SIAM J. SCI. COMPUT.
Vol. 30, No. 2, pp. 916–936

CONSISTENCY ON DOMAIN BOUNDARIES FOR LINEAR PDAE SYSTEMS*

JENS NEUMANN† AND CONSTANTINOS C. PANTELIDES‡

Abstract. This paper considers the determination of consistent variations of the dependent variables over the boundary of the domain of definition of a linear system of partial differential and algebraic equations (PDAEs). Particular emphasis is placed on the specifications ("boundary conditions") imposed on different parts of the boundary and their consistency with the underlying PDAE system and with each other. Specifications imposed on overlapping parts of the boundary (e.g., faces with common edges, or edges with common vertices) often lead to inconsistencies (corner singularities) that are not trivial to detect, especially in PDAEs involving three or more dimensions. A symbolic/numerical algorithm is proposed for the analysis of PDAE systems defined over finite hyperrectangular domains of arbitrary dimensions.

Key words. partial differential and algebraic equations, boundary conditions, corner singularities, consistency

AMS subject classifications. 35G15, 34A25, 32W50, 15A30, 05C20

DOI. 10.1137/060650088

1. Introduction. This paper is concerned with mixed systems of linear partial differential and algebraic equations (PDAEs) of arbitrary order involving arbitrary numbers of dependent variables u and independent variables x. Thus, each equation is of the general form

$$\sum_k A_k \frac{\partial^{\sum_j \alpha_{kj}}}{\partial x_1^{\alpha_{k1}} \partial x_2^{\alpha_{k2}} ... \partial x_{N_d}^{\alpha_{k,N_d}}} u(x) = f(x), \tag{1}$$

involving an arbitrary number of terms k, each of nonnegative order α_{kj} with respect to independent variable x_j, $j = 1, \ldots, N_d$, where N_d is the number of independent variables. A_k are constant matrices, and $f(x)$ is a nonlinear forcing function.

The independent variables vary over a finite or semi-infinite domain $\{x \mid x_j \in [x_j^L, x_j^U],\ j = 1, \ldots, N_d\}$, where x_j^L and x_j^U denote lower and upper bounds on independent variable x_j. A point x^* is a corner of the domain if, for every independent variable j, x_j^* is equal to either x_j^L or x_j^U. The present paper is restricted to PDAE systems where all boundary conditions are imposed at a given corner x^* and/or on higher-dimensional parts of the domain (e.g., edges and faces of a 3-dimensional domain) which contain that particular corner. Hence, we do not consider problems where some boundary conditions are imposed on a particular hyperrectangle and others on a different hyperrectangle that is parallel to the first one (e.g., on two parallel faces of a 3-dimensional domain).

*Received by the editors January 16, 2006; accepted for publication (in revised form) July 30, 2007; published electronically March 5, 2008. This research was supported by the United Kingdom's Engineering and Physical Sciences Research Council (EPSRC) under Platform Grant GR/N08636.
http://www.siam.org/journals/sisc/30-2/65008.html

†Centre for Process Systems Engineering, Department of Chemical Engineering, Imperial College London, South Kensington Campus, London SW7 2AZ, UK. Current address: BASF AG, D-67056 Ludwigshafen, Germany (jens.neumann@basf.com).

‡Corresponding author. Centre for Process Systems Engineering, Department of Chemical Engineering, Imperial College London, South Kensington Campus, London SW7 2AZ, UK (c.pantelides@imperial.ac.uk).

To motivate the consistency issues of interest to this paper, consider the following PDAE system:

$$\frac{\partial u_1}{\partial x_1} = u_1 + u_2, \tag{2}$$

$$\frac{\partial u_1}{\partial x_2} = u_1 - 2u_2, \tag{3}$$

$$\frac{\partial u_2}{\partial x_1} + \frac{\partial u_2}{\partial x_2} = 2u_1 + u_2 - 2u_3, \tag{4}$$

involving the three dependent variables $u_1(x_1, x_2)$, $u_2(x_1, x_2)$, and $u_3(x_1, x_2)$. Suppose we wish to impose on this system boundary conditions of the form $u_2(x_1^*, x_2) = g(x_2)$ and $u_2(x_1, x_2^*) = h(x_1)$, where x_j^* indicates one of the bounds of the independent variable x_j. We then need to be concerned with questions such as the following.

1. Is it, in principle, possible to impose such boundary conditions for some suitable choice of the functions $g(.)$ and $h(.)$?
2. How many additional boundary conditions are necessary?
3. Beyond the obvious corner condition $g(x_2^*) = h(x_1^*)$, what other relations, if any, would need to be satisfied by functions $g(.)$ and $h(.)$ to ensure consistency?

As is well known in the case of mixed systems of ordinary differential and algebraic equations (DAEs), consistent initial values of the variables may need to satisfy not only the original equations but also additional consistency conditions derived by one or more differentiations of some of the system equations with respect to the independent variable. In their pioneering work, Martinson and Barton [4] showed that similar concepts apply to PDAE systems. Consider, in particular, a part of the domain boundary on which the values of a given subset of the independent variables are fixed. Then, in addition to the original PDAEs, the dependent variables may need to satisfy conditions derived by partial differentiation of these PDAEs with respect to these fixed independent variables. Moreover, Martinson and Barton [4] demonstrated that algorithms for the identification of consistency conditions in DAE systems, such as the graph-theoretical algorithm proposed by Pantelides [6], could easily be extended to be used in the context of PDAEs.

In this paper, we focus on the issue of consistency between specifications imposed on different parts of the domain boundary, and also between these conditions and the original PDAEs. One of the key problems in this context is that specifications being imposed on overlapping parts of the boundary (e.g., faces with common edges, or edges with common vertices) may lead to inconsistencies, also known as "corner singularities." The latter are not trivial to detect, especially in PDAEs involving three or more dimensions, and may cause complications in numerical solution algorithms which need special handling (see, for example, Flyer and Fornberg [1]).

Section 2 introduces the basic concept of the domain digraph and a notation that permits the clear and unambiguous description of PDAE analysis algorithms. An overview of the algorithm is provided in section 3. Section 4 deals with one major step of the algorithm, namely, the identification of the consistency conditions that are associated with each arc in the domain digraph; this is achieved by extending an algorithm for index reduction in linear DAE systems and makes use of symbolic fraction-free Gaussian elimination that can deal with operator matrices. Section 5 shows that there may be further restrictions which arise from interactions among two or more overlapping parts of the boundary. These can be identified only by *simultaneously* considering differentiations of the equations on each overlapping part of the boundary with respect to a *different* independent variable. Section 6 considers the second major step of the algorithm, namely that of forming a complete system at each node of the domain digraph via additional specifications ("boundary conditions"), and of deriving consistency relations that must be satisfied by specifications imposed at different nodes in order to avoid corner singularities.

2. Basic concepts.

2.1. The domain digraph. The boundary of a multidimensional hyperrectangular domain comprises a number of lower-dimensional domains, each of which can be characterized in terms of the set $I^F \subseteq \{1, \ldots, N_d\}$ of independent variables that are fixed at given values:

$$x_l = x_l^\star \quad \forall l \in I^F. \tag{5}$$

The original multidimensional domain (i.e., the one over which the PDAE system is defined) has $I^F = \emptyset$.

The *level* of a domain α is the cardinality of the corresponding set I_α^F. For an N_d-dimensional problem, the number of domains of level k ($k = 0, \ldots, N_d$) is $\binom{N_d}{k}$. A domain α is said to be a *child* of a domain β if $I_\beta^F \subset I_\alpha^F$ and $card(I_\alpha^F) = card(I_\beta^F) + 1$. Conversely, domain β is said to be a *parent* of α. The independent variable which is included in I_α^F but not in I_β^F is denoted as $x_{\beta\alpha}$.

A domain may have more than one parent; for example, a domain with $I^F = \{1, 2, 3\}$ has three parents, namely $\{1, 2\}$, $\{1, 3\}$, and $\{2, 3\}$. We use $\mathcal{P}_\alpha$ to denote the set of parents of a domain α. Conversely, a domain may have multiple children; for example, in a 3-dimensional case, domain $\{1\}$ has two children, namely $\{1, 2\}$ and $\{1, 3\}$.

The above definitions lead to a natural representation of the domains of interest in the form of a directed graph ("digraph"), where each node corresponds to a different domain (i.e., one with a distinct I^F) and an arc is drawn from one domain to another if the former is a parent of the latter. The form of this graph for the case $N_d = 3$ is shown in Figure 1. In general, we will use the set I^F to label the corresponding domain node in the digraph. Each arc entering a particular domain node will be labelled by the latter's I^F, with the element that distinguishes the origin from the destination node being marked by an asterisk. For example, $\{1, 2^\star, 3\}$ denotes the arc from domain node $\{1, 3\}$ to domain node $\{1, 2, 3\}$.

