A note on POD model reduction methods for DAEs

Falk Ebert*

Berlin, December 20, 2006

Contents

AMS Subject Classification: 65L20, 65L80
Keywords: differential-algebraic equations, POD, model reduction

Abstract

We review some known results for POD model reduction applied to ODEs. Then, these results are generalized to several types of DAEs. We provide algorithms for the model reduction and error bounds for the reduced order models. Some limits of the approach are pointed out and alternative methods for reduced order subspace approximation are presented. The POD approach is tested and evaluated for a medium sized DAE example from multibody dynamics.

*Institut für Mathematik, MA 3-3, Technische Universität Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany, ebert@math.tu-berlin.de. Supported by Matheon, the DFG Research Center "Mathematics for key technologies".

1 Introduction

The proper orthogonal decompositon (POD) model order reduction approach has been a useful tool in many applications, cf. [21, 25, 27]. Originating from data representation problems, it has offered possiblilities for the state space reduction of ordinary differential equations of the form

$$\dot{y} = f(t, y), \tag{1}$$

where $y \in \mathcal{C}^1([0, t_{end}] \to \mathbb{R}^n)$,with given $y(0) = y_0$ and where f satisfies a Lipschitz condition with respect to the second argument. Its mode of application is rather general, which distinguishes this method from Krylov-space based methods or Balanced Truncation, which all are intended for linear, time-invariant systems

$$E\dot{x} = Ax + Bu, \tag{2a}$$

$$y = Cx, \tag{2b}$$

where $E, A \in \mathbb{R}^{n,n},\ B \in \mathbb{R}^{n,m},\ C \in \mathbb{R}^{p,n}$ and usually $m, p \ll n$, with possibly singular E. The functions u and y are called control and output of the system, respectively. For a detailed study of model order reduction of systems of the form (2) see [3].

The great disadvantage of POD based model reduction is that in order to be able to reduce the size of a system the solution of the system, or an approximation thereof has to be known a priori. Only recently in [26], error bounds and a perturbation analysis for POD model reduction have been presented. Still, there POD is only applied to ordinary differential equations (ODEs). However, the state of the art in modelling constrained multibody systems, electrical circuits and chemical processes among many others leads to differential-algebraic equations (DAEs),

$$0 = F(t, \dot{x}, x). \tag{3}$$

DAEs not only model the dynamic behaviour of a system but explicitly contain constraints on the system. This significantly distinguishes them from ODEs and requires specific treatment, both analytically and numerically, see [9, 12, 15, 19, 23, 24]. The index of a DAE is a measure of how much the DAE deviates from an ODE and it also serves as an indicator of the difficulties involved in the numerical solutions of DAEs. While many index concepts exist, under common assumptions they appear to be more or less equivalent. Hence, here we will limit our considerations to the differentiation- or d-index introduced in [11]. Recent results in the study of some structured DAEs suggest that the appearance of DAEs of index higher than one may be avoided already in the modelling phase or the equation setup for certain systems, namely multibody systems and electrical circuits, see [7, 29]. In other cases it is possible to algebraically reduce the index of the DAE to one, see [19]. Thus, we will restrict ourselves to the study of DAEs of differentiation index 1.

The present paper tries to present methods and error bounds for the application of POD model order reduction to some types of DAEs. We will first construct an error bound for the ODE case. This error bound is simpler than the one presented in [26], but possibly also less accurate. This result will then be applied to semi-explicit and to linear implicit DAEs. For the latter case, an algorithm will be provided that lists the steps necessary for the reduction of these DAEs. Also, results from model order reduction for linear systems with constant coefficients will be applied to nonlinear DAEs to present an alternative to the often costly computation of snapshots for the POD approach.

2 POD model reduction for ODEs

2.1 The POD method in data representation

The POD method has been conceived as a method for finding approximating subspaces for sets of data points. Here, the task is to find an optimal approximating subspace for the trajectory x, i.e. to determine a projector $\mathbb{P}$ of rank m such that

$$\|x - \mathbb{P}x\|_2^2 = \int_0^{t_{end}} \|x(t) - \mathbb{P}x(t)\|_2^2 \, dt \tag{4}$$

is minimized. A necessary condition for the existence of the integral is that $x \in L_2([0, t_{end}] \to \mathbb{R}^n)$, but since the considered trajectories are solutions of ODEs or DAEs of d-index one, we assume $x \in \mathcal{C}^1([0, t_{end}] \to \mathbb{R}^n)$, cf. [19], which for finite time intervals is contained within the respective L_2 space. Equation (4) means that $\mathbb{P}$ projects x on a subspace of dimension m and $\mathbb{P}$ has to be found such that x deviates from this subspace as little as possible.
For the solution of this problem, one needs to consider the correlation matrix (or Gramian)

$$M = \int_0^{t_{end}} x(t)x^T(t) \, dt.$$

This matrix is symmetric and positive semidefinite by construction. Hence, it has an orthogonal set of eigenvectors. Let

$$\begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \begin{bmatrix} P^T \\ Q^T \end{bmatrix} \tag{5}$$

be the spectral decomposition of M, with $\Lambda_1 = \text{diag}(\lambda_1, \ldots, \lambda_m)$ and $\Lambda_2 = \text{diag}(\lambda_{m+1}, \ldots, \lambda_n)$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Note that, since M is symmetric positive definite, the spectral decomposition (5) and the singular value decomposition of M coincide.
The optimal approximating subspace of dimension m is given by the columns of P and the projector $\mathbb{P}$ in (4) can be computed as $\mathbb{P} = PP^T$, cf. [18]. The minimum in (4) is then

$$\|x - \mathbb{P}x\|_2^2 = \sum_{i=m+1}^{n} \lambda_i = \text{trace } \Lambda_2.$$

Depending on what error is 'acceptable', m can be chosen such that $\|x - \mathbb{P}x\|_2^2$ lies below a prescribed threshold.

2.2 Subspace approximation for ODEs

The idea behind POD model reduction for ODEs is to restrict the differential equation to the approximating subspace spanned by the orthogonal columns of P. This approach, known as Galerkin projection, has been investigated e.g. in [2, 26].
Given an ordinary differential equation of the form

$$\dot{y} = f(t, y), \tag{6}$$

with initial value $y(0) = y_0$, where f is continuous in t and Lipschitz continuous with respect to y, then a unique continuously differentiable solution exists on the bounded interval $[0, t_{end}]$. Assuming that we know the solution or an approximation thereof, we can compute the Gramian and its decomposition (5). Then, the Galerkin projection is obtained by setting $y = Pu$ and multiplying (6) with P^T.

$$\dot{u} = P^T f(t, Pu) \tag{7}$$

With P constant, the new right hand side retains the smoothness properties of f and so the reduced state u lies in $\mathcal{C}^1([0, t_{end}]) \to \mathbb{R}^m)$, cf. [4], which hopefully is of much smaller dimension than n.
For the development of an error bound, we first restrict ourselves to the simplified case of a linear ODE with constant coefficients,

$$\dot{y} = Ay + \hat{f}(t), \quad y(0) = y_0. \tag{8}$$

The Galerkin projection of (8) is

$$\dot{u} = P^T A P u + P^T \hat{f}(t), \quad u(0) = u_0 = P^T y_0. \tag{9}$$

We are interested in the error of the approximated solution $\hat{y} = Pu$ with respect to the solution y of (8).

$$\begin{aligned}
\|y - \hat{y}\|_2^2 &= \|y - Pu\|_2^2 \\
&= \int_0^{t_{end}} (y(t) - Pu(t))^T (y(t) - Pu(t)) dt \\
&= 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T (\frac{d}{ds} y(s) - P\frac{d}{ds} u(s)) ds \right) dt
\end{aligned}$$

Replacing $\frac{d}{ds}y(s)$ and $\frac{d}{ds}u(s)$ by the expressions (8) and (9) yields

$$\|y - \hat{y}\|_2^2 = 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T (Ay(s) + \hat{f}(s)) - PP^T APu(s) - PP^T \hat{f}(s))) ds \right) dt.$$

We set

$$\mathbb{P} = PP^T, \quad I - \mathbb{P} = \mathbb{Q} = QQ^T, \tag{10}$$

with Q as in (5). By construction, $\mathbb{P}$ is a projector and $\mathbb{Q}$ its complementary projector. Then, we have

$$\begin{aligned}
\|y - \hat{y}\|_2^2 &= 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T (\mathbb{P} + \mathbb{Q})(Ay(s) + \hat{f}(s)) - \mathbb{P}APu(s) - \mathbb{P}\hat{f}(s)) ds \right) dt \\
&= 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T (\mathbb{P}A(y(s) - Pu(s)) + \mathbb{Q}(Ay(s) + \hat{f}(s))) ds \right) dt \\
&= 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T \mathbb{P}A(y(s) - Pu(s)) \, ds \right) dt \\
&\quad + 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T \mathbb{Q}(Ay(s) + \hat{f}(s)) ds \right) dt
\end{aligned}$$

With (8) and $P^T\mathbb{Q} = 0$, the second integral simplifies to $2\int_0^{t_{end}} \left(\int_0^t y^T(s)\mathbb{Q}\frac{d}{ds}y(s)\, ds \right) dt$. We set

$$e^2(t) = \int_0^t (y(s) - \hat{y}(s))^T (y(s) - \hat{y}(s)) \, ds$$

where obviously $e^2(0) = 0$ and obtain

$$e^2(t_{end}) = 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T \mathbb{P}A(y(s) - Pu(s)) \, ds \right) dt + 2\int_0^{t_{end}} \left(\int_0^t y^T(s)\mathbb{Q}\frac{d}{ds}y(s) \, ds \right) dt.$$

