Efficient Numerical Integration of Dynamical Systems based on Structural-Algebraic Regularization avoiding State Selection

Lena Scholz Andreas Steinbrecher
Technical University Berlin, Department of Mathematics
Str. des 17. Juni 136, 10623 Berlin

Abstract

Differential-algebraic equations naturally arise in the modeling of dynamical processes, in particular using MODELICA as modeling language. In general, the model equations can be of higher index, i.e., they can contain *hidden constraints* which lead to instabilities and order reductions in the numerical integration. Therefore, a regularization or remodeling of the model equations is required. One way to obtain the required information on the hidden constraints is a structural analysis based on the sparsity pattern of the system. For the determination of a regular index-reduced system formulation then, usually, a crucial step is the so-called *state selection*. In this paper, we will present a new approach for the remodeling of dynamical systems that uses the information obtained from the structural analysis to construct a regularized overdetermined system formulation. This overdetermined system can then be solved using specially adapted numerical integrators, in such a way that the state selection can be performed within the numerical integrator during runtime of the simulation.

Keywords: DAEs; regularization; structural analysis; overdetermined system; state selection

1 Introduction

The MODELICA language is a common tool for modeling of dynamical processes. In general, the model equations that describe the dynamical process consist of differential equations in combination with algebraic constraints, i.e., we have to deal with so-called *differential-algebraic equations* (DAEs).
The solutions of such systems have to satisfy the algebraic constraints, but, in general, not all constraints are stated in an explicit way. In particular, if the resulting system of DAEs is of higher index there exist so-called *hidden constraints* and the numerical treatment leads to instabilities, inconsistencies and possibly non-convergence of the numerical methods, see [2, 4, 6, 8]. Thus, a *regularization* or *remodeling* of the model equations is required to guarantee stable and robust numerical computations, see also [3, 6, 8, 15].

The current state of the art in many modeling and simulation tools to deal with high index DAEs is to use some kind of structural analysis based on the sparsity pattern of the system. Here, generic structural information is used to identify the constraints, to determine the index of the system, and to compute an index-reduced system model. Hereby, a crucial step is the so-called *state selection* that is required in order to introduce new algebraic variables (the so-called *dummy derivatives*) for the selected differential components of the DAE system in order to obtain a regular index-reduced formulation.

In this paper, we present a new regularization approach for the remodeling of dynamical systems that uses the information provided by the structural analysis, in particular by the Signature Method [12], to construct an overdetermined system regularization that can be solved using a specially adapted numerical integrator. This approach has the great advantage that the problem of state selection can be moved within the numerical integrator and can therefore be performed during the runtime of the simulation.

In the following, we consider *quasi-linear DAEs* of the form

$$E(x,t)\dot{x} = k(x,t), \tag{1}$$

on the domain $\mathbb{I} = [t_0, t_f]$ with initial values $x(t_0) = x_0 \in \mathbb{R}^n$, where $E \in \mathscr{C}(\mathbb{R}^n \times \mathbb{I}, \mathbb{R}^{n,n})$ is called the *leading matrix* of the quasi-linear DAE and $k \in \mathscr{C}(\mathbb{R}^n \times \mathbb{I}, \mathbb{R}^n)$ its *right-hand side*. Furthermore, $x : \mathbb{I} \to \mathbb{R}^n$ represent the *unknown variables*. The DAE system (1) is assumed to be uniquely solvable and nonredundant. Furthermore, we assume that the rank of the leading matrix E is constant for

DOI
10.3384/ECP140961171

all $(x,t) \in \mathbb{R}^n \times \mathbb{I}$ and that the rank of the partial derivatives of the (hidden) constraints with respect to x is constant for all consistent $(x,t) \in \mathbb{R}^n \times \mathbb{I}$. Note that in general these assumptions are not necessary and can be relaxed, see [16].

