1000.0x + u \\ bx + u &= 0. \end{aligned}$$

The system is unstable when the constraint is not included in the pDAE and the ODE is integrated (u being handled by the optimizer in a dynamic optimization problem). But the system is stable for $b = 10000$ when integrated as an index 1 DAE.

We measure the change in local stability due to addition of one constraint and its corresponding algebraic variables via the *logarithmic norm* [10]. Further details and results on the use of the logarithmic norm in partitioning are given in [17]. Since stability as well as structural considerations are an integral part of our partitioning strategy, we briefly describe the stability estimates here.

The logarithmic norm has been used extensively in nonlinear stability analysis for numerical ODEs [10] and is defined as follows.

Definition 1 (Logarithmic Norm.) *The logarithmic norm for a square matrix H in real or complex space is defined as*

$$\mu(H) := \lim_{\epsilon \to 0, \epsilon > 0} \frac{||I + \epsilon H|| - 1}{\epsilon}. \tag{9}$$

Part of the usefulness of the logarithmic norm as a stability measure arises because for any square matrix $H \in \mathcal{R}^{n \times n}$

$$\max_{\lambda} \quad \Re(\lambda(H)) \leq \mu(H). \tag{10}$$

Hence an ODE $\dot{x} = Hx$ is locally stable if $\mu(H) \leq 0$. The logarithmic norm has many other properties that are useful for analysis [10], [17] and as we shall see later it is closely related to the pseudoeigenvalues [12] of the matrix H, which have also been widely used to measure stability for both differential systems and numerical solution methods.

Since we will be examining stability for one constraint and its corresponding algebraic variable at a time, the logarithmic norm of a rank 1 matrix $u_1 u_2^T$ (where u_1 and u_2 are unit vectors of dimension n) is of particular interest and is given by

$$\mu(u_1 u_2^T) = \max(0.5(u_1^T u_2 \pm 1), 0) \quad \text{for} \quad n > 2 \tag{11a}$$

$$\mu(u_1 u_2^T) = \max(0.5(u_1^T u_2 \pm 1)) \quad \text{for} \quad n = 2. \tag{11b}$$

3.1 Index 1 Stability

The UODE (6) for an index 1 subsystem is locally stable if the largest real part of the eigenvalues of $H - QC_v^{-1}C_1$ is less than or equal to zero. For practical computations we require it to be less than a positive real number (TOL) of suitable size. In terms of the logarithmic norm, the index 1 pDAE subsystem is stable if

$$\mu(H - QC_v^{-1}C_1) \leq 0. \tag{12}$$

An unstable ODE ($\mu(H) > 0$) or a pDAE already in place can be stabilized by appending index 1 constraints if for the resulting system

$$\mu(H - QC_v^{-1}C_1) \leq \mu(H) + \mu(-QC_v^{-1}C_1) \leq 0. \tag{13}$$

For a single algebraic constraint, Q is a column vector (say, $\tilde{Q}$), C_1 is a row vector (say, $\tilde{C}_1$) and C_v is a scalar represented by c_v. Let $u_Q = \frac{-\tilde{Q}}{||\tilde{Q}||_2}$ and $u_{C_1}^T = \frac{\text{sign}(c_v)\tilde{C}_1}{||\tilde{C}_1||_2}$ and $\gamma = \frac{||\tilde{Q}||_2.||\tilde{C}_1||_2}{|c_v|} \geq 0$. Then $\mu(H - \tilde{Q}c_v^{-1}\tilde{C}_1) = \mu(H + \gamma u_Q u_{C_1}^T)$. Computationally, the condition for stability is

$$\mu(H + \gamma u_Q u_{C_1}^T) \leq \mu(H) + \mu(\gamma u_Q u_{C_1}^T) = \mu(H) + \tilde{\gamma} \leq TOL, \tag{14}$$

where $\tilde{\gamma}$ is defined by

$$\mu(\gamma u_Q u_{C_1}^T) = \tilde{\gamma} = \gamma \max\left(0, \frac{(u_{C_1}^T u_Q \pm 1)}{2}\right) \tag{15}$$

for $n_x > 2$ (where n_x is the dimension of x). For $n_x = 2$, the maximum is taken over $\frac{(u_{C_1}^T u_Q \pm 1)}{2}$ only.

Next we compute (15) for each index 1 constraint being appended to a group of already existing index 1 constraints. Thus, let

$$\Psi_1 \equiv C_1 x + C_v v + c_c v_1 + r_1(t) = 0$$

be the existing group of constraints to which a constraint of the form

$$\tilde{C}_1 x + c_v v_1 + c_r v + r_{11}(t) = 0$$

is considered for being appended. This is equivalent to appending the constraint

$$\bar{\Psi}_1 \equiv \bar{C}_1 x + \bar{C}_v v_1 + \bar{r}_1(t) = 0$$

to the UODE

$$\dot{x} = (H - QC_v^{-1}C_1)x + Tw(t) + r(t)$$

where

$$\bar{C}_1 = \tilde{C}_1 - c_r C_v^{-1} C_1 \tag{16a}$$

$$\bar{C}_v = c_v - c_r C_v^{-1} c_c \tag{16b}$$

and $\bar{r}_1(t)$ represents the terms that are function of time only. While evaluating (15) u_{C_1} is computed as $u_{\bar{C}_1} = \frac{\text{sign}(\bar{C}_v)C_1}{||\bar{C}_1||_2}$. Then $\tilde{\gamma}$ for second or later index 1 constraints is computed as

$$\tilde{\gamma}_{\text{append}} = \frac{||\tilde{Q}||_2 ||\bar{C}_1||_2}{|\bar{C}_v|} \max\left(0, \frac{(u_{\bar{C}_1}^T u_Q \pm 1)}{2}\right) \tag{17}$$

(for $n_x > 2$), where $\tilde{Q}$ is the column vector in the differential equations corresponding to the algebraic variable chosen for the index 1 equation under consideration.

3.2 Index 2 Stability

We have obtained the stability estimate for index 2 pDAEs (5) as $\mu(\bar{H}) + 2\mu(-T(C_2T)^{-1}\dot{C}_2) \leq TOL$, where $\bar{H} = H - QC_v^{-1}C_1$ is the stability matrix of the index 0 or index 1 subsystem. Since for all practical purposes H is the dominant matrix term in $\bar{H}$, a parameter k is estimated as $p_{H,\max}^T R p_{H,\max}$ where $p_{H,\max}$ is the normalized eigenvector corresponding to the largest eigenvalue of H and $R = (I_{n_x} - T(C_2T)^{-1}C_2)$. Then we obtain the stability condition

$$\mu(\bar{H}) + 2\mu(-T(C_2T)^{-1}\dot{C}_2) \leq \frac{TOL}{|p_{H,\max}^T R p_{H,\max}|} = \frac{TOL}{|k|}. \tag{18}$$

For $H = 0$, $|k|$ is taken as 1. Any index 2 algebraic constraint which is a candidate for being appended to the already constituted pDAE system is checked for $T = C_2^T$ (it has been shown that index 2 constraints do not destabilize the pDAE if $T = C_2^T$ [17]), whether C_2 is a constant (when C_2 is a constant, $-T(C_2T)^{-1}\dot{C}_2$ disappear) and for $(C_2H + C_2)R = 0$, failing which the check (18) is done.