Differentiation with respect to t_{end} yields

$$\frac{d}{dt_{end}} e^2(t_{end}) = 2\int_0^{t_{end}} (y(s) - Pu(s))^T \mathbb{P}A(y(s) - Pu(s)) \, ds + 2\int_0^{t_{end}} y^T(s)\mathbb{Q}\frac{d}{ds}y(s) \, ds.$$

The term $(y - Pu)^T\mathbb{P}A(y - Pu)$ can be bounded by $\mu(\mathbb{P}A)(y - Pu)^T(y - Pu)$. Here, $\mu(A)$ represents the *logarithmic norm* of a matrix associated with a matrix norm $\|\cdot\|$ as introduced in [13]

$$\mu(A) = \lim_{h\to 0, h>0} \frac{\|I + hA\| - 1}{h}.$$

In the case of the $\|\cdot\|_2$ norm, it satisfies $\mu(A) = \max_{x\neq 0} \frac{x^T A x}{x^T x}$.
Using $(y-Pu)^T \mathbb{P} A (y-Pu) \leq \mu(\mathbb{P}A)(y-Pu)^T(y-Pu)$ and with the definition of $e^2(t)$ we can find an upper bound for $e^2(t_{end})$ as

$$\begin{aligned} e^2(t_{end}) &\leq \exp\left(2\mu(\mathbb{P}A)t_{end}\right) e^2(0) + \exp(2\mu(\mathbb{P}A)t_{end}) \int_0^{t_{end}} \exp(-2\mu(\mathbb{P}A)t) \left(2\int_0^t y^T(s)\mathbb{Q}\frac{d}{ds}y(s)\, ds\right) dt, \\ &= \exp\left(2\mu(\mathbb{P}A)t_{end}\right) e^2(0) + \exp(2\mu(\mathbb{P}A)t_{end}) \int_0^{t_{end}} \exp(-2\mu(\mathbb{P}A)t) y^T(t)\mathbb{Q}y(t)\, dt. \end{aligned}$$

For details on bounds of the solution of ODEs see e.g. [32].
Since $\mu(\mathbb{P}A)$ may be negative, the term $\exp(-2\mu(\mathbb{P}A)t)$ cannot be bounded by 1 but instead by $\max(1, \exp(-2\mu(\mathbb{P}A)t_{end}))$. Hence,

$$\exp(2\mu(\mathbb{P}A)t_{end}) \int_0^{t_{end}} \exp(-2\mu(\mathbb{P}A)t) y^T(t)\mathbb{Q}y(t)\, dt \quad \leq \quad \max(1, \exp(2\mu(\mathbb{P}A)t_{end})) \int_0^{t_{end}} y^T(t)\mathbb{Q}y(t)\, dt.$$

Setting $\varepsilon = \int_0^{t_{end}} y(t)^T \mathbb{Q} y(t)\, dt = \sum_{i=m+1}^n \lambda_i$, the error can be bounded as

$$\|y - \hat{y}\|_2 \leq \sqrt{\exp\left(2\mu(\mathbb{P}A)t_{end}\right) \|\mathbb{Q}y_0\|_2^2 + \max(\exp\left(2\mu(\mathbb{P}A)t_{end}\right), 1)\varepsilon}. \tag{11}$$

In the general case, the exact solution y will usually not be available for the determination of the subspace P and some approximate solution $\bar{y}$ has to be used. In general, ε will then be larger than the optimal $\sum_{i=m+1}^n \lambda_i$, depending on the difference of y and $\bar{y}$. Assuming that an approximation $\bar{y}$ has been used to determine the projector $\mathbb{P}$ and that the correct solution y differs from $\bar{y}$ as $y = \bar{y} + \delta$, then the following holds,

$$\begin{aligned} \|y - \mathbb{P}y\|_2 &= \|\mathbb{Q}(\bar{y} + \delta)\|_2 \\ &\leq \|\mathbb{Q}\bar{y}\|_2 + \|\mathbb{Q}\delta\|_2 = \sqrt{\sum_{i=m+1}^n \lambda_i} + \|\mathbb{Q}\delta\|_2 =: \sqrt{\varepsilon}. \end{aligned}$$

If y_0 lies in the subspace P, then projection does not change the initial value $y(0)$, i.e., $\mathbb{Q}y_0 = 0$ and (11) simplifies to

$$\|y - \hat{y}\|_2 \leq \sqrt{\max(\exp\left(2\mu(\mathbb{P}A)t_{end}\right), 1)\varepsilon}. \tag{12}$$

In the general nonlinear case (6), the approach is almost identical. We have

$$\begin{aligned} \|y - \hat{y}\|_2^2 &= \|y - Pu\|_2^2, \\ &= 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T (\frac{d}{ds}y(s) - P\frac{d}{ds}u(s))ds\right) dt, \\ &= 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T (f(s, y(s)) - \mathbb{P}f(s, Pu(s)))ds\right) dt, \\ &= 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T \mathbb{P}(f(s, y(s)) - f(s, Pu(s)))ds\right) dt \\ &\quad + 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T \mathbb{Q}(f(s, y(s)) - \mathbb{P}f(s, Pu(s)))ds\right) dt. \end{aligned} \tag{13}$$

The second summand simplifies to

$$\begin{aligned} 2\int_0^{t_{end}} \left(\int_0^t (y(s) - Pu(s))^T \mathbb{Q}(f(s, y(s)) - \mathbb{P}f(s, Pu(s)))ds\right) dt &= 2\int_0^{t_{end}} \left(\int_0^t y^T(s)\mathbb{Q}f(s, y)ds\right) dt, \\ &= 2\int_0^{t_{end}} \left(\int_0^t y^T(s)\mathbb{Q}\frac{d}{ds}y(s)\right) dt, \end{aligned}$$

while, using the mean value theorem, one can express $f(t,y)-f(t,Pu) = f(t,y)-f(t,\hat{y}) = f_y(t,\xi)(y-\hat{y}) = f_y(t,\xi)(y-Pu)$ for some $\xi \in [y,\hat{y}]$. Here, $\xi \in [y,\hat{y}]$ means that $\xi = y + \theta(\hat{y}-y)$ with $\theta \in [0,1]$. Then, the first summand in (13) can be written as

$$\begin{aligned} 2\int_0^{t_{end}} &\left(\int_0^t (y(s)-Pu(s))^T \mathbb{P}(f(s,y(s))-f(s,Pu(s)))ds\right)\,dt \\ &= 2\int_0^{t_{end}}\left(\int_0^t (y(s)-Pu(s))^T\mathbb{P}f_y(s,\xi(s))(y(s)-Pu(s))ds\right)dt. \end{aligned}$$

Then, one can proceed as in the linear case using

$$\hat{\mu} = \max_{t\in[0,t_{end}],\ \xi\in[y,\hat{y}]} \mu\left(f_y(t,\xi)\right),$$

where $[y,\hat{y}]$ is the convex hull of y and $\hat{y}$, to obtain the error bound

$$\|y-\hat{y}\|_2 \le \sqrt{\exp\left(2\hat{\mu}t_{end}\right)\|\mathbb{Q}y_0\|_2^2 + \max(\exp\left(2\hat{\mu}t_{end}\right),1)\varepsilon}. \tag{14}$$

or in the case of no projection error of the initial value

$$\|y-\hat{y}\|_2 \le \sqrt{\max(\exp\left(2\hat{\mu}t_{end}\right),1)\varepsilon}. \tag{15}$$

3 POD model reduction for semi-explicit DAEs

For the treatment of DAEs let us first consider linear semi-explicit DAEs in the form

$$\dot{y} = A_{1,1}y + A_{1,2}z + f_1, \tag{16a}$$
$$0 = A_{2,1}y + A_{2,2}z + f_2, \tag{16b}$$

with consistent starting values $y(0) = y_0,\ z(0) = z_0$, i.e. fulfilling (16b). We assume furthermore that the system is of d-index 1 , i.e. $A_{2,2}$ is nonsingular, cf. [19]. Since the differential and algebraic equations in this case are separated, it is possible to perform model reduction only on the dynamic part of the system which is given by

$$\dot{y} = (A_{1,1} - A_{1,2}A_{2,2}^{-1}A_{2,1})y + f_1 - A_{1,2}A_{2,2}^{-1}f_2, \tag{17a}$$

while the algebraic condition relates z and y via

$$z = -A_{2,2}^{-1}(A_{2,1}y + f_2). \tag{17b}$$

It is important, that no reduction is performed on the algebraic equations, since this may mean that algebraic equations representing physical laws are no longer fulfilled. In the higher index case, some of the differential equations will turn out to be algebraic equations, so called hidden constraints, cf. e.g. [19], and should thus be identified first.