2.2. Domain equations. We wish to determine a set of equations $\mathcal{F}_\alpha$ that describe each and every domain α in the domain digraph. In general, each domain node represents a restriction[1] of its parent nodes. Consequently, any equation that applies to a domain node's parent will also apply to the node itself. Moreover, as has already been mentioned in the introduction, the following hold:

- The restriction of a particular domain β to form a child domain α may result in consistency conditions that apply in domain α. Consequently, each arc $\beta \to \alpha$ in the domain digraph is potentially associated with a set of consistency conditions $\mathcal{C}_{\beta\alpha}$.
- There may be additional conditions arising from interactions of the equations describing different parent domains β of the domain α. These are obtained

[1] Obtained by fixing one of the independent variables that is still free.

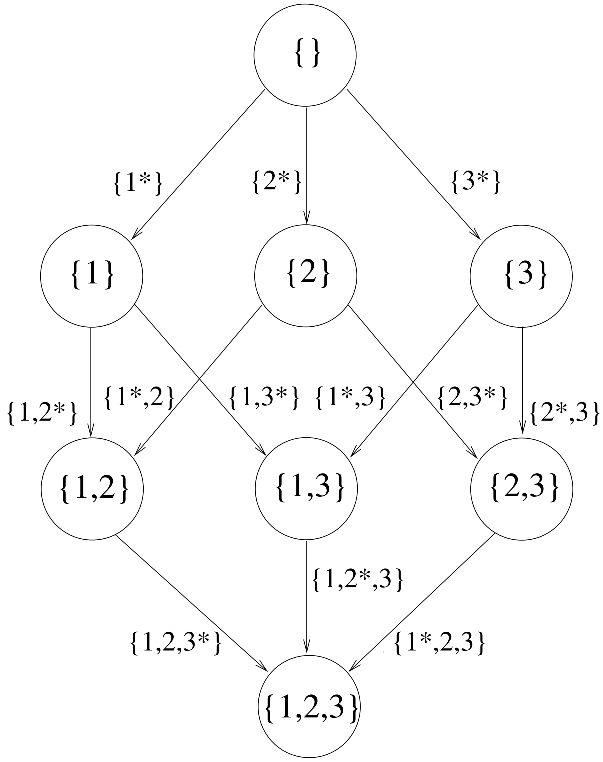

FIG. 1. *Domain digraph for 3-dimensional problems.*

from the first- and higher-order partial derivatives of the equations $\mathcal{F}_\alpha$ with respect to the variable $x_{\beta\alpha}$ that is being restricted on arc $\beta \to \alpha$. We denote these differentiated equations as $\mathcal{D}_{\beta\alpha}$.

The above considerations imply that the set of equations $\mathcal{F}_\alpha$ will include the union of the equations $\mathcal{F}_\beta$ for all parent domains $\beta \in \mathcal{P}_\alpha$, the consistency conditions $\mathcal{C}_{\beta\alpha}$ for all arcs $\beta\alpha$ entering node α for all $\beta \in \mathcal{P}_\alpha$, and the additional conditions $\mathcal{D}_{\beta\alpha}$. These may not be sufficient to determine all the variables that characterize the system behavior in this domain, and additional specification equations ("boundary conditions") $\mathcal{S}_\alpha$ may need to be imposed. In general, these are algebraic or partial differential equations relating subsets of the variables in the domain node. Overall, then, the set of equations that describe a node α will be given by

$$\mathcal{F}_\alpha = \cup_{\beta\in\mathcal{P}_\alpha} (\mathcal{F}_\beta \cup \mathcal{C}_{\beta\alpha} \cup \mathcal{D}_{\beta\alpha}) \cup \mathcal{S}_\alpha. \tag{6}$$

Since a domain node may have multiple parents, all of which are derived from the original (unrestricted) multidimensional domain, the equations that a node inherits from its parents may involve certain redundancy. It is, of course, desirable to identify and remove such redundant equations. More important than redundancies, however, are potential inconsistencies (i.e., corner singularities) which may arise because of specifications $\mathcal{S}_\beta$ and $\mathcal{S}_{\beta'}$ imposed on two different parents β and β' of the same domain node α. In such cases, we need to derive a set of conditions $\mathcal{R}_\alpha$ that must be satisfied by the sets of specifications $\mathcal{S}_\beta$ and $\mathcal{S}_{\beta'}$ so that no such inconsistencies can arise. We will call these *specification consistency relations* (SCRs).

A trivial example of an SCR is the case of a certain dependent variable $u(x_1, x_2)$ being specified as two different functions $f(x_1)$ and $g(x_2)$ on two intersecting edges $x_2 = x_2^\star$ and $x_1 = x_1^\star$ of a 2-dimensional domain. A potential inconsistency may arise at the corner $(x_1^\star, x_2^\star)$ between the two edges if the specified functions do not yield the same value for the dependent variable. Thus, an appropriate specification consistency relation that would exclude such an inconsistency would be $f(x_1^\star) = g(x_2^\star)$. The common approach of dealing with this complication is not to enforce one of the two functions at the corner, e.g., by accepting that $u(x_1, x_2^*) = f(x_1)$ for $x_1 > x_1^*$ but not for $x_1 = x_1^*$. This is equivalent to a modified boundary condition $u(x_1, x_2^*) = \tilde{f}(x_1)$, where the new function $\tilde{f}$ is defined as $\tilde{f}(x_1) \equiv f(x_1)$ for all $x_1 > x_1^*$; $\tilde{f}(x_1^*) = g(x_2^*)$. However, this does not necessarily satisfy higher-order SCRs, and the resulting corner singularities may lead to severe numerical problems.

2.3. Domain variables. The system behavior in each domain node is described by the dependent variables and their partial derivatives expressed as functions of the subset of the independent variables that are *not* fixed in this node, i.e., $\{1, \ldots, N_d\} \setminus I^F$. These functions are to be determined by the solution of a set of (generally, partial differential) equations.

In determining a complete solution over a given domain α, a dependent variable u_i and its partial derivative $\partial u_i/\partial x_j$ with respect to an independent variable x_j that is already fixed (i.e., $j \in I_\alpha^F$) have to be treated as separate dependent variables (cf. Martinson and Barton [4]). This is because, if independent variable x_j is fixed, then u_i is no longer a function of x_j over this part of the boundary and it is no longer possible to deduce the function $\partial u_i/\partial x_j$ by differentiating u_i with respect to x_j. Thus, the set of distinct dependent variables (DDVs) for a given node α comprises the original dependent variables and the partial derivatives of the dependent variables with respect to any independent variable(s) x_i such that $i \in I_\alpha^F$.

For example, consider a 4-dimensional problem involving a dependent variable u_1. Now consider the part of the boundary on which independent variables x_1 and x_3 have been fixed, i.e., a face (x_2, x_4). This corresponds to a domain node α such that $I_\alpha^F = \{x_1, x_3\}$. The DDV set for this node may contain[2] the following variables:

$$
\begin{aligned}
&u_1(x_1^\star, x_2, x_3^\star, x_4),\\
&\partial u_1/\partial x_1(x_1^\star, x_2, x_3^\star, x_4),\ \partial u_1/\partial x_3(x_1^\star, x_2, x_3^\star, x_4),\\
&\partial^2 u_1/\partial x_1^2(x_1^\star, x_2, x_3^\star, x_4),\ \partial^2 u_1/\partial x_3^2(x_1^\star, x_2, x_3^\star, x_4),\ \partial^2 u_1/\partial x_1 \partial x_3(x_1^\star, x_2, x_3^\star, x_4).
\end{aligned}
$$

All of these are quite distinct as far as node α is concerned. For example, take $u_1(x_1^\star, x_2, x_3^\star, x_4)$. It cannot be differentiated with respect to x_1 in order to obtain $\partial u_1/\partial x_1(x_1^\star, x_2, x_3^\star, x_4)$. Consequently, the above set of six DDVs can be denoted as $v_i(x_2, x_4)$, $i = 1, \ldots, 6$, which have to be determined from the solution of the system of equations $\mathcal{F}_\alpha$ given by (6).

On the other hand, partial derivatives with respect to x_2 do not form distinct dependent variables. For example, $\partial u_1/\partial x_2(x_1^\star, x_2, x_3^\star, x_4)$ can be obtained by differentiating $u_1(x_1^\star, x_2, x_3^\star, x_4)$ with respect to x_2. Similarly, $\partial^3 u_1/\partial x_3^2 \partial x_4(x_1^\star, x_2, x_3^\star, x_4)$ can be obtained by differentiating $\partial^2 u_1/\partial x_3^2(x_1^\star, x_2, x_3^\star, x_4)$ with respect to x_4. In

[2]Not all of the variables listed here will necessarily occur in a particular system of interest. Conversely, some systems may involve even higher-order partial derivatives. Note, however, that if a higher-order derivative is a DDV, then all lower-order derivatives leading up to it are also considered to be DDVs.

fact, $\partial u_1/\partial x_2(x_1^\star, x_2, x_3^\star, x_4)$ and $\partial^3 u_1/\partial x_3^2 \partial x_4(x_1^\star, x_2, x_3^\star, x_4)$ can be written simply as $\partial v_1/\partial x_2$ and $\partial v_5/\partial x_4$, respectively.