In the case of a single index 2 constraint being checked for stability, an analytical expression for $\mu(-T(C_2T)^{-1}\dot{C}_2)$ is obtained. In this case C_2 is a row vector (denoted by $\tilde{C}_2$), T is a column vector (denoted by $\tilde{T}$) and $R = I - \frac{u_T u_{C_2}^T}{u_{C_2}^T u_T}$ where $u_{C_2} = \tilde{C}_2^T/||\tilde{C}_2||_2$ and $u_T = \tilde{T}/||\tilde{T}||_2$. Then $||R|| = \frac{1}{u_{C_2}^T u_T}$. The change in the stability measure by appending a single index 2 constraint, which can be estimated by

$$\mu(\bar{H}) + 0.5||\dot{u}_{C_2}||_2 + \max\left(\frac{-\text{sign}(\tilde{C}_2\tilde{T})\dot{\tilde{C}}_2\tilde{T} \pm ||\dot{\tilde{C}}_2||_2||\tilde{T}||_2}{2|\tilde{C}_2\tilde{T}|}, 0\right) \leq \frac{TOL}{|k|} \tag{19}$$

(for $n_x > 2$) is sufficient for determining the stability when including a single index 2 algebraic constraint in the pDAE.

3.3 A Practical Criterion for Constraint Inclusion

For some types of systems, in particular non-linear highly oscillatory systems such as the stiff spring pendulum and wheel set on rails examples described in [17] the logarithmic norm of the implicit ODE subsystem may already be large. Thus the criteria (14) and (19) would partition all constraints out of the DAE, even if they do not further increase the logarithmic norm. Hence in practice we relax the criteria to require only that the constraints do not increase the logarithmic norm by much. Thus, instead of checking simply if $\mu \leq TOL$, the stability check for accepting a constraint is modified as

$$\mu \leq \max\left(\frac{TOL}{|t_{\text{final}} - t_0|}, 3TOL_0\right) \tag{20}$$

where TOL_0 is the logarithmic norm of the unconstrained dynamic system.

3.4 Physically Important Constraints

Some constraints may represent physically important relationships, and hence should be included in the pDAE regardless of their effect on the index or stability (although the index should be checked and if necessary reduced for numerical solution). For example, scleronomic constraints describe physical or geometrical features of a mechanical system. In [17] we demonstrate that for the crane model of section 5.7, failure to include these constraints in the pDAE can lead to unphysical results in the simulation and non-convergence in dynamic optimization. In contrast, path constraints can be added or deleted from the pDAE. Thus we allow in our software for the user to specify which constraints and variables *must* be included in the pDAE.

4 The Partitioning Algorithm

Based on considering one algebraic constraint at a time, a partitioning algorithm that returns a locally (at the point of linearization) index 2 or lower pDAE is designed. Constraints are examined for their impact on structure and stability. In this section we discuss the essential components.

4.1 Design of the Algorithm

So far we have discussed only the tests for index 1 or index 2 system To build the complete pDAE system, strategies for choice of algebraic variable and a global stability check are needed. The algorithm is designed to proceed by examining one constraint at a time for its structure, stability and physical significance (if known).

4.1.1 Step One: Linearization

The partitioning algorithm extracts a pDAE subsystem from the original DAEs linearized at a particular value of t. Hence a pre-processing step for linearizing the DAEs is needed before the partitioning algorithm can start working. The pre-processing step involves application of automatic differentiation software (such as ADIFOR [4]) to the procedure evaluating the residuals ($F = 0$) for the DAEs to generate the procedures for evaluating $JPRIME0 := \frac{\partial F}{\partial \dot{x}}$, $J0 := \frac{\partial F}{\partial x}$ and $JDOT0 := d\left(\frac{\partial F}{\partial x}\right)/dt$, all evaluated at $t = t_k$.

4.1.2 Step Two: Identification of Differential Equations and Variables

The next step in the algorithm is to obtain the mass matrix $JPRIME1$ from $JPRIME0$. This is the sub-matrix of rows and columns of $JPRIME0$ that are not null vectors. The corresponding columns and rows in $JPRIME0$ which contain at least one non-zero entry are marked as candidate differential variables and differential equations (respectively).

If $JPRIME1$ is square and non-singular, the differential equations and differential variables have been identified and we can directly move on to consider the algebraic equations and variables for appending index 1 and/or index 2 constraints to the pDAE sub-system. If $JPRIME1$ is full row rank with more columns than rows, then a search for the deficient number (i.e., the number by which columns exceed rows in $JPRIME1$) of index 1 algebraic equations involving only the candidate differential variables is done before considering other algebraic equations and variables in the original system. This limited search for index 1 algebraic equations is similar to the general search (described in the next step) for index 1 and 2 algebraic equations but involves only the candidate differential variables and stability is checked only at the end of examining the candidate index 1 algebraic equations. The classification of the variables corresponding to the index 1 algebraic equations appended from the limited search is revised as index 1 algebraic variables.

It may be mentioned here that for a small number of equations one can use the LU-decomposition with partial pivoting for identifying differential equations. The rows pivoted to the bottom and corresponding to the zero-rows of the upper triangular matrix are constraint equations. But this involves $O(n_x^3)$ computational cost as opposed to a search for non-zero entries which loops over all the equations and variables requiring $O(n_x^2)$ operations.

Depending upon the dimensions of the mass matrix, three possible cases may arise. These are:
(a) Square: $JPRIME1$ is a square matrix.
(b) Fat rectangle: $JPRIME1$ has number of columns less than number of rows.
(c) Thin rectangle: $JPRIME1$ has number of columns greater than number of rows.
Depending upon the rank of the mass matrix, two possible cases may arise in each of the above three cases.
(a) Full rank: The mass matrix has a rank $= \min(n_{\text{rows}}, n_{\text{columns}})$.
(b) Rank deficient: The mass matrix has a rank less than $\min(n_{\text{rows}}, n_{\text{columns}})$.