For POD model reduction, we need to construct the correlation matrix with respect to y only, because we only want to reduce the state space of y. We proceed as in the ODE case by constructing the correlation matrix

$$M_y = \int_0^{t_{end}} y(t)y^T(t)\,dt,$$

and factorizing it as

$$M_y = \begin{bmatrix} \tilde{P} & \tilde{Q} \end{bmatrix} \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \begin{bmatrix} \tilde{P} \\ \tilde{Q} \end{bmatrix}.$$

In order to keep the initial value consistent, we add the initial value to the projection subspace with the Gram-Schmidt method, cf. [14],

$$\begin{aligned} p_0 &= \tilde{Q}\tilde{Q}^T y_0, \\ P &= \left[\frac{p_0}{\|p_0\|_2}, \tilde{P}\right]. \end{aligned}$$

The reduced system is given by

$$\begin{aligned} \dot{u} &= P^T A_{1,1} P u + P^T A_{1,2} \hat{z} + f_1 \\ 0 &= A_{2,1} P u + A_{2,2} \hat{z} + f_2. \end{aligned}$$

Then, we can formally obtain the reduced ODE for u as

$$\dot{u} = P^T (A_{1,1} - A_{1,2} A_{2,2}^{-1} A_{2,1}) P u + P^T (f_1 - A_{1,2} A_{2,2}^{-1} f_2). \tag{18}$$

As the initial value is projected onto itself, we can use the error estimate (12) and obtain

$$\|y - \hat{y}\|_2 = \|y - Pu\| \leq \sqrt{\max(\exp(2\mu(\mathbb{P}(A_{1,1} - A_{1,2}A_{2,2}^{-1}A_{2,1})t_{end}), 1)\varepsilon}.$$

For the error in the algebraic variable $\hat{z} = -A_{2,2}^{-1}(A_{2,1}\hat{y} + f_2)$, we obtain

$$\|z - \hat{z}\|_2 = \|A_{2,2}^{-1}A_{2,1}(y - \hat{y})\|_2 \leq \|A_{2,2}^{-1}A_{2,1}\|_2 \|(y - \hat{y})\|_2.$$

Hence, for $x = [y^T,\ z^T]^T$ and $\hat{x} = [\hat{y}^T,\ \hat{z}^T]^T$ we have the error estimate

$$\|x - \hat{x}\|_2 \leq \sqrt{(1 + \|A_{2,2}^{-1}A_{2,1}\|_2^2) \max(\exp(2\mu(\tilde{A})t_{end}), 1))\varepsilon},$$

with $\tilde{A} = \mathbb{P}(A_{1,1} - A_{1,2}A_{2,2}^{-1}A_{2,1})$.
In the semi-explicit nonlinear case

$$\dot{y} = f(t, y, z) \tag{19a}$$

$$0 = g(t, y, z) \tag{19b}$$

we have to require that g_z is nonsingular in order to have a d-index 1 system. Then, by use of the Implicit Function Theorem, one can locally conclude the existence of a function $z = \varphi(t, y)$. Therefore, we can set $\tilde{f}(t, y) = f(t, y, \varphi(y))$, where

$$\tilde{f}_y(t, y) = f_y(t, y, \varphi(y)) - f_z(t, y, \varphi(y)) \left(g_z(t, y, \varphi(y))\right)^{-1} f_y(t, y, \varphi(y)).$$

Thus, for the differential equation

$$\dot{y} = \tilde{f}(t, y),$$

we can apply POD model reduction

$$\dot{u} = P^T \tilde{f}(t, Pu).$$

Using (15) we have the error bound

$$\|y - \hat{P}u\|_2 = \|y - \hat{y}\|_2 \leq \sqrt{\max(\exp\left(2\hat{\mu}t_{end}\right), 1)\varepsilon}, \tag{20}$$

where

$$\begin{aligned} \hat{\mu} &= \max_{\substack{t\in[0,t_{end}] \\ \eta\in[y,\hat{y}]}} \mu\left(\tilde{f}_y(t, \eta)\right), \\ &= \max_{\substack{t\in[0,t_{end}] \\ \eta\in[y,\hat{y}] \\ \zeta\in[z,\hat{z}]}} \mu\left((f_y(t, \eta, \zeta) - f_z(t, \eta, \zeta)(g_z(t, \eta, \zeta))^{-1} g_y(t, \eta, \zeta))\right). \end{aligned}$$

For the error in z we have

$$\begin{aligned}
z - \hat{z} &= \varphi(t, y) - \varphi(t, \hat{y}), \\
&\leq \max_{\substack{t\in[0,t_{end}],\\ \eta\in[y,\hat{y}]}} \varphi_y(t, \eta)(y - \hat{y}), \\
&\leq \max_{\substack{t\in[0,t_{end}],\\ \eta\in[y,\hat{y}]}} -(g_z(t, \eta, \varphi(t, \eta)))^{-1} g_y(t, \eta, \varphi(t, \eta))(y - \hat{y}).
\end{aligned}$$

Setting $\hat{\nu} = \max\limits_{\substack{t\in[0,t_{end}]\\ \eta\in[y,\hat{y}]\\ \zeta\in[z,\hat{z}]}} -(g_z(t, \eta, \zeta))^{-1} g_y(t, \eta, \zeta)$, we have an error bound for $x = \begin{bmatrix} y \\ z \end{bmatrix}$ given by

$$\|x - \hat{x}\|_2 \leq \sqrt{(1 + \hat{\nu}^2) \max(\exp(2\hat{\mu} t_{end}), 1)\varepsilon}. \tag{21}$$

Note, that these error bounds are closely related to the asymptotic stability of the considered DAE as it has recently been derived in [20]. It can be observed that the drift between two solutions $x = \begin{bmatrix} y \\ z \end{bmatrix}$ and $\hat{x} = \begin{bmatrix} \hat{y} \\ \hat{z} \end{bmatrix}$ of (19) can be bounded by (21) if the difference of the starting values $\|y - \hat{y}\|_2^2$ is at most ε .

4 POD model reduction for linear implicit DAEs

Finally, we will consider the case of linear implicit DAEs of d-index 1

$$E(t)\dot{x} = f(t, x). \tag{22}$$

The matrix $E(t)$ is allowed to be singular but assumed to be continuously differentiable and of constant rank r. Here, d-index 1 means that the matrix $[E(t),\ f_x(t, x)]$ has full rank in the neighborhood of the solution trajectory x. In [19, 28], it has been shown that for smooth $E(t)$ of constant rank r smooth invertible matrices $S(t)$ and $T(t)$ exist such that

$$\begin{bmatrix} I_r & \\ & 0 \end{bmatrix} = S(t)E(t)T(t). \tag{23}$$

One possible factorization of this structure can be obtained from the *orthogonal standard form* in [19]. We will subsequently drop the explicit time dependency of S, E and T.

Setting $x = T \begin{bmatrix} y \\ z \end{bmatrix}$ in (22), one obtains

$$E\left(T \frac{d}{dt} \begin{bmatrix} y \\ z \end{bmatrix} + \dot{T} \begin{bmatrix} y \\ z \end{bmatrix}\right) = f\left(t, T \begin{bmatrix} y \\ z \end{bmatrix}\right).$$

Multiplication with $S(t)$ and reordering of the terms yields

$$\begin{aligned}
SET \frac{d}{dt} \begin{bmatrix} y \\ z \end{bmatrix} &= S\left(f\left(t, T \begin{bmatrix} y \\ z \end{bmatrix}\right) - E\dot{T} \begin{bmatrix} y \\ z \end{bmatrix}\right), \\
\begin{bmatrix} I_r & \\ & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} y \\ z \end{bmatrix} &= \tilde{f}\left(t, \begin{bmatrix} y \\ z \end{bmatrix}\right).
\end{aligned} \tag{24}$$

For this equation we can use the error estimate (21). The problem is that usually the formulation (24) is not used for the integration of the DAE, but rather the original equation (22). Hence, generally, x will be available, while y and z are not. The task is to determine the correlation matrix M_y with the help of the computed solution x.

$$\begin{aligned}
M_y &= \int_0^{t_{end}} y(t)y^T(t)\,dt \\
\begin{bmatrix} M_y & 0 \\ 0 & 0 \end{bmatrix} &= \int_0^{t_{end}} \begin{bmatrix} y \\ 0 \end{bmatrix} \begin{bmatrix} y^T & 0 \end{bmatrix} dt \\
&= \int_0^{t_{end}} \begin{bmatrix} I_r & \\ & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \begin{bmatrix} y^T & z^T \end{bmatrix} \begin{bmatrix} I_r & \\ & 0 \end{bmatrix} dt \\
&= \int_0^{t_{end}} S(t)E(t)T(t) \begin{bmatrix} y \\ z \end{bmatrix} \begin{bmatrix} y^T & z^T \end{bmatrix} T^T(t)E^T(t)S^T(t)dt.
\end{aligned}$$

With $x = T \begin{bmatrix} y \\ z \end{bmatrix}$ and (23), we have

$$\begin{bmatrix} M_y & 0 \\ 0 & 0 \end{bmatrix} = \int_0^{t_{end}} S(t)E(t)x(t)x^T(t)E^T(t)S^T(t)dt. \tag{25}$$

We thus can state the following POD algorithm.

Algorithm 4.1 (POD for linear implicit DAEs of d-index 1)**.**
Given the solution x of the DAE (22), we perform the following steps.