3. Algorithm overview.

Given a particular domain node α, we wish to determine the complete set of equations $\mathcal{F}_\alpha$ that describe the system behavior on this domain. We also need to derive a set of SCRs $\mathcal{R}_\alpha$ that must be satisfied by any specification equations imposed on the parents and/or other ancestors of node α in order to avoid corner singularities.

PROCEDURE ProcessNode ([in] α, [out] $\mathcal{F}_\alpha$, [out] $\mathcal{S}_\alpha$, [out] $\mathcal{R}_\alpha$)

1. Set $\mathcal{F}_\alpha := \emptyset$, $\mathcal{S}_\alpha := \emptyset$, $\mathcal{R}_\alpha := \emptyset$.
2. FOR EACH parent $\beta \in \mathcal{P}_\alpha$ DO:
   (a) From system $\mathcal{F}_\beta$, derive consistency conditions $\mathcal{C}_{\beta\alpha}$.
   (b) Augment system of equations $\mathcal{F}_\alpha := \mathcal{F}_\alpha \cup \mathcal{F}_\beta \cup \mathcal{C}_{\beta\alpha}$.
   (c) Augment system of equations $\mathcal{F}_\alpha := \mathcal{F}_\alpha \cup_{\nu=1}^{m^*_{\beta\alpha}} \frac{\partial^\nu}{\partial x^\nu_{\beta\alpha}} (\mathcal{F}_\beta \cup \mathcal{C}_{\beta\alpha})$.
3. Analyze the system $\mathcal{F}_\alpha$ to determine
   (a) set of redundant equations,
   (b) set of potentially inconsistent specifications; add specification consistency relations to $\mathcal{R}_\alpha$.

   Discard both redundant and potentially inconsistent specifications from set $\mathcal{F}_\alpha$ (and from set $\mathcal{S}_\alpha$ if appropriate).
4. If system $\mathcal{F}_\alpha$ is of incomplete rank, then
   (a) request one or more new specification equations $\mathcal{S}$;
   (b) set $\mathcal{F}_\alpha := \mathcal{F}_\alpha \cup \mathcal{S}$;
   (c) set $\mathcal{S}_\alpha := \mathcal{S}_\alpha \cup \mathcal{S}$;
   (d) GO TO step 3.
5. END PROCEDURE ProcessNode.

FIG. 2. *Overview of algorithm for processing a domain node.*

An overview of the proposed algorithm is shown in Figure 2. Step 1 initializes the set of equations $\mathcal{F}_\alpha$ associated with the given domain node α to the empty set. The sets of specifications $\mathcal{S}_\alpha$ that may need to be added at this node and SCRs $\mathcal{R}_\alpha$ that may be identified at this node are also initialized to empty sets.

Step 2(a) constructs the set of consistency conditions associated with node α in accordance with (6). In particular, it considers all parent nodes β of α and derives the consistency conditions associated with the restriction of the domain from β to α along the arc $\beta \to \alpha$; this may involve differentiating some of the equations in domain β with respect to the independent variable that is being fixed in going from β to α. This part of the algorithm is considered in detail in section 4. As indicated by step 2(b), the system of equations describing the current node α will include the union of the systems $\mathcal{F}_\beta$ describing its parent nodes, plus the consistency conditions $\mathcal{C}_{\beta\alpha}$ associated with the arc from each one of the parent nodes.

As we shall see in section 5, the first- and higher-order partial derivatives of the equations $\mathcal{F}_\beta$ and the consistency conditions $\mathcal{C}_{\beta\alpha}$ with respect to the independent variable $x_{\beta\alpha}$ may result in additional consistency conditions when equations arising from more than one parent node are considered simultaneously. Therefore, at step 2(c) we augment the system $\mathcal{F}_\alpha$ with these additional derivatives up to an order $m^*_{\beta\alpha}$; the choice of the latter is considered in section 5. Which of these differentiations lead to useful consistency conditions is determined at step 3.

Step 3 analyzes $\mathcal{F}_\alpha$ to identify any redundant equations. It also identifies any potentially inconsistent equations and derives SCRs that can prevent any such inconsistencies from arising. Both redundant and inconsistent equations are removed from the system of equations describing the current node; SCRs are also recorded as part of the set $\mathcal{R}_\alpha$ so that they can form part of the output of procedure ProcessNode. The algorithm used for performing this step is considered in detail in section 6.

The system of equations $\mathcal{F}_\alpha$ resulting from step 3 may be incomplete in the sense that it is not sufficient to determine the variation of all dependent variables over the part of the boundary corresponding to node α. Consequently (step 4 of the algorithm), it may be necessary to introduce to it additional equations $\mathcal{S}$ (i.e., boundary conditions); this represents information that cannot be derived from the original equations in the mathematical model and, consequently, requires interaction with the model's developer. In such cases, these equations are added to the system $\mathcal{F}_\alpha$ and are also recorded in the set $\mathcal{S}_\alpha$ so that they can form part of the output of procedure ProcessNode. In general, there is no guarantee that any such specifications will be nonredundant or even consistent with the equations already in $\mathcal{F}_\alpha$. Consequently, the redundancy and consistency analysis of step 3 has to be repeated until no further modifications to the set $\mathcal{F}_\alpha$ are found to be necessary.

The form of step 2 suggests that the domain digraph must be traversed in a breadth-first manner, i.e., all parent nodes of node α need to be resolved before considering α itself. It is assumed that the original set of PDAEs, $\mathbf{F}$ describing the system behavior over the N_d dimensional domain is a complete and nonredundant set of equations, and consequently, the level 0 node is simply characterized by the original equations. This leads to the overall algorithm outlined in Figure 3. In addition to the sets of equations $\mathcal{F}_\alpha$ that characterize each and every domain α, the algorithm provides a complete set of all specifications $\mathcal{S}$ imposed on the system and the set $\mathcal{R}$ of all consistency relations that these specifications must satisfy.

PROCEDURE AnalyzePDAESystem

1. Set $\mathcal{F}_{\{\}} := \{\mathbf{F}\}$, $\mathcal{S} := \emptyset$, $\mathcal{R} := \emptyset$.
2. FOR level $k := 1$ TO N_d DO
   (a) FOR each domain node α of level k DO
      i. ProcessNode (α, $\mathcal{F}_\alpha$, $\mathcal{S}_\alpha$, $\mathcal{R}_\alpha$);
      ii. Set $\mathcal{S} := \mathcal{S} \cup \mathcal{S}_\alpha$; $\mathcal{R} := \mathcal{R} \cup \mathcal{R}_\alpha$.
3. END PROCEDURE AnalyzePDAESystem.

FIG. 3. *Overall PDAE system analysis algorithm.*

4. Consistency conditions associated with domain digraph arcs.

In the context of step 2 of the algorithm in Figure 2, the arc $\beta \to \alpha$ is considered only after a complete and nonredundant set of equations $\mathcal{F}_\beta$ describing a parent node β has been established by the earlier application of procedure ProcessNode to node β. For linear PDAE systems, $\mathcal{F}_\beta$ can be written in the form

$$\mathbf{A}\frac{\partial^m \mathbf{v}}{\partial x_{\beta\alpha}^m} + \sum_{k=0}^{m-1} \mathbf{B}_k \frac{\partial^k \mathbf{v}}{\partial x_{\beta\alpha}^k} = \mathbf{f}, \tag{7}$$

where $\mathbf{v}$ denote the DDVs associated with node β (cf. section 2.3) and $x_{\beta\alpha}$ is the independent variable that is being restricted in the arc from the parent node β to the child node α, i.e., $I_\alpha^F = I_\beta^F \cup \{j\}$. Here, m denotes the order of the highest partial derivative with respect to $x_{\beta\alpha}$. The matrices $\mathbf{A}$ and $\mathbf{B}_k$, $k = 0, \dots, m-1$, are square and may involve partial derivative operators with respect to independent variables $x_{j'}$, $j' \notin I_\alpha$.

The set of consistency conditions can be determined by applying consistent initialization algorithms for DAEs to the system (7). Most of these algorithms require reformulation of this order m system to an equivalent first-order system. However, Neumann [5] has presented an algorithm that can be applied directly to high-order linear DAEs by extending the algorithm of Gear and Petzold [2] for first-order systems. Briefly, the algorithm applies Gaussian elimination with pivoting to the leading matrix $\mathbf{A}$; if this results in any zero rows, then these are differentiated with respect to the independent variable to yield consistency relations that must be satisfied by the dependent variables. The procedure is then repeated until matrix $\mathbf{A}$ is found to be of full rank, at which point the algorithm terminates.

The main complication in applying this type of algorithm to the PDAE system (7) arises from the fact that matrices $\mathbf{A}$ and $\mathbf{B}_k$, $k = 0, \dots, m-1$, in general involve partial derivative operators. We therefore need to employ symbolic Gaussian elimination (see, for instance, Golub and Van Loan [3]), which may be extended to deal with such operator-valued coefficient matrices. Row additions and subtraction are straightforward. Multiplication of a row with a partial differentiation operator corresponds to partially differentiating that particular equation with respect to the independent variable direction that is specified in the operator. As division by operators is not defined, we employ *fraction-free* Gaussian elimination, where the need for division is eliminated by multiplying the rows concerned by their largest common denominator and then subtracting them from each other.