2 Structural Analysis of DAEs

In many simulation environments like DYMOLA, OPENMODELICA or MAPLESIM a structural analysis is used to reduce the index of the DAE system relying on its sparsity structure, e.g., there are various versions and extensions of *Pantelides Algorithm* [10] in combination with the *Dummy Derivative Approach* [9], or the *Signature Method* [12]. These structural approaches have the great advantage that fast and efficient linear optimization algorithms based on graph theoretical concepts can be used and further structural information like a block lower triangular form of the system can be extracted which is essential for efficient and fast computations.

In this section, we will shortly review the basic steps of the Signature Method (Σ-method) introduced in [12]. For ease of representation we write the model equations (1) as

$$F(t,x,\dot{x}) = 0 \tag{2}$$

with $F(t,x,\dot{x}) := E(x,t)\dot{x} - k(x,t)$, where $F \in \mathscr{C}(\mathbb{I} \times \mathbb{R}^n \times \mathbb{R}^n, \mathbb{R}^n)$, and we denote by F_i the components of the vector F and by x_j the components of the vector x. Then, the Σ-method consists of the following steps:

1. Built the *signature matrix* $\Sigma = [\sigma_{ij}]_{i,j=1,\dots,n}$

$$\sigma_{ij} := \begin{cases} \text{highest order of derivative of } x_j \text{ in } F_i, \\ -\infty \text{ if } x_j \text{ does not occur in } F_i. \end{cases}$$

2. Find a *highest value transversal* (HVT) of Σ, i.e., a *transversal* T of Σ

$$T = \{(1,j_1),(2,j_2),\dots,(n,j_n)\},$$

where $(j_1,\dots,j_n)$ is a permutation of $(1,\dots,n)$, with maximal value $Val(T) = \sum_{(i,j)\in T} \sigma_{ij}$.

3. Compute the *offsets vectors* c and d with $c_i \geq 0$ such that

$$\begin{aligned} d_j - c_i &\geq \sigma_{ij} \text{ for all } i,j = 1,\dots,n, \\ d_j - c_i &= \sigma_{ij} \text{ for all } (i,j) \in T. \end{aligned} \tag{3}$$

4. Form the Σ-*Jacobian* $\mathfrak{J} = [\mathfrak{J}_{ij}]_{i,j=1,\dots,n}$, with

$$\mathfrak{J}_{ij} := \begin{cases} \frac{\partial F_i}{\partial x_j^{(\sigma_{ij})}} & \text{if } d_j - c_i = \sigma_{ij}, \\ 0 & \text{otherwise.} \end{cases}$$

5. Built the *reduced derivative array* $\mathscr{F}(t,\mathscr{X}) = 0$ consisting of

$$\begin{aligned} F_i(t,x,\dot{x}) &= 0, \\ \frac{d}{dt}F_i(t,x,\dot{x}) &= 0, \\ &\vdots \\ \frac{d^{(c_i)}}{dt^{(c_i)}}F_i(t,x,\dot{x}) &= 0 \end{aligned}$$

for all $i = 1,\dots,n$ with

$$\mathscr{X} = [x_1,\dot{x}_1,\dots,x_1^{(d_1)},\dots,\ x_n,\dot{x}_n,\dots,x_n^{(d_n)}]^T.$$

6. Success check: if the algebraic system $\mathscr{F}(t^*,\mathscr{X}^*) = 0$ has a solution $(t^*,\mathscr{X}^*) \in \mathbb{I} \times \mathbb{R}^{n+\Sigma_{i=1}^n d_i}$ and $\mathfrak{J}$ is nonsingular at $(t^*,\mathscr{X}^*)$, then the Σ-method succeeds.

If the Σ-method succeeds, it allows to determine the *structural index* of the DAE as

$$\nu_S := \max_i c_i + \begin{cases} 0 & \text{if all } d_j > 0, \\ 1 & \text{if some } d_j = 0. \end{cases}$$

We call $\mathfrak{J}$ the Σ-Jacobian since it is in general not the analytical Jacobian, but defined by the offset vectors. The HVT as well as the offset vectors can be computed efficiently by solving a *linear programming problem* (LPP) and the corresponding dual problem, see [12]. Note that usually there is not only one uniquely determined HVT, and also the offset vectors c and d are not uniquely defined by the conditions (3). However, there exists a unique element-wise smallest solution of the dual problem, the so-called *canonical offsets*, that is independent of the chosen HVT.

