

Differential Equivalence for Linear Differential Algebraic Equations

Questa è la versione sottoposta a revisione paritaria (postprint) della seguente opera:

Original

Differential Equivalence for Linear Differential Algebraic Equations / Tognazzi, S.; Tribastone, M.; Tschaikowski, M.; Vandin, A.. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 0018-9286. - 67:7(2022), pp. 3484-3493. [10.1109/TAC.2021.3108530]

Availability:

This version is available at: 20.500.11771/21359

Publisher:

Published

DOI:10.1109/TAC.2021.3108530

Terms of use:

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. (https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

Differential Equivalence for Linear Differential Algebraic Equations

Stefano Tognazzi¹, Mirco Tribastone², Max Tschaikowski³ and Andrea Vandin⁴

Abstract—Differential-algebraic equations (DAEs) are a widespread dynamical model that describes continuously evolving quantities defined with differential equations, subject to constraints expressed through algebraic relationships. As such, DAEs arise in many fields ranging from physics, chemistry, and engineering. In this paper we focus on linear DAEs, and develop a theory for their minimization up to an equivalence relation. We present differential equivalence, which relates DAE variables that have equal solutions at all time points (thus requiring them to start with equal initial conditions) and extends the line of research on bisimulations developed for Markov chains and ordinary differential equations. We apply our results to the electrical engineering domain, showing that differential equivalence can explain invariances in certain networks as well as analyze DAEs which could not be originally treated due to their size.

I. INTRODUCTION

Differential-algebraic equations (DAEs) are a popular model of dynamical systems across many branches of science and engineering. They often arise when describing physical quantities that evolve continuously and deterministically according to an ordinary differential equation (ODE), together with constraints that express properties such as conservation of energy. An example is *RLC* electric circuits, where components such as capacitors and inductors are associated with first-order linear differential equations while the algebraic constraints impose Kirchhoff's conservation laws for voltages in closed loops and currents at each junction [1]. A more general approach that leads to DAEs as the underlying dynamics is that of *bond graphs* (e.g., [2]), a unifying model for networked physical systems such as electrical, mechanical, and hydraulic networks.

Solving DAEs is difficult [3]. Compared to ODEs, the initial condition may not be chosen arbitrarily (provided that the solution exists), but it has to be *consistent*, i.e., it has to satisfy the algebraic constraints. This leads to expensive implicit schemes that require the solution of a system of equations at each time step, thus motivating a large body of literature on model reduction techniques for DAEs (cf. [4], [5], [6] and references therein). Such techniques are appealing for numerical purposes since they were proven to be effective for reducing models of real large-scale systems. However, the reduction is approximate in general and the reduced model may not carry

physical intelligibility (because its variables represent a linear transformation of the original state space). This limits their use when the modeler strives to gain mechanistic insights from the model, for example to explain symmetric behavior in a design [7], [8].

In this paper we present a technique for reducing linear DAE systems which is both exact and *structure-preserving*, in the sense that the reduction is obtained through a quotienting up to an equivalence that relates variables that have the same solutions at all time points. Our line of research can be related to aggregation of Markov chains [9], [10] and differential equivalence (DE) for ODE systems [11]. The notion of DE can be naturally extended to linear DAEs by working on an equivalent linear ODE system into which a linear DAE system can be transformed (see Section III). Unfortunately, such a transformation requires the solution of an ill-conditioned problem [12], [13].

We circumvent this issue by developing in Section IV a criterion for differential equivalence over the *numerical solution scheme* of the DAE system, instead of the DAE system itself. The idea is to find a *numeric DE* by relating numerical solutions of variables that are equal at all steps, through inspection of the explicit update law of the numerical solution. Using the popular backward differentiation formulae as the numerical solver [14], we prove that a numeric DE is a DE of the original DAE system, independently of the discretization time step used in the numerical solver. Moreover, we show that computing the coarsest numeric DE through a partition refinement algorithm reduces to computing the coarsest DE of a linear ODE system that is related to the explicit update law of the numerical solver (hence, it can be computed in polynomial time [15], [16]).

We apply our theory to benchmark DAEs from electric circuit theory by showing that numeric DE is i) effective in discovering symmetries in the topology of the electrical networks and ii) allows one to analyze circuits that would otherwise issue out-of-memory errors.

In the special case of semi-explicit linear DAE systems of index one, it is well-known that index reduction can be accomplished by inverting the matrix describing the algebraic constraints in question. Unfortunately, matrix inversion may suffer from numerical errors in case of large-scale systems and miss symmetries that would be otherwise present. We address this by presenting a novel sufficient condition for DE that can be computed efficiently and that does not require matrix inversion. The scalability of the approach is demonstrated on large-scale models for which the computation of numeric GE causes out-of-memory errors. We also perform an empirical study on the speedup obtainable by combining our approach with the state-of-the-art modeling language OpenModelica [17] and demonstrate runtime gains of up to 3 orders of magnitude.

The present work extends [18] by the aforementioned results

¹ Stefano Tognazzi is with University of Konstanz, Germany: stefano.tognazzi@uni-konstanz.de

² Mirco Tribastone is with IMT Lucca, Italy: mirco.tribastone@imtlucca.it

³ Max Tschaikowski is with Aalborg University, Denmark: tschaikowski@cs.aau.dk

⁴ Andrea Vandin is with Sant'Anna School for Advanced Studies, Pisa, Italy and DTU Technical University of Denmark: andrea.vandin@santannapisa.it

This work was supported in part by the PRIN project SEDUCE, no. 2017TWRNCB, the Poul Due Jensen Foundation grant no. 883901, and the DFF RP1 project REDUCTO no. 9040-00224B.

for semi-explicit linear DAE systems and the corresponding experiments, including the empirical study of OpenModelica. Further, the proofs of all statements are provided.

Paper outline. Section II sets the scene by presenting background knowledge on DE and linear DAEs. Building on that, Section III introduces GDE, while Section IV introduces numeric GDE and an efficient algorithm for its computation. Instead, Section V devises an efficiently computable sufficient condition for GDE, while Section VI presents a thorough evaluation of numeric GDE. To this end, we apply it to benchmark models from practice, carry out a scalability analysis of the aforementioned sufficient condition and study the speedup obtainable by combining our approach with the state-of-the-art modeling language OpenModelica. After providing further discussion on related work in Section VII, the paper concludes in Section VIII.

II. PRELIMINARIES

Notation. The derivative of variable x w.r.t. variable y is denoted by $\partial_y x$. The set of variables is given by $\{x_i \mid i \in \mathcal{S}\}$, where \mathcal{S} is some finite index set; partitions of \mathcal{S} are denoted by \mathcal{H} and \mathcal{G} . A partition \mathcal{H} of \mathcal{S} refines a partition \mathcal{G} if and only if each block of \mathcal{H} is a subset of some block of \mathcal{G} . We write B^A for the set of functions from A to B . Following the standard notation, \mathbb{C} and x^T denote the complex numbers and the transpose of a vector $x \in \mathbb{C}^{\mathcal{S}}$, respectively; instead, for an $i \in \mathcal{S}$, $e_i \in \mathbb{R}^{\mathcal{S}}$ is such that $e_i(j) = 1$ if $i = j$ and 0 otherwise. For a given function $f : A \rightarrow B$ and a set $C \subseteq A$, $f(C)$ defines the image of f under C , i.e., $f(C) = \{f(x) \mid x \in C\}$. The subspace spanned by the columns of $C \in \mathbb{R}^{\mathcal{S} \times \mathcal{S}}$ is denoted by $\langle C \rangle$.

Differential Equivalence. The following notion from [11] is used to study a family of positive forward invariant sets [19].

Definition 1. Fix a linear ODE system $\partial_t x = Ax + b$ with $A \in \mathbb{R}^{\mathcal{S} \times \mathcal{S}}$ and $b \in \mathbb{R}^{\mathcal{S}}$.¹

- For a partition \mathcal{H} of \mathcal{S} , we define the algebraic variety $\mathcal{U}_{\mathcal{H}} = \{x \in \mathbb{R}^{\mathcal{S}} \mid x_i - x_j = 0 \text{ for all } H \in \mathcal{H}, i, j \in H\}$.
- A partition \mathcal{H} of \mathcal{S} is a differential equivalence (DE) if $x(0) \in \mathcal{U}_{\mathcal{H}}$ and $\partial_t x = Ax + b$ implies $x(t) \in \mathcal{U}_{\mathcal{H}}$ for all $t > 0$.

DE enjoys a geometric characterization in terms of positive forward invariant sets as stated next.

Theorem 1 (Geometric Differential Equivalence). *Given an ODE system $\partial_t x = Ax + b$, a partition \mathcal{H} of \mathcal{S} is called geometric differential equivalence (GDE) if $A(\mathcal{U}_{\mathcal{H}}) + b \subseteq \mathcal{U}_{\mathcal{H}}$. In particular, \mathcal{H} is a DE if and only if \mathcal{H} is a GDE.*

Proof: See [20, Theorem 3] or [19, Section 10]. ■

The notion of DE allows for the following model reduction.

Theorem 2. *Given an ODE system $\partial_t x = Ax + b$, let $\mathcal{H} = \{H_1, \dots, H_m\}$ be a DE partition.*

- For $H \in \mathcal{H}$, let $i_H \in H$ be the representative of H and $S_0 = \{i_H \mid H \in \mathcal{H}\}$. Set further $e_H \in \mathbb{R}^{\mathcal{S}}$ by $e_H(i) = 1$

if $i \in H$ and zero otherwise. Then, the reduced ODE system $\partial_t \hat{x} = \hat{A}\hat{x} + \hat{b}$ is given by $\hat{A} = S_l A S_r$, $\hat{b} = S_l b$, where $\hat{A} \in \mathbb{R}^{S_0 \times S_0}$ and $\hat{b} \in \mathbb{R}^{S_0}$ and the transformation matrices $S_l \in \mathbb{R}^{S_0 \times \mathcal{S}}$, $S_r \in \mathbb{R}^{\mathcal{S} \times S_0}$ are defined via $S_l = (e_{i_{H_1}}, \dots, e_{i_{H_m}})^T$ and $S_r = (e_{H_1}, \dots, e_{H_m})$.