4.1. Example. As an example of the derivation of consistency conditions using the algorithm described in this section, we consider the time-dependent Navier–Stokes equations for creeping, isothermal flow in a rectangular 2-dimensional duct.

$$\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_3} = 0, \tag{8}$$

$$\alpha_1 \frac{\partial u_1}{\partial x_1} + \frac{\partial u_3}{\partial x_2} - \alpha_2 \left(\frac{\partial^2 u_1}{\partial x_2^2} + \frac{\partial^2 u_1}{\partial x_3^2} \right) = 0, \tag{9}$$

$$\alpha_1 \frac{\partial u_2}{\partial x_1} + \frac{\partial u_3}{\partial x_3} - \alpha_2 \left(\frac{\partial^2 u_2}{\partial x_2^2} + \frac{\partial^2 u_2}{\partial x_3^2} \right) = 0. \tag{10}$$

There are three independent variables, namely variable x_1 describing the time domain and variables x_2 and x_3 describing the two spatial dimensions. The dependent variables u_1 and u_2 correspond to the velocities in the x_2 and x_3 spatial direction, respectively, and u_3 to the pressure.

In the remainder of this section, we consider the derivation of the consistency conditions for two of the arcs in the domain digraph, namely $\{1^\star\}$ and $\{1, 2^\star\}$.

4.2. Consistency conditions for arc $\{1^\star\}$. Arc $\{1^\star\}$ originates at node $\{\}$, which is described by the original system equations; the corresponding DDVs are simply the original dependent variables, i.e., u_1, u_2, and u_3. We start by writing the system in the form (7) with $x_{\beta\alpha} \equiv x_1$:

$$\begin{bmatrix} 0 & 0 & 0 \\ \alpha_1 & 0 & 0 \\ 0 & \alpha_1 & 0 \end{bmatrix} \frac{\partial}{\partial x_1} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{bmatrix} \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} & 0 \\ \gamma & 0 & \frac{\partial}{\partial x_2} \\ 0 & \gamma & \frac{\partial}{\partial x_3} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = 0, \tag{11}$$

where $\gamma \equiv -\alpha_2 \left(\frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}\right)$.

We note that the third column of matrix $\mathbf{A}$ above is entirely zero. In order to avoid unnecessary differentiations (see Neumann [5]), we replace variable u_3 by $\frac{\partial \hat{u}_3}{\partial x_1}$ and rewrite the above in the form

$$\begin{bmatrix} 0 & 0 & 0 \\ \alpha_1 & 0 & \frac{\partial}{\partial x_2} \\ 0 & \alpha_1 & \frac{\partial}{\partial x_3} \end{bmatrix} \frac{\partial}{\partial x_1} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} + \begin{bmatrix} \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} & 0 \\ \gamma & 0 & 0 \\ 0 & \gamma & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} = 0. \tag{12}$$

Gaussian elimination with row pivoting applied to the leading matrix $\mathbf{A}$ leads to the following system:

$$\begin{bmatrix} \alpha_1 & 0 & \frac{\partial}{\partial x_2} \\ 0 & \alpha_1 & \frac{\partial}{\partial x_3} \\ 0 & 0 & 0 \end{bmatrix} \frac{\partial}{\partial x_1} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} + \begin{bmatrix} \gamma & 0 & 0 \\ 0 & \gamma & 0 \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} = 0. \tag{13}$$

Thus, the rank of $\mathbf{A}$ is equal to 2, and, in accordance with the Gear and Petzold [2] algorithm, the last equation has to be differentiated with respect to x_1 to yield the consistency condition

$$\frac{\partial^2 u_1}{\partial x_1 \partial x_2} + \frac{\partial^2 u_2}{\partial x_1 \partial x_3} = 0. \tag{14}$$

System (13) is modified to

$$\begin{bmatrix} \alpha_1 & 0 & \frac{\partial}{\partial x_2} \\ 0 & \alpha_1 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} & 0 \end{bmatrix} \frac{\partial}{\partial x_1} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} + \begin{bmatrix} \gamma & 0 & 0 \\ 0 & \gamma & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} = 0. \tag{15}$$

We now apply further row operations to the leading matrix above to bring it to upper-triangular form by eliminating the two subdiagonal elements in the last row. In accordance with fraction-free symbolic Gaussian elimination, this involves multiplying the first row by $\partial/\partial x_2$, the second row by $\partial/\partial x_3$, and the last row by α_1, and then subtracting the first and second rows from the last one. The final result is

$$\begin{bmatrix} \alpha_1 & 0 & \frac{\partial}{\partial x_2} \\ 0 & \alpha_1 & \frac{\partial}{\partial x_3} \\ 0 & 0 & -\frac{\partial^2}{\partial x_2^2} - \frac{\partial^2}{\partial x_3^2} \end{bmatrix} \frac{\partial}{\partial x_1} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} + \begin{bmatrix} \gamma & 0 & 0 \\ 0 & \gamma & 0 \\ -\frac{\partial \gamma}{\partial x_2} & -\frac{\partial \gamma}{\partial x_3} & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \hat{u}_3 \end{pmatrix} = 0. \tag{16}$$

As the leading matrix $\mathbf{A}$ is nonsingular, the algorithm can terminate, having identified a single consistency relation (14).

4.3. Consistency conditions for arc $\{1, 2^\star\}$. We now turn to consider arc $\{1, 2^\star\}$. The DDVs for the parent node $\{1\}$ (cf. section 2.3) include the original variables u_1, u_2, and u_3, as well as those partial derivatives of these variables with respect to the fixed independent variable x_1 that actually occur in the system, namely $\partial u_1/\partial x_1$ and $\partial u_2/\partial x_1$. We label these five variables (in the order listed, with the value of x_1 fixed at $x_1^\star$) as $v_i(x_2, x_3)$, $i = 1, \ldots, 5$, and rewrite the system equations (8)–(10)

and the consistency condition (14) as

$$\frac{\partial v_1}{\partial x_2} + \frac{\partial v_2}{\partial x_3} = 0, \tag{17}$$

$$\alpha_1 v_4 + \frac{\partial v_3}{\partial x_2} - \alpha_2 \left(\frac{\partial^2 v_1}{\partial x_2^2} + \frac{\partial^2 v_1}{\partial x_3^2} \right) = 0, \tag{18}$$

$$\alpha_1 v_5 + \frac{\partial v_3}{\partial x_3} - \alpha_2 \left(\frac{\partial^2 v_2}{\partial x_2^2} + \frac{\partial^2 v_2}{\partial x_3^2} \right) = 0, \tag{19}$$

$$\frac{\partial v_4}{\partial x_2} + \frac{\partial v_5}{\partial x_3} = 0. \tag{20}$$

The above four system equations are not sufficient to determine uniquely the initial state of the system at $x_1 = 0$. An additional specification equation[3] needs to be provided to allow the determination of $v_i(x_2, x_3)$, $i = 1, \ldots, 5$, for all x_2, x_3. A formal analysis of the issue of additional specification equations is provided in section 6. For the moment, the reader is asked to accept that a complete set of equations can be obtained by specifying the initial variation of u_1 over the entire spatial domain:

$$v_1(x_2, x_3) = g(x_2, x_3) \quad \forall x_2, x_3, \tag{21}$$

where $g(.,.)$ is a given function.

The system of equations (17)–(21) is a system of five PDAEs in the five dependent variables $v_i(x_2, x_3)$ over the 2-dimensional domain (x_2, x_3). To establish the consistency conditions for arc $\{1, 2^\star\}$, we place the above system in the standard form (cf. (7))

$$\mathbf{A}\frac{\partial^2 \mathbf{v}}{\partial x_2^2} + \mathbf{B_1}\frac{\partial \mathbf{v}}{\partial x_2} + \mathbf{B_0}\mathbf{v} = \mathbf{f}, \tag{22}$$

where the left-hand side matrices and right-hand side vector are given by

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ -\alpha_2 & 0 & 0 & 0 & 0 \\ 0 & -\alpha_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{B_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$
$$\mathbf{B_0} = \begin{pmatrix} 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 \\ -\alpha_2 \frac{\partial^2}{\partial x_3^2} & 0 & 0 & \alpha_1 & 0 \\ 0 & -\alpha_2 \frac{\partial^2}{\partial x_3^2} & \frac{\partial}{\partial x_3} & 0 & \alpha_1 \\ 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \mathbf{f} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ g(x_2, x_3) \end{pmatrix}. \tag{23}$$

We note that matrix $\mathbf{A}$ has three zero columns, corresponding to variables v_3, v_4, and v_5. In order to avoid unnecessary differentiations (cf. Neumann [5]), we introduce a new variable $\hat{v}_3$ such that $\partial \hat{v}_3 / \partial x_2 \equiv v_3$; similarly, we define $\hat{v}_4$ such that $\partial \hat{v}_4 / \partial x_2 \equiv v_4$. If we now replace v_3 and v_4 in the vector of dependent variables by $\hat{v}_3$ and $\hat{v}_4$, respectively, the third and fourth columns of $\mathbf{B_1}$ (which are nonzero) will

[3] Since the independent variable x_1 is time, this is actually an initial condition.

become the corresponding columns of $\mathbf{A}$. Since the last column of $\mathbf{B_1}$ is also zero, we need to define a new variable $\hat{v}_5$ such that $\partial^2 \hat{v}_5/\partial x_2^2 \equiv v_5$; this has the effect of moving the last column of $\mathbf{B_0}$ (which is nonzero) into the last column of A.