Only in the cases where $JPRIME1$ is a square or a fat rectangle with full rank can the algorithm perform partitioning directly on the original equations and variables. For all rank deficient cases and the case where $JPRIME1$ is a thin rectangular matrix, the original equations and variables undergo transformation through a singular value decomposition of $JPRIME0$ before being partitioned: $SVD(JPRIME0) = USV^T$. In the original system of equations, substitutions $x = Vy$, $y = V^T x$

and $\dot{y} = V^T \dot{x}$ (assuming that V^T is constant) are used to transform the original input into a new system of the form $S\dot{y} + U^T J0 V y + ... = 0$.

The partitioning code is implemented in such a way that it can detect the rank deficiency or the thin rectangular structure of $JPRIME0$ and is stopped and restarted with application of the partitioning algorithm to the transformed DAE system in y. The matrix S is a diagonal matrix and the application of partitioning algorithm produces a full rank square $JPRIME1$ for the transformed system. The accepted pDAE obtained from the transformed system, if determined to be solvable, and the information about its stability may be used, in combination with some physical and/or mathematical intuition about the problem, to guide the selection of differential and algebraic variables in the original system.

Illustration A. Consider the input with

$$JPRIME0 = \begin{pmatrix} 2 & -2 & 0 \\ -2 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$JPRIME1 = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$$

$$J0 = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0.1 & 0 \\ -1 & 0 & 0 \end{pmatrix}.$$

The rank deficient square matrix condition is met and this system undergoes an SVD. The new system in

$$y = \begin{pmatrix} 0.7071 & -0.7071 & 0 \\ -0.7071 & -0.7071 & 0 \\ 0 & 0 & 1 \end{pmatrix} x$$

is partitioned. In Step 3, the last two variables are returned as index 1 algebraic variables and the last two equations as index 1 algebraic equations.

Illustration B. Consider the system with

$$JPRIME0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 7 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$J0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 3 \\ 1 & 0 & 0 & 4 & 4 \\ 0 & 9 & 5 & 0 & 5 \\ 0 & 5 & 3 & 3 & 3 \end{pmatrix},$$

from which the mass matrix is obtained as

$$JPRIME1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 5 & 7 & 3 \end{pmatrix}.$$

The fat rectangle with full row rank condition is satisfied. The first two equations are identified as differential equations. The first four variables are possible differential or algebraic variables to be identified in Step 3. The second and the third variables are reclassified as algebraic index 1 variables and two more algebraic equations (fourth and fifth) are appended as index 1 algebraic equations following limited search similar to Step 3.

4.1.3 Step Three: Identification of Index 1 Equations and Variables

In this step, we search for index one equations and variables, one constraint at a time. When the constraint under consideration has the option of choosing from more than one algebraic variable to which it can be linked, the search for the dominant algebraic variable corresponding to the constraint must be done in an economical way. If more than one yet unidentified algebraic variable satisfies the index-one condition in Lemma (2), then the problem arises of deciding on a variable that corresponds to the algebraic equation under consideration. The direct way of associating the possible variables is to mark all the possible variables and then go on to the next equation and so on. After the search for index 1 or index 2 equations ends, all possible combinations of the variables are explored and the one that gives the pDAE with the lowest logarithmic norm is associated with the index 1 subsystem. But for large systems this method will be computationally very expensive. Hence a trade-off policy is adopted. The following explains the policy for both index 1 and index 2 systems.

Policy 1. For small systems, all possible algebraic variables corresponding to the current algebraic equation under consideration for entry into the index 1 subsystem are tried out. Thus, for all algebraic variables that satisfy the condition in Lemma (2) the logarithmic norm for the index 1 subsystem (up to and including the current algebraic equation) is evaluated. The one that gives the lowest logarithmic norm is associated with the index 1 equation under consideration. This is feasible only for very small systems since a complete search involves exponential computational complexity.

Policy 2. For a very large system with many coupled terms across the algebraic variables, the above policy may prove to be expensive. In those cases, the policy is to look for a variable moderately connected with the differential variables in the current equation. This can be done by directly starting the search for a non-zero scalar $\frac{||\tilde{C}_1||_2}{c_v} \leq 0.5\left(\min_v\left(\frac{||\tilde{C}_1||_2}{c_v}\right) + \max_v\left(\frac{||\tilde{C}_1||_2}{c_v}\right)\right)$ (for index 2, substitute $\tilde{C_2T}$ in place of c_v) where the minimum and the maximum are taken over the possible algebraic variables in the equation being considered. If this choice yields an unacceptable logarithmic norm, the next higher c_v ($\tilde{C_2T}$) is considered and so on. Since the logarithmic norm is inversely proportional to c_v ($\tilde{C_2T}$ for index 2), the larger ratios need not be explored. The scalars c_v or $\tilde{C_2T}$ may be suitably scaled over the range of their values. This policy was implemented in the test program that produced the results in the numerical examples.

Policy 3.Pivoting. If $J_a := \frac{\partial F_a}{\partial x_a}$ denotes the matrix formed from the rows corresponding to the algebraic equations and columns corresponding to the algebraic variables in the Jacobian $\frac{\partial F}{\partial x}$, then for detecting index 1 equations and corresponding index 1 variables the entry with the maximum absolute value in J_a is tested for invertibility as in Lemma (2). Obviously for the first index 1 algebraic equation and its corresponding index 1 variable the entry chosen must be non-zero. For the subsequent equations and variables the entry with the maximum absolute value is tried first. If the test for index 1 fails or the equation-variable combination fail to pass the logarithmic norm-based stability criterion, the entry with the second largest absolute value is tried. After an equation and its corresponding algebraic variable have been selected the corresponding row and column are removed from J_a. The search terminates when no suitable pivot is found. The search for index 1 system terminates when no suitable pivots are found. For index 2 systems the remaining equations are used to form a matrix $J_{a2} := \frac{\partial F_{a2}}{\partial x_d}\frac{\partial F_d}{\partial x_{a2}}$, where the subscript $a2$ denotes the remaining algebraic equations/variables after the selection of index 1 subsystems. A pivoting scheme similar to the search of index 1 subsystem is applied to J_{a2} to find the index 2 subsystem.

It may be noted that the last two policies are heuristic. Since a complete search for a subsystem takes exponential time, a suitable heuristic policy is needed to make the partitioning algorithm practically applicable by making it polynomial time. The last two policies return an index 2 or lower pDAE subsystem if there is one, but does not guarantee the selection of largest possible index 2 or lower pDAE subsystem.

The heuristics work best when C_v ($\tilde{C_2T}$ for index 2) has, for some permutation, a dominant lower triangular structure. Hence, it is desirable to evaluate the logarithmic norm of the fully assembled pDAE to verify that the system is stable. For very large systems, this check should be run every time a fixed number of constraints has been appended.