- The reduced model $\partial_t \hat{x} = \hat{A}\hat{x} + \hat{b}$ can be computed in polynomial time. Moreover, for any $x(0) \in \mathcal{U}_{\mathcal{H}}$, the solution of $\partial_t x = Ax + b$ is uniquely related to the solution of $\partial_t \hat{x} = \hat{A}\hat{x} + \hat{b}$ via $\hat{x}(0) = S_l x(0)$. In particular, $\hat{x}(t) = S_l x(t)$ and $x(t) = S_r \hat{x}(t)$ for all $t \geq 0$.

Proof: See [16]. ■

The coarser a GDE partition \mathcal{H} is (i.e., the fewer blocks it has), the smaller is the reduced system induced by \mathcal{H} . For a given partition \mathcal{G} , this motivates to ask whether there exists a coarsest GDE partition \mathcal{H} that refines some arbitrary partition \mathcal{G} . The next result ensures that this is indeed the case. In fact, the next result states more. By building upon efficient algorithmic approaches, it states that the coarsest GDE partition exists and that it can be computed in polynomial time.

Theorem 3. *Fix a linear ODE system $\partial_t x = Ax + b$ and let \mathcal{G} be some partition of \mathcal{S} . Then, there exists a coarsest GDE \mathcal{H} partition that refines \mathcal{G} and it can be computed in polynomial time (following common terminology, a partition \mathcal{H} refines a partition \mathcal{G} when for any block $H \in \mathcal{H}$, there is a block $G \in \mathcal{G}$ such that $H \subseteq G$).*

Proof: See [16]. ■

For instance, $\{\{1, 2\}, \{3\}\}$ is a GDE for the ODE system $\partial_t x = Ax$, where A is given below. Matrix $\hat{A} = S_l A S_r$ describing the underlying reduced ODE system $\partial_t \hat{x} = \hat{A}\hat{x}$ is provided as well.

$$A = \begin{pmatrix} 1 & 2 & 5 \\ 2 & 1 & 5 \\ 3 & 4 & 7 \end{pmatrix}, \quad \hat{A} = \begin{pmatrix} 3 & 5 \\ 7 & 7 \end{pmatrix},$$

$$S_l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad S_r = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

It can be shown that the coarsest GDE partition that refines $\{\{1, 2, 3\}\}$ is $\{\{1, 2\}, \{3\}\}$. This trivially implies that the coarsest GDE partition that refines $\{\{1, 2\}, \{3\}\}$ is $\{\{1, 2\}, \{3\}\}$ itself. Instead, the coarsest GDE partition that refines, e.g., $\{\{1\}, \{2, 3\}\}$ is the trivial GDE partition $\{\{1\}, \{2\}, \{3\}\}$ (the partition consisting of singleton blocks is always a GDE).

Linear DAEs. Systems of linear differential algebraic equations (DAEs) satisfy the form $E\partial_t x = Ax + b$ where $E, A \in \mathbb{R}^{\mathcal{S} \times \mathcal{S}}$ and $b \in \mathbb{R}^{\mathcal{S}}$. In case E is invertible, a linear DAE system can be directly recast into a linear ODE system via $\partial_t x = E^{-1}Ax + E^{-1}b$. If E is not invertible, however, the transformation corresponds to the computation of the Kronecker normal form, an instance of index reduction [14], [21], which is ill-conditioned in general [13], unless additional assumptions are imposed [12].

We summarize several facts from the theory of DAEs [14].

Definition 2. *A DAE system $E\partial_t x = Ax + b$ is called regular if, for every initial condition $x(0) \in \mathbb{R}^{\mathcal{S}}$, there exists either a unique solution or no solution at all.*

¹Sometimes, ODE systems of the form $\partial_t x = Ax + b$ are also called affine or linear nonhomogeneous.

Theorem 4. A DAE system $E\partial_t x = Ax + b$ is regular if and only if $A - \lambda E$ is invertible for at least one $\lambda \in \mathbb{C}$. The set of initial conditions for which a regular DAE admits solutions, \mathcal{D} , is an affine subspace of \mathbb{R}^S . For any regular DAE system $E\partial_t x = Ax + b$, there exist $\hat{A} \in \mathbb{R}^{S \times S}$ and $\hat{b} \in \mathbb{R}^S$ such that for any $x(0) \in \mathcal{D}$, the solutions of $\partial_t \hat{x} = \hat{A}\hat{x} + \hat{b}$ and $E\partial_t x = Ax + b$ coincide when subject to $x(0) = \hat{x}(0)$.

Proof: See Chapter 2 in [14]. ■

The above results ensure that any linear DAE system $E\partial_t x = Ax + b$ can be expressed as a linear ODE system $\partial_t x = \hat{A}x + \hat{b}$. As mentioned before, the computation of \hat{A} , \hat{b} and the domain \mathcal{D} relies on the Kronecker normal form [12], [13].

Example 1. Consider the DAE system $E\partial_t x = Ax$ where

$$E = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 2 \\ -3 & 0 & -9 \end{pmatrix}, \quad A = \begin{pmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 2 & -2 \end{pmatrix} \quad (1)$$

Then, it can be proven that $E\partial_t x = Ax$ admits a unique solution only when $x(0) \in \mathcal{D} = \langle (1, 0, 0)^T, (0, -2, 1)^T \rangle$, where $\langle S \rangle$ denotes the vector space spanned by the vectors in S . Moreover, the corresponding solution satisfies $\partial_t x = \hat{A}x$, where

$$\hat{A} = \begin{pmatrix} 3 & \xi & 2(\xi + 1) \\ 2 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \quad (2)$$

for any number ξ . Specifically, matrix \hat{A} is not unique.

III. DIFFERENTIAL EQUIVALENCE FOR DIFFERENTIAL ALGEBRAIC EQUATIONS

We start by lifting the notion of DE to linear DAEs.

Definition 3. Let $E\partial_t x = Ax + b$ be a regular DAE. A partition \mathcal{H} of S is called a DE of $E\partial_t x = Ax + b$ if $\mathcal{U}_{\mathcal{H}} \cap \mathcal{D} \neq \emptyset$ and, for any initial condition $x(0) \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$, it holds that $x(t) \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$ for all $t > 0$.

In contrast to the ODE case, we now have to account for the domain \mathcal{D} . This is because only initial conditions $x(0)$ from \mathcal{D} induce a unique solution of $E\partial_t x = Ax + b$. Moreover, it already suffices to ask for $x(t) \in \mathcal{U}_{\mathcal{H}}$ for all $t > 0$ instead of $x(t) \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$ for all $t > 0$. This is because any solution x of a regular DAE system $E\partial_t x = Ax + b$ has to be contained in \mathcal{D} , see Theorem 4.

Example 2. Consider the running example (1) and let $\mathcal{H} = \{\{1, 2\}, \{3\}\}$. With this, for any initial condition $(2\alpha, 2\alpha, -\alpha) \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D} = \langle (2, 2, -1)^T \rangle$, the solution of the DAE system is given by $x(t) = e^{2t}(2\alpha, 2\alpha, -\alpha)$, where \mathcal{D} is as in Example 1. Overall, \mathcal{H} is a DE of (1).

Recall that any linear DAE $E\partial_t x = Ax + b$ can be expressed in terms of some ODE system $\partial_t x = \hat{A}x + \hat{b}$. Since any such ODE system is only uniquely determined on the domain \mathcal{D} , a lifting of GDE to DAEs has so to take \mathcal{D} into account. The next example demonstrates this.

Example 3. Consider example (1). From Example 2, we know that $\mathcal{H} = \{\{1, 2\}, \{3\}\}$ is a DE of $E\partial_t x = Ax + b$.

- Then $\hat{A}(\mathcal{U}_{\mathcal{H}}) \not\subseteq \mathcal{U}_{\mathcal{H}}$ if \hat{A} is as in (2) with $\xi = 0$.
- Then $\hat{A}(\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$ if \hat{A} is as in (2) with $\xi = -1$.

Regardless of the value of ξ in (2), however, it holds true that $\hat{A}(\mathcal{U}_{\mathcal{H}} \cap \mathcal{D}) \subseteq \mathcal{U}_{\mathcal{H}}$.

Motivated by the above example, GDE is lifted to the domain of DAEs as follows.

Definition 4. Let $E\partial_t x = Ax + b$ be a regular DAE and let $\partial_t x = \hat{A}x + \hat{b}$ be a corresponding ODE system. We call \mathcal{H} geometric differential equivalence (GDE) of $E\partial_t x = Ax + b$ if $\hat{A}(\mathcal{U}_{\mathcal{H}} \cap \mathcal{D}) + \hat{b} \subseteq \mathcal{U}_{\mathcal{H}}$ and $\mathcal{U}_{\mathcal{H}} \cap \mathcal{D} \neq \emptyset$.

The following generalization of Theorem 1 states that GDE characterizes DE on the domain of DAEs.

Theorem 5. Let $E\partial_t x = Ax + b$ be a regular DAE and let $\partial_t x = \hat{A}x + \hat{b}$ be an underlying ODE system. A partition \mathcal{H} is a DE if and only if it is a GDE.