- Construct smooth nonsingular matrices $S = [S_1\ S_2]$ and $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$ such that (23) is fulfilled.
- Compute $M_y = \int_0^{t_{end}} S_1 E x x^T E^T S_1^T\, dt$.
- Compute a decomposition

$$M_y = \begin{bmatrix} P^T \\ Q^T \end{bmatrix} \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \begin{bmatrix} P & Q \end{bmatrix}.$$

- Set

$$P_L := S^T \begin{bmatrix} P & 0 \\ 0 & I_{n-r} \end{bmatrix}, \qquad P_R := T \begin{bmatrix} P & 0 \\ 0 & I_{n-r} \end{bmatrix}.$$

- The reduced order DAE is then given by

$$P_L^T E \frac{d}{dt}(P_R x_{red}) = P_L^T f(t, P_R x_{red}),$$

  with initial value $x_{red}(0) = \begin{bmatrix} P^T & 0 \\ 0 & I_{n-r} \end{bmatrix} T^{-1}(0)x_0$.

Again, it has to be noted that it is usually unrealistic to assume that the exact solution x is known. However, if a sufficiently accurate set of snapshots exists such that the approximating subspace can be computed, then the POD method is applicable. The set of snapshots may be obtained empirically, using the underlying physics or from simulation. Another approach will be presented in the following.

5 Alternative computation of the correlation matrix

Besides the approximation of the correlation matrix via snapshots, another possibility for the computation of dominant subspaces for model order reduction is known for linear dynamic and descriptor systems with constant coefficients. Instead of approximating the finite integral $M = \int_0^{t_{end}} x(t)x^T(t)dt$ with quadrature rules, it is computed analytically. Suppose that $x(t)$ solves a homogeneous descriptor system of the form

$$E\frac{d}{dt}x = Ax, \quad x(0) = x_0, \tag{26}$$

where we assume that the pencil (E, A) is of index 1. For the generalized Lyapunov equation

$$EMA^T + AME^T = E(x(t_{end})x^T(t_{end}) - x_0x_0^T)E^T. \tag{27}$$

it is known, that it has a solution M if for the finite eigenvalues of λ_j and λ_k of the pencil (E, A) it holds that $\bar{\lambda}_j + \lambda_k \neq 0$ for all $j, k = 1, \dots, n$. Furthermore, the term EME^T is unique, cf. [31]. Equation (27) can be obtained as follows

$$\frac{d}{dt}Ex(t)x^T(t)E^T = Ex(t)x^T(t)A^T + Ax(t)x^T(t)E^T.$$

After integration, we obtain

$$\int_0^{t_{end}} (\frac{d}{dt}Ex(t)x^T(t)E^T) \quad = \quad E\int_0^{t_{end}} x(t)x^T(t)dtA^T + A\int_0^{t_{end}} x(t)x^T(t)dtE^T$$

which is the assertion. For the numerical solution of (27), we apply the algorithm proposed in [30]. For simplicity, we assume that E is in the form $\begin{bmatrix} E_1 \\ 0 \end{bmatrix}$ with E_1 of full rank. This can usually be obtained by a simple transformation of the system from the left. Then EME^T is of the form $\begin{bmatrix} E_1ME_1^T & 0 \\ 0 & 0 \end{bmatrix}$ and we are only interested in the computation of $E_1ME_1^T$. We apply the QZ-algorithm, cf. [14], to the pencil (E^T, A^T) and obtain

$$QE^TZ = \begin{bmatrix} E_{11} & E_{12} \\ 0 & 0 \end{bmatrix}, \quad QA^TZ = \begin{bmatrix} A_{11} & \tilde{A}_{12} \\ 0 & A_{22} \end{bmatrix} \tag{28}$$

with orthogonal Q and Z_1 and E_{11}, A_{11}, A_{22} quasi upper triangular. Here, *quasi upper triangular* means that the considered matrices only have 1×1 or 2×2 blocks on the diagonal. We set

$$Y = \begin{bmatrix} I & -E_{11}^{-1}E_{12} \\ 0 & I \end{bmatrix}, \quad Y^{-1} = \begin{bmatrix} I & E_{11}^{-1}E_{12} \\ 0 & I \end{bmatrix}$$

and multiply the matrices (28) with Y from the right

$$QE^TZY = \begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix}, \quad QA^TZY = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}. \tag{29}$$

We transform (27) with Y^TZ^T from the left and with ZY from the right and insert $I = Q^TQ$.

$$\begin{aligned} &Y^TZ^TEQ^TQMQ^TQA^TZY + Y^TZ^TAQ^TQMQ^TQE^TZY \\ &\qquad = Y^TZ^TEQ^TQ(x(t_{end})x^T(t_{end}) - x(t_0)x^T(t_0))Q^TQE^TZY \end{aligned}$$

We set $\widehat{M} = QMQ^T$ and $\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = Qx$ and with (29) we obtain

$$\begin{aligned} &\begin{bmatrix} E_{11}^T & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \widehat{M}_{11} & \widehat{M}_{12} \\ \widehat{M}_{21} & \widehat{M}_{22} \end{bmatrix}\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} + \begin{bmatrix} A_{11}^T & 0 \\ A_{12}^T & A_{22}^T \end{bmatrix}\begin{bmatrix} \widehat{M}_{11} & \widehat{M}_{12} \\ \widehat{M}_{21} & \widehat{M}_{22} \end{bmatrix}\begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix} \\ &\qquad = \begin{bmatrix} E_{11}^T(\hat{x}_1(t_{end})\hat{x}_1^T(t_{end}) - \hat{x}_1(0)\hat{x}_1^T(0))E_{11} & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned} \tag{30}$$

Equation (30) can now be solved blockwise for $\widehat{M}_{11}$, $\widehat{M}_{12}$, $\widehat{M}_{21}$ while $\widehat{M}_{22}$ can be chsoen arbitrarily. However, we will see that only $\widehat{M}_{11}$ is needed for the computation of $E_1 M E_1^T$. The equation for $\widehat{M}_{11}$

$$E_{11}^T \widehat{M}_{11} A_{11} + A_{11}^T \widehat{M}_{11} E_{11} = E_{11}^T (\hat{x}_1(t_{end}) \hat{x}_1^T(t_{end}) - \hat{x}_1(t_0) \hat{x}_1^T(t_0)) E_{11}$$

is again a generalized Lyapunov equation but in the convenient form where E_{11} and A_{11} are of full rank and quasi upper triangular. Hence, the system is easily solvable by standard techniques such as Hammarling's algorithm, cf. [17, 22]. Now, for the computation of $E_1 M E_1^T$, with the above definitions, we have

$$\begin{aligned}
\begin{bmatrix} E_1 \\ 0 \end{bmatrix} M \begin{bmatrix} E_1^T & 0 \end{bmatrix} &= ZY^{-T} Y^T Z^T \begin{bmatrix} E_1 \\ 0 \end{bmatrix} Q^T Q M Q^T Q \begin{bmatrix} E_1^T & 0 \end{bmatrix} Z Y Y^{-1} Z^T \\
&= ZY^{-T} \begin{bmatrix} E_{11}^T & 0 \\ 0 & 0 \end{bmatrix} \widehat{M} \begin{bmatrix} E_{11} & 0 \\ 0 & 0 \end{bmatrix} Y^{-1} Z^T \\
&= ZY^{-T} \begin{bmatrix} E_{11}^T \widehat{M}_{11} E_{11} & 0 \\ 0 & 0 \end{bmatrix} Y^{-1} Z^T.
\end{aligned}$$

With

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

and the definition of Y we have that

$$M_y = E_1 M E_1^T = \begin{bmatrix} Z_{11} & Z_{12} \end{bmatrix} \begin{bmatrix} E_{11}^T \\ E_{12}^T \end{bmatrix} \widehat{M}_{11} \begin{bmatrix} E_{11} & E_{12} \end{bmatrix} \begin{bmatrix} Z_{11}^T \\ Z_{12}^T \end{bmatrix}.$$

In this final result, we see that Y does not have to be computed. The only matrices that enter the computation of M_y are E_{11}, E_{12}, A_{11}, Q and Z.

Remark 5.1. The Lyapunov approach works fine on systems of the form (26), provided that the system size is small enough so that LAPACK, [1], can be applied. However several problems arise when applying the technique to practical problems. First, the state $x(t_{end})$ is usually unknown and if it is obtained by numerical simulation, then all intermediate steps may already be used to compute an approximation to the correlation matrix with quadrature rules. The approach makes sense, if it can be expected that the system goes into a steady state within the interval $[0, t_{end}]$ and that the steady state can be computed without simulating the system.
For the system (26), $x = 0$ is always a steady state. If additionally, the matrix A is nonsingular, then it is also unique. Hence, if we set $x(t_{end}) = 0$, then the generalized Lyapunov equation (27) simplifies to

$$EMA^T + AME^T = -E x_0 x_0^T E^T.$$

The linearization of an autonomous nonlinear system around a steady state also has the advantage, that the arising linear system is in the form (26) while retaining the index of the nonlinear system, cf. Theorem 1 in [10] or [29].

For autonomous systems, we can thus state the following algorithm.

Algorithm 5.1 (Computation of the correlation matrix M_y for autonomous linear implicit DAEs with steady state)**.**
Given a DAE of the form $E\dot{x} = f(x)$ with initial value x_0 we perform the following steps.

- Compute $SE = \begin{bmatrix} E_1 \\ 0 \end{bmatrix}$
- Compute a steady state $\bar{x}$ via $0 = f(\bar{x})$.