In summary, the PDE system (17)–(21) can be expressed in terms of the modified set of variables $\hat{\mathbf{v}}$ related to $\mathbf{v}$ via $\mathbf{v} = (\hat{v}_1,\ \hat{v}_2,\ \partial\hat{v}_3/\partial x_2,\ \partial\hat{v}_4/\partial x_2,\ \partial^2\hat{v}_5/\partial x_2^2)^T$. Its standard form $\mathbf{A}\partial^2\hat{\mathbf{v}}/\partial x_2^2 + \mathbf{B_1}\partial\hat{\mathbf{v}}/\partial x_2 + \mathbf{B_0}\hat{\mathbf{v}} = \mathbf{f}$ is given by the following matrices and right-hand side vector, shown in the compact form $(\mathbf{A}|\mathbf{B_1}|\mathbf{B_0}|\mathbf{f})$:

$$
\left(\begin{array}{ccccc|ccccc|ccccc|c}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 \\
-\alpha_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha_2 & 0 & 0 & \alpha_1 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & g(x_2, x_3)
\end{array}\right). \tag{24}
$$

Performing Gaussian elimination with row pivoting on the leading matrix $\mathbf{A}$ transforms the above to

$$
\left(\begin{array}{ccccc|ccccc|ccccc|c}
-\alpha_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha_2 & 0 & 0 & \alpha_1 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & g(x_2, x_3)
\end{array}\right). \tag{25}
$$

The rank of the leading matrix is 3. Thus, the last two equations are to be differentiated with respect to x_2, yielding the equations

$$
\frac{\partial^2\hat{v}_1}{\partial x_2^2} + \frac{\partial^2\hat{v}_2}{\partial x_2\partial x_3} = 0 \qquad \text{at} \quad x_2 = x_2^\star \quad \forall x_3, \tag{26}
$$

$$
\frac{\partial\hat{v}_1}{\partial x_2} = \frac{\partial}{\partial x_2} g(x_2, x_3) \qquad \text{at} \quad x_2 = x_2^\star \quad \forall x_3. \tag{27}
$$

If the last two equations are replaced by their differentiated forms, the system matrix becomes

$$
\left(\begin{array}{ccccc|ccccc|ccccc|c}
-\alpha_2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \alpha_1 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha_2 & 0 & 0 & \alpha_1 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_2} g(x_2, x_3)
\end{array}\right). \tag{28}
$$

Once more, Gaussian elimination with row pivoting is applied to the leading matrix, transforming the system to the form

$$
\left(\begin{array}{ccccc|ccccc|ccccc|c}
1 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha_2 & 0 & 0 & \alpha_1 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & \alpha_2\frac{\partial}{\partial x_3} & 0 & \alpha_1 & 0 & -\alpha_2\frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_2} g(x_2, x_3)
\end{array}\right). \tag{29}
$$

The leading matrix is still rank deficient, and the last equation needs to be differentiated again, yielding

$$\frac{\partial^2 \hat{v}_1}{\partial x_2^2} = \frac{\partial^2}{\partial x_2^2} g(x_2, x_3) \qquad \text{at} \quad x_2 = x_2^\star \quad \forall x_3, \tag{30}$$

which results in the updated system matrix
(31)

$$\left(\begin{array}{ccccc|ccccc|ccccc|c}
1 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha_2 & 0 & 0 & \alpha_1 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & -\alpha_2 \frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & \alpha_2 \frac{\partial}{\partial x_3} & 0 & \alpha_1 & 0 & -\alpha_2 \frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial^2}{\partial x_2^2} g(x_2, x_3)
\end{array}\right).$$

Gaussian elimination with row pivoting of the leading matrix gives
(32)

$$\left(\begin{array}{ccccc|ccccc|ccccc|c}
1 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha_2 & 0 & 0 & \alpha_1 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & -\alpha_2 \frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & \alpha_2 \frac{\partial}{\partial x_3} & 0 & \alpha_1 & 0 & -\alpha_2 \frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial^2}{\partial x_2^2} g(x_2, x_3)
\end{array}\right).$$

The rank of the leading matrix is still 4. Thus, the last equation has to be differentiated again, yielding

$$-\frac{\partial^3 \hat{v}_2}{\partial x_2^2 \partial x_3} = \frac{\partial^3}{\partial x_2^3} g(x_2, x_3) \qquad \text{at} \quad x_2 = x_2^\star \quad \forall x_3, \tag{33}$$

which leads to the updated system matrix
(34)

$$\left(\begin{array}{ccccc|ccccc|ccccc|c}
1 & 0 & 0 & 0 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\alpha_2 & 0 & 0 & \alpha_1 & 0 & 0 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & -\alpha_2 \frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & \alpha_2 \frac{\partial}{\partial x_3} & 0 & \alpha_1 & 0 & -\alpha_2 \frac{\partial^2}{\partial x_3^2} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{\partial}{\partial x_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial^3}{\partial x_2^3} g(x_2, x_3)
\end{array}\right).$$

At this stage, the leading matrix becomes nonsingular, and the algorithm can terminate. The analysis has demonstrated that, on any boundary with fixed values of the independent variables x_1 and x_2, the system-dependent variables must obey the four consistency conditions (26), (27), (30), and (33). In terms of the original system variables, these can be written as

$$\frac{\partial^2 u_1}{\partial x_2^2} + \frac{\partial^2 u_2}{\partial x_2 \partial x_3} = 0 \qquad \text{at} \quad x_1 = x_1^\star,\ x_2 = x_2^\star \quad \forall x_3, \tag{35}$$

$$\frac{\partial u_1}{\partial x_2} = \frac{\partial}{\partial x_2} g(x_2, x_3) \qquad \text{at} \quad x_1 = x_1^\star,\ x_2 = x_2^\star \quad \forall x_3, \tag{36}$$

$$\frac{\partial^2 u_1}{\partial x_2^2} = \frac{\partial^2}{\partial x_2^2} g(x_2, x_3) \qquad \text{at} \quad x_1 = x_1^\star,\ x_2 = x_2^\star \quad \forall x_3, \tag{37}$$

$$-\frac{\partial^3 u_2}{\partial x_2^2 \partial x_3} = \frac{\partial^3}{\partial x_2^3} g(x_2, x_3) \qquad \text{at} \quad x_1 = x_1^\star,\ x_2 = x_2^\star \quad \forall x_3. \tag{38}$$

We note that these are not necessarily *all* the consistency conditions that have to be satisfied at node $\{1, 2\}$ of the domain digraph: this node has two parent nodes, namely $\{1\}$ and $\{2\}$. Here we have considered the arc $\{1, 2^\star\}$ leading to the node of interest from parent $\{1\}$. It may also be possible to derive additional consistency conditions by considering the arc $\{1^\star, 2\}$ originating from parent $\{2\}$. Of course, these two sets of consistency conditions, as well as any boundary specifications imposed at the two parent nodes, may involve a certain degree of redundancy and/or potential inconsistency when put together in a single system of equations describing the child node $\{1, 2\}$. Resolving such issues in a formal manner is the topic of the next two sections.

5. Additional differentiations of parent node equations. As described in section 4, given a node α, the derivatives of the equations of a parent node β with respect to the independent variable $x_{\beta\alpha}$ may imply certain consistency conditions regarding the DDVs of α. These conditions can be determined in a relatively straightforward manner by considering each parent node β independently of any other parent. However, it is also possible that additional conditions may arise from interactions between equations belonging to different parents.

5.1. Example. In order to understand the origins of this complication, consider a 1-dimensional transient heat conduction equation of the form

$$\frac{\partial u}{\partial x_1} - \frac{\partial^2 u}{\partial x_2^2} = 0, \tag{39}$$

where x_1 typically corresponds to time and x_2 to a spatial position. As is well known, this equation requires one initial condition and two boundary conditions. Here we choose conditions of the form

$$u = g(x_2) \qquad \text{at} \quad x_1 = x_1^\star \quad \forall x_2, \tag{40}$$

and

$$u = h(x_1) \qquad \text{at} \quad x_2 = x_2^\star \quad \forall x_1, \tag{41}$$

$$\frac{\partial u}{\partial x_2} = k(x_1) \qquad \text{at} \quad x_2 = x_2^\star \quad \forall x_1. \tag{42}$$

In a heat conduction context, these correspond to specifications of the initial temperature throughout the 1-dimensional domain, and of the temperature and heat flux at one end of the domain ($x_2 = x_2^*$). Consider, for example, a rod of finite length made of thermally conducting material, one end of which is immersed in a medium of given temperature $h(x_1)$ at all times x_1 (e.g., a water/ice mixture at 0°C). Then the above equations define the engineering problem that one would have to solve to determine the rate at which heat would have to be introduced at the *other* end of the rod (e.g., via an electric heating coil) in order to deliver a desired heat flux $k(x_1)$ into the surrounding medium at x_2^*.