Proof: Let us assume that \mathcal{H} is a GDE, i.e., we have that $\hat{A}(\mathcal{U}_{\mathcal{H}} \cap \mathcal{D}) + \hat{b} \subseteq \mathcal{U}_{\mathcal{H}}$. Thanks to Theorem 4, this implies $\hat{A}(\mathcal{U}_{\mathcal{H}} \cap \mathcal{D}) + \hat{b} \subseteq \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$. Moreover, it holds that $\partial_t x = \hat{A}x + \hat{b}$. Hence, for any $x(0) \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$, we conclude that the Euler method, if applied to $\partial_t x = \hat{A}x + \hat{b}$, yields a numerical solution that remains in $\mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$. By letting the time step of the Euler method converge to zero, we observe that the same applies to the true solution x . Let us now assume that \mathcal{H} is a DE. Then, for any $x(0) \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$, the solution x remains in $\mathcal{U}_{\mathcal{H}}$. Hence, $\frac{1}{\Delta t}(x(0 + \Delta t) - x(0)) \in \mathcal{U}_{\mathcal{H}}$ for any $\Delta t > 0$ because $\mathcal{U}_{\mathcal{H}}$ is a linear subspace. By considering $\Delta t \rightarrow 0$ and noting that x satisfies the ODE system $\partial_t x = \hat{A}x + \hat{b}$, we obtain $\hat{A}(x(0)) + \hat{b} \in \mathcal{U}_{\mathcal{H}}$. ■

Note that the constraint $\mathcal{U}_{\mathcal{H}} \cap \mathcal{D} \neq \emptyset$ in Definition 4 ensures that there actually exist initial conditions to which a GDE partition can be applied.

We proceed by providing the notion of a reduced model underlying a GDE.

Definition 5. For a regular DAE system $E\partial_t x = Ax + b$, let $\partial_t x = \hat{A}x + \hat{b}$ be an underlying ODE system and let $\mathcal{H} = \{H_1, \dots, H_m\}$ be a DE partition. Let $\mathcal{S}_0, \{i_H \mid H \in \mathcal{H}\}, S_l$ and S_r be as in Theorem 2. Then, the reduced DAE system $\underline{E}\partial_t \underline{x} = \underline{A}\underline{x} + \underline{b}$ and the reduced ODE system $\partial_t \underline{x} = \underline{\hat{A}}\underline{x} + \underline{\hat{b}}$ are given by $\underline{E} = S_l E S_r$, $\underline{A} = S_l A S_r$, $\underline{b} = S_l b$, $\underline{\hat{A}} = S_l \hat{A} S_r$, $\underline{\hat{b}} = S_l \hat{b}$ and the reduced domain $\underline{\mathcal{D}} = \{S_l x \mid x \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}\}$.

The following can be shown.

Theorem 6. Given a regular DAE system $E\partial_t x = Ax + b$ and let $\partial_t x = \hat{A}x + \hat{b}$ be a corresponding ODE system. Then, for a DE \mathcal{H} , the reduced models $\underline{E}\partial_t \underline{x} = \underline{A}\underline{x} + \underline{b}$ and $\partial_t \underline{x} = \underline{\hat{A}}\underline{x} + \underline{\hat{b}}$ can be computed in polynomial time. Moreover, for any $x(0) \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$, the solution of $E\partial_t x = Ax + b$ is uniquely related to the solution of $\underline{E}\partial_t \underline{x} = \underline{A}\underline{x} + \underline{b}$ and $\partial_t \underline{x} = \underline{\hat{A}}\underline{x} + \underline{\hat{b}}$ via $\underline{x}(0) = S_l x(0)$. In particular, $\underline{x}(t) = S_l x(t)$ and $x(t) = S_r \underline{x}(t)$ for all $t \geq 0$.

Proof: Follows immediately from the definition of DE and Theorem 4. ■

Example 4. From Example 2, we know that $\mathcal{H} = \{H_1, H_2\} = \{\{1, 2\}, \{3\}\}$ is a DE of (1). Together with $i_{H_1} = 1$ and

$i_{H_2} = 3$, we conclude that

$$\underline{E} = \begin{pmatrix} 0 & 1 \\ -3 & -9 \end{pmatrix} \quad \underline{A} = \begin{pmatrix} -1 & 0 \\ 2 & -2 \end{pmatrix} \quad \hat{\underline{A}} = \begin{pmatrix} 3 & 2 \\ -1 & 0 \end{pmatrix}$$

$$S_l = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad S_r = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \underline{\mathcal{D}} = \langle \begin{pmatrix} 2 \\ -1 \end{pmatrix} \rangle$$

Since the solution of the running example is given by $x(t) = e^{2t}(2\alpha, 2\alpha, -\alpha)^T$ for all $(2\alpha, 2\alpha, -\alpha)^T \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$, it holds that $S_l x(t) = e^{2t}(2\alpha, -\alpha)^T = \underline{x}(t)$. A direct computation confirms that \underline{x} solves $\underline{E}\underline{x} = \underline{A}\underline{x}$ and $\underline{x} = \hat{\underline{A}}\underline{x}$ for all $\underline{x}(0) = (2\alpha, -\alpha)^T \in \underline{\mathcal{D}}$. Moreover, $x(t) = S_r \underline{x}(t)$ for all $\underline{x}(0) \in \underline{\mathcal{D}}$.

IV. NUMERIC GEOMETRIC DIFFERENTIAL EQUIVALENCE

While GDE extends to linear DAEs, finding GDE partitions hinges on the availability of \mathcal{D} . Since the computation of \mathcal{D} relies on the solution of an ill-conditioned problem [13], this section introduces numeric GDE.

Backward Differentiation Formulae. A popular family of numerical schemes to solve linear DAE systems are the so-called backward differential formulae (BDF) [22]. The general idea is, as in the case of all numerical schemes, to approximate the true solution of a given linear DAE system $E\partial_t x = Ax + b$ on an interval $[0; T]$ by a sequence of points $x[0], x[1], \dots, x[M]$ such that $x[i]$ approximates $x(t_i)$, where $\Delta t = T/M$ and $t_i = i\Delta t$ for all $0 \leq i \leq M$.

The next result [14, Theorem 5.24] ensures 1) that the BDF solution $(x[i])_i$ is well-defined if the time step Δt is sufficiently small and; 2) that the numeric solution $(x[i])_i$ converges to the solution of $E\partial_t x = Ax + b$ if $x(0) = x[0] \in D$.

Theorem 7. *Given a regular linear DAE system $E\partial_t x = Ax + b$ and an initial condition $x(0) \in \mathbb{R}^S$, fix $1 \leq k \leq 6$ and set $x[-6] = x[-5] = \dots = x[-1] = 0$. Then, there exist $\Delta t_0 > 0$ and $C > 0$ that do not depend on $\Delta t > 0$ and which satisfy the following.*

1) *For all $0 < \Delta t \leq \Delta t_0$ there exists a unique $x[i]$ solving the linear system of equations*

$$(A - \frac{\alpha(k,0)}{\Delta t} E)x[i] = E \left(\sum_{l=1}^k \frac{\alpha(k,l)}{\Delta t} x[i-l] \right) - b$$

for all $1 \leq i \leq M$, where the k -BDF coefficients $\alpha(k, \cdot)$ are given in [14];

2) *If x is the solution subject to a consistent initial condition $x(0) \in D$, we have $\max_{0 \leq i \leq M} \|x(t_i) - x[i]\|_{\infty} \leq C\Delta t^k$ for all $0 < \Delta t \leq \Delta t_0$.*

Proof: See Theorem 5.24 in [14]. ■

Remark 1. *The BDF schemes apply an auto-correction in the case an inconsistent initial condition $x(0)$ is provided. In particular, if $x(0) \notin D$ and $\nu \geq 1$ denotes the DAE index [14] of $E\partial_t x = Ax + b$, Theorem 7 remains true and additionally it holds [14, Remark 5.25] that $x[i] \in \mathcal{D}$ for all $\nu \leq i \leq M$.*

Armed with the BDF scheme and the convergence result of Theorem 7, we introduce numeric DE and numeric GDE. Intuitively, numeric DE and numeric GDE take the roles of DE and GDE, respectively, if Δt is sufficiently small. This is

because the ground truth given in terms of the DAE system $E\partial_t x = Ax + b$ and the underlying analytical solution x is replaced, in practical computations, with the BDF scheme of Theorem 7 and the numerical solution $(x[i])_i$.

Definition 6. *Fix a regular DAE system $E\partial_t x = Ax + b$, time step $\Delta t > 0$ and $1 \leq k \leq 6$ such that $(A - \frac{\alpha(k,0)}{\Delta t} E)$ is invertible, set $\Phi(x) := (A - \frac{\alpha(k,0)}{\Delta t} E)^{-1} E x - (A - \frac{\alpha(k,0)}{\Delta t} E)^{-1} b$. A partition \mathcal{H} is called*

- *numeric DE if, for any $x[0] \in \mathcal{U}_{\mathcal{H}}$, it holds that $x[i] \in \mathcal{U}_{\mathcal{H}}$ for all $1 \leq i \leq M$;*
- *numeric GDE if $\Phi(\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$.*

Note that $x[i] = \Phi(\sum_{l=1}^k \frac{\alpha(k,l)}{\Delta t} x[i-l])$ for all $1 \leq i \leq M$. As for the algebraic notions, the following result can be stated for the numeric ones.