If the Σ-method succeeds for a given system (2) at a consistent point, the canonical offset vector c gives the required information which equations have to be differentiated and how many times in order to be able to extract all hidden constraints. Thus, the reduced derivative array $\mathscr{F}$ can be obtained by adding the derivatives of F_i up to order c_i to the original system for all $i = 1,\dots,n$.

Example 2.1 *We illustrate the steps of the Σ-method for the example of the simple pendulum of mass $m = 1$,* *length $\ell > 0$ under gravity $\mathfrak{g}$, see also [12]. The system equations are given by*

$$\begin{aligned}
F_1(t,x,\dot{x}) &= \dot{p}_1 - q_1 &&= 0,\\
F_2(t,x,\dot{x}) &= \dot{p}_2 - q_2 &&= 0,\\
F_3(t,x,\dot{x}) &= \dot{q}_1 + 2p_1\lambda &&= 0,\\
F_4(t,x,\dot{x}) &= \dot{q}_2 + 2p_2\lambda + \mathfrak{g} &&= 0,\\
F_5(t,x,\dot{x}) &= p_1^2 + p_2^2 - \ell^2 &&= 0,
\end{aligned} \tag{4}$$

with $x = \begin{bmatrix} p_1 & p_2 & q_1 & q_2 & \lambda \end{bmatrix}^T$. The signature matrix for this system is given by

$$\Sigma = \begin{bmatrix} \colorbox{lightgray}{1} & - & \colorbox{cyan}{0} & - & - \\ - & \colorbox{cyan}{1} & - & \colorbox{lightgray}{0} & - \\ 0 & - & \colorbox{lightgray}{1} & - & \colorbox{cyan}{0} \\ - & 0 & - & \colorbox{cyan}{1} & \colorbox{lightgray}{0} \\ \colorbox{cyan}{0} & \colorbox{lightgray}{0} & - & - & - \end{bmatrix},$$

where the two possible HVTs are marked by gray and blue boxes. (Here, the entry $-$ stands for $-\infty$.) The canonical offset vectors are given by $c = [1,1,0,0,2]$ and $d = [2,2,1,1,0]$ (independently of the chosen HVT). The corresponding Σ-Jacobian is given by

$$\mathfrak{J} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 2p_1 \\ 0 & 0 & 0 & 1 & 2p_2 \\ 2p_1 & 2p_2 & 0 & 0 & 0 \end{bmatrix}$$

and the reduced derivative array takes the form

$$\mathscr{F}(t,\mathscr{X}) = \begin{bmatrix} \dot{p}_1 - q_1 \\ \ddot{p}_1 - \dot{q}_1 \\ \dot{p}_2 - q_2 \\ \ddot{p}_2 - \dot{q}_2 \\ \dot{q}_1 + 2p_1\lambda \\ \dot{q}_2 + 2p_2\lambda + \mathfrak{g} \\ p_1^2 + p_2^2 - \ell^2 \\ 2p_1\dot{p}_1 + 2p_2\dot{p}_2 \\ 2p_1\ddot{p}_1 + 2\dot{p}_1^2 + 2p_2\ddot{p}_2 + 2\dot{p}_2^2 \end{bmatrix} = 0. \tag{5}$$

Thus, the Σ-Jacobian $\mathfrak{J}$ is nonsingular at every consistent point and the Σ-method succeeds with $\nu_S = \max_i c_i + 1 = 3$.

The information provided by the HVT and the offset vectors can also be used to introduce new algebraic variables for selected differential variables yielding an extended square regularized system, for details see also [13]. However, it may happen that the success check of the Σ-method fails as can be seen in the following example.

