ORDER REDUCTION FOR SECOND ORDER SYSTEMS

Jens Bastian and Joachim Haase, Fraunhofer IIS Dresden, Germany

Corresponding Author: Jens Bastian
Fraunhofer Institute for Integrated Circuits (IIS), Branch Lab Design Automation (EAS)
Zeunerstr. 38, 01069 Dresden, Germany
Phone: +49 351/4640-836, Fax: +49 351/4640-703
email: Jens.Bastian@eas.iis.fhg.de

Abstract. In this article a projection method for order reduction of large systems of linear equations is presented. Such systems arise for instance through the semi-discretization of partial differential equations from the electrical, thermal or mechanical domain. The proposed algorithm extends the efficient nodal order reduction method (ENOR) that was developed for second order systems. Our method ensures an exact solution for the static case and provides good approximations in the frequency and the time domain. The reduced systems can be used for behavioral models for system simulation. Examples from several domains demonstrate the power of our algorithm.

1 Introduction

During the design of systems with analog and digital, electrical and heterogenous components, behavioral modeling with languages such as VHDL-AMS is a useful tool [10], [5]. The foundations of this way are hierarchical representations of heterogeneous systems using a Kirchhoffian network approach [6], [13]. This approach is widely used in MEMS design, for instance. An important topic with this approach is the modeling of components of these systems [11]. In the top-down design step micromechanical components of such systems can be specified using predefined primitives as masses, springs, beams, gaps and so on. These elements can be easily combined with models of other electrical and nonelectrical components. Thus, the system behavior can be evaluated and the components can be specified. In many cases the real design of the nonelectrical components is carried out with the help of simulation programs that can solve systems of partial differential equations (PDE) such as the FEM tool ANSYS [2]. But these programs only help to describe components and not the whole system. Thus for bottom-up verification of the whole system models of the designed nonelectrical components are needed [9]. Order reduction methods allow to derive descriptions from linear PDEs that can be used to establish behavioral models. This paper presents an improved order reduction method and its application on behavioral modeling.

As mentioned above, components of microsystems can often be described by partial differential equations such as electrical interconnects, micromechanical sensors or heat conduction. The semi-discretization of these PDEs usually leads to large systems of ordinary differential equations (ODE) of the form

\[ M\ddot{x} + D\dot{x} + Kx = Bu \]  
\[ x_a = B_a^T x \]  

with \( x(t) \in \mathbb{R}^N, M, D, K \in \mathbb{R}^{N \times N}, B \in \mathbb{R}^{N \times m}, u(t) \in \mathbb{R}^m, B_a \in \mathbb{R}^{N \times p}. \) \( N \) can be large (\( 10^3 \ldots 10^5 \ldots \)).

The interpretation of the matrices depends on the application. For mechanical systems, \( M, D, \) and \( K \) are the mass, damping and stiffness matrices. In thermal problems, \( M = 0, D \) contains the heat capacitances and \( K \) the conductivities. \( B \) is the (generalized) incidence matrix of points with applied forces, thermal input, currents etc., \( u. \) Generalized means, that the elements of \( B \) do not have to be 1 but can be some weighting factors e.g. for spatial thermal input. \( x \) is the vector of displacements, temperatures, voltages, currents etc. \( B_a \) is the (generalized) incidence matrix of observation points \( x_a. \)

Often, especially for components in system simulation, the terminals are at the same time the observation points, \( B = B_a. \)

Due to constraints on computational time and memory when using these models of components in system simulation, the large systems (1,2) are approximated by systems of considerably lower dimension,

\[ \tilde{M}\ddot{x} + \tilde{D}\dot{x} + \tilde{K}\tilde{x} = \tilde{B}u \]  
\[ x_a = \tilde{B}_a^T \tilde{x} \]
with $\tilde{M}, \tilde{D}, \tilde{K} \in \mathbb{R}^{n \times n}$, $\tilde{B} \in \mathbb{R}^{n \times m}$, $\tilde{B}_a \in \mathbb{R}^{n \times p}$, $\hat{x} \in \mathbb{R}^n$ and $n \ll N$, usually $n = 10 \ldots 100$. This is called order reduction.

Often, $M$, $D$, and $K$ are symmetric and positive semidefinite. When reduced-order models of components are used in system simulation, stability of the whole simulation can only be guaranteed if the reduced-order models preserve the passivity of the original components.

2 Order reduction by projection methods

During the 1990s projection methods became a popular and powerful tool for order reduction of large linear systems. Here a first order system

\begin{align*}
C\dot{x} + Gx &= Bu \\
x_a &= B_a^T x
\end{align*}

(i.e. $M = 0$ in (1)) is projected into lower dimensional subspaces of $\mathbb{R}^N$,

\begin{align*}
W^T CV \dot{x} + W^T GV \hat{x} &= W^T Bu \\
x_a &= B_a^T V \hat{x}
\end{align*}

with $V, W \in \mathbb{R}^{N \times n}$.