Illustration C. Say we are following policy 3 to choose variables and equations and have a $JPRIME1 =$ identity matrix of dimension 3 and

$$J0 = \begin{pmatrix} 1 & 0 & 0 & 2 & 3 \\ 0 & 1 & 0 & 4 & 5 \\ 0 & 0 & 1 & 6 & 7 \\ 1 & 0 & 0 & 3 & 5 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

and the first three variables and equations are differential variables and equations. Then the fifth variable and the fourth equation are selected as a possible index 1 equation and variable and the matrix C_v (a scalar at this stage) is 5. During the stability check, the logarithmic norm is evaluated with $q = (3, 5, 7)^T$ and $c1 = (1, 0, 0)^T$. If this partitioning is deemed unstable, the procedure selects $C_v = 3$ as the next best pivot and remembers that the fourth equation-fifth variable pair is structurally index 1 but was rejected as unstable. Then the fourth equation and fourth variable combination is similarly tried out. If stable, the combination is accepted into the pDAE by marking the third variable and equation as algebraic. In case this combination is deemed unstable too, the procedure stops searching for a pair for the third equation and marks the third variable as a rejected index 1 variable and the fourth equation as a rejected index 1 equation. Next, we examine the final equation in the DAE in a similar way. For this C_v grows to a two-by-two matrix: $\begin{pmatrix} 3 & 5 \\ 1 & 0 \end{pmatrix}$.

4.1.4 Step Four: Identification of Index 2 Equations and Variables

If constraints and variables remain following Steps 2 and 3, we search for index 2 equations and variables. This procedure is exactly similar to the index 1 case except that instead of the matrix C_v, the matrix C2*T is formed from J0 and during stability check we need to form C2DOT $= \dot{C}_2$ from JDOT for computing the logarithmic norm in (19). It may be noted here that for systems with index 2 constraints coupled with index 1 variables the present algorithm revises the matrix C_2 as

$$C_2 - C_3 C_v^{-1} C_1,$$

where C_3 is the non-invertible coupling matrix in the linearized constraint of the form $C_2 x + C_3 v + r_2(t) = 0$. Consequently C2DOT is revised as

$$\dot{C}_2 - \dot{C}_3 C_v^{-1} C_1 + C_3 C_v^{-1} \dot{C}_v C_v^{-1} C_1 - C_3 C_v^{-1} \dot{C}_1$$

for evaluation of logarithmic norms. The matrices $\dot{C}_3$, $\dot{C}_1$ and $\dot{C}_v$ are formed from JDOT by selecting the appropriate rows and columns and hence does not need an extra pass through ADIFOR.

4.1.5 Rejecting Constraints During Check Back for Stability

The algorithm has the limitation that during stability analysis the effect of the terms (corresponding to the new algebraic variable being included in the pDAE) coupling existing constraints and the constraint under consideration are ignored. Hence after accepting a few constraints a check back for the overall stability of the pDAE is needed. While performing the local stability check for each algebraic equation, the new algebraic variable corresponding to this equation must be picked in such a way that the local stability check is a fair estimate of the stability of the pDAE and that the global check pointing will not change the built-up pDAE unless absolutely necessary.

If it is found during checking back that the global logarithmic norm of the pDAE is greater than TOL, then some of the constraints must be rejected to keep the system stable. In keeping with

Policy 2, for index 1 constraints we leave out the last appended constraint and the algebraic variable having the maximum $\frac{||\tilde{C}_1||_2}{|c_v|}$ (scaled over the range of values c_v takes in the problem) and check for overall stability in terms of the recomputed overall logarithmic norm. If the system is still unstable, the procedure is repeated on the next constraint on the block of constraints being checked. This is done until a stable index 1 pDAE is found. For an index 2 system, the same thing is done on the constraint having the largest $\frac{||\dot{\tilde{C}}_2||_2}{|\tilde{C}_2\tilde{T}|}$ in the block of constraints being checked and variables with maximum $\frac{||\dot{\tilde{C}}_2||_2}{|\tilde{C}_2\tilde{T}|}$ are rejected.

4.1.6 Update of Inverted Matrices

For detecting index 1 or index 2 structure using Lemma (2) or (4) it is needed to update the inverted c_v or C_2T matrix. This is done using the formula for inverting a partitioned matrix. The inversion of the updated matrix $\tilde{A} = \begin{pmatrix} A & \tilde{A}(1:n,n+1) \\ \tilde{A}(n+1,1:n) & \tilde{A}_{n+1,n+1} \end{pmatrix}$ (where A and $\tilde{A}$ are $n \times n$ and $n+1 \times n+1$ square matrices respectively) can be expressed in terms of A and the new row and column. (The notation $A_{i,j}$ denotes the entry at the ith row and the jth column of matrix A and $A(i:k,j:l)$ denotes the block formed by rows i to k and columns j to l of matrix A.) Using the standard formula for inverting a partitioned matrix, we obtain

$$\tilde{A}^{-1} := \begin{pmatrix} A^{-1} - A^{-1}\tilde{A}(1:n,n+1)\tilde{A}^{-1}(n+1,1:n) & -A^{-1}\tilde{A}(1:n,n+1)\tilde{A}^{-1}_{n+1,n+1} \\ -\tilde{A}^{-1}_{n+1,n+1}\tilde{A}(n+1,1:n)A^{-1} & (\tilde{A}_{n+1,n+1} - \tilde{A}(n+1,1:n)A^{-1}\tilde{A}(1:n,n+1))^{-1} \end{pmatrix}.$$

4.2 Correctness of the Algorithm

The algorithm $PARTITION$ identifies index 1 and index 2 structures and gives an estimate of the stability of the resultant pDAE.

Theorem 1 (Structure with Single Constraint) *The algorithm $PARTITION$ selects a single isolated constraint so that the resulting pDAE has index less than or equal to 2 and is stable locally at $t = t_k$.*

Proof. Since the algorithm uses the results in Lemma (2) for index 1 and structure and Lemma (4) for index 2 structure, the constraints entering the pDAE are index 1 and index 2 respectively. The check in equation (19) has been shown to be a sufficient condition for the stability of a single isolated constraint being appended to the unconstrained ODE part of the pDAE. □

Theorem 2 (Overall Structure of the pDAE) *The algorithm $PARTITION$ appends a cluster of constraints based on index 1 or index 2 structure such that the overall pDAE has index less than or equal to 2 and uses a stability estimate to output a pDAE which is stable at $t = t_k$ for clusters of constraints appearing in disjoint blocks.*

Proof. Since the results in Lemma (2) for index 1 structure and Lemma (4) for index 2 structure update and check the global invertibility of the C_v and C_2T matrices respectively for all the constraints collected including the last appended constraint, the overall index 1 or index 2 structure is maintained in the pDAE being constructed.