Example 2.2 *Consider the simple DAE system*

$$\begin{aligned}
\dot{x}_1 &= x_3 + b_1\\
\dot{x}_2 &= x_4 + b_2\\
0 &= x_2 + x_3 + x_4 + b_3\\
0 &= -x_1 + x_3 + x_4 + b_4
\end{aligned} \tag{6}$$

which is regular and of differentiation index (d-index) 3. If we apply the Σ-method to system (6)*, we get the signature matrix*

$$\Sigma = \begin{bmatrix} \colorbox{lightgray}{1} & - & 0 & - \\ - & \colorbox{lightgray}{1} & - & 0 \\ - & 0 & \colorbox{lightgray}{0} & 0 \\ 0 & - & 0 & \colorbox{lightgray}{0} \end{bmatrix} \tag{7}$$

with marked HVT on the diagonal and canonical offset vectors $c = [0,0,0,0]$ and $d = [1,1,0,0]$. The corresponding Σ-Jacobian is given by

$$\mathfrak{J} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & -1 \end{bmatrix} \tag{8}$$

and $\mathfrak{J}$ is singular, i.e., the Σ-method fails.

Example 2.2 shows that for regular and therefore uniquely solvable systems the structural analysis can fail. In the following, systems for which the success check fails since the Σ-Jacobian is singular will be called *structurally singular*[1]. It has been shown in [13] that this is the case for certain coupled systems that are obtained by coupling semi-explicit d-index 1 subsystems, when the coupling results in redundancies or in an increase in the index. Nevertheless, the structural approach works well in many cases and for many important structures as e.g. systems in Hessenberg form, see [12].

Remark 2.3 It has been shown in [12] that Pantelides Algorithm [10] and the Signature Method described above are essentially equivalent in the sense that if they can both be applied and they both succeed (or converge) they result in the same structural index and the offset vector $c = [c_i]$ corresponds to the number of differentiations for each equation F_i as determined by Pantelides Algorithm. The advantage of using the Signature Method is the fast and efficient computation of the offset vectors via LPPs and the direct success

[1] Note that the term *structurally singular* is also used with a different meaning in other areas of research.

check (i.e., checking the regularity of the Σ-Jacobian) that allows us to use the results for further treatment. Note that Pantelides Algorithm will not converge in cases where the success check of the Signature Method fails.

3 Regularization using Overdetermined Formulations

Regularization approaches for high index DAEs like the Dummy Derivatives Approach [9] or index reduction by Minimal Extension [7] consist of adding the hidden constraints to the system equation and the selection of certain differential components that can then be replaced by new algebraic variables in order to lower the index of the system and to obtain a new regular index-reduced system formulation. Hereby, a problem is that the choice of states that are selected can change during the numerical integration (e.g., if the pendulum moves from the vertical to the horizontal position). Thus, if the state selection is performed outside the numerical integrator this often is computational inefficient. In the following, we will present a regularization of quasi-linear DAEs (1) that are of higher index, i.e., that contain hidden constraints. This regularization is based on an overdetermined system formulation in order to overcome the difficulties in the numerical simulation.

If the structural analysis presented in Section 2 succeeds, the offset vector c gives us the required information about the hidden constraints in the system. If the success check of the Σ-method fails, we can use the procedure proposed in [14, 15] to determine the hidden constraints of a quasi-linear DAE (1).

Remark 3.1 For structurally singular systems in semi-explicit form arising in coupled systems of DAEs a combined structural-algebraic approach has been proposed in [13] that can be applied in cases where the success check fails, but nevertheless allows us to use certain information provided by the structural analysis. In this way, the determination of the hidden constraints can be improved.

Let us denote the hidden constraints by

$$0 = h(x,t), \tag{9}$$

where $h\,:\,\mathbb{R}^n \times \mathbb{I} \to \mathbb{R}^{n_c}$ with $\operatorname{rank}\left(\frac{\partial h}{\partial x}(x,t)\right) = const.$ for all consistent $(x,t) \in \mathbb{R}^n \times \mathbb{I}$. Adding the hidden constraints to the quasi-linear DAE (1) leads to the overdetermined DAE

$$\begin{aligned} E(x,t)\dot{x} &= k(x,t), & \text{(10a)} \\ 0 &= h(x,t) & \text{(10b)} \end{aligned}$$

consisting of the original quasi-linear DAE (1) and all hidden constraints (9). This overdetermined formulation (10) is equivalent to the original DAE (1) in the sense that both have the same solution set. Note that the unknowns x are unchanged, i.e., a transformation of the state variables is not necessary and the number of unknowns is not increased (in contrast to the dummy derivative approach). The overdetermined formulation (10) has the advantage that all constraints are stated in explicit form, i.e., no hidden constraints exist anymore. A further advantage of the overdetermined formulation (10) is the fact that it is not necessary to apply analytical manipulations for the determination of a square and uniquely solvable system of DAEs (provided that consistent initial values are given).