In fact, the application of the algorithm of section 4 to (39) determines that there are no consistency conditions associated with either arc $\{1^*\}$ or arc $\{2^*\}$. Node $\{1\}$ has two DDVs, namely $u(x_1^*, x_2)$ and $\partial u/\partial x_1(x_1^*, x_2)$, and these are determined by the system of two equations (39) and (40). On the other hand, node $\{2\}$ has three DDVs, namely $u(x_1, x_2^*)$, $\partial u/\partial x_2(x_1, x_2^*)$, and $\partial^2 u/\partial x_2^2(x_1, x_2^*)$, determined by (39), (41), and (42).

We now consider the corner node $\{1,2\}$. The application of the algorithm of section 4 identifies two consistency relations associated with arc $\{1,2^*\}$, namely

$$\frac{\partial u}{\partial x_2} = g'(x_2) \quad \forall x_2, \tag{43}$$

$$\frac{\partial^2 u}{\partial x_2^2} = g''(x_2) \quad \forall x_2. \tag{44}$$

An additional consistency condition is associated with arc $\{1^*,2\}$:

$$\frac{\partial u}{\partial x_1} = h'(x_1) \quad \forall x_2. \tag{45}$$

Overall, the corner node $\{1,2\}$ is described by the four DDVs $[u,\ \frac{\partial u}{\partial x_2},\ \frac{\partial^2 u}{\partial x_2^2},\ \frac{\partial u}{\partial x_1}]$, all evaluated at $(x_1^\star, x_2^\star)$, and these have to satisfy the seven equations (39)–(45). This is possible only if the functions $g(.)$, $h(.)$, and $k(.)$ used in the specifications (40)–(42) satisfy certain fairly obvious consistency relations:[4]

- $g(x_2^*) = h(x_1^*)$, arising from the uniqueness of the value of $u(x_1^*, x_2^*)$;
- $g'(x_2^*) = k(x_1^*)$, arising from the uniqueness of the value of $\partial u/\partial x_2(x_1^*, x_2^*)$;
- $h'(x_1^*) - g''(x_2^*) = 0$, arising from enforcing (39) at the corner.

Like all algorithms of this kind (e.g., those used for the analysis of DAE system initial conditions), the algorithm of section 4 aims to determine the *minimal* required set of equation differentiations. For example, in considering arc $\{1^*,2\}$, the differentiation of (39) with respect to x_1 was deemed to be unnecessary since enforcing the resulting equation at the corner yields

$$\frac{\partial^2 u}{\partial x_1^2}(x_1^*, x_2^*) - \frac{\partial^3 u}{\partial x_1 \partial x_2^2}(x_1^*, x_2^*) = 0, \tag{46}$$

which does not, in itself, restrict the admissible values of the four DDVs listed above. It is always possible to satisfy this equation by choosing an appropriate value for $\partial^3 u/\partial x_1 \partial x_2^2(x_1^*, x_2^*)$. Similarly, in considering arc $\{1,2^*\}$, the differentiation of (39) twice with respect to x_2 would yield

$$\frac{\partial^3 u}{\partial x_2^2 \partial x_1}(x_1^*, x_2^*) - \frac{\partial^4 u}{\partial x_2^4}(x_1^*, x_2^*) = 0, \tag{47}$$

which again does not, in itself, restrict the admissible values of the DDVs. However, if one combines (46) and (47), one obtains the relationship

$$\frac{\partial^2 u}{\partial x_1^2}(x_1^*, x_2^*) = \frac{\partial^4 u}{\partial x_2^4}, \tag{48}$$

which, when combined with (41) and (40), yields a new SCR,[5]

$$h''(x_1^*) = g''''(x_2^*), \tag{49}$$

which is not implied by any of the others that have already been identified. In other words, equation differentiations which are not useful in the context of any individual arc $\beta \to \alpha$ may actually lead to additional SCRs if considered simultaneously.

[4] A formal method for the derivation of such relations will be presented in section 6.

[5] We recall that SCRs are conditions that need to be satisfied by specifications already imposed on higher-dimensional parts of the domain corresponding to ancestors of the node α under consideration.

5.2. Maximum number of required differentiations. The above example indicates that some SCRs may arise from linear combinations of the partial derivatives of the parent node equations $\mathcal{F}_\beta$ with respect to the independent variable $x_{\beta\alpha}$ being restricted along the arcs $\beta \to \alpha$ for different parent nodes $\beta \in \mathcal{P}_\alpha$.

Standard algorithms for the identification of consistency conditions for DAE systems consider differentiations of the system equations with respect to a single independent variable. These algorithms can identify which differentiations will lead to useful consistency conditions *before* actually carrying out these differentiations. For example, the Gear–Petzold algorithm for linear DAEs of the form $\mathbf{A}\dot{x}+\mathbf{B}x = f(t)$ applies Gaussian elimination to the lead matrix $\mathbf{A}$; only if this indicates that the matrix is rank deficient does the algorithm differentiate the equations corresponding to the zero rows. Similarly, the algorithm by Pantelides [6] uses graph theoretical concepts to identify subsets of k equations, differentiation of which would introduce only $k-1$ new variables; only then are these equation subsets differentiated. As pointed out by Martinson and Barton [4] and as we have already seen in section 4 of this paper, this approach carries over to the PDAE counterparts of these DAE algorithms.

Unfortunately, the a priori identification of equations to be differentiated does not seem to be possible here as we are considering simultaneously the differentiation of equations from more than one parent node β, each with respect to a different independent variable $x_{\beta\alpha}$. Instead, we adopt a different approach whereby we first perform the differentiations and then we determine a posteriori which, if any, of the resulting differentiated equations lead to SCRs. This can be determined by considering the linear system comprising these differentiated equations, the parent node equations $\mathcal{F}_\beta$, and any consistency conditions $\mathcal{C}_{\beta\alpha}$ associated with the arcs $\beta \to \alpha$. More specifically, Gaussian elimination can identify that some of these linear equations will be inconsistent unless the corresponding elements of the right-hand side vector are zero, which leads directly to the SCRs.

Before we proceed to consider the SCR identification more formally in section 6, we need to determine the maximum number of times that the equations of each parent node β need to be differentiated with respect to $x_{\beta\alpha}$. We note that here we are concerned with SCRs corresponding to potential inconsistencies between differentiated equations from different parents. Such inconsistencies can arise only if the differentiation(s) of an equation from a parent β with respect to $x_{\beta\alpha}$ gives rise to one or more variables which are in common with those arising from differentiation(s) of at least one equation from a different parent β' with respect to $x_{\beta'\alpha}$. For example, the SCR (49) arose because the differentiated equations (46) and (47) both involved the variable $\partial^3 u/\partial x_1 \partial x_2^2$.

The above reasoning leads to the following simple algorithm for determining the maximum number of differentiations that need to be taken into account.

Given a node α, consider each parent node $\beta \in \mathcal{P}_\alpha$.

1. For each dependent variable u_i, determine the order $m_{i\beta\alpha}$ of the highest-order partial derivative with respect to $x_{\beta\alpha}$ appearing in any equation of any other parent node $\beta' \in \mathcal{P}_\alpha$, $\beta' \neq \beta$.
2. For each dependent variable u_i, determine the order $\mu_{i\beta\alpha}$ of the lowest-order partial derivative with respect to $x_{\beta\alpha}$ occurring in the equations describing node β.
3. The maximum number of differentiations with respect to $x_{\beta\alpha}$ which need to be considered is given by $m^*_{\beta\alpha} = \max_i(m_{i\beta\alpha} - \mu_{i\beta\alpha})$.

It is worth noting the following:

- The equations in parent node β are the only ones that will be differentiated with respect to $x_{\beta\alpha}$. Consequently, differentiations of equations in any *other* parent node β' will not result in an increase in the order of any partial derivatives they contain with respect to $x_{\beta\alpha}$. Thus, step 1 of the above algorithm produces, for each dependent variable u_i, the maximum order $m_{i\beta\alpha}$ for a partial derivative with respect to $x_{\beta\alpha}$ that can arise from any other parent β'.

- Now consider an equation of parent node β. Suppose this equation involves a μ_βth-order partial derivative of a dependent variable u_i with respect to $x_{\beta\alpha}$. Then, differentiating this equation by more than $m_{i\beta\alpha} - \mu_{i\beta\alpha}$ times with respect to $x_{\beta\alpha}$ will give rise to a partial derivative of order exceeding the maximum order $m_{i\beta\alpha}$ which can possibly appear in any equation derived from other parent nodes β'. Thus, such differentiations are unnecessary as they cannot result in any overlap in the partial derivatives of u_i arising from differentiations from different parent nodes.
- The same dependent variable u_i may appear in more than one term of the same or different equations, each of a different order with respect to $x_{\beta\alpha}$. Hence, step 2 sets $\mu_{i\beta\alpha}$ to be the lowest of all these orders.
- Inconsistencies leading to SCRs may potentially arise from different dependent variables occurring in the same equation. Hence, $m^*_{\beta\alpha}$ is determined by considering the maximum number of differentiations which would lead to a potential overlap between partial derivatives of at least one dependent variable.