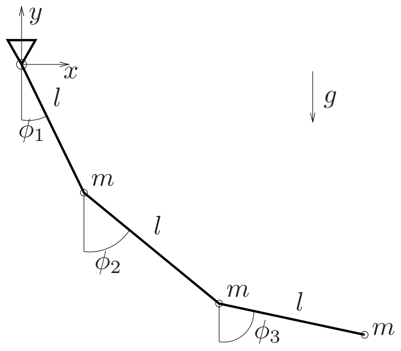

Figure 1: pendulum schematics

- Compute $A = f_x(\bar{x})$.
- Compute a QZ-decomposition of the pencil $((SE)^T, (SA)^T)$ as

$$(Q(SE)^T Z, Q(SA)^T Z) = \left(\begin{bmatrix} E_{11} & E_{12} \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \right),$$

  with

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$

- If the pencil (E_{11}, A_{11}) is unstable ABORT, otherwise compute, e.g. by the Hammarling method, the solution $\widehat{M}_{11}$ of the generalized Lyapunov equation

$$E_{11}^T \widehat{M}_{11} A_{11} + A_{11}^T \widehat{M}_{11} E_{11} = -E_{11}^T Q_1 x_0 x_0^T Q_1^T E_{11}.$$

- Compute the correlation matrix M_y as

$$M_y = Z_1 \begin{bmatrix} E_{11}^T \\ E_{12}^T \end{bmatrix} \widehat{M}_{11} \begin{bmatrix} E_{11} & E_{12} \end{bmatrix} Z_1^T.$$

Remark 5.2. If not the matrix M_y itself is needed but only a square root factor R_y such that $M_y = R_y^T R_y$, then if U_y with $U_y^T U_y$ is a Cholesky factor of $\widehat{M}_{11}$, R_y can be computed as $R_y = U_y \begin{bmatrix} E_{11} & E_{12} \end{bmatrix} Z_1^T$.

6 Multibody system example

6.1 Derivation of the equations of motion for an n-links chain

As a numerical example we consider an n-link chain modeled by an n-element pendulum as depicted in Fig. 6.1. This n-element pendulum is composed of n identical stiff massless rods of length l joined in series. At the end of each rod, i.e. in the joints and at the end of the bottom rod, a point mass m will be attached. The first rod is assumed to be attached to a joint in the origin. We denote the position of the i-th joint by $p_i = [x_i \ y_i]^T$. The position of the $(i+1)$-th joint can be obtained by rotating the vector $[0 \ l]^T$ about p_i by the angle ϕ_{i+1}. Thus, every position p_i can be expressed as

$$p_i = \left(\sum_{j=0}^{i} R(\phi_j) \right) \begin{bmatrix} 0 \\ -l \end{bmatrix},$$

with

$$R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

The velocity $\dot{p}_i$ is then given as

$$\dot{p}_i = \left(\sum_{j=0}^{i} \dot{\phi}_j Q(\phi_j) \right) \begin{bmatrix} 0 \\ -l \end{bmatrix},$$

with

$$Q(\phi) = \frac{\partial}{\partial \phi} P(\phi) = \begin{bmatrix} -\sin\phi & -\cos\phi \\ \cos\phi & -\sin\phi \end{bmatrix}.$$

To derive the equations of motion we will use the Lagrangian formalism, cf. [5]. To this end we define the Lagrangian as $L = T - U$ with kinetic energy

$$\begin{aligned} T &= \sum_{i=1}^{n} \frac{m}{2} \dot{p}_i^T \dot{p}_i \\ &= \frac{m}{2} \sum_{i=1}^{n} \begin{bmatrix} 0 \\ -l \end{bmatrix}^T \left(\sum_{j=1}^{i} \dot{\phi}_j Q^T(\phi_j) \right) \left(\sum_{j=1}^{i} \dot{\phi}_j Q(\phi_j) \right) \begin{bmatrix} 0 \\ -l \end{bmatrix} \end{aligned}$$

and potential energy

$$\begin{aligned} U &= mg \sum_{i=1}^{n} y_i \\ &= -mgl \sum_{i=1}^{n} \sum_{j=1}^{i} \cos\phi_j. \end{aligned}$$

The equations of motion for this system are obtained by applying the Euler-Lagrange operator to L,

$$0 = \left(\frac{d}{dt} \left(\frac{\partial}{\partial \dot{\Phi}} \right) - \frac{\partial}{\partial \Phi} \right) L(\dot{\Phi}, \Phi, t)$$

where $\Phi = [\phi_1, \ \ldots \ , \phi_n]^T$. We have

$$\begin{aligned} \frac{\partial}{\partial \dot{\phi}_k} T &= \frac{\partial}{\partial \dot{\phi}_k} \left(\frac{m}{2} \sum_{i=1}^{n} \begin{bmatrix} 0 \\ -l \end{bmatrix}^T \left(\sum_{j=1}^{i} \dot{\phi}_j Q^T(\phi_j) \right) \left(\sum_{j=1}^{i} \dot{\phi}_j Q(\phi_j) \right) \begin{bmatrix} 0 \\ -l \end{bmatrix} \right) \\ &= m \sum_{i=1}^{n} \begin{bmatrix} 0 \\ -l \end{bmatrix}^T \frac{\partial}{\partial \dot{\phi}_k} \left(\sum_{j=1}^{i} \dot{\phi}_j Q^T(\phi_j) \right) \left(\sum_{j=1}^{i} \dot{\phi}_j Q(\phi_j) \right) \begin{bmatrix} 0 \\ -l \end{bmatrix}. \end{aligned}$$

As $\dot{\phi}_k$ will not appear in the inner sums until the index $i = k$, we can let the index i of the outer sum start at $i = k$,

$$\frac{\partial}{\partial \dot{\phi}_k} T = m \sum_{i=k}^{n} \sum_{j=1}^{i} \begin{bmatrix} 0 \\ -l \end{bmatrix}^T Q^T(\phi_k) Q(\phi_j) \begin{bmatrix} 0 \\ -l \end{bmatrix} \dot{\phi}_j.$$

A simple computation shows that

$$\begin{bmatrix} 0 \\ -l \end{bmatrix}^T Q^T(\phi_k) Q(\phi_j) \begin{bmatrix} 0 \\ -l \end{bmatrix} = \begin{bmatrix} 0 \\ -l \end{bmatrix}^T \begin{bmatrix} \cos(\phi_k - \phi_j) & \sin(\phi_k - \phi_j) \\ -\sin(\phi_k - \phi_j) & \cos(\phi_k - \phi_j) \end{bmatrix} \begin{bmatrix} 0 \\ -l \end{bmatrix} = l^2 \cos(\phi_k - \phi_j).$$

Hence,

$$\frac{\partial}{\partial \dot\phi_k} T \quad = \quad ml^2 \sum_{i=k}^{n} \sum_{j=1}^{i} \cos(\phi_k - \phi_j)\dot\phi_j$$

and taking the time derivative

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial}{\partial \dot\phi_k} T\right) &= ml^2 \sum_{i=k}^{n} \sum_{j=1}^{i} \frac{d}{dt}(\cos(\phi_k - \phi_j)\dot\phi_j), \\ &= ml^2 \sum_{i=k}^{n} \sum_{j=1}^{i} (\cos(\phi_k - \phi_j)\ddot\phi_j - \sin(\phi_k - \phi_j)\dot\phi_j(\dot\phi_k - \dot\phi_j)). \end{aligned} \tag{31}$$

The derivative of T with respect to ϕ_k is given as

$$\begin{aligned} \frac{\partial}{\partial \phi_k} T &= \frac{\partial}{\partial \phi_k}\left(\frac{m}{2}\sum_{i=1}^{n}\left[\begin{array}{c} 0 \\ -l \end{array}\right]^T \left(\sum_{j=1}^{i}\dot\phi_j Q^T(\phi_j)\right)\left(\sum_{j=1}^{i}\dot\phi_j Q(\phi_j)\right)\left[\begin{array}{c} 0 \\ -l \end{array}\right]\right) \\ &= m\sum_{i=1}^{n}\left[\begin{array}{c} 0 \\ -l \end{array}\right]^T \frac{\partial}{\partial \phi_k}\left(\sum_{j=1}^{i}\dot\phi_j Q^T(\phi_j)\right)\left(\sum_{j=1}^{i}\dot\phi_j Q(\phi_j)\right)\left[\begin{array}{c} 0 \\ -l \end{array}\right]. \end{aligned}$$

Again, ϕ_k will not appear in the inner sum before the index $i = k$ of the outer sum, hence,

$$\begin{aligned} \frac{\partial}{\partial \phi_k} T &= m\sum_{i=k}^{n}\sum_{j=1}^{i}\dot\phi_k \left[\begin{array}{c} 0 \\ -l \end{array}\right]^T (\frac{\partial}{\partial \phi_k} Q^T(\phi_k))Q(\phi_j)\left[\begin{array}{c} 0 \\ -l \end{array}\right]\dot\phi_j \\ &= m\sum_{i=k}^{n}\sum_{j=1}^{i}\dot\phi_k \left[\begin{array}{c} 0 \\ -l \end{array}\right]^T (-R(\phi_j))Q(\phi_j)\left[\begin{array}{c} 0 \\ -l \end{array}\right]\dot\phi_j, \end{aligned}$$

because $\frac{\partial}{\partial \phi}Q(\phi) = -R(\phi)$. The product $-R(\phi_k)^T Q(\phi_j)$ can be evaluated as

$$-R(\phi_k)^T Q(\phi_j) \quad = \quad \left[\begin{array}{cc} -\sin(\phi_k - \phi_j) & \cos(\phi_k - \phi_j) \\ -\cos(\phi_k - \phi_j) & -\sin(\phi_k - \phi_j) \end{array}\right],$$

and thus,

$$\left[\begin{array}{c} 0 \\ -l \end{array}\right]^T (-R(\phi_j))^T Q(\phi_j)\left[\begin{array}{c} 0 \\ -l \end{array}\right] \quad = \quad -l^2 sin(\phi_k - \phi_j),$$

and

$$\frac{\partial}{\partial \phi_k} T \quad = \quad -ml^2 \sum_{i=k}^{n}\sum_{j=1}^{i}\dot\phi_k\dot\phi_j \sin(\phi_k - \phi_j). \tag{32}$$