Theorem 8. *Given a regular DAE system $E\partial_t x = Ax + b$, $1 \leq k \leq 6$ and $\Delta t > 0$ such that $(A - \frac{\alpha(k,0)}{\Delta t} E)$ is invertible, let Φ be as in Definition 6.*

- *\mathcal{H} is numeric DE if and only if \mathcal{H} is numeric GDE.*
- *For any partition \mathcal{G} , the coarsest numeric GDE \mathcal{H} which refines \mathcal{G} is the coarsest DE of $\partial_t x = \Phi(x)$ that refines \mathcal{G} ; \mathcal{H} can be computed in polynomial time.*

Proof: Let \mathcal{H} be a numeric GDE. Since $\mathcal{U}_{\mathcal{H}}$ is a linear space, this yields $\Phi(\sum_{l=1}^k \frac{\alpha(k,l)}{\Delta t} x[i-l]) = x[i] \in \mathcal{U}_{\mathcal{H}}$ if $x[i-1], \dots, x[i-k] \in \mathcal{U}_{\mathcal{H}}$. Recalling that $x[-6] = \dots = x[-1] = 0 \in \mathcal{U}_{\mathcal{H}}$ by definition, we infer that $x[i] \in \mathcal{U}_{\mathcal{H}}$ for all $1 \leq i \leq M$ when $x[0] \in \mathcal{U}_{\mathcal{H}}$. Conversely, if \mathcal{H} is a numeric DE, we have $\Phi(\frac{\alpha(k,1)}{\Delta t} x[0]) = \Phi(\sum_{l=1}^k \frac{\alpha(k,l)}{\Delta t} x[i-l]) = x[1] \in \mathcal{U}_{\mathcal{H}}$ for any $x[0] \in \mathcal{U}_{\mathcal{H}}$ because $x[-6] = \dots = x[-1] = 0$. Since this holds for any $x[0] \in \mathcal{U}_{\mathcal{H}}$ and $\mathcal{U}_{\mathcal{H}}$ is a linear subspace, this implies that $\Phi(x[0]) \in \mathcal{U}_{\mathcal{H}}$ for any $x[0] \in \mathcal{U}_{\mathcal{H}}$. The second statement follows from Theorem 1 and 3. ■

Theorem 8 states that a numeric GDE of the DAE system $E\partial_t x = Ax + b$ is a DE of the ODE system $\partial_t x = \Phi(x)$. Hence, for any partition \mathcal{G} of S , we can use the polynomial time algorithm of Theorem 3 to compute the coarsest numeric GDE that refines \mathcal{G} .

Numeric GDE allows to speed up numerical computations by invoking the reduced numerical mapping.

Definition 7. *Given a regular DAE system $E\partial_t x = Ax + b$, $1 \leq k \leq 6$ and $\Delta t > 0$ such that $(A - \frac{\alpha(k,0)}{\Delta t} E)$ is invertible, assume that \mathcal{H} is a numeric GDE. Then, the underlying reduced mapping $\underline{\Phi} : \mathbb{R}^{S_0} \rightarrow \mathbb{R}^{S_0}$ is defined via*

$$x \mapsto (S_l (A - \frac{\alpha(k,0)}{\Delta t} E)^{-1} E S_r) x - S_l (A - \frac{\alpha(k,0)}{\Delta t} E)^{-1} b$$

with S_l, S_r and S_0 as in Definition 5.

Similarly to GDE and DE, the following can be shown.

Theorem 9. *Given a regular DAE system $E\partial_t x = Ax + b$, $1 \leq k \leq 6$, $\Delta t > 0$ such that $(A - \frac{\alpha(k,0)}{\Delta t} E)$ is invertible and some partition \mathcal{G} , the coarsest numeric GDE \mathcal{H} that refines \mathcal{G} and the underlying reduced numerical mapping $\underline{\Phi}$ from Definition 7 can be computed in polynomial time. Moreover, with $\underline{x}[i] = \underline{\Phi}(\sum_{l=1}^k \frac{\alpha(k,l)}{\Delta t} \underline{x}[i-l])$, it holds that $x[i] \in S_r \underline{x}[i]$ for all $1 \leq i \leq M$, provided that $x[0] = S_r \underline{x}[0]$.*

Proof: Follows from the definition of numeric GDE and by noting that matrix inversion can be done in cubic time. ■

On the Relation of Numeric and Algebraic Notions. Despite numeric DE and numeric GDE provide an efficient model reduction technique for practical computations, it is interesting to ask whether the convergence of the BDF scheme toward the true analytical solution, ensured by Theorem 7, can be used to tie GDE to numeric GDE.

The next result is well-known.

Lemma 1. *Assume that $X_\lambda \in \mathbb{R}^{S \times S}$ invertible for all $\lambda_0 \leq \lambda < \infty$ and that $\lambda \mapsto X_\lambda^{-1}$ is differentiable on $[\lambda_0; \infty)$. Then $\partial_\lambda(X_\lambda^{-1}) = -X_\lambda^{-1} \cdot \partial_\lambda X_\lambda \cdot X_\lambda^{-1}$ for all $\lambda_0 \leq \lambda < \infty$.*

Proof: Please refer to introductory texts in linear algebra. ■

With Lemma 1, the following pivotal auxiliary statement can be established.

Proposition 1. *Given a regular DAE system $E\partial_t x = Ax + b$ and $1 \leq k \leq 6$, let $\Delta t_0 > 0$ be such that $(A - \frac{\alpha(k,0)}{\Delta t} E)$ is invertible for all $0 < \Delta t \leq \Delta t_0$. Then, for all $\lambda \geq \lambda_0 = \frac{\alpha(k,0)}{\Delta t_0}$, the matrices $R_\lambda = (A - \lambda E)^{-1} E$ and $S_\lambda = (A - \lambda E)^{-1}$ are well-defined and characterized by the Riccati matrix differential equations $\partial_\lambda R_\lambda = R_\lambda^2$ and $\partial_\lambda S_\lambda = R_\lambda S_\lambda$ on $[\lambda_0; \infty)$.*

Proof: Since $(A - \lambda E)^{-1}$ is invertible for all $\lambda_0 \leq \lambda < \infty$, Cramer's rule ensures that $\lambda \mapsto (A - \lambda E)^{-1}$ is well-defined and differentiable on $[\lambda_0; \infty)$. With this, Lemma 1 implies that

$$\begin{aligned} \partial_\lambda((A - \lambda E)^{-1} E) &= \partial_\lambda((A - \lambda E)^{-1}) E \\ &= \left(- (A - \lambda E)^{-1} \cdot \partial_\lambda(A - \lambda E) \cdot (A - \lambda E)^{-1} \right) E \\ &= \left((A - \lambda E)^{-1} \cdot E \cdot (A - \lambda E)^{-1} \right) E \\ &= ((A - \lambda E)^{-1} E)^2, \end{aligned}$$

thus showing $\partial_\lambda R_\lambda = R_\lambda^2$. Since the above calculation implies also $\partial_\lambda S_\lambda = R_\lambda S_\lambda$, the theorem of Picard-Lindelöf completes the proof. ■

With this, we are in a position to show the following.

Proposition 2. *Given a regular DAE system $E\partial_t x = Ax + b$ and $1 \leq k \leq 6$, let $\Delta t_0 > 0$ be such that $(A - \frac{\alpha(k,0)}{\Delta t} E)$ is invertible for all $0 < \Delta t \leq \Delta t_0$. Then, \mathcal{H} is numeric GDE for all $0 < \Delta t \leq \Delta t_0$ whenever \mathcal{H} is numeric GDE for Δt_0 .*

Proof: By assumption, it holds that $(A - \lambda_0 E)^{-1} b \in \mathcal{U}_{\mathcal{H}}$ and $R_{\lambda_0}(\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$ for $\lambda_0 = \frac{\alpha(k,0)}{\Delta t_0}$, where $R_\lambda := (A - \lambda E)^{-1} E$ is well-defined for all $\lambda \geq \lambda_0$, see Proposition 1. With this, we fix an $f : \mathbb{R}^{S \times S} \rightarrow \mathbb{R}^{S \times S}$ such that

- a) $f(R) = R^2$ for any R such that $\|R\| \leq 2C$, where $C := \max_{\lambda_0 \leq \lambda \leq \lambda_1} \|R_\lambda\|$ for some arbitrary $\lambda_1 > \lambda_0$;
- b) f is globally Lipschitz continuous on $\mathbb{R}^{S \times S}$.

The existence of such an f follows from Kirszbraun's theorem (or via smooth cutoff functions). Thanks to b), this allows us to infer that the Euler method, if applied to $\partial_\lambda R_\lambda = f(R_\lambda)$ on $[\lambda_0; \lambda_1]$, converges for vanishing step size τ (recall that standard convergence results of Euler's method require global rather than local Lipschitzianity of the drift [22]). In particular, for sufficiently small $\tau > 0$, we obtain $\|R[i]\| \leq \frac{3}{2}C$, where

$R[0], R[1], \dots, R[\kappa]$ denotes the underlying Euler sequence with $\kappa = \lceil (\lambda_1 - \lambda_0) / \tau \rceil$. Together with property a), this implies that $f(R[i]) = R[i]^2$ for all $1 \leq i \leq \kappa$ which, in turn, yields $R[i](\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$ for all $1 \leq i \leq \kappa$ by a proof by induction over i and the fact that $R_{\lambda_0}(\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$. A combination of all observations then yields $R_\lambda(\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$ for all $\lambda \in [\lambda_0; \lambda_1]$.

We are left with showing that $(A - \lambda E)^{-1} b \in \mathcal{U}_{\mathcal{H}}$ for all $\lambda_0 \leq \lambda \leq \lambda_1$. To this end, we define $S_\lambda := (A - \lambda E)^{-1}$ as in Proposition 1 and recall that $\partial_\lambda S_\lambda = R_\lambda S_\lambda$. The Euler sequence underlying this ordinary matrix differential equation is given, for any time step $\tau > 0$, by $S[i+1] = S[i] + \tau R[i] S[i]$ where $S[0] = S_{\lambda_0}$ and $R[i] = R_{\tau i + \lambda_0}$. Starting with the assumption that $S[0]b \in \mathcal{U}_{\mathcal{H}}$, a proof by induction over i ensures that $S[i]b \in \mathcal{U}_{\mathcal{H}}$ for all $i \geq 0$ because $R_\lambda(\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$ for all $\lambda_0 \leq \lambda \leq \lambda_1$. By replacing the original drift by some function g that satisfies similar properties to a) and b) from above, one can infer that $S_\lambda b \in \mathcal{U}_{\mathcal{H}}$ for all $\lambda \geq \lambda_0$. ■

Proposition 2 states that the variable equivalences identified in the BDF scheme by a numeric GDE do not depend on the chosen time step Δt . Armed with this, the next major result establishes that numeric GDE implies GDE.