Example 3.2 *For the simple pendulum the hidden constraints can be derived from the reduced derivative array* (5) *and consists of*

$$\begin{bmatrix} p_1^2 + p_2^2 - \ell^2 \\ 2p_1q_1 + 2p_2q_2 \\ -4p_1^2\lambda + 2q_1^2 - 4p_2^2\lambda - 2p_2\mathfrak{g} + 2q_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{11}$$

4 Numerical Approach

Unfortunately it is impossible to model and integrate the overdetermined formulation (10) within the common MODELICA frameworks. Therefore, the above described approach has been incorporated into a prototype MODELICA framework named `MPSSim` (`Multi-Physics System Simulation`). Here, a direct numerical integrator has been adapted for the overdetermined regularization (10).

In the following, the used adapted numerical integration scheme is exemplary illustrated for the implicit Euler method. In this case, the discretization of the overdetermined system (10) leads to the overdetermined nonlinear system

$$0 = \begin{bmatrix} E(x_k,t_k)(x_k - x_{k-1}) - \tau_k k(x_k,t_k) \\ -\tau_k h(x_k,t_k) \end{bmatrix} \tag{12}$$

to determine the next iterate x_k. Here τ_k denotes the stepsize in the integration step $k = 1,\ldots,N$ in the Euler scheme, t_k the discrete time point, and x_k the approximation of the solution $x(t_k)$ at the point t_k.

The nonlinear system (12) is no longer exactly solvable because of discretization and rounding errors during the numerical integration. Therefore, it is only possible to find an approximation $\tilde{x}_k$ which minimizes the residual $r \neq 0 \in \mathbb{R}^{n+n_C}$ with

$$r = \begin{bmatrix} r_D \\ r_C \end{bmatrix} = \begin{bmatrix} E(\tilde{x}_k,t_k)(\tilde{x}_k - x_{k-1}) - \tau_k k(\tilde{x}_k,t_k) \\ -\tau_k h(\tilde{x}_k,t_k) \end{bmatrix}$$

in a certain sense. In general, such an approximation results in a residual $r_C \neq 0$, which in turn leads to unfulfilled constraints, i.e., $0 \neq h(x_k,t_k)$, not even within machine precision. This would lead to the typical difficulties in the numerical integration of higher index DAEs, i.e., instabilities, convergence problems, inconsistencies, or the solution drifts away from the original solution manifold.

In order to avoid these problems it is necessary to make sure that the constraints are always satisfied during numerical integration. This can be achieved if the nonlinear system (12) is treated separately such that the next iterate x_k satisfies the lower part, i.e., the constraints, exactly or within a prescribed precision, while x_k yields a minimal residual in the upper part, i.e., in the differential part. The described numerical approach is implemented in the software package `QUALIDAES` (`QUAsi LInear DAE Solver`). This software package is suited for the direct numerical integration of regularized overdetermined model equations and is based on the 3-stage implicit Runge-Kutta method of type Radau IIa of order 5, see [5, 6]. `QUALIDAES` is integrated as numerical solver into the `MPSSim` framework. In the current version the user has to provide the model equations already given in overdetermined regularized form (10) formulated as MODELICA model. Then, using the translator `MO2FOR` [1] a FORTRAN source code is generated that can be used to solve the model equations with the solver `QUALIDAES`. The FORTRAN source code is automatically compiled and linked to the solver `QUALIDAES`. In Figure 1 the approach for the numerical treatment of models defined in MODELICA using `MPSSim` is illustrated. Note that within this framework, it is not necessary to determine a dynamic (state) selector, since this is achieved automatically within the separated treatment of (12) by its numerical solution, as described above. For a convenient usage also a graphical user interface (GUI) has been implemented in `Matlab` (see Figure 2) allowing the graphical representation of the obtained numerical results and can be used for further post-processing.