Once we determine $m^*_{\beta\alpha}$, we apply this number of differentiations with respect to $x_{\beta\alpha}$ to the equations (7) describing node β to obtain an extended set of equations that needs to be considered in the context of its child node α:

$$\mathbf{A}\frac{\partial^{m+\nu}\mathbf{v}}{\partial x_{\beta\alpha}^{m+\nu}} + \sum_{k=0}^{m-1}\mathbf{B}_k\frac{\partial^{k+\nu}\mathbf{v}}{\partial x_{\beta\alpha}^{k+\nu}} = \mathbf{f}, \quad \nu = 0, \dots, m^*_{\beta\alpha}. \tag{50}$$

We note that if the application of the algorithm of section 4 to the original system has identified that (7) gives rise to any consistency conditions $\mathcal{C}_{\beta\alpha}$ associated with arc $\beta \to \alpha$, then similar consistency conditions will be associated with the extended system (50) for each value of ν, since the lead matrix A is the same for all such systems. Therefore, these additional consistency conditions are simply the first $m^*_{\beta\alpha}$ partial derivatives of $\mathcal{C}_{\beta\alpha}$ with respect to $x_{\beta\alpha}$.

In summary, taking each parent node $\beta \in \mathcal{P}_\alpha$ in sequence, we use the above algorithm to determine $m^*_{\beta\alpha}$. The equations that need to be considered in the context of the child node α are $\mathcal{F}_\beta$, $\mathcal{C}_{\beta\alpha}$, and their first $m^*_{\beta\alpha}$ derivatives with respect to $x_{\beta\alpha}$.

5.3. Illustrative example. We reconsider the corner node $\{1, 2\}$ of the heat conduction example of section 5.1 in the context of the algorithm of section 5.2. As has already been noted, parent node $\{1\}$ is described by (39) and (40). We need to determine which differentiations, if any, of these equations with respect to x_2 may potentially be useful. To do this, we note that the highest-order partial derivative with respect to x_2 in all other parent nodes (i.e., node $\{2\}$ in this case) is 2. We also note that the lowest-order partial derivative with respect to x_2 in (39) and (40) is 0. Therefore, $m^*_{\{1,2^*\}} = 2 - 0 = 2$.

Therefore, in analyzing the corner node $\{1, 2\}$ (cf. section 6), we will have to consider up to two differentiations with respect to x_2 of (39) and (40) which describe node $\{1\}$, as well as the consistency relations (43) and (44) associated with arc $\{1, 2^*\}$.

The first such differentiation yields

$$\begin{bmatrix} 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \frac{\partial u}{\partial x_2} \\ \frac{\partial^2 u}{\partial x_2 \partial x_1} \\ \frac{\partial^2 u}{\partial x_2^2} \\ \frac{\partial^3 u}{\partial x_2^3} \end{pmatrix} = \begin{bmatrix} 0 \\ g'(x_2) \\ g''(x_2) \\ g'''(x_2) \end{bmatrix} \quad \forall x_2, \tag{51}$$

and the second yields

$$\begin{bmatrix} 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \frac{\partial^2 u}{\partial x_2^2} \\ \frac{\partial^3 u}{\partial x_2^2 \partial x_1} \\ \frac{\partial^3 u}{\partial x_2^3} \\ \frac{\partial^4 u}{\partial x_2^4} \end{pmatrix} = \begin{bmatrix} 0 \\ g''(x_2) \\ g'''(x_2) \\ g''''(x_2) \end{bmatrix} \quad \forall x_2. \tag{52}$$

Parent node $\{2\}$ is described by (39), (41), and (42). We need to determine which differentiations, if any, of these equations with respect to x_1 may potentially be useful. We note that the highest-order partial derivative with respect to x_1 in all other parent nodes (i.e., node $\{1\}$ in this case) is 1. Also, the lowest-order partial derivative with respect to x_1 in node $\{2\}$ is 0. Therefore, $m^*_{\{1^*,2\}} = 1 - 0 = 1$, which means that we have to consider the first partial derivatives of the node equations (39), (41), and (42) and the consistency condition (45) with respect to x_1, yielding

$$\begin{bmatrix} 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \frac{\partial u}{\partial x_1} \\ \frac{\partial^2 u}{\partial x_1 \partial x_2} \\ \frac{\partial^3 u}{\partial x_1 \partial x_2^2} \\ \frac{\partial^2 u}{\partial x_1^2} \end{pmatrix} = \begin{bmatrix} 0 \\ h'(x_1) \\ k'(x_1) \\ h''(x_1) \end{bmatrix} \quad \forall x_1. \tag{53}$$

6. Redundancy and inconsistency analysis at domain digraph nodes. Section 4 established an algorithm for the determination of the consistency conditions $\mathcal{C}_{\beta\alpha}$ that are associated with any arc in the domain digraph for a given PDE system. Once these consistency equations are determined for all arcs $\beta \to \alpha$ entering a particular node α of the domain digraph, a set of equations describing node α may be assembled from the equations $\mathcal{F}_\beta$ describing all parent nodes $\beta \in \mathcal{P}_\alpha$ and the consistency conditions $\mathcal{C}_{\beta\alpha}$. Moreover, as described in section 5, the first $m^*_{\beta\alpha}$ partial derivatives of $\mathcal{F}_\beta$ and $\mathcal{C}_{\beta\alpha}$ with respect to independent variable $x_{\beta\alpha}$ may also need to be considered.

All of the above may be assembled into a single set of linear equations of the form

$$\mathbf{D}\mathbf{v} = \mathbf{f}, \tag{54}$$

where $\mathbf{D}$ is a matrix whose elements may, in general, be operator-valued, $\mathbf{v}$ are the DDVs for node α (cf. section 2.3), and $\mathbf{f}$ is a right-hand side vector. Both $\mathbf{v}$ and $\mathbf{f}$ are functions of those independent variables which are not fixed at node α. We can then apply Gaussian elimination with row and column pivoting to the above system to bring it to the form

$$\begin{bmatrix} \tilde{\mathbf{D}}^{[1]} \\ \mathbf{0}^{[1]} \\ \mathbf{0}^{[2]} \end{bmatrix} \tilde{\mathbf{v}} = \begin{bmatrix} \tilde{\mathbf{f}}^{[1]} \\ \tilde{\mathbf{f}}^{[2]} \\ \mathbf{0} \end{bmatrix}, \tag{55}$$

where $\tilde{\mathbf{D}}^{[\mathbf{1}]}$ is an upper triangular matrix with nonzero diagonal elements and $\tilde{\mathbf{v}}$ represents a reordering of $\mathbf{v}$ reflecting any column pivoting operations performed during the Gaussian elimination.

Equation (55) involves three block rows. The last block row, namely $\mathbf{0}^{[2]}\tilde{\mathbf{v}} = \mathbf{0}$, corresponds to redundant equations. In general, this redundancy arises because all parent nodes β of node α are derived from the same root node (i.e., the one corresponding to the original PDAE system), which leads to some overlap among the equations describing these nodes β. Also, the differentiations considered in section 5 may introduce some additional redundancy arising from overlaps between, on one hand, the partial derivatives of $\mathcal{F}_\beta$ with respect to $x_{\beta\alpha}$ and, on the other hand, the consistency conditions $\mathcal{C}_{\beta\alpha}$ and their partial derivatives with respect to $x_{\beta\alpha}$. This is inevitable since $\mathcal{C}_{\beta\alpha}$ are themselves derived from $\mathcal{F}_\beta$ by differentiation with respect to $x_{\beta\alpha}$ (cf. section 4). In any case, the equations corresponding to these zero rows convey no useful information and can simply be discarded at this stage.

The middle block row of (55), namely $\mathbf{0}^{[1]}\tilde{\mathbf{v}} = \tilde{\mathbf{f}}^{[2]}$, corresponds to potential inconsistencies. Assuming that the original PDAE system was well-posed (therefore involving a consistent set of equations), such inconsistencies may have arisen only from specifications imposed at different parent nodes, or earlier ancestors of node α. The only possible way in which these potential inconsistencies can be avoided is by ensuring that the *specification consistency relations* $\tilde{\mathbf{f}}^{[2]} = 0$ are satisfied.

Assuming that specifications at earlier nodes can be adjusted to satisfy the SCRs, then the first block row of (55), namely $\tilde{\mathbf{D}}^{[\mathbf{1}]}\tilde{\mathbf{v}} = \tilde{\mathbf{f}}^{[1]}$, provides a set of linearly independent equations that can be used for the determination of the DDVs $\mathbf{v}$. In most cases, the number of these equations will be smaller than that of the DDVs, and, consequently, they will have to be supplemented by additional specification equations $\mathcal{S}_\alpha$ imposed at the current node α.