Theorem 10. *Given a regular DAE system $E\partial_t x = Ax + b$ and $1 \leq k \leq 6$, let $\Delta t_0 > 0$ be such that $(A - \frac{\alpha(k,0)}{\Delta t} E)$ is invertible for all $0 < \Delta t \leq \Delta t_0$. If \mathcal{H} is a numeric GDE, then it is also a GDE.*

Proof: Assume that \mathcal{H} is a numeric GDE and let $1 \leq \nu \leq |\mathcal{S}|$ be the DAE index of $E\partial_t x = Ax + b$. Then, by noting that $x[i] = \Phi(\sum_{l=1}^k \frac{\alpha(k,l)}{\Delta t} x[i-l])$, Remark 1 and $x[0] \in \mathcal{U}_{\mathcal{H}}$ yield $x[\nu] \in \mathcal{U}_{\mathcal{H}} \cap \mathcal{D}$, thus showing $\mathcal{U}_{\mathcal{H}} \cap \mathcal{D} \neq \emptyset$. Moreover, thanks to Theorem 8 and Proposition 2, it holds that \mathcal{H} is a numeric DE for all $0 < \Delta t < \Delta t_0$. By letting $\Delta t \rightarrow 0$, this implies that \mathcal{H} is a DE. With this, Theorem 5 yields the claim. ■

Since Δt_0 is bounded by the inverse of the modulus of the largest generalized eigenvalue of the matrix pencil $A - \lambda E$, one can bound Δt_0 using the generalized Schur decomposition [23].

Remark 2. *Numeric GDE is only a sufficient condition for GDE in general. Indeed, for (1), the matrix $(A - \lambda E)$ is invertible if and only if $\lambda^2 - 3\lambda + 2 \neq 0$. Hence, $(A - \lambda E)$ is invertible for all $\lambda \geq 3$, meaning that λ_0 and Δt_0 can be set to 3 and $\frac{1}{3}$, respectively. A computation of $(A - \frac{1}{3} E)^{-1} E$ reveals that the coarsest numeric GDE for Δt_0 (hence, by Theorem 2, for all $0 < \Delta t \leq \Delta t_0$) is the trivial partition $\{\{1\}, \{2\}, \{3\}\}$. At the same time, Example 2 establishes that $\{\{1, 2\}, \{3\}\}$ is a GDE of (1).*

Even though we expect numeric GDE to coincide with GDE for most practical models, the above remark demonstrates that numeric GDE is in general stricter than GDE. At the same time, the coarsest numeric GDE partition can be computed in polynomial time, while GDE relies on the solution of an ill-conditioned problem [13].

V. NUMERIC GEOMETRIC DIFFERENTIAL EQUIVALENCE AND SEMI-EXPLICITNESS

We next assume that we are given a linear DAE system in semi-explicit form, i.e., we consider

$$\partial_t x = A \begin{pmatrix} x \\ y \end{pmatrix} + b \quad 0 = (B_1 \ B_2) \begin{pmatrix} x \\ y \end{pmatrix} + c, \quad (3)$$

where x and y are conveniently called state and algebraic variables, respectively. Additionally, we assume that the system has DAE index one. In this case, it is known [24, Section 2.2] that B_2 is invertible and that the index reduction of (3) is given by the linear ODE system

$$\partial_t x = A \begin{pmatrix} x \\ y \end{pmatrix} + b \quad \partial_t y = -B_2^{-1} B_1 \left[A \begin{pmatrix} x \\ y \end{pmatrix} + b \right] \quad (4)$$

That is, if $(x(0), y(0))$ is a consistent initial condition of (3), then the corresponding solution of (3) can be obtained by solving (4).

The above discussion shows that the coarsest DE of (4) gives rise to a DE of (3). It has to be noted, however, that the computation of (4) relies on matrix inversion, which may suffer from truncation errors in the case of large-scale systems.

In order to establish our next result, the following auxiliary statement will be needed.

Lemma 2. *Fix an invertible matrix B . Then, \mathcal{H} is a DE of $\partial_t x = Bx$ if and only if \mathcal{H} is a DE of $\partial_t x = B^{-1}x$.*

Proof: It suffices to show only one direction, so let us assume that \mathcal{H} is a DE of $\partial_t x = Bx$. For every $H \in \mathcal{H}$, let e_H denote the vector with $e_H(i) = 1$ if $i \in H$ and $e_H(i) = 0$ otherwise. Since \mathcal{H} is a DE, it holds that $y_H := Be_H \in \mathcal{U}_{\mathcal{H}}$ for all $H \in \mathcal{H}$. The regularity of B , instead, ensures that $\{y_H \mid H \in \mathcal{H}\}$ is a basis of $\mathcal{U}_{\mathcal{H}}$. Using this and the fact that $B^{-1}y_H = e_H \in \mathcal{U}_{\mathcal{H}}$ for all $H \in \mathcal{H}$, we infer that $B^{-1}(\mathcal{U}_{\mathcal{H}}) \subseteq \mathcal{U}_{\mathcal{H}}$, thus showing the claim. ■

Armed with Lemma 2, we next provide a result that allows one to obtain a DE of (4) without having to invert a matrix.

Theorem 11. *Consider the auxiliary ODE system*

$$\partial_t x = A \begin{pmatrix} x \\ y \end{pmatrix} + b \quad \partial_t y = (B_1 \ B_2) \begin{pmatrix} x \\ y \end{pmatrix} \quad (5)$$

which arises from (3) by dropping c and by replacing the algebraic constraint with a differential constraint. If $\mathcal{H} = (\mathcal{H}', \mathcal{G}')$ is a DE of (5) such that \mathcal{H}' and \mathcal{G}' are partitions of state and algebraic variables, respectively, then \mathcal{H} is also a DE of (4).

Proof: Let us assume that $\mathcal{H} = (\mathcal{H}', \mathcal{G}')$ is a DE partition of (5), where \mathcal{H}' and \mathcal{G}' are partitions on the index sets of x and y , respectively. By considering the functions

$$x \mapsto (B_1 \ B_2) \begin{pmatrix} x \\ 0 \end{pmatrix}, \quad y \mapsto (B_1 \ B_2) \begin{pmatrix} 0 \\ y \end{pmatrix},$$

we infer that $B_1(\mathcal{U}_{\mathcal{H}'}) \subseteq \mathcal{U}_{\mathcal{H}'}$ and $B_2(\mathcal{U}_{\mathcal{G}'}) \subseteq \mathcal{U}_{\mathcal{G}'}$. Hence, Lemma 2 yields $B_2^{-1}(\mathcal{U}_{\mathcal{G}'}) \subseteq \mathcal{U}_{\mathcal{G}'}$. This, in turn, allows us to conclude that $B_2^{-1}B_1[A(\mathcal{U}_{\mathcal{H}'}, \mathcal{U}_{\mathcal{G}'}) + b] \subseteq \mathcal{U}_{\mathcal{G}'}$ because $A(\mathcal{U}_{\mathcal{H}'}, \mathcal{U}_{\mathcal{G}'}) + b \subseteq \mathcal{U}_{\mathcal{H}'}$ by assumption. Overall, we obtain

$$(A + b, B_2^{-1}B_1A + B_2^{-1}B_1b)(\mathcal{U}_{\mathcal{H}'}, \mathcal{U}_{\mathcal{G}'}) \subseteq (\mathcal{U}_{\mathcal{H}'}, \mathcal{U}_{\mathcal{G}'}),$$

thus showing the claim. We end the proof by pointing out that the result can be also shown in the case when c is *not* dropped. This may, however, reduce the applicability. Indeed, in the case one requires that $(x, y) \in (\mathcal{U}_{\mathcal{H}'}, \mathcal{U}_{\mathcal{G}'})$ implies

$$(B_1 \ B_2) \begin{pmatrix} x \\ y \end{pmatrix} + c \in \mathcal{U}_{\mathcal{G}'},$$

picking $(x, y) = 0$ ensures that $c \in \mathcal{U}_{\mathcal{G}'}$. Hence, not dropping c is equivalent to imposing the additional requirement $c \in \mathcal{U}_{\mathcal{G}'}$ in the premise of the theorem. ■

Theorem 11 provides an efficient way of finding DE partitions of large-scale DAE systems. In particular, the underlying computational steps do not involve inversion of matrices or any other numerical operations that may suffer from non-negligible truncation errors. At the same time, the coarsest DE of (4) must not be necessarily the coarsest DE of (3). To see this, consider the trivial DAE system $\partial_t x_1 = x_1$, $y_1 = x_1$. Then, it holds true that $\{\{1, 1\}\}$ is a DE which cannot be found using Theorem 11 because the latter requires the state variables and algebraic variables to be put into different blocks. This being said, we wish to stress that the separation of state and algebraic variables cannot be dropped in Theorem 11. Indeed, $\{\{1, 1\}\}$ cannot be a DE of the DAE system

$$\partial_t x_1 = 2x_1 - y_1 \quad 0 = 2x_1 - y_1 \quad (6)$$

because $x_1 \neq y_1$. At the same time, $\{\{1, 1\}\}$ is a DE of the ODE system arising from (6) by replacing $0 = 2x_1 - y_1$ with $\partial_t y_1 = 2x_1 - y_1$.

VI. EVALUATION

We next evaluate our results on benchmark models from the electric engineering literature. Section VI-A focuses on numeric GDE as implemented in a Matlab prototype, while Section VI-B focuses on semi-explicit DAEs to which the results from Section V apply. For the latter, we use the DE for ODE systems [16] as implemented in the tool ERODE [27], and report the analysis speedups for OpenModelica [17], to which ERODE offers exporting capabilities.