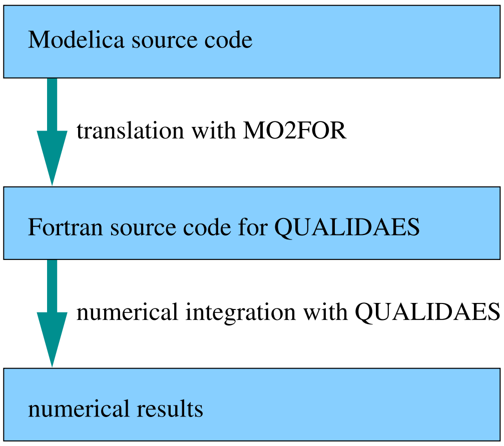

Modelica source code
translation with MO2FOR
Fortran source code for QUALIDAES
numerical integration with QUALIDAES
numerical results

Figure 1: Scheme of `MPSSim`

5 Numerical Example

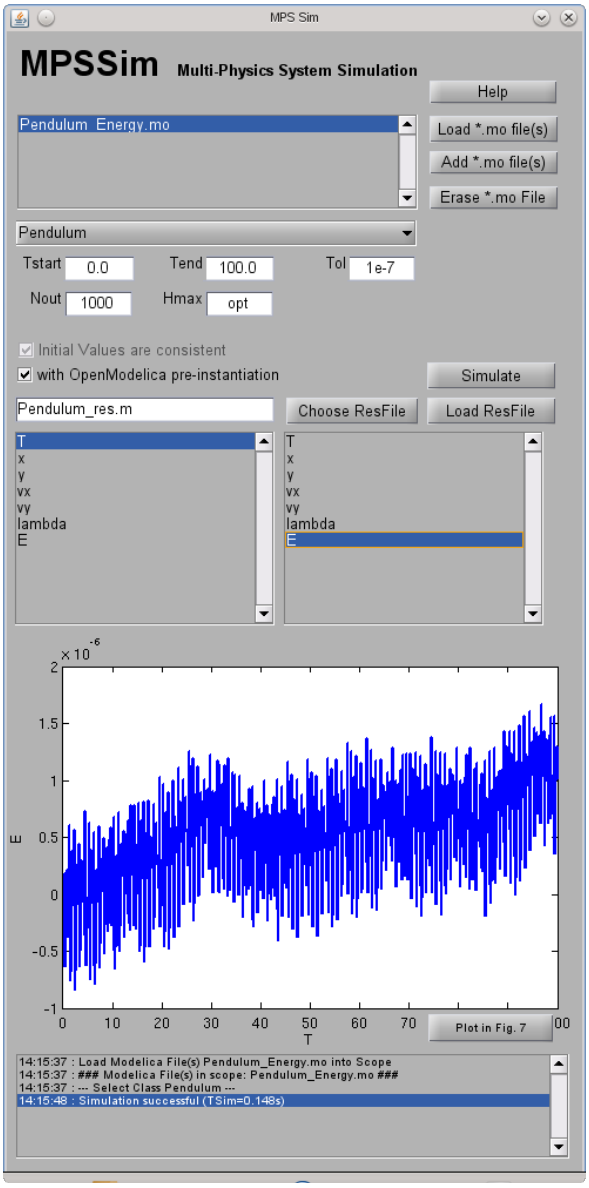
To show the promising performance of integrating a DAE system using `MPSSim` with an overdetermined system formulation we have compared the simulation of the simple pendulum equations given in Example 2.1 using `MPSSim`, `MapleSim`, `Dymola` and `OpenModelica`. In order to have a measurement for the error we include another equation in the system describing the total energy