The potential energy U is independent of $\dot\Phi$, so we only need to consider $\frac{\partial}{\partial \Phi} = \left[\begin{array}{ccc} \frac{\partial}{\partial \phi_1} & \dots & \frac{\partial}{\partial \phi_n} \end{array}\right]^T$,

$$\begin{aligned} \frac{\partial}{\partial \phi_k} U &= \frac{\partial}{\partial \phi_k}(-mgl\sum_{i=1}^{n}\sum_{j=1}^{i}\cos\phi_j) \\ &= mgl\sum_{i=k}^{n}\sin\phi_k = mgl(n-k+1)\sin\phi_k. \end{aligned} \tag{33}$$

Collecting equations (31), (32) and (33) we obtain

$$\begin{aligned}\left(\frac{d}{dt}\left(\frac{\partial}{\partial\dot\phi_k}\right)-\frac{\partial}{\partial\phi_k}\right)(T-U) &= ml^2\sum_{i=k}^{n}\sum_{j=1}^{i}(\cos(\phi_k-\phi_j)\ddot\phi_j-\sin(\phi_k-\phi_j)\dot\phi_j(\dot\phi_k-\dot\phi_j))\\ &\quad+ml^2\sum_{i=k}^{n}\sum_{j=1}^{i}\dot\phi_k\dot\phi_j\sin(\phi_k-\phi_j)\\ &\quad+mgl(n-k+1)\sin\phi_k.\end{aligned}$$

This can be simplified to

$$\begin{aligned}\left(\frac{d}{dt}\left(\frac{\partial}{\partial\dot\phi_k}\right)-\frac{\partial}{\partial\phi_k}\right)(T-U) &= ml^2\sum_{i=k}^{n}\sum_{j=1}^{i}(\cos(\phi_k-\phi_j)\ddot\phi_j+\sin(\phi_k-\phi_j)\dot\phi_j^2)\\ &\quad+mgl(n-k+1)\sin\phi_k.\end{aligned}$$

Divided by ml, the equations of motion are given as

$$\begin{aligned}0 &= \left(\frac{d}{dt}\left(\frac{\partial}{\partial\dot\Phi}\right)-\frac{\partial}{\partial\Phi}\right)L(\dot\Phi,\Phi,t)\\ &= M(\Phi)\ddot\Phi+D(\Phi)\dot\Phi^2+f(\Phi),\end{aligned}\tag{34}$$

with

$$\dot\Phi^2=\begin{bmatrix}\dot\phi_1^2\\ \vdots\\ \dot\phi_1^2\end{bmatrix},\quad M=[\mu_{i,j}]_{i,j=1}^n,\quad D=[\delta_{i,j}]_{i,j=1}^n,\quad f=[f_i]_{i=1}^n,$$

$$\mu_{i,j}=l\cos(\phi_i-\phi_j)(n+1-\max(i,j)),\quad \delta_{i,j}=l\sin(\phi_i-\phi_j)(n+1-\max(i,j)),\quad f_i=g(n-k+1)\sin\phi_i.$$

The equations of motion (34) are an implicit system of second order ordinary differential equations. For the treatment with many common solvers, we need to transform it into an explicit ODE of first order,

$$\frac{d}{dt}\begin{bmatrix}\Phi\\ \dot\Phi\end{bmatrix}=\begin{bmatrix}\dot\Phi\\ -M^{-1}(D\dot\Phi^2+f)\end{bmatrix}.\tag{35}$$

As this equation describes an undamped system, high frequency components will remain in the solution. However, the high frequent oscillations are usually not present in real life multi body systems. Additionally they slow down numerical integration methods. It is, thus, suggested to introduce some artificial damping, e.g. by adding a velocity dependent force term

$$\frac{d}{dt}\begin{bmatrix}\Phi\\ \dot\Phi\end{bmatrix}=\begin{bmatrix}\dot\Phi\\ -M^{-1}(D\dot\Phi^2+d\dot\Phi+f)\end{bmatrix}.\tag{36}$$

with small non-negative d.

6.2 Stiff n-element pendulum

The behaviour of the uncostrained pendulum chain is essentially chaotic and numerical simulations suggest that the angles between adjacent pairs of rods ϕ_i and ϕ_{i+1} appear to be mostly independent. We try to change this behaviour by introducing additional stiffness into the equations via rotational springs in the joints. These springs will be installed in all interior joints, i.e. not in the joint, where the chain is fastened and not at the end of the final link. In this way, large differences the angles of adjacent links

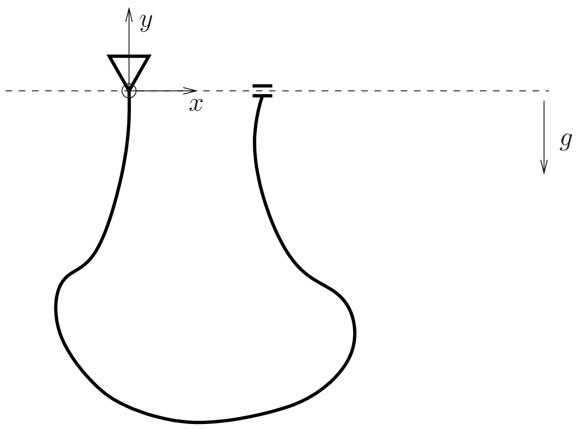

Figure 2: constrained pendulum

will be penalized by an increase of energy in the spring between both links. This changes the potential energy of the system to

$$U = -mgl \sum_{i=1}^{n} \sum_{j=1}^{i} \cos \phi_j + \frac{\kappa}{2} \sum_{i=1}^{n-1} (\phi_i - \phi_{i+1})^2.$$

The spring constant κ should be chosen depending on n. We suggest that κ is set to

$$\kappa = \kappa_0 n^4.$$

This choice reflects the bending moment of a beam which changes with the fourth power of the scaling factor if the beam is scaled uniformly, cf. [8]. In this way a chain with more elements will have a comparable overall stiffness. The equations of motion of the unconstrained pendulum chain (36) then change to

$$\frac{d}{dt} \begin{bmatrix} \Phi \\ \dot{\Phi} \end{bmatrix} = \begin{bmatrix} \dot{\Phi} \\ -M^{-1}(D\dot{\Phi}^2 + d\dot{\Phi} + K\Phi + f) \end{bmatrix}, \tag{37}$$

with the previously introduced identifiers and additionally

$$K = \kappa \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}.$$

6.3 Constrained n-element pendulum

We now additionally require that the last mass of the pendulum is attached to a translational joint keeping it at the same height as the origin as in Fig. 6.3.
This can be expressed by the constraint $0 = y_n = l \sum_{i=1}^{n} \cos \phi_i$. Then, in the formulation of constrained multi-body systems, Equation (35) becomes

$$\frac{d}{dt}\Phi = \dot{\Phi}, \tag{38a}$$
$$\frac{d}{dt}\dot{\Phi} = -M^{-1}(D\dot{\Phi}^2 + d\dot{\Phi} + K\Phi + f - G^T\lambda), \tag{38b}$$
$$0 = g(\Phi), \tag{38c}$$

with $g(\Phi) = \sum_{i=1}^{n} \cos\phi_i$ and $G = \frac{\partial}{\partial\Phi} g(\Phi) = -[\sin\phi_1 \ \ldots \ \sin\phi_n]$. The resulting DAE (38) has d-index 3 and is, thus, badly suited for numerical integration. The reason for the bad numerical behaviour is that the DAE (38) contains so-called hidden constraints $0 = g^I(\Phi, \dot\Phi)$ and $0 = g^{II}(\Phi, \dot\Phi, \lambda)$ that are obtained by differentiating $0 = g(\Phi)$ twice,

$$\begin{aligned}
g^I(\Phi, \dot\Phi) &= \frac{d}{dt} g(\Phi) = -l \sum_{i=1}^{n} \dot\phi_i \sin\phi_i, \\
g^{II}(\Phi, \dot\Phi, \lambda) &= \frac{d^2}{dt^2} g(\Phi) = -l \sum_{i=1}^{n} (\ddot\phi_i \sin\phi_i + \dot\phi_i^2 \cos\phi_i).
\end{aligned}$$

The dependency of g^{II} on λ can be seen by replacing $\ddot\phi_i$ with the right hand side in (38c). These two constraints are needed for a numerically stable computation of (38). But, simply adding g^I and g^{II} to (38) would produce a system with redundant equations. According to [29], this redundancy can be eliminated by multiplying (38a) and (38b) with $\Pi(\Phi)$ such that $\begin{bmatrix} \Pi \\ G \end{bmatrix}$ is square and invertible. For an even number n it is possible, that all rods of the chain are in vertical position pointing either up or down. Then the chain would be in a singular position and the DAE becomes underdetermined. The consequence of this is that $G = 0$ and no Π can be determined that makes $\begin{bmatrix} \Pi \\ G \end{bmatrix}$ regular. This situation can effectively be avoided by choosing an odd number n of elements. Then, it is impossible for all rods to be in a vertical position without violating the constraint.