A. Evaluation of Numeric GDE

In this section we compare the analysis runtime of the original models against those of the reduced ones (including minimization runtime) using the `ode15s` solver of Matlab, capable of processing linear (and nonlinear) DAE systems with DAE index at most one. Experiments have been performed on a 2.6 GHz Intel running Matlab 2016b; the Matlab command `ode15s` has been invoked with standard settings.

a) *H-tree model:* We start by considering a power distribution network from [25] which has been adapted by removing inductances. The network has a hierarchical tree topology (H-tree), depicted in Figure 1. At each depth i , all branches have equal resistances and capacitances R_i and C_i , respectively, whose values are given in Table II. For depths $i \leq 4$, the values were taken from [25], while values for depths $i \geq 5$ have been extrapolated. An H-tree model of depth N results in a DAE system with $2^N + 1$ variables, given by the

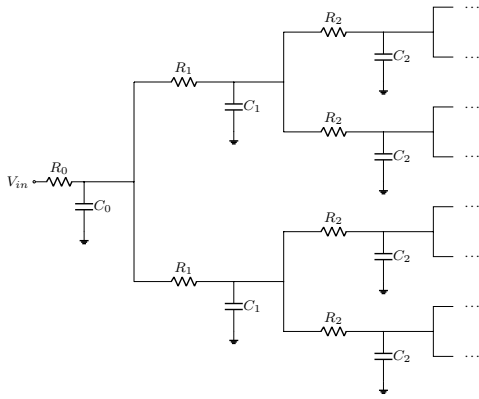


Fig. 1: H-tree network adapted from [25].

| N | $Model$ | | $Num. GDE$ | | |
|-----|-----------------|-------------|-----------------|------------|--------------------|
| | $ \mathcal{S} $ | $Sol. (s)$ | $ \mathcal{H} $ | $Red. (s)$ | $Red. \& Sol. (s)$ |
| 3 | 9 | 8.86E-2 | 5 | 4.38E-3 | 7.38E-3 |
| 4 | 17 | 1.44E-1 | 6 | 9.83E-3 | 1.56E-2 |
| 5 | 33 | 1.61E-1 | 7 | 1.15E-2 | 2.02E-2 |
| 6 | 65 | 1.72E-1 | 8 | 2.86E-2 | 3.46E-2 |
| 7 | 129 | 1.77E-1 | 9 | 3.70E-2 | 4.56E-2 |
| 8 | 257 | 2.39E-1 | 10 | 5.50E-2 | 6.61E-2 |
| 9 | 513 | 2.86E-1 | 11 | 8.19E-2 | 9.84E-2 |
| 10 | 1025 | 1.04E+0 | 12 | 1.98E-1 | 2.24E-1 |
| 11 | 2049 | 5.05E+0 | 13 | 5.40E-1 | 5.88E-1 |
| 12 | 4097 | 3.77E+1 | 14 | 9.41E-1 | 1.03E+0 |
| 13 | 8193 | 6.27E+2 | 15 | 2.07E+0 | 2.23E+0 |
| 14 | 16385 | <i>O.M.</i> | 16 | 5.03E+0 | 5.35E+0 |

TABLE I: Numerical results for H-trees.

| i | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R_i | 6.37 | 12.75 | 25.50 | 50 | 100 | 200 | 400 | 800 | 1600 | 3200 | 6400 | 12800 | 25600 |
| C_i | 0.300 | 0.130 | 0.140 | 0.070 | 0.070 | 0.035 | 0.035 | 0.018 | 0.018 | 0.009 | 0.009 | 0.005 | 0.005 |

TABLE II: Resistances and capacitances of the H-tree networks at depths i .

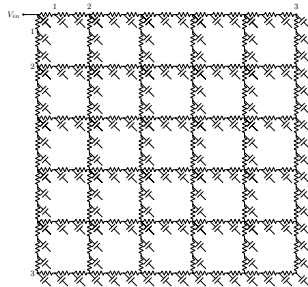


Fig. 2: Mesh networks adapted from [26].

| N | $Model$ | | $Num. GDE$ |
|-----|-----------------|-----------------|------------|
| | $ \mathcal{S} $ | $ \mathcal{H} $ | $Red. (s)$ |
| 4 | 106 | 56 | 2.48E-1 |
| 5 | 157 | 82 | 3.06E-1 |
| 6 | 218 | 113 | 3.51E-1 |
| 7 | 289 | 149 | 6.41E-1 |
| 8 | 370 | 190 | 1.05E+0 |
| 9 | 461 | 236 | 1.70E+0 |
| 10 | 561 | 287 | 4.26E+0 |

TABLE III: Numerical results for meshes.

voltages across each capacitor and by the voltage V_{in} , which is assumed to have constant value of 5.0V.

Table I considers H-tree networks with increasing depth N (column $|\mathcal{S}|$ gives the state space size). Numeric GDE confirms the symmetry discussed in [25], namely that the voltages across the capacitors of same depth are equal at all time points, resulting in $N + 2$ partition blocks (column $|\mathcal{H}|$).

Column $Sol.$ provides the computation time of the original model via `ode15s` in the case where the time horizon was set to 2 time units. Column $Red.$ provides the time necessary for the computation of the coarsest numeric GDE. We can see that this is always at least one order of magnitude smaller than the time to analyze the original model. To allow for a fair comparison, column $Red. \& Sol.$ provides the cumulative runtime for the computation of the coarsest numeric GDE and the solution of the underlying reduced DAE system via `ode15s`. Numeric GDE always led to better performance. In addition, it allowed us to analyze the case $N = 14$, whose full model could not be solved by `ode15s` due to out of memory errors.

b) Mesh networks: Figure 2 shows a square transmission line mesh, adapted from [26], where each line has $N = 5$ segments of 3 resistor/capacitor series each.² The variables of

²Every capacitor is connected to the ground, which is not explicitly drawn to avoid clutter.

the DAE system represent the voltages at each capacitor.

Table III considers seven variants of the network for N ranging from 4 to 10 for resistances and capacitances equal to 1 and 0.1, respectively. Columns $|\mathcal{S}|$, $|\mathcal{H}|$, and $Red.$ are as in Table I. We do not provide the runtime for analyzing the original and reduced DAEs because both the original and the reduced DAE systems had index higher than one, which cannot be handled by `ode15s`. We wish to point out, however, that the reduced numerical model from Theorem 9 could be, in principle, derived. It is interesting to note that, for each N , the size of the largest numeric GDE is $N + 1 + (|\mathcal{S}| - N)/2$. This has a geometric interpretation in that it relates (voltages of) capacitor nodes placed symmetrically with respect to the main diagonal of the network (Figure 2 marks nodes within representative equivalence classes with the same number).

c) Further benchmarks: Table IV considers further benchmarks of real-world electrical circuits available at <https://sites.google.com/site/rommes/software>. All models are given in the form $E\partial_t x(t) = Ax(t) + Bu(t)$ where $x(t) \in \mathbb{R}^n$ is the system's state, $u(t)$ represents some m -dimensional input and $B \in \mathbb{R}^{n \times m}$. In order to obtain symmetries that are valid for any input signal u , we started from an input partition \mathcal{G} whose blocks of variables are *affected in the same way* by the input, that is, two variable indices i, j were put into the same block of \mathcal{G} if and only if $e_i^T(Bu) = e_j^T(Bu)$ for all u (which is

| Model | | Num. GDE | | |
|------------------|------|-----------------|-----------------|----------|
| Name | Ref. | $ \mathcal{S} $ | $ \mathcal{H} $ | Red. (s) |
| bips98_606 | [28] | 7 135 | 5 656 | 5.29E+3 |
| bips98_1142 | [28] | 9 735 | 7 225 | 1.08E+4 |
| bips98_1450 | [28] | 11 305 | 8 115 | 1.73E+4 |
| nopss_11k | [29] | 11 685 | 8 015 | 1.95E+4 |
| mimo46x46_system | [30] | 13 250 | 9 132 | 5.35E+4 |
| bips07_1693 | [28] | 13 275 | 9 073 | 7.59E+4 |
| mimo8x8_system | [31] | 13 309 | 9 070 | 5.14E+4 |

TABLE IV: Further circuit benchmarks: numerical results.

equivalent to saying that i, j were put into the same block if and only if $e_i^T(Be_k) = e_j^T(Be_k)$ for all $1 \leq k \leq m$). For the experiments, we considered the input $b = B\mathbf{1}$, with $\mathbf{1}$ being the m -dimensional vector whose coordinates are all one. This led to systems of the form $E\partial_t x = Ax + b$.

As in the case of Table III, we only provide column *Red.* in Table IV because the original and reduced DAE systems had index greater than one. We remark that all instances could be reduced to about 70% of their original size.

B. Evaluation of semi-explicit Models

In this section we study the applicability of the results from Section V to the benchmarks in Section VI-A, and report on possible speedup of OpenModelica.

a) *Evaluation of semi-explicit Models:* We start computing numeric GDE using DE for a subset of the benchmarks from Section VI-A given as semi-explicit DAEs, with a focus on the speed up in the reduction obtained thanks to Theorem 11. In particular, since all considered models are linear, for computing DE partitions we can use the polynomial time and space algorithms from [16], [15].