The check back using equation (18) gives a somewhat conservative estimate of the stability of the cluster of constraints appended. If the check back is done on all the constraints appended so far then the stability estimate is a global sufficient condition, though more conservative than the single isolated constraint case. If the check back is limited to a small block of constraints, then the estimate (18) is sufficient when the cluster blocks are disjoint, i. e., when algebraic variable vectors do not have coupling terms between them (C_v, Q, and T all have block diagonal structures). In case C_v, Q and T are dense matrices with significant off-block-diagonal terms, the check back for stability over only the cluster block ensures stability locally limited only to the particular block. □.

4.3 Properties and Limitations of the *PARTITION* Algorithm

1. For a DAE system with a very large number of constraints and tightly coupled algebraic variables between them, policy 2 may end up excluding a few eligible constraints.
2. The results for index and stability are local, i.e., based on the frozen coefficient local system.
3. The algorithm may not be able to distinguish some algebraic variables and constraints from differential equations and variables in the case in which SVD is performed in Step 2.
4. Physically meaningful constraints index 3 or higher are not directly included with the pDAE. Such systems should be reduced to lower index (i.e., index 2 or lower) before input into the partitioning algorithm.
5. Systems with a local singularity at the initial time may yield a misleading result unless they are also analyzed at a later time.
6. If the local matrix pencil is singular, the algorithm will simply stop, i.e., it does not attempt to analyze singular systems.

5 Examples of Partitioning

In this section we illustrate the working of the algorithm with several examples. All the examples were run on a 180 MHZ IP32 Processor SGI O2 computer (FPU: MIPS R5000 and CPU: MIPS R5000).

The first two examples were run just to test whether the algorithm can successfully identify index 1 or index 2 constraints. To keep the tests simple inclusion or rejection of the constraint based on stability estimates was not considered.

5.1 A Simple Example

Consider the equation

$$F_1 := \dot{x} + \dot{y} - f_1(t) = 0.$$

A path described by

$$F_2 := \sin(t)x + ty - f_2(t) = 0$$

is appended to the system and F_1 and F_2 are input to the *PARTITION* algorithm with $t = 0$. This system has no clearly defined differential or algebraic variable. Further, the system Jacobian does not have full rank at $t = 0$. The algorithm returns an undecided or incomplete pDAE at $t = 0$. A perturbation of $t = 0.01$ is chosen and *PARTITION* is called again to run at $t = 0.01$. This time F_1 is listed as a differential equation with y as differential variable and F_2 is returned as an index 1 algebraic constraint equation with x as the corresponding variable.

5.2 A Non-Solvable Example

The following input

$$\begin{aligned} \dot{x}_1 + \dot{x} - f_1(t) &= 0 \\ x_1 + x_2 - f_2(t) &= 0 \end{aligned}$$

causes the partition algorithm to stop without giving any output since the Jacobian of the system is singular. Thus systems in non-solvable form are checked for and the output is safe-guarded from having a singular Jacobian.

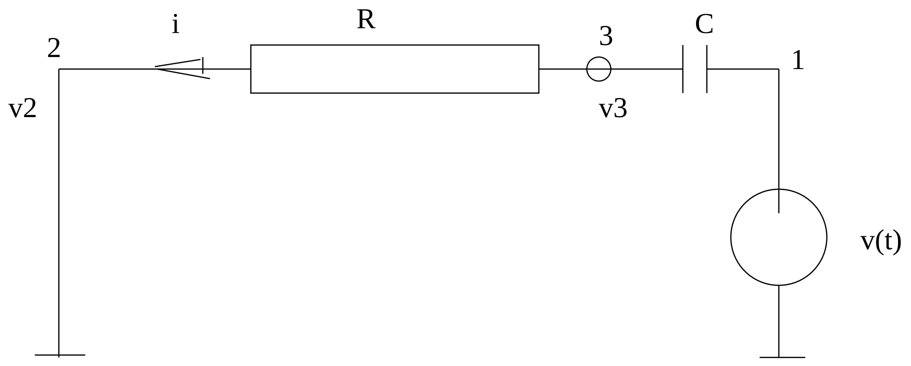

Figure 1: Linear electrical circuit

5.3 A General Example

The DAE system

$$\begin{aligned} F_1 := \sin(x_2)\dot{x_1} + \cos(\dot{x_3})\dot{x_2} + \dot{x_3} + x_4 x_1 + 23 x_5 x_2 = 0 \\ F_2 := t^2 x_2 + t x_1 = 0 \\ F_3 := x_1^2 + \sin(t) x_3 x_2 - \cos(6t) = 0 \\ F_4 := x_3 + x_5 = 0 \\ F_5 := x_2 x_4 + \sin(0.5t) = 0 \end{aligned}$$

is used as an input to the *PARTITION* algorithm. At $t = 0.0$ the pDAE is output as differential equation list = $[F_1]$; differential variable list = $[x_3]$; index 1 equation list = $[F_3, F_4, F_5]$; index 1 variable list = $[x_1, x_4, x_5]$. The output at $t = 0.001$ is obtained as differential equation list = $[F_1]$; differential variable list = $[x_3]$; index 1 equation list = $[F_2, F_3, F_4, F_5]$; index 1 variable list = $[x_1, x_2, x_4, x_5]$. Since the last output is the union of the two outputs, it is taken as the final partition. The index 2 equation and variable lists are returned empty.

The structure of this example is such that there are multiple ways in which the index 1 algebraic variables can be chosen. The algorithm *PARTITION* suggests only one of the many possible ways. This example has been used to test whether the present algorithm can correctly identify an index 1 system even when the index 1 algebraic variables do not occur explicitly in the input equations.

5.4 Modified Nodal Analysis of a Linear Electrical Circuit

Existing structure-based partitioning algorithms fail to take into consideration the singularity of C and confuse this index 1 system with an arbitrarily higher index system [18]. In contrast, the partitioning algorithm described in this paper transforms the equations and the variables through a singular value decomposition and identifies the index 1 structure. The following example taken from [18] illustrates a typical situation.

Consider the circuit equations obtained from a Modified Nodal Analysis (MNA) of a linear electric circuit (Figure (1))

$$\begin{pmatrix} C & -C & 0 \\ -C & C & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{v}_1 \\ \dot{v}_3 \\ \dot{i} \end{pmatrix} + \begin{pmatrix} 0 & 0 & -1 \\ 0 & \frac{1}{R} & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_3 \\ i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ v(t) \end{pmatrix},$$

with $C = 2$ pF and $R = 10$ Ohm. The DAEs can be transformed into

$$\begin{pmatrix} 4 \times 10^{-12} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\hat{v}}_1 \\ \dot{\hat{v}}_3 \\ \dot{\hat{i}} \end{pmatrix} + \begin{pmatrix} 0.05 & 0.05 & -0.7071 \\ -0.05 & -0.05 & -0.7071 \\ -0.7071 & 0.7071 & 0 \end{pmatrix} \begin{pmatrix} \hat{v}_1 \\ \hat{v}_3 \\ \hat{i} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \bar{v}(t) \end{pmatrix}$$

and is identified as an index 1 system by the *PARTITION* algorithm. The logarithmic norm of the index 1 DAE system is computed as -5×10^{10} which is small enough to make it acceptable.