Rational interpolation is achieved if the column spaces of $V$ and $W$ span unions of Krylov subspaces [8]. Bases of these subspaces are usually calculated by Lanczos or Arnoldi algorithms [8], [7]. An overview of reduced order modeling can be found in [3].

To use these methods for second order systems (1), they have to be transformed into a first order system as follows. The substitution

$$y = \dot{x}$$

is for a non singular $K$ equivalent to

$$K \dot{x} - Ky = 0.$$ 

Then (1) is equivalent to

\begin{equation}
\begin{bmatrix}
M & 0 \\
0 & K
\end{bmatrix}
\begin{bmatrix}
\frac{dy}{dt} \\
x
\end{bmatrix} +
\begin{bmatrix}
D & K \\
-K & 0
\end{bmatrix}
\begin{bmatrix}
y \\
x
\end{bmatrix} =
\begin{bmatrix}
B \\
0
\end{bmatrix} u.
\end{equation}

The formulation (7) is selected so that (8) corresponds to (5) with $C \geq 0$ and $G + G^T \geq 0$. Thus (8) fulfills the necessary condition for passivity of (5), cf. [7].

Transforming (1) to (8) has some disadvantages:

- The dimension of the system matrices doubles.
- Symmetry and definiteness properties are lost.
- Often, the elements of $M$ and $K$ differ by orders of magnitude and thus $C$ and $G$ are poorly balanced.

In 1999 the ENOR algorithm [12] was developed for RLC circuits which suits well for second order systems. Unfortunately, due to its derivation from the modified nodal analysis equations, a division by the expansion point (frequency in the Laplace domain) occurs in the algorithm. Therefore, 0 cannot be used as expansion point in the Laplace domain and the static case cannot be reproduced exactly.

3 A modified projection method for second order systems

The algorithm presented in this article is similar to ENOR but has some improvements. After Laplace transformation into frequency domain the system can be expanded about the Laplace variable $s_0 = 0$ which ensures an exact solution for the static case. Furthermore, the system can be expanded at several points in the frequency domain simultaneously, so the original system is well approximated in a wide range of the frequency domain and in the time domain.
3.1 Derivation of the method

After transforming (1) into Laplace domain we obtain

\[(s^2 M + sD + K)X(s) = BU(s)\].

The projection matrix will be determined from the pulse response \(U(s) = I_{m \times m}\) which leads to

\[(s^2 M + sD + K)X(s) = B.\]

The general case can be found by superposition. Now \(X\) is expanded about the point (angular frequency) \(s_0\), i.e. in powers of \(z = s - s_0\),

\[\begin{align*}
(s^2 M + sD + K) & (X_0 + (s - s_0)X_1 + (s - s_0)^2X_2 + \cdots) = B
\end{align*}\]

which leads to

\[\begin{align*}
((s_0 + z)^2 M + (s_0 + z)D + K) & (X_0 + zX_1 + z^2X_2 + \cdots) = B.
\end{align*}\]

After equating powers of \(z^k\) we obtain the block moments \(X_k\) by iteratively solving the resulting systems of linear equations,

\[\begin{align*}
(s_0^2 M + s_0D + K)X_0 &= B \\
(s_0^2 M + s_0D + K)X_1 + (2s_0D + D)X_0 &= 0 \\
(s_0^2 M + s_0D + K)X_2 + (2s_0D + D)X_1 + M X_0 &= 0 \\
&\vdots
\end{align*}\]

The projection matrix \(V\) is constructed as an orthogonal basis of the \(X_k\),

\[V = \text{orth} \{X_0, X_1, \ldots\}.\]

Orthogonalization is performed “on the fly” during the algorithm. Otherwise, the \(X_k\) would contain only information about the eigenspace of the dominant eigenvalue after few iterations. Orthogonal projection with the obtained \(V\) finally leads to the reduced system (3.4) with

\[x = V \hat{x}, \quad \hat{x} = V^T x,\]

and \(\hat{M} = V^T MV, \hat{D} = V^T DV, \hat{K} = V^T KV, \hat{B} = V^T B, \hat{B}_d = V^T B_d.\)

The reduced system is passive if \(M, D,\) and \(K\) are positive semidefinite.

3.2 Algorithm

Comparing the coefficients of \(z^k\) leads to the subsequent algorithm to reduce (1). The \(l\) expansion points are summarized in \(s_0\). The input parameter \(n\) is an upper bound for the dimension of the reduced system. The tolerance \(\epsilon_u\) is used for deflation. Details follow below the algorithm.