$$E = \frac{1}{2}m(q_1^2 + q_2^2) + m\mathfrak{g}p_2$$

that should be preserved for all $t \in \mathbb{I}$ and every solution of the system (4). We use a gravitational constant of $\mathfrak{g} = 13.7503716373294544\frac{m}{s^2}$ to ensure a time period of $T = 2s$ for the motion of the pendulum and a mass of $m = 1kg$ as well as a length of $\ell = 1m$. At first we simulate the system for $t \in [0s, 100s]$ with given (fixed) consistent initial conditions

$$\begin{aligned} p_1(0) = 1, \quad p_2(0) = 0, \quad q_1(0) = 0, \\ q_2(0) = 0, \quad \lambda(0) = 0, \quad E(0) = 0, \end{aligned}$$

and a prescribed error tolerance of 10^{-7} (for both the absolute and relative error). In the simulation with `MPSSim` we solve the overdetermined formulation (4) together with (11) containing all hidden constraints, while the other simulation tools use the original d-index 3 formulation (4) and the index reduction is performed within the tool using different index reduction strategies. The values $E(t_f)$ of the total energy at the final time point $t_f = 100s$ together with the required CPU times needed for the integration are listed in Table 1. In `Dymola` a modified version of the multi-step solver Dassl is used. Here, quite a large number of state selections are required (alternating selecting the



Pendulum_Energy.mo

Pendulum

Tstart 0.0 Tend 100.0 Tol 1e-7

Nout 1000 Hmax opt

☑ Initial Values are consistent

☑ with OpenModelica pre-instantiation

Pendulum_res.m

```
14:15:37 : Load Modelica File(s) Pendulum_Energy.mo into Scope
14:15:37 : ### Modelica File(s) in scope: Pendulum_Energy.mo ###
14:15:37 : --- Select Class Pendulum ---
14:15:48 : Simulation successful (TSim=0.148s)
```

Figure 2: Matlab-GUI of MPSSim

states p_1 and q_1, or p_2 and q_2). In MapleSim we use once the provided solver CK45 suited for semi-stiff problems, which is usually less accurate but faster, and once the Rosenbrock method suited for stiff systems, which yields a higher accuracy at the expense of more required CPU time. Furthermore, the corresponding results obtained for a long time simulation for $t \in [0s, 1000s]$ are listed in Table 2. Note that OpenModelica fails to integrate the system in this case.

Simulation tool	$E(t_f)$	CPU time
MPSSim	$1.13 \cdot 10^{-6}$	$0.148s$
MapleSim (Rosenbrock)	$0.00 \cdot 10^{-0}$	$4.484s$
MapleSim (CK45)	$1.50 \cdot 10^{-5}$	$1.604s$
Dymola	$2.14 \cdot 10^{-3}$	$0.890s$
OpenModelica	$-1.60 \cdot 10^{-3}$	$2.938s$

Table 1: Simulation result of the pendulum equation with energy conservation for $t \in [0s, 100s]$

Simulation tool	$E(t_f)$	CPU time
MPSSim	$1.89 \cdot 10^{-5}$	$1.364s$
MapleSim (Rosenbrock)	$5.00 \cdot 10^{-6}$	$45.358s$
MapleSim (CK45)	$1.49 \cdot 10^{-4}$	$15.645s$
Dymola	$2.14 \cdot 10^{-2}$	$8.830s$
OpenModelica	—	—

Table 2: Simulation result of the pendulum equation with energy conservation for $t \in [0s, 1000s]$

Comparing the obtained results one can see that the projection strategy onto the constraint manifold used within MapleSim yields an accurate numerical solution, while the numerical results obtained with Dymola are less accurate. The numerical results obtained with MPSSim are accurate within the range of the prescribed error tolerance at low computational costs. However, note that using MPSSim only the costs for the numerical integration of the overdetermined system are measured, while the CPU times of the other tools also contains the costs for index reduction, state selection, projection, and further transformations.