5.5 Anisotropic Biphasic Theory Equations

The system consists of partial differential-algebraic equations (PDAE) described in detail in [3]. The spatial discretization is obtained using the Finite Element Method and the spatially discretized system is used as input to the *PARTITION* algorithm for obtaining the desired information about the DAE system. The model equations in one dimension are

$$\text{Momentum Balance:} \quad -R\frac{\partial u_{zz}}{\partial t} + \frac{\partial(\sigma_{zz}\theta)}{\partial z} = 0 \tag{21a}$$

$$\text{Mass Balance:} \quad \frac{\partial \theta}{\partial t} + \theta\frac{\partial^2 u_{zz}}{\partial t \partial z} = 0 \tag{21b}$$

$$\text{Constitutive Equation:} \quad \frac{\partial \sigma_{zz}}{\partial t} + \frac{\sigma_{zz}}{L} - \beta\frac{\partial^2 u_{zz}}{\partial t \partial z} = 0 \tag{21c}$$

where R is the resistivity constant, L the relaxation time constant and β is the viscosity. The test system uses a one-dimensional confined compression ramp problem spatially discretized into 2-node finite elements. The dependent variables are longitudinal strain (u_{zz}), stress (σ_{zz}) and volume fraction(θ). The physical space (z) is discretized into 20 elements. At every node the three equations in (21) are considered in the following order : momentum balance, mass balance and the constitutive equation. The longitudinal strain rate $\frac{\partial u_{zz}}{\partial t}$ is considered lumped at each node. All other variables are spatially discretized using linear shape functions. The variables occur in the order of strain in gel, then volume fraction of collagen, and stress in gel. The boundary condition is imposed at the first node by replacing the momentum balance equation with $u_{zz} - d = 0$, d being related to the displacement of the plunger applying pressure on the sample. The spatial discretization leads to a DAE system of 63 equations. The spatially discretized DAE system is in the following form. The boundary condition at the first node is obtained as

$$u_{zz1} - d = 0.$$

For node 2 to the last-but-one node:

$$M_0\dot{u}_{zz} + f_0(\sigma_{zz}, \theta) = 0.$$

For node 1 to the last-but-one node:

$$M_1\dot{\theta} + M_2\dot{u}_{zz} = 0$$
$$M_3\dot{\sigma}_{zz} + M_4\sigma_{zz} + f_1(\dot{u}_{zz}) = 0.$$

For the last node:

$$F_b(u_{zz\text{LastNode}}, \sigma_{zz\text{LastNode}}) = 0.$$

In the above equations M_i ($i = 1, 2, 3, 4$) are the coefficient matrices obtained from the Finite Element discretization and F_b is the boundary condition at the end node not in contact with the

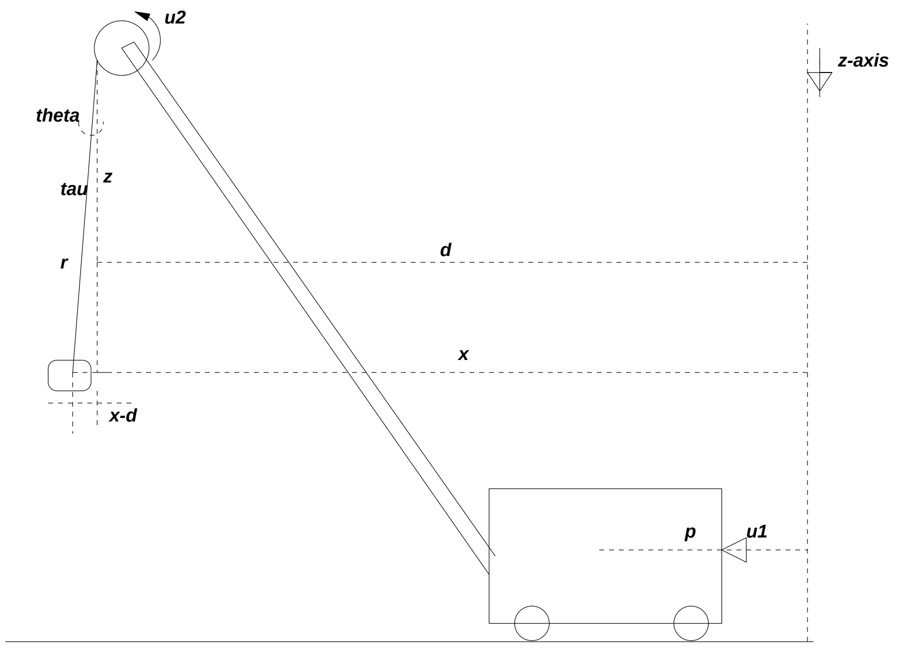

Figure 2: Planar Crane Model

plunger. The matrix M_0 is a diagonal matrix in which alternate entries are zero. Thus, at every other node the momentum balance equation is an algebraic equation.

The initial conditions for $t = 0$ are suitably input to the algorithm. The algorithm $PARTITION$ classifies equations 1, 7, 13, 19, 25, 31, 37, 43, 49, 55 as index 1 and the rest as differential equations. The corresponding index 1 variables are 1, 6, 12, 18, 24, 30, 36, 42, 48 and 54. Physically this means that the algorithm selects the stress in gel at every other node (i.e., second, fourth and so on) as an index 1 algebraic variable and the stress balance equation in the next node (i.e., third, fifth and so on) as the corresponding index 1 equation. The strain at first node and the related boundary condition at that node are selected as index 1 algebraic variable and index 1 algebraic equation respectively. The logarithmic norm of the linearized underlying ODE at $t = 0$ was estimated as -45.547 in 0.1699 CPU seconds.