1: Input: \(M, D, K, B, s_0, n, \epsilon_u\)
2: Output: \(V\)
3: Set \(V = []\) and \(k = 0.\)
4: for \(j = 1, \ldots, l\) do
5: Calculate \(\hat{K}^{(j)} = \left(s_0^{(j)}\right)^2 M + s_0^{(j)}D + K\) and \(\hat{D}^{(j)} = 2s_0^{(j)} M + D.\)
6: Factorize \(\hat{K}^{(j)}\)
7: Solve \(\hat{K}^{(j)} X_0^{(j)} = B.\)
8: Set \(V = \begin{bmatrix} V \ X_0^{(j)} \end{bmatrix}.\)
9: Set \(X_0^{(j)} = 0.\)
10: end for
11: Calculate \(V = \text{orth} V.\)
12: for \(j = 1, \ldots, l\) do
13: Adjust columns of \(X_0^{(j)}\) to \(V.\)
end for
loop
for j = 1, ..., l do
Solve $K^{(j)} X^{(j)} = -\dot{D}^{(j)} X^{(j)}_{k-1} - MX^{(j)}_{k-2}$.
Set $V = \begin{bmatrix} V & X^{(j)}_k \end{bmatrix}$.
end for
Calculate $V = \text{orth} V$.
for j = 1, ..., l do
Adjust columns of $X^{(j)}_k$, $X^{(j)}_{k-1}$, and $X^{(j)}_{k-2}$ to $V$.
end for
if coldim $V \geq n$ or $\forall j : \text{coldim} X^{(j)}_k = 0$ then
Terminate.
end if
end loop

The expansion is simultaneously performed about $l$ points $s_0 = [s^{(1)}_0 \ldots s^{(l)}_0]$. If there is a $j$ with $s^{(j)}_0 = 0$, then the reduced system will be statically exact. Larger $s_0$ provide adaptation of moments for higher frequencies which results in a better approximation in the time domain.

During the orthogonalization in steps 11 and 21 of the algorithm, columns which are linearly dependent on earlier columns are deleted. This is called exact deflation. In finite-precision arithmetic, vectors that are in some sense “almost” linearly dependent on earlier vectors are also deflated. This is called inexact deflation determined by $\epsilon_u$, a parameter specified by the user. For instance, when using QR decomposition, all columns $q^k$ with $\|q^k\| < \epsilon_u$ are deflated.

In step 6, symmetric matrices are factorized by Cholesky decomposition, others by LU factorization. Indeed, if too much fill-in is generated, iterative methods such as CG for symmetric matrices and GMRES for others must be used. To obtain fast convergence, or convergence at all, incomplete factorizations are performed as a good preconditioner.

The orthogonalization in steps 11 and 21 can be performed by the modified Gram-Schmidt algorithm, by QR factorization (Householder), or by calculating an SVD.

Additionally, the projection matrix $V$ generated by our algorithm can be combined with projection matrices calculated with other methods such as modal reduction. This means, an orthonormal basis of the columns of the considered matrices is constructed and used as the new projection matrix.

4 Examples

4.1 Micromechanical sensor

![Basic structure of an acceleration sensor](image)
In figure 1 the basic structure of a micromechanical acceleration sensor is depicted. The depicted element is the basic component of a surface-micromachined microresonator. It consists of the shuttle mass and the beam springs that are fixed at the anchor. The springs are compliant in the \( y \) direction, but stiff in the \( x \) direction. The structure is similar to that used in familiar acceleration sensors [1], [4].

The FEM program ANSYS was used for the design of the component. The model exported from ANSYS consists of 3019 nodes with each 6 degrees of freedom (3 translational and 3 rotational). The sensor is clamped in the fixation points. Considering these Dirichlet boundary conditions we have to reduce a system of the dimension \( N = 18102 \). The behavioral model shall describe the movement of the mass, if it is stimulated in \( y \) direction.

![Figure 2: Frequency spectrum of displacement (in \( \mu \text{m} \)) in \( y \) direction](image)

The original system was reduced to a system of dimension \( n = 6 \). Results of the small signal simulation are shown in figure 2. It represents the deflection of the seismic mass with respect to the frequency of the excitation force in the center of the mass. The expansion was performed about \( s_0 = \begin{bmatrix} 0 & 10^5 & 10^6 \end{bmatrix} \). The dashed line is the frequency response of the original and the dotted line the response of the reduced system. The good accuracy of the reduced-order model can also be seen in figure 3 where the relative error is depicted.

![Figure 3: Relative error of the frequency spectrum of the reduced-order model](image)
4.2 Thermal example

The second example is a simple model of heat conduction in a chip. The temperature is constant 273.15 K at one end. At the other end a heat flow jump from 0 to 1 W is spatially inducted. The system is modeled with ANSYS and has the dimension $N = 26320$ after eliminating the Dirichlet boundary conditions.

![Temperature of an observation point](image)

The original system was reduced to $n = 4$. The expansion was performed about $s_0 = 0$. In figure 4 the temperature history in one of the observation points is depicted. The solid line corresponds to the original system. The dashed line corresponds to the reduced system with expansion points $s_0 = [0 \ 10^3]$ and the dotted line to the reduced system with expansion point $s_0 = 10^3$. Both reduced systems have the dimension $n = 8$. As it can be seen, the static case is not reflected by the second system. It is advantageously to select $s_0 = 0$ as an expansion point.

5 Conclusions

In this article a projection method for order reduction of large systems of linear ordinary differential equations was presented. Like ENOR it is based on moment matching techniques. By expanding about different frequency points simultaneously, the original systems are well approximated in time and frequency domain and exact for the static case. Some typical examples demonstrate the capability of our algorithm.

References