As the entries of G change over time, the matrix Π has to be determined repeatedly during the integration. Thus, $\Pi(\Phi)$ may change discontinuously over time, but since the rank of Π is constant, this incurs no numerical problems. Following the approach in [29], the system

$$\begin{aligned}
\Pi \frac{d}{dt} \Phi &= \Pi \dot\Phi, && (39a) \\
\Pi \frac{d}{dt} \dot\Phi &= -\Pi M^{-1}(D\dot\Phi^2 + d\dot\Phi + K\Phi + f - G^T\lambda), && (39b) \\
0 &= g(\Phi), && (39c) \\
0 &= g^I(\Phi, \dot\Phi), && (39d) \\
0 &= g^{II}(\Phi, \dot\Phi, \lambda), && (39e)
\end{aligned}$$

is of d-index 1. The system (39) is thus well suited for numerical integration with common solvers for stiff ODEs, cf. [9, 16, 19].

The constant κ_0 can be used to emphasize certain aspects of the model. For small κ_0, the constrained chain will be acting in a very chaotic way due to the high degree of nonlinearity in the system, whereas for high stiffness, the pendulum chain behaves like a bending beam, which can basically be approximated by a linear model. Higher stiffness has another advantage. If the chain rests more or less in horizontal position, then the constraint matrix G becomes more predictable. Assuming that the chain is oriented to the right, in horizontal position, all angles are approximately $\frac{\pi}{2}$ and $G \approx [-1, \ldots, -1]$. For this case, the matrix Π can be chosen constant, e.g as

$$\Pi = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix}. \quad (40)$$

6.4 Linearized model of the constrained pendulum chain

For some applications a nonlinear model may not be suitable. In such cases it is common practice to linearize the model equations around a steady state, cf. [10]. Such a steady state of the equations of the stiff constrained pendulum chain (38) can be computed by solving the system

$$
\begin{aligned}
0 &= K\bar{\Phi} + f(\bar{\Phi}) - G^T(\bar{\Phi})\bar{\lambda}, \\
0 &= g(\bar{\Phi})
\end{aligned}
$$

for $\begin{bmatrix} \bar{\Phi} \\ \bar{\lambda} \end{bmatrix}$. By construction $\dot{\bar{\Phi}} = 0$. We then set $\dot{\Phi} = \dot{\bar{\Phi}} + \delta\dot{\Phi},\ \Phi = \bar{\Phi} + \delta\Phi,\ \lambda = \bar{\lambda} + \delta\lambda$ in (38) and after dropping higher order terms in a Taylor expansion, we obtain the linearized system

$$
\begin{aligned}
\frac{d}{dt}\delta\Phi &= \delta\dot{\Phi}, && \text{(41a)} \\
\frac{d}{dt}\delta\dot{\Phi} &= -M(\bar{\Phi})^{-1}(d\delta\dot{\Phi} + K\delta\Phi + f_\Phi(\bar{\Phi})\delta\Phi - G^T(\bar{\Phi})\delta\lambda), && \text{(41b)} \\
0 &= G(\bar{\Phi})\delta\Phi. && \text{(41c)}
\end{aligned}
$$

According to [10], this system is of differentiation index 3 and we have to add the hidden constraints

$$
\begin{aligned}
0 &= G(\bar{\Phi})\delta\dot{\Phi}, && \text{(41d)} \\
0 &= G(\bar{\Phi})\frac{d}{dt}\delta\dot{\Phi} = -G(\bar{\Phi})M(\bar{\Phi})^{-1}(d\delta\dot{\Phi} + K\delta\Phi + f_\Phi(\bar{\Phi})\delta\Phi - G^T(\bar{\Phi})\delta\lambda). && \text{(41e)}
\end{aligned}
$$

For this system we apply the selection procedure as in (39) with selector Π as in (40) and obtain a system of the form $E\dot{x} = Ax,\ x(0) = x_0$ with

$$
E = \begin{bmatrix} \Pi & & \\ & \Pi & \\ & & 0 \\ & & 0 \\ & & 0 \end{bmatrix},\ A = \begin{bmatrix} 0 & \Pi & 0 \\ -\Pi M^{-1}(\bar{\Phi})(K + f_\Phi(\bar{\Phi})) & -\Pi M^{-1}(\bar{\Phi})d & \Pi M^{-1}(\bar{\Phi})G^T(\bar{\Phi}) \\ G(\bar{\Phi}) & 0 & 0 \\ 0 & G(\bar{\Phi}) & 0 \\ -G(\bar{\Phi})M^{-1}(\bar{\Phi})(K + f_\Phi(\bar{\Phi})) & -G(\bar{\Phi})M^{-1}(\bar{\Phi})d & G(\bar{\Phi})M^{-1}(\bar{\Phi})G^T(\bar{\Phi}) \end{bmatrix} \tag{42}
$$

$$
x^T = [\delta\Phi^T,\ \delta\dot{\Phi}^T,\ \delta\lambda] \quad \text{and} \quad x_0^T = [\Phi^T(0) - \bar{\Phi}^T,\ \dot{\Phi}^T(0) - \dot{\bar{\Phi}}^T,\ \lambda(0) - \bar{\lambda}]. \tag{43}
$$

7 Simulation results

We have investigated the model described by equation (39) with Π as in (40), with stiffness parameter $\kappa_0 = 2\cdot 10^{-2}$ and damping $d = 1$. With this set of parameters the system behaves highly nonlinear and the solution gets close to a singular state, i.e. a state where the motion of the system becomes undetermined, e.g. and inverted pendulum. The dimension $n = 51$ has been chosen such that the number of differential equations in (38) is 100. The DAE has been integrated using the implicit Euler scheme with constant stepsize 10^{-2} on the time interval [0, 10], i.e. in 1000 steps. The nonlinear systems arising in each time step were solved with reasonable accuracy using one Newton iteration. The Euler scheme uses the exact mass matrix but only a finite-difference approximation of the Jacobian. Unfortunately, even if the state-space of the reduced order model is much smaller than for the original model, the work for the evaluation of the right-hand side of (39) is not reduced, hence the main work load depends on the number of function evaluations. However, computing the Jacobian by finite differences requires a number of function evaluations proportional to the size of the system and becomes cheaper as the system size decreases. Gains in computation time will mostly be attributed to this fact.

For the determination of the Galerkin projectors we have tested two approaches. The first one uses the implicit Euler method, a shorter time interval and a larger stepsize to obtain a set of snapshots. These are then used to generate an approximation to the correlation matrix. For a second approach, we linearize the model around a steady state and compute the Gramian of the resulting linear constant coefficient model via an appropriate Lyapunov equation.

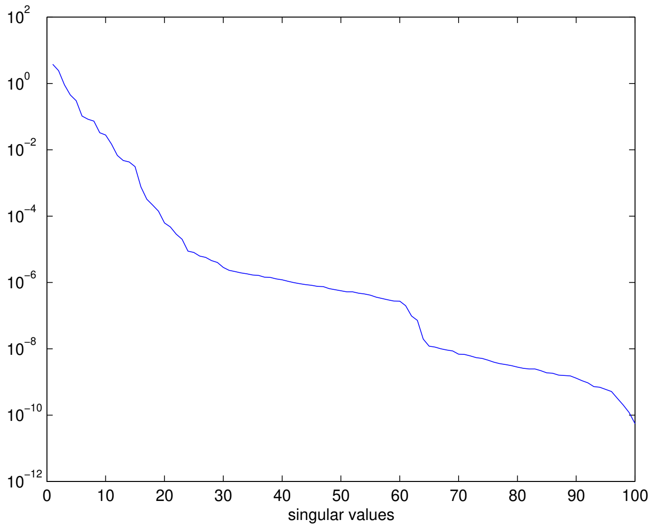

Figure 3: Decay of singular values of the snapshot matrix

7.1 Simulation with snapshots

For the generation of snapshots we have integrated (37) on the shorter time interval [0, 5] with 100 steps only. The computation of the snapshots took 800 seconds and the computation of the full model with 1000 steps took about 7,800 seconds on a Pentium 4 desktop PC with 1.4 GHz and 1 GByte RAM using MatLab7. Computation time for the Galerkin projectors took less than a second and was usually negligible compared to the computation of the snapshots.
If we set $X = [\tilde{x}(0), \ldots, \tilde{x}(5)]$, where $\tilde{x}(t)$ is the numerical approximation to $x(t)$, then $\tilde{M}_x = XX^T$ yields a good approximation to the correlation matrix M_x with respect to the solution $x(t)$ on [0, 5]. With (25), the reduced correlation matrix M_y can easily be obtained via $M_y = S_1 E M_x E^T S_1^T$ since $E = \begin{bmatrix} \Pi \\ 0 \end{bmatrix}$ is constant and $S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix}$ can be chosen as identity. For numerical reasons we did not compute the eigenvalues of the matrix $\tilde{M}_y$ but the singular value decomposition of $E_1 X$. By definition, the singular values of $E_1 X$ are the roots of the eigenvalues of $\tilde{M}_y = E_1 X X^T E_1^T$ and the computation of the singular values introduces less roundoff. In Figure 3 the decay of singular values is depicted. It is visible, that the singular values decay rapidly to a magnitude of 10^{-6} and much slower afterwards. Hence, it can be expected that a reduction of the systems dimension to 20 yields a good approximation while a higher dimension does not improve the accuracy significantly. In Table 7.1 computational test results are listed. For different tolerances *tol*, all singular values σ_i with $\sigma_i/\sigma_1 < tol$ have been dropped. The remaining number of singular values determines the order of the reduced system and the respective singular vectors are used for the construction of P_L and P_R. In the table only the order of the reduced dynamic part is listed. The actual system contains three more states due to three constraints. The listed computation times only show the time for the computation of the reduced order systems, i.e. without the time needed for the computation of the snapshots and the Galerkin projection. The last line shows the obtained relative errors with respect to the solution of the non-reduced system. For the tolerance 10^{-6} the system ran into a singular position such that the matrix $\begin{bmatrix} \Pi \\ G \end{bmatrix}$ became singular and the simulation was aborted.