5.6 Planar Model of a Crane

This example is a planar rigid body model of a crane manipulating a payload, which must be lifted along a specified trajectory [6]. The crane is a mechanical system in planar Cartesian co-ordinates, x and z, denoting the horizontal and vertical positions of the payload. The other state variables are d, the horizontal distance traveled by the crane trolley from the origin and r, the paid out cable length. The equations are

$$\begin{aligned}
M_2\ddot{x} &= -\tau\sin(\theta) && \text{(22a)}\\
M_2\ddot{z} &= -\tau\cos(\theta) + mg && \text{(22b)}\\
M_1\ddot{d} &= -C_1\dot{d} + u_1 + \tau\sin(\theta) && \text{(22c)}\\
J\ddot{r} &= -C_2\dot{r} - C_3u_2 + C_3^2\tau && \text{(22d)}\\
0 &= \theta - \tan^{-1}((x-d)/z) && \text{(22e)}\\
0 &= r^2 - (x-d)^2 - z^2 && \text{(22f)}\\
x &= \phi_1(t) && \text{(22g)}
\end{aligned}$$

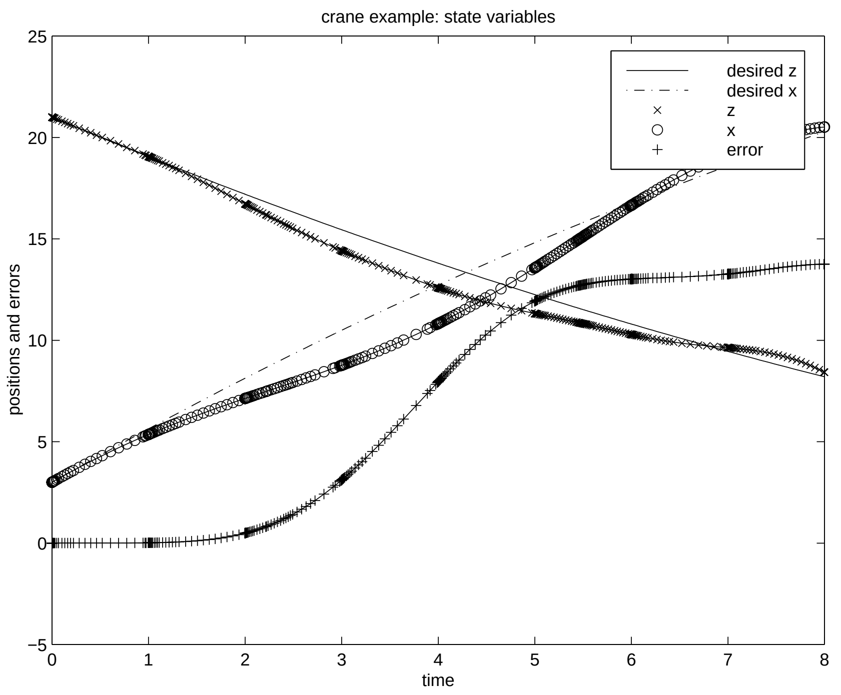

Figure 3: Crane problem using equations (22a)-(22e) and (23) in the pDAE.

$$z = \phi_2(t) \tag{22h}$$

where M_1, M_2, m are the masses of the trolley, cable and payload system and the payload alone, respectively. C_1, C_2, C_3 are constants, and J is the mass moment of inertia for the pulley paying out the cable. The algebraic variables are τ, the tension in the cable and θ, the cable angle with vertical. The horizontal force driving the trolley (u_1) and torque driving the pulley in the winch (u_2) are the two control variables. Equations (22g) and (22h) describe the specified path of the payload.

For the crane model, constraints (22e) and (22f) are both scleronomic constraints, i.e., they describe some geometric structure of the physical problem. Constraint (22e) introduces the measurement of θ, the angle by which the payload deviates from the vertical. Constraint (22f) connects the payload to the pulley, by relating its horizontal and vertical positions to the length of the cable paid out.

Applying the algorithm to this model, constraint (22e) is included in the pDAE since it satisfies the index one structure and stability criterion. The resulting UODE in x does not have large eigenvalues in $\mathcal{C}^+$ at $t = t_0$. For a practical physical application, the eigenvalues of the pDAE system can be expected to be of moderate size and the system can be expected to be stable. The constraint equation (22f) is an index 3 scleronomic constraint (known from the physics of the problem) and is reduced to its index 2 form via one differentiation of constraint (22f) with respect to t

$$r\dot{r} - (x-d)(\dot{x}-\dot{d}) - z\dot{z} = 0 \tag{23}$$

or to index 1 via two differentiations of constraint (22f) with respect to t. (if the available DAE integration software can integrate index 1 or 0 systems only). The other constraints (pre-programmed trajectory of the payload) (22g and 22h) are handled from the SQP method.

In the crane model the partition algorithm detects and accepts the index-1 constraint (22e) at $t = 0$. The value of TOL was set at 20.0. The interval of integration is from $t_0 = 0$ to $t_{\text{final}} = 8$. The value of TOL_0 in (20) was set at 0. The initial values are $x = 3.0, \dot{x} = 2.7, z = 21.0, \dot{z} = -2.0, d =$

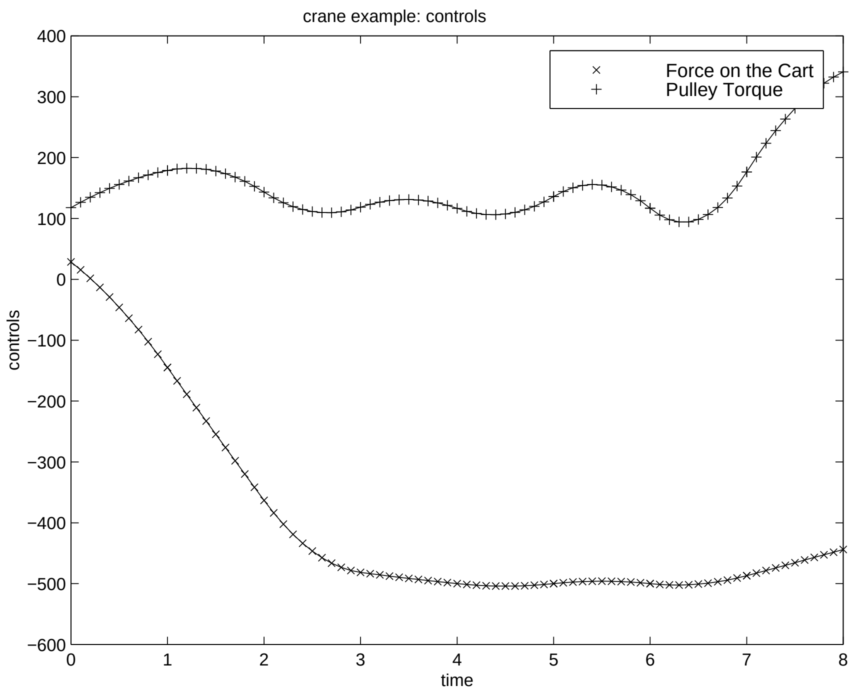

Figure 4: Crane controls using equations (22a)-(22e) and (23) in the pDAE.