The results in Section V can be applied to the H-tree networks because their underlying DAEs have index one. Here we consider H-tree instances of depth up to 19, starting from the largest network considered in Section VI-A, depth 14. The values of the resistances and capacitances for the new models, reported in Table V, have been extrapolated as done in Section VI-A. Table VI compares the reduction runtimes of numeric GDE and DE. Differently from Table I, we do not provide information about analysis runtime of the original and reduced DAEs. Focusing on the H-tree network of depth 14, we can see that the computed numeric GDE and DE coincide, with the latter computation being two orders of magnitude faster. More importantly, when computing numeric GDE for larger models our prototypal implementation fails due to out of memory problems. Instead, as studied in [15], DE can be applied to models with millions of variables using the tool ERODE [27] (in case of ODE systems with polynomial right-hand sides which cover in particular linear drifts). Here we are interested in identifying the largest model instance for which DE has a runtime similar to the of numeric GDE for the one of depth 14. From the last row of Table VI we can see that we have to consider an H-tree instance of depth 19 in order to get a runtime of about 5 seconds for DE.

We end the section by pointing out that Theorem 11 cannot be applied to the other benchmark models of Section VI-A

| i | 15 | 16 | 17 | 18 | 19 |
|-------|--------|---------|---------|---------|---------|
| R_i | 51 200 | 102 400 | 204 800 | 409 600 | 819 200 |
| C_i | 0.0025 | 0.0025 | 0.0012 | 0.0012 | 0.0006 |

TABLE V: Specifications for larger H-tree networks.

| N | Model | | Num. GDE | | DE | |
|-----|-----------------|-----------------|----------|-----------------|----------|--|
| | $ \mathcal{S} $ | $ \mathcal{H} $ | Time (s) | $ \mathcal{H} $ | Time (s) | |
| 14 | 16 385 | 16 | 5.03E+0 | 16 | 8.70E-2 | |
| 15 | 32 769 | —O.M.— | — | 17 | 1.91E-2 | |
| 16 | 65 537 | —O.M.— | — | 18 | 4.12E-1 | |
| 17 | 131 073 | —O.M.— | — | 19 | 8.98E-1 | |
| 18 | 262 145 | —O.M.— | — | 20 | 1.09E+0 | |
| 19 | 524 289 | —O.M.— | — | 21 | 5.23E+0 | |

TABLE VI: DE and Num. GDE runtimes on H-trees.

because those are instances of high index problems, i.e., they have DAE indices higher than one.

b) *Speedup of OpenModelica:* State-of-the-art modeling languages for complex systems such as OpenModelica perform a number of preprocessing steps aimed at simplifying a model by means of index reduction [17]. Here we illustrate that, in general, DE constraints $x_i = x_j$ (where $H \in \mathcal{H}$ and $i, j \in H$) are not discovered by the preprocessing of OpenModelica and, if added manually, can be used to speed it up. To this end, we analyze models from the largest family considered in this paper: the H-tree instances. In particular, we consider those with at least 50 variables, and compute GDEs using DE, therefore exploiting the results from Section V. The results are provided in Table VII, where the blocks of columns *Original Model*, *DE Constraints* and *DE-Reduced Model* refer, respectively, to the original models, to the models enriched with the DE constraints of the computed DE, and to the DE-reduced models. For each block, we provide a column showing the runtime of each corresponding internal phase reported by OpenModelica. In particular, the first phase (backend) is the one devoted to model optimization. For all experiments we provide the average runtime over 5 independent runs with time horizon 1, using OpenModelica’s default index reduction method *dynamic state selection* [32]³. We used OpenModelica v1.14.1 (64-bit) running on a Win10 (64-bit) machine with a 2.8 GHz Intel CPU and 8GB of RAM.

The largest model that OpenModelica could handle within a time out of 4 hours that we imposed is the one with depth 14. Instead, it was not even possible to load models with depths above 16 due to memory constraints. We note that for models up to depth 10, the overall runtime (*Total*) is governed by the compile phase, while the runtime of the backend phase grows significantly for larger models, and becomes dominant for depths 13 and 14. Interestingly, the simulation phase is the one with lowest runtime for models of depth greater than 9.

Considering the models enriched with DE constraints, we can see that OpenModelica was able to handle also the models that previously failed due to time out. Indeed, the DE constraints

³In particular, we used the following OpenModelica settings for parameter simulationOptions: startTime = 0.0, stopTime = 1.0, numberOfIntervals = 500, tolerance = 1e-006, method = 'dassl', options = "", outputFormat = 'mat', variableFilter = '.*', cflags = "", simflags = ""

| N | Original Model | | | | DE Constraints | | | | DE-Reduced Model | | | |
|-----|----------------|---------|------------|---------|----------------|---------|------------|---------|------------------|---------|------------|---------|
| | Backend | Compile | Simulation | Total | Backend | Compile | Simulation | Total | Backend | Compile | Simulation | Total |
| 6 | 3.19E-2 | 6.57E+0 | 5.65E-1 | 7.73E+0 | 7.58E-3 | 6.66E+0 | 5.50E-1 | 7.84E+0 | 5.42E-3 | 6.57E+0 | 5.74E-1 | 7.90E+0 |
| 7 | 1.23E-1 | 6.73E+0 | 5.99E-1 | 8.06E+0 | 1.05E-2 | 6.61E+0 | 5.60E-1 | 7.73E+0 | 6.50E-3 | 6.55E+0 | 5.63E-1 | 7.62E+0 |
| 8 | 1.48E-1 | 6.87E+0 | 6.07E-1 | 8.26E+0 | 1.48E-2 | 6.62E+0 | 5.70E-1 | 7.75E+0 | 6.80E-3 | 6.71E+0 | 5.73E-1 | 8.03E+0 |
| 9 | 5.63E-1 | 7.83E+0 | 6.15E-1 | 9.86E+0 | 2.58E-2 | 6.58E+0 | 5.70E-1 | 7.75E+0 | 7.92E-3 | 6.57E+0 | 5.52E-1 | 7.64E+0 |
| 10 | 1.94E+0 | 1.09E+1 | 8.77E-1 | 1.49E+1 | 4.33E-2 | 6.73E+0 | 5.70E-1 | 7.95E+0 | 7.98E-3 | 6.61E+0 | 5.49E-1 | 7.80E+0 |
| 11 | 1.31E+1 | 2.44E+1 | 1.53E+0 | 4.08E+1 | 1.42E-1 | 6.64E+0 | 6.30E-1 | 8.14E+0 | 9.90E-3 | 6.56E+0 | 5.66E-1 | 7.71E+0 |
| 12 | 9.80E+1 | 5.28E+1 | 3.24E+0 | 1.57E+2 | 1.96E-1 | 6.63E+0 | 6.40E-1 | 8.19E+0 | 9.54E-3 | 6.56E+0 | 5.47E-1 | 7.68E+0 |
| 13 | 8.45E+2 | 1.68E+2 | 1.21E+1 | 1.03E+3 | 3.23E-1 | 6.65E+0 | 7.00E-1 | 8.86E+0 | 1.03E-2 | 6.56E+0 | 5.39E-1 | 7.67E+0 |
| 14 | 8.12E+3 | 6.34E+2 | 1.12E+2 | 8.88E+3 | 7.36E-1 | 6.56E+0 | 8.50E-1 | 1.00E+1 | 1.07E-2 | 6.57E+0 | 5.47E-1 | 7.75E+0 |
| 15 | Time Out | | | | 1.67E+0 | 6.66E+0 | 1.12E+0 | 1.22E+1 | 1.23E-2 | 6.58E+0 | 5.80E-1 | 7.69E+0 |
| 16 | Time Out | | | | 3.52E+0 | 6.66E+0 | 1.72E+0 | 1.77E+1 | 1.22E-2 | 6.62E+0 | 5.84E-1 | 7.84E+0 |
| 17 | Out of Memory | | | | Out of Memory | | | | 1.25E-2 | 6.58E+0 | 5.61E-1 | 7.72E+0 |
| 18 | Out of Memory | | | | Out of Memory | | | | 1.30E-2 | 6.61E+0 | 5.33E-1 | 7.73E+0 |
| 19 | Out of Memory | | | | Out of Memory | | | | 1.43E-2 | 6.59E+0 | 5.55E-1 | 7.72E+0 |

TABLE VII: Numerical results of H-trees for OpenModelica. All runtimes are seconds. The DE runtime for models with $N \geq 14$ can be found in Table VI, while DE runtimes for smaller models are in the order of few milliseconds.

synthesized by DE allow for much faster analysis, leading to a runtime 3 orders of magnitude smaller for the model with depth 14. We can see that all three phases of backend, compilation and simulation greatly took advantage by the addition of the DE constraints. In particular, the runtime of the backend phase grows slowly with the model size, while those of compilation and simulation remain essentially constant. This can be explained by the fact that the backend phase is able to reduce the model to the size of the computed DE partition ($i+2$ for models of depth i , see Tables I, and VI). This, however, does not help analyzing models of depth greater than 16 whose loading still fails due to out of memory errors.

Finally, all DE-reduced models could be analyzed in essentially constant time (at most 8 seconds). Despite averaging the analysis over 5 runs, the runtime is not monotonically increasing with the model size. This is due to the quite small and similar size of the obtained reduced models.

VII. FURTHER DISCUSSION

In [33], the author proposes a technique for the minimization of polynomial ODE systems in the spirit of DE but which does not apply to DAE systems. Instead, the recent work [34] extends classic bisimulation relations for linear systems [35], [36], [37] to linear DAE systems. While similar in style, [34] is not comparable to DE. This is because [34] cannot be used to show that certain variables enjoy identical solutions at all time points; conversely, unlike [34], DE cannot be used to show that a linear observation map (i.e., $y = Cx$ for some matrix C) coincides with the solution of the reduced system. The symmetry property underlying DE can be aligned to Φ -related vector fields [38]. However, DE imposes constraints on initial conditions and establish a reduction of a system instead of relating two given systems, see Section VII.

The idea of obtaining symmetries of a dynamical system by studying symmetries of its numerical approximation is related to geometric numerical integration. There, one studies which symmetries are preserved by numerical schemes. If applied to ODE systems, for instance, Runge-Kutta schemes are known to preserve linear invariants [39, Section IV.1]. However, the results do not carry over to DAE systems in general. Indeed,

the implicit Euler scheme for DAE systems does not preserve linear invariants in general, as will be shown later, cf. Remark 2. A known class of DAE systems whose solutions are known to preserve certain invariants are, for instance, constrained mechanical systems [39, Section VII.1].