tolerance	1E-2	1E-4	1E-6	1E-8	1E-10	1E-12
red. order	8	16	28	62	94	100
comp. time (s)	0.9E3	1.5E3	-	4.8E3	7.0E3	7.5E3
rel. error	1.1E-4	3.9E-6	-	1.3E-7	1.3E-7	1.3E-7

Table 1: test results using snapshots

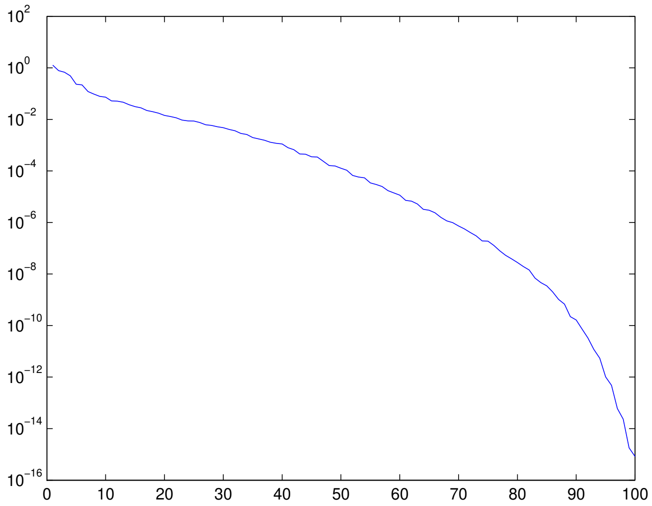

Figure 4: Decay of singular values of the square root factor

7.2 Correlation matrix via Lyapunov equation

In order to compute the correlation matrix M_y with the help of a generalized Lyapunov equation, we have lineraized the nonlinear system around its steady state and obtained a system $E\dot{x} = Ax$ with E and A as in (42). The computation of the steady state took 17 seconds with a stabilized Newton method and the solution of the generalized Lyapunov equation using Algorithm 5.1 took another 1.7s. As in the case of the computation of the correlation matrix with snapshots, in order to reduce roundoff we computed square root factors of the solution of the generalized Lyapunov equation. The decay of singular values of the square root factor of M_y is depicted in Figure 4. It can be observed that they decay much slower than for the computation with snapshots. Tests confirm that the rate of decay is related to the damping parameter d in a way that high damping leads to rapidly decaying singular values. With the Lyapunov approach, the correlation matrix is computed on an infinite time interval and slowly decaying parts of the system dynamics become more important than for the correlation matrix on a short time interval.
The test results for different drop tolerances for singular values are listed in Table 7.2. For the computation with tolerance 10^{-2} the system again ran into a singular state and the simulation failed.

tolerance	1E-2	1E-4	1E-6	1E-8	1E-10	1E-12
red. order	21	49	67	82	90	94
comp. time (s)	-	4.3E3	5.6E3	6.6E3	7.1E3	7.9E3
rel. error	-	1.2E-4	1.8E-6	3.1E-6	2.3E-7	1.3E-7

Table 2: test results using subspace computed via a Lyapunov equation

8 Summary

We have shown that the POD model order reduction method may sucessfully be applied to reduce the dimension of moderately large scale DAEs. The main idea behind the approach was to split equations and unknowns and to perform a reduction only on the dynamic part of the DAE. This already shows the limits of the approach. In order to make the method efficient, it must be reasonably easy to separate differential and algebraic parts of the DAE. Furthermore, as only the dynamic part is reduced this approach makes sense only if the dynamic part is considerably larger than the algebraic part.
A known problem for POD methods, the generation of the appropriate projection subspaces has been adressed in two ways. First, a set of snapshots has been computed by a low accuracy simulation of the original system to approximate the correlation matrix of the solution. In a different approach, a linearization of the system around its steady state has been computed. With tools from linear model order reduction, the Gramian of the linear system has been computed and used instead of the correlation matrix. With both methods it was possible to compute systems of smaller dimension that reproduced the solution of the large system with sufficient accuracy.
The construction of the projection subspaces with snapshots generated by simulation took a comparably large amount of computation time but produced good approximations. The computation of the subspaces using a generalized Lyapunov equation was comparably fast, but the decay of singular values was much slower and thus the approximation quality was less than for the snapshot case. If the reduced model is to be used repeatedly, the one-time computation of a large set of snapshots pays off as a good approximating subspace of low dimension can be expected. If the system is solved with changing parameters, e.g. in optimization problems, then the Lyapunov approach might be faster than several time-consuming snapshot generations.

References

[1] E. Anderson and et.al. *LAPACK Users' Guide*. SIAM, 3rd edition, 1999.

[2] A.C. Antoulas. *Lectures on the Approximation of Large-Scale Dynamical Systems*. SIAM Book Series: Advances in Design and Control, 2002.

[3] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, PA, 2005.

[4] V.I. Arnold. *Ordinary differential equations*. Nauka, Moscow, 1971.

[5] V.I. Arnold. *Mathematical Methods of Classical Mechanics*. Springer Verlag, 1978.

[6] S. Bächle. Index reduction for differential-algebraic equations in circuit simulation. Technical Report Matheon 141, Technical University of Berlin, 2004.

[7] S. Bächle and F. Ebert. Element-based topological index reduction for differential-algebraic equations in circuit simulation. In *SCEE 2006 proceedings*, 2006.

[8] H. Birnbaum and N. Denkmann. *Taschenbuch der Technischen Mechanik*. Verlag Harri Deutsch, Frankfurt/Main, 1997.

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

[10] S.L. Campbell. Linearization of DAEs along trajectories. *Zeitschrift für Angewandte Mathematik und Physik*, 46:70–84, 1995.

[11] S.L. Campbell and C.W. Gear. The index of general nonlinear DAEs. *Numerische Mathematik*, 72(2):173–196, 1995.

[12] S.L. Campbell and E. Griepentrog. Solvability of general differential algebraic equations. *SIAM J. Sci. Comput.*, 16:257–270, 1995.

[13] G. Dahlquist. Stability and error bounds in the numerical integration of ordinary differential equations, 1959.

[14] G.H. Golub and C.F. van Loan. *Matrix Computations*. The Johns Hopkins University Press Baltimore and London, 3 edition, 1996.

[15] E. Griepentrog and R. März. *Differential-Algebraic Equations and Their Numerical Treatment*. Teubner Verlagsgesellschaft, Leipzig, 1986.

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

[17] S.J. Hammarling. Numerical solution of the stable non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2:303–323, 1982.

[18] C. Homescu, L.R. Petzold, and R. Serban. Error estimation for reduced order models of dynamical systems. Technical report, Lawrence Livermore National Laboratory, 2003.

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

[20] P. Kunkel and V. Mehrmann. Stability properties of differential-algebaric equations and spin-stabilized discretizations. Technical Report Matheon 356, 2006.

[21] H.V. Ly and H.T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and computer modelling*, 33(1-3):223–236, 2001.

[22] T. Penzl. Numerical solution of generalized Lyapunov equations. *Adv. Comput. Math.*, 8(1-2):33–48, 1998.

[23] L. Petzold. Differential/algebraic equations are not ODEs. *SIAM Sci. Stat.*, 3:367–384, 1982.

[24] P.J. Rabier and W.C. Rheinboldt. A general existence and uniqueness theorem for implicit differential algebraic equations. *Differential and Integral Equations*, 4:563–582, 1991.

[25] M. Rathinam and L. Petzold. Dynamic iteration using reduced order models: a method for simulation of large scale modular systems. *SIAM Journal on Numerical Analysis*, 40(4):1446–1474, 2002.

[26] M. Rathinam and L. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 40(4):1893–1925, 2003.

[27] S.S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 5(24):425–448, 2000.

[28] W.C. Rheinboldt. On the computation of multi-dimensional solution manifolds of parametrized equations. *Numerische Mathematik*, 53:165–181, 1988.

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

[30] T. Stykel. Numerical solution and perturbation theory forgeneralized Lyapunov equations. *Linear Algebra Appl.*, 349:155–185, 2002.

[31] T. Stykel. Generalized Lyapunov Equations for Descriptor Systems: Stability and Inertia Theorems. Technical report, Preprint SFB393/00-38, TU-Chemnitz, October 2000.

[32] W. Walter. *Differential- und Integral-Ungleichungen*. Springer-Verlag, 1964.