$1.0, \dot{d} = 2.7, r = ((x-d)^2 + z^2)^{0.5}, \dot{r} = -2.0, \tau = 980.0, u_1 = 0, u_2 = 0, \theta = \tan^{-1}\left(\frac{(x-d)}{z}\right)$. The logarithmic norm of the stability matrix H for the crane model at $t = 0$ is 0.5 and with the addition of the index-1 constraint (22e) it goes up to 0.82794. This is less than $\max(TOL/8, TOL_0)$ (see equation (20)) and the constraint (22e) is accepted. The angle of deflection from vertical θ is chosen as the index 1 variable.

The search for index 2 constraints returns the empty set. In order to search for index 3 scleronomic constraints, the remaining algebraic equations are differentiated using automatic differentiation once with respect to time. The index-2 search algorithm is re-applied and the algorithm returns the differentiated form of constraint (22f), i. e., (23) as an index-2 constraint suitable for inclusion in the pDAE system. Here $|k| = 0.9999$. The logarithmic norm of the index 2 pDAE stability matrix (using equation(19)) is 1.61502 and the tension in the cable τ is returned as the corresponding index 2 variable. The matrix S from the stability analysis changes very slowly in the neighborhood of $t = 0$ since $T = (0, 0, 0, 0, -\sin(\theta), -\cos(\theta), \sin(\theta), C_3^2)^T$ remains almost unchanged due to very small changes in $\theta \approx 0$ (i. e. an almost vertical cable undergoing only small swings). Hence the logarithmic norm changes only slightly even after the constraint (23) has been appended to the pDAE in its differentiated form. During check back, the overall logarithmic norm of the accepted pDAE system is computed as 1.09717, which is smaller than $\max(TOL/8, TOL_0)$ and is acceptable. Building the whole system takes 9.82×10^{-2} CPU seconds on a 180 MHZ IP32 Processor SGI O2 computer.

Even though the constraint (23) slightly raises the logarithmic norm estimate for the index 2 pDAE system thus constructed, this partition leads to a more physical and well conditioned problem. The plots in Figures (3) and (4) are the results obtained from dynamic optimization using a multiple shooting type scheme with DASPK3.0 [16] as the DAE integrator and SNOPT as the NLP method [15]. In this case the path constraints are chosen as $p_1 \equiv x - (-0.0675t^2 + 2.7t + 3.0) = 0$ and $p_2 \equiv z - (-21.0 - 0.05t^2 + 2.0t) = 0$. The optimizer SNOPT stops at an acceptable point which cannot be improved further after 4 major iterations (with 188, 182, 112 and 1 minor iterations). The objective function is given by the error integral $\int_{t_0}^{t_{\text{final}}}(p_1^2 + p_2^2)^{0.5}dt$. The time interval is $0 \leq t \leq 8$. The controls (u_1 and u_2) have been modeled as quadratic polynomials over each of the 8 shooting intervals used for the problem. The tolerances for DASPK3.0 were $rtol = 10^{-7}$ and $atol = 10^{-7}$ and all tolerances for SNOPT were 10^{-5}. The control and state continuity constraints across the shooting intervals have been satisfied to less than or equal to 0.1 when the optimizer has stopped. The time taken for solving this problem is 1421.91 seconds.

References

[1] Ascher, U. M., and Petzold, L. R., Stability of computational methods for constrained dynamics systems, *SIAM J. Sci. Comput.*, 14(1), pp. 95-120, January 1993.

[2] Ascher, U.M. and Petzold, L.R., *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, SIAM Publications, Philadelphia, PA. 1998.

[3] Barocas, V., *Anisotropic Biphasic Modeling of Cell-Collagen Mechanical Interactions in Tissue Equivalents*, PhD Thesis, Department of Chemical Engineering and Material Science, University of Minnesota, 1996.

[4] Bischof, C., Carle, A., Peyvand, K., Mauer, A. and Hovland, P., *ADIFOR 2.0 User's Guide,* ANL/MCS-TM-192. Argonne National Laboratory, Argonne, IL 60439, 1995.

[5] Brenan, K.E., Campbell, S.L., and Petzold, L.R., *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, SIAM, second edition, 1996.

[6] Cambell, S. L., High Index Differential Algebraic Equations. *Mech. Struct. and Mach.*, 23(2), pp. 199-222, 1995.

[7] Duff, I. S. and Gear, C. W., Computing structural index, *SIAM J. Algeb. Discr. Meth.*, 7, pp. 315-330, 1981.

[8] Feehery, W. F., Banga, J. R., Barton, P. I., A novel approach to dynamic optimization of ODE and DAE systems as high index problems, AIChE Annual Meeting, Miami Beach, FL, 1995.

[9] Gill, P. E., Murray, W., and Saunders, M. A., *User's Guide to SNOPT5.0: A FORTRAN Package for Large-Scale Nonlinear Programming*, April, 1997.

[10] Hairer, E., Nørsett, S. P., Wanner, G., *Solving Ordinary Differential Equations: Vol 1*, Springer-Verlag, 1986.

[11] Hairer, E. and Wanner, G., *Solving Ordinary Differential Equations: Differential-Algebraic and Stiff Systems: Vol. 2.* , Springer-Verlag, 1991.

[12] Higham, D. J. and Trefethen, L. N., Stiffness of ODEs, *BIT* 33, pp. 285-303, 1993.

[13] Jarvis, R. B. and Pantelides, C. C., A differentiation-free algorithm for solving high-index systems. *Paper 146g, Session on "Applied Mathematics and Computer Methods-II", AIChE 1992 Annual Meeting, Miami Beach*, 1-6 November, 1992.

[14] Mattson, S. E. and Söderlind, G., Index reduction in differential-algebraic equations using dummy derivatives, *SIAM J. Sci. Comput.*, 14(3), pp. 677-692, May 1993.

[15] Petzold, L., Rosen, B., Gill. P., Jay, L. O., Leonard, M. and Sharma, V., An SQP method for the optimal control of large-scale dynamical systems, to appear, *J. Comp. Appl. Math.*

[16] Petzold, L. and Li, S., DASPK3.0 User's Guide, Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA, website: http://www.engineering.ucsb.edu/∼cse

[17] Raha, S. and Petzold, L. R., Constraint partitioning for stability in path-constrained dynamic optimization problems, submitted.

[18] Reiszig, G., Martinson, W. S. and Barton, P. I., Differential-algebraic equations of index 1 may have an arbitrarily high structural index, to appear, *SIAM J. Sci. Comput.*

[19] Reymond, J.D., *Implementation des methodes Radau IIA d'ordre 7 et 9,* Diploma thesis, Univ. Geneva, 1989.

[20] Vassiliadis, V. S., Sargent, R. W. H., and Pantelides, C., Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints, *IEC Research*, 33, pp. 2123-2133, 1994.