Since GDE is an extension of DE, we next relate DE and bisimulation [35], [36], [37]. Essentially, DE provides a reduction of a *single* system, while bisimulation relates *two* given systems. These two notions can be compared when the observation maps are not restricted to linear functions. In this case, the original system and its DE reduction can be shown to be bisimilar. For instance, $\{\{1, 2\}\}$ is a DE partition of $\dot{x}_1 = x_1$, $\dot{x}_2 = x_2$ because $x_1(t) = x_2(t)$ for all $t > 0$ if $x_1(0) = x_2(0)$. The corresponding DE reduction from Definition 5 is $\dot{x}_1 = x_1$. By fixing the quotient system $\dot{y}_1 = y_1$ and the observation maps $(x_1, x_2) \mapsto x_1$, $y_1 \mapsto y_1$, it can be shown that $\mathcal{B} = \{((x_1, x_2), y_1) \mid x_1 = x_2 = y_1\}$ is a bisimulation. However, for this comparison to be made, one needs the appropriate observation maps, which are induced by the DE reduction. Further, the coarsest DE reduction can be computed in polynomial time for nonlinear ODE systems [16].

VIII. CONCLUSION

This paper introduces geometric differential equivalence (GDE), a notion for the reduction of differential-algebraic equations (DAEs) that relates variables when they have equal solutions at all time points. Unlike related work on ordinary differential equations, GDE requires a careful treatment of the domain on which the DAE is defined. Model reductions underlying GDE can be computed in polynomial time and were obtained for a number of benchmark models of electrical circuits. Developing approximate notions is part of future work.

REFERENCES

- [1] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. 6, pp. 504–509, 1975.
- [2] P. J. Gawthrop and G. P. Bevan, "Bond-graph modeling," *IEEE Control Systems*, vol. 27, no. 2, pp. 24–45, 2007.
- [3] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.

- [4] V. Mehrmann and T. Stykel, "Balanced Truncation Model Reduction for Large-Scale Systems in Descriptor Form," in *Dimension Reduction of Large-Scale Systems*, 2005, pp. 83–115.
- [5] T. Reis and T. Stykel, "Balanced truncation model reduction of second-order systems," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 14, no. 5, pp. 391–406, 2008.
- [6] P. Benner and T. Stykel, *Model Order Reduction for Differential-Algebraic Equations: A Survey*. Springer, 2017, pp. 107–160.
- [7] M. Sunnaker, G. Cedersund, and M. Jirstrand, "A method for zooming of nonlinear models of biochemical systems," *BMC Systems Biology*, vol. 5, no. 1, p. 140, 2011.
- [8] M. Apri, M. de Gee, and J. Molenaar, "Complexity reduction preserving dynamical behavior of biochemical networks," *Journal of Theoretical Biology*, vol. 304, pp. 16 – 26, 2012.
- [9] A. Abate, L. Brim, M. Ceska, and M. Z. Kwiatkowska, "Adaptive aggregation of markov chains: Quantitative analysis of chemical reaction networks," in *CAV*, 2015, pp. 195–213.
- [10] P. Buchholz, "Exact and Ordinary Lumpability in Finite Markov Chains," *Journal of Applied Probability*, vol. 31, no. 1, pp. 59–75, 1994.
- [11] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Symbolic computation of differential equivalences," in *POPL*, 2016.
- [12] J. Demmel and B. Kågström, "The Generalized Schur Decomposition of an Arbitrary Pencil $A-\lambda B$: Robust Software with Error Bounds and Applications. Part I: Theory and Algorithms," *ACM Trans. Math. Softw.*, vol. 19, p. 160174, 1993.
- [13] P. Kunkel and V. Mehrmann, "A new look at pencils of matrix valued functions," *Linear Algebra and its Applications*, vol. 212–213, pp. 215 – 248, 1994.
- [14] —, *Differential-Algebraic Equations. Analysis and Numerical Solution*. Zurich, Switzerland: EMS Publishing House, 2006.
- [15] L. Cardelli, M. Tribastone, M. Tschaikowski, and A. Vandin, "Efficient syntax-driven lumping of differential equations," in *TACAS*, vol. 9636, 2016, pp. 93–111.
- [16] —, "Maximal aggregation of polynomial dynamical systems," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 114, no. 38, pp. 10029–10034, 2017.
- [17] P. Fritzson, *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*, 2nd ed. Wiley-IEEE Press, 2014.
- [18] S. Tognazzi, M. Tribastone, M. Tschaikowski, and A. Vandin, "Backward invariance for linear differential algebraic equations," in *CDC*, 2018, pp. 3771–3776. [Online]. Available: <https://doi.org/10.1109/CDC.2018.8619710>
- [19] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.
- [20] M. Tschaikowski and M. Tribastone, "Approximate reduction of heterogeneous nonlinear models with differential hulls," *IEEE Trans. Automat. Contr.*, vol. 61, no. 4, pp. 1099–1104, 2016.
- [21] C. C. Pantelides, "The Consistent Initialization of Differential-Algebraic Systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 9, no. 2, pp. 213–231, 1988.
- [22] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1971.
- [23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.
- [24] K. Brenan, S. Campbell, S. Campbell, and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, ser. Classics in Applied Mathematics. SIAM, 1996.
- [25] J. Rosenfeld and E. G. Friedman, "Design Methodology for Global Resonant H-Tree Clock Distribution Networks," *IEEE Trans. VLSI Syst.*, vol. 15, no. 2, pp. 135–148, 2007.
- [26] B. N. Sheehan, "ENOR: model order reduction of RLC circuits using nodal equations for efficient factorization," in *Proceedings 1999 Design Automation Conference*, 1999, pp. 17–21.
- [27] L. Cardelli, M. Tribastone, A. Vandin, and M. Tschaikowski, "ERODE: A tool for the evaluation and reduction of ordinary differential equations," in *Tools and Algorithms for the Construction and Analysis of Systems — 23rd International Conference, TACAS*, 2017. [Online]. Available: <http://cse.lab.imtlucca.it/~mirco.tribastone/papers/tacas2017.pdf>
- [28] F. D. Freitas, J. Rommes, and N. Martins, "Gramian-based reduction method applied to large sparse power system descriptor models," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1258–1270, 2008.
- [29] J. Rommes, N. Martins, and F. Freitas, "Computing rightmost eigenvalues for small-signal stability assessment of large scale power systems," in *IEEE PES General Meeting*, July 2010, pp. 1–1.
- [30] N. Martins, P. C. Pellanda, and J. Rommes, "Computation of transfer function dominant zeros with applications to oscillation damping control of large power systems," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1657–1664, 2007.
- [31] J. Rommes and N. Martins, "Efficient computation of multivariable transfer function dominant poles using subspace acceleration," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1471–1483, 2006.
- [32] S. Mattsson, S. E. Mattsson, H. Olsson, H. Elmqvist, and D. A. Lund, "Dynamic selection of states in dymola," in *Proceedings of Modelica Workshop*, 2000.
- [33] M. Boreale, "Algebra, coalgebra, and minimization in polynomial differential equations," in *FOSSACS*, 2017, pp. 71–87.
- [34] N. Y. Megawati and A. van der Schaft, "Bisimulation equivalence of differential-algebraic systems," *Int. J. Control*, vol. 91, no. 1, pp. 45–56, 2018.
- [35] A. J. van der Schaft, "Equivalence of dynamical systems by bisimulation," *IEEE Transactions on Automatic Control*, vol. 49, 2004.
- [36] G. J. Pappas, "Bisimilar linear systems," *Automatica*, vol. 39, no. 12, pp. 2035–2047, 2003.
- [37] A. Girard and G. Pappas, "Approximate Bisimulations for Nonlinear Dynamical Systems," in *CDC*, 2005, pp. 684–689.
- [38] G. J. Pappas, G. Lafferriere, and S. Sastry, "Hierarchically consistent control systems," *IEEE TAC*, vol. 45, no. 6, pp. 1144–1160, 2000.
- [39] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, 2006.



Stefano Tognazzi is a post doctoral researcher at the University of Konstanz. He received his Master and Ph.D. in Computer Science from the University of Udine, Italy and IMT Lucca, Italy, in 2015 and 2018, respectively.



Mirco Tribastone is a Professor at IMT Lucca, Italy. Prior to joining IMT Lucca he was Associate Professor at the University of Southampton, UK, and Assistant Professor at the Ludwig-Maximilians University of Munich, Germany. He received his Ph.D. in Computer Science from the University of Edinburgh, UK, in 2010. He graduated in Computer Engineering at the University of Catania, Italy.



Max Tschaikowski is a Poul Due Jensen Associate Professor at Aalborg University, Denmark. Prior to it, he was a Lise Meitner Fellow at TU Wien, Austria, an Assistant Professor at IMT Lucca, Italy, a Research Fellow at the University of Southampton, UK, and a Research Assistant at the Ludwig-Maximilians University in Munich, Germany. He was awarded a Diplom in mathematics (equivalent to a Master) and a Ph.D. in computer science by the LMU in 2010 and 2014, respectively.



Andrea Vandin is a tenure-track Assistant Professor at Sant'Anna School for Advanced Studies, Pisa, Italy, and an Adjunct Associate Professor at DTU Technical University of Denmark. Prior to it he was an Associate Professor at DTU and an Assistant Professor at IMT Lucca, Italy. In 2013–2015 he was a Senior Research Assistant at the University of Southampton, UK. He received his PhD in Computer Science and Engineering from IMT Lucca, Italy. He graduated in Computer Science at the University of Pisa, Italy.