6 Conclusions

In this article we have discussed the efficient and robust numerical simulation of dynamical systems that are modeled with Modelica. We have presented a regularization method for quasi-linear DAEs that is based on an overdetermined system formulation that is obtained by adding all hidden constraints explicitly to the original model equation. The information on the hidden constraints can be obtained from a structural analysis of the system. If a structural analysis cannot be applied these information can be obtained in an analytical way. The overdetermined system formulation can then directly be integrated using a specially adapted numerical integrator. The great advantage of the direct discretization of the overdetermined formulation is the fact that it is not necessary to determine a selector analytically in advance and that the number of unknowns in the DAE is not increased. A further advantage of an overdetermined regularization with respect to the numerical integration is the possibility to add solution invariants, e.g., mass, impulse or energy conservation laws, to the constraints, which often stabilizes numerical integration. Performing the state selection within the numerical integrator also allows us to switch between different state selections and also opens the door to handle structure varying system models [11]. Currently, no MODELICA simulation framework is able to handle overdetermined system formulations. Therefore, a prototype MODELICA framework `MPSSim` is presented that includes a translator `MO2FOR` that is used to translate an overdetermined system model provided in MODELICA into FORTRAN source code which can then be integrated using the software package `QUALIDAES`. `MPSSim` is still at an early state of development and will be continuously improved.

Acknowledgements

This work has been supported by the European Research Council through Advanced Grant *MODSIM-CONMP*.

References

[1] R. Altmeyer and A. Steinbrecher. Regularization and numerical simulation of dynamical systems modeled with Modelica. Preprint 29-2013, Institut für Mathematik, TU Berlin, 2013.

[2] K.E. Brenan, S.L. Campbell, and L.R. Petzold. *Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*, volume 14 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 1996.

[3] C.W. Gear. Differential-algebraic equation index transformations. *SIAM Journal on Scientific and Statistic Computing*, 9:39–47, 1988.

[4] E. Griepentrog and R. März. *Differential-Algebraic Equations and Their Numerical Treatment*, volume 88 of *Teubner-Texte zur Mathematik*. BSB B.G.Teubner Verlagsgesellschaft, Leipzig, 1986.

[5] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. Springer-Verlag, Berlin, Germany, 1989.

[6] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems*. Springer-Verlag, Berlin, Germany, 2nd edition, 1996.

[7] P. Kunkel and V. Mehrmann. Index reduction for differential-algebraic equations by minimal extension. *Zeitschrift für Angewandte Mathematik und Mechanik*, 84(9):579–597, 2004.

[8] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations. Analysis and Numerical Solution*. EMS Publishing House, Zürich, Switzerland, 2006.

[9] S. Mattsson and G. Söderlind. Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal on Scientific and Statistic Computing*, 14:677–692, 1993.

[10] C.C. Pantelides. The consistent initialization of differential-algebraic systems. *SIAM Journal on Scientific and Statistic Computing*, 9:213–231, 1988.

[11] P. Pepper, A. Mehlhase, Ch. Höger, and L. Scholz. A compositional semantics for Modelica-style variable-structure modeling. In P. Fritzson F.E. Cellier, D. Broman and E.A. Lee, editors, *4th International Workshop on Equation-Based Object-oriented Modeling Languages and Tools (EOOLT 2011)*, number 56 in Linköping Electronic Conference Proceedings, pages 45–54, Zurich, Switzerland, September 2011. September 5, 2011.

[12] J. Pryce. A simple structural analysis method for DAEs. *BIT Numerical Mathematics*, 41:364–394, 2001.

[13] L. Scholz and A. Steinbrecher. A combined structural-algebraic approach for the regularization of coupled systems of DAEs. Preprint 30-2013, Institut für Mathematik, TU Berlin, 2013.

[14] A. Steinbrecher. Analysis of quasi-linear differential-algebraic equations. Preprint 11-2006, Institut für Mathematik, TU Berlin, 2006.

[15] A. Steinbrecher. *Numerical Solution of Quasi-Linear Differential-Algebraic Equations and Industrial Simulation of Multibody Systems*. PhD thesis, Technische Universität Berlin, 2006.

[16] A. Steinbrecher. Deflating type regularization method for quasi-linear differential-algebraic equations. Preprint, Institut für Mathematik, TU Berlin, 2013. In preparation.