The specifications $\mathcal{S}_\alpha$ will have to be linearly independent both of each other and with respect to the rows of $\tilde{\mathbf{D}}^{[\mathbf{1}]}$. From the mathematical point of view, the simplest way of ensuring this is to specify the last $n^v_\alpha - n^D_\alpha$ elements of the vector $\tilde{\mathbf{v}}$ as given functions of the independent variables that are not fixed at node α, where n^v_α is the number of DDVs $\mathbf{v}$ and n^D_α is the number of rows in $\tilde{\mathbf{D}}^{[\mathbf{1}]}$ (i.e., the rank of the original matrix $\mathbf{D}$). However, such a choice of specifications may not be the most appropriate or convenient from the physical point of view. Instead, it may be desirable to specify some other elements of the vector $\mathbf{v}$ or, indeed, to impose specifications involving linear relations among two or more elements of this vector. In such cases, one will have to augment the system $\tilde{\mathbf{D}}^{[\mathbf{1}]}\tilde{\mathbf{v}} = \tilde{\mathbf{f}}^{[1]}$ with these $n^v_\alpha - n^D_\alpha$ proposed specification equations and then perform the necessary Gaussian elimination on the corresponding rows to ensure that linear independence has been preserved.

For illustration purposes, consider once more the heat conduction equation (39) introduced in section 5. Here, the system (54) for the corner node $\{1, 2\}$ comprises

- from parent node $\{1\}$, the heat conduction equation (39) and the specification (40);
- from arc $\{1, 2^\star\}$, the consistency conditions (43) and (44);
- from parent node $\{2\}$, the heat conduction equation (39) and the specifications (41) and (42);
- from arc $\{1^\star, 2\}$, the consistency condition (45);
- the potentially constraining differentiated equations (51)–(53).

The vector of DDVs is given by

$$
\mathbf{v}_{\{1,2\}} \equiv \begin{pmatrix} u(x_1^\star, x_2^\star) \\ \frac{\partial u}{\partial x_1}(x_1^\star, x_2^\star) \\ \frac{\partial u}{\partial x_2}(x_1^\star, x_2^\star) \\ \frac{\partial^2 u}{\partial x_2^2}(x_1^\star, x_2^\star) \\ \frac{\partial^2 u}{\partial x_1 \partial x_2}(x_1^\star, x_2^\star) \\ \frac{\partial^3 u}{\partial x_2^3}(x_1^\star, x_2^\star) \\ \frac{\partial^3 u}{\partial x_1 \partial x_2^2}(x_1^\star, x_2^\star) \\ \frac{\partial^4 u}{\partial x_2^4}(x_1^\star, x_2^\star) \\ \frac{\partial^2 u}{\partial x_1^2}(x_1^\star, x_2^\star) \end{pmatrix}. \tag{56}
$$

The overall system is given by 20 linear equations:

$$
\begin{bmatrix}
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}
\begin{pmatrix} u(x_1^\star, x_2^\star) \\ \frac{\partial u}{\partial x_1}(x_1^\star, x_2^\star) \\ \frac{\partial u}{\partial x_2}(x_1^\star, x_2^\star) \\ \frac{\partial^2 u}{\partial x_2^2}(x_1^\star, x_2^\star) \\ \frac{\partial^2 u}{\partial x_1 \partial x_2}(x_1^\star, x_2^\star) \\ \frac{\partial^3 u}{\partial x_2^3}(x_1^\star, x_2^\star) \\ \frac{\partial^3 u}{\partial x_1 \partial x_2^2}(x_1^\star, x_2^\star) \\ \frac{\partial^4 u}{\partial x_2^4}(x_1^\star, x_2^\star) \\ \frac{\partial^2 u}{\partial x_1^2}(x_1^\star, x_2^\star) \end{pmatrix}
=
\begin{pmatrix} 0 \\ g(x_2^\star) \\ g'(x_2^\star) \\ g''(x_2^\star) \\ 0 \\ h(x_1^\star) \\ k(x_1^\star) \\ h'(x_1^\star) \\ 0 \\ g'(x_2^\star) \\ g''(x_2^\star) \\ g'''(x_2^\star) \\ 0 \\ g''(x_2^\star) \\ g'''(x_2^\star) \\ g''''(x_2^\star) \\ 0 \\ h'(x_1^\star) \\ k'(x_1^\star) \\ h''(x_1^\star) \end{pmatrix}. \tag{57}
$$

We now perform Gaussian elimination on the above system, bringing it to the form

$$
\left[\begin{array}{ccccccccc}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\mathbf{v} =
\left[\begin{array}{c}
g(x_2^\star) \\ 0 \\ g'(x_2^\star) \\ g''(x_2^\star) \\ 0 \\ g'''(x_2^\star) \\ 0 \\ g''''(x_2^\star) \\ -g''''(x_2^\star) \\
\hdashline
h'(x_1^\star) - g''(x_2^\star) \\ h(x_1^\star) - g(x_2^\star) \\ k(x_1^\star) - g'(x_2^\star) \\ k'(x_1^\star) - g'''(x_2^\star) \\ h''(x_1^\star) - g''''(x_2^\star) \\
\hdashline
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{array}\right], \tag{58}
$$

which is consistent with the general form of (55). In particular, the last block row indicates that there are six redundant equations. This is simply a consequence of the fact that the original equation (39) was included twice in system (57) since it forms part of the equations that describe both node $\{1\}$ and node $\{2\}$. Moreover, the differentiated systems (51)–(53) introduce another five redundant equations.

The second block row of (58) indicates that there is potential for inconsistency between the specifications (40), (41), and (42) imposed at the parent nodes $\{1\}$ and $\{2\}$, respectively. Thus, corner singularities will arise if the choice of the mathematical functions $g(x_2)$, $h(x_1)$, and $k(x_1)$ fails to satisfy one or more of the following relations, obtained by setting the corresponding elements of the right-hand side vector in this block row to zero:

$$h'(x_1^\star) - g''(x_2^\star) = 0, \tag{59}$$

$$h(x_1^\star) - g(x_2^\star) = 0, \tag{60}$$

$$k(x_1^\star) - g'(x_2^\star) = 0, \tag{61}$$

$$k'(x_1^\star) - g'''(x_2^\star) = 0, \tag{62}$$

$$h''(x_1^\star) - g''''(x_2^\star) = 0. \tag{63}$$

7. Summary and conclusions. This paper has presented an algorithm for the determination of consistent variations of the dependent variables over the boundary of a linear PDAE system.

The algorithm is applicable to systems involving any number of independent variables. By making use of the domain digraph introduced in section 2.1, it systematically considers all types of the boundary, starting with the higher-dimensional ones, and moving towards lower-dimensional ones until it reaches a single point.

The first major component of the algorithm involves the determination of any "hidden" consistency conditions that are associated with each arc in the domain digraph (cf. section 4). The linearity of the systems considered allows, in principle, the application of standard algorithms for the analysis of DAE initial conditions for this purpose. Here we make use of the extended form of these algorithms (see Neumann [5]), which makes them directly applicable to higher-order systems without having to reformulate them to first-order ones. The main difference between the algorithms for DAE and PDAE systems in this context is the need for symbolic operator-based Gaussian elimination in the latter case.

The second major component of our algorithm (cf. section 6) is the formal identification of specification consistency relations which need to be satisfied by the specifications ("boundary conditions") imposed on different parts of the domain boundary. Some SCRs can be identified only by considering simultaneously differentiations of different equations with respect to different independent variables (cf. section 5). This kind of complication is unique to PDAE systems and cannot be handled by simple extension of DAE consistent initialization algorithms. In any case, failure to satisfy SCRs gives rise to corner singularities, and our algorithm provides a systematic approach to identifying the latter for general PDAE systems.

A limitation of the analysis presented in this paper is that it does not distinguish between different instances of the same node in the domain digraph. For example, in a 3-dimensional domain, there are actually two faces corresponding to each of the three level-1 nodes. Hence, in general, it is not possible to consider problems in which some of the boundary conditions are imposed on, say, face (x_1^*, x_2, x_3), while others are imposed on face (x_1^{**}, x_2, x_3) where $x_1^* \neq x_1^{**}$.

Acknowledgment. The authors wish to thank an anonymous referee whose counterexample to their original algorithm led them to derive the additional algorithmic steps presented in section 5 of this paper.

REFERENCES

[1] N. Flyer and B. Fornberg, *Accurate numerical resolution of transients in initial-boundary value problems for the heat equation*, J. Comput. Phys., 184 (2003), pp. 526–539.

[2] C. W. Gear and L. R. Petzold, *ODE methods for the solution of differential/algebraic systems*, SIAM J. Numer. Anal., 21 (1984), pp. 716–728.

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1996.

[4] W. S. Martinson and P. I. Barton, *A differentiation index for partial differential-algebraic equations*, SIAM J. Sci. Comput., 21 (2000), pp. 2295–2315.

[5] J. Neumann, *Consistency Analysis of Systems of Partial and Ordinary Differential and Algebraic Equations*, Ph.D. thesis, Imperial College, London, 2004.

[6] C. C. Pantelides, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 213–231.