Model Order and Terminal Reduction Approaches via Matrix Decomposition and Low Rank Approximation

Peter Benner and André Schneider

Abstract We discuss methods for model order reduction (MOR) of linear systems with many input and output variables, arising in the modeling of linear (sub) circuits with a huge number of nodes and a large number of terminals, like power grids. Our work is based on the approaches SVDMOR and ESVDMOR proposed in recent publications [1–5]. In particular, we discuss efficient numerical algorithms for their implementation. Only by using efficient tools from numerical linear algebra, these methods become applicable for truly large-scale problems.

1 Introduction

Nowadays, MOR is an important and conventional step in the preprocessing of circuit simulation. The original model resulting from methods like modified nodal analysis has to be simplified due to its complexity. One issue of this simplification for VLSI design is the MOR of parasitic linear interconnect circuits. These circuits form substructures in the design of ICs and contain linear elements with comparatively little or no influence on the result of the simulation.

In some applications, the structure of these parasitic linear subcircuits has recently changed in the following sense. So far, the number of elements in these interconnect circuits was significantly larger than the number of connections to the whole circuit, the so-called pins or terminals. This assumption is no longer valid in all cases. Circuits with a lot of elements need extra power supply networks, so-called power grids [6,7]. In clock distribution networks, the clock signal is distributed from a common point to all the elements that need it for synchronization [8]. For simulating these circuits new methods are needed. Often, a lot of their terminals behave similar so that it is possible to compress the input-/output matrices in such a way that

Peter Benner, André Schneider
Fakultät für Mathematik, Technische Universität Chemnitz, 09107 Chemnitz, Germany, e-mail: benner@mathematik.tu-chemnitz.de, andre.schneider@mathematik.tu-chemnitz.de

J. Roos and Luis R.J. Costa (eds.), *Scientific Computing in Electrical Engineering SCEE 2008*, Mathematics in Industry 14, DOI 10.1007/978-3-642-12294-1_64,

the I/O behavior can be realized through a few so-called virtual inputs/outputs [1–5]. As a consequence we deal with these virtual terminals, the number of which is much less than the original number of terminals. This allows the use of well known MOR methods like balanced truncation or Krylov subspace methods to reduce the number of inner nodes.

The intention of this paper is to explain the existing (E)SVDMOR approaches [1,4] and show improvements within the implementation in particular for large scale systems. In the following section, we review the fundamentals of the underlying approaches. We introduce the moments of a transfer function of the circuit describing system and show how to use the information in these moments in order to reduce the number of terminals. Later, we point out the weak point of the algorithm for really large scale systems and present a solution for this problem. After the introduction of this efficient algorithm to achieve a very compact model we show and discuss first numerical results in Section 3.

2 SVDMOR and ESVDMOR

Recent studies [1–5] have shown that we can make use of correlations between the plurality of input and output terminals. We use the singular value decomposition (SVD) based method SVDMOR [1,5] as well as the extended version of SVDMOR, the so-called ESVDMOR [2–5], which is the foundation for our work and will be explained in the following.

2.1 Extended-SVDMOR

We assume that the linear system to be reduced has the following transfer function in frequency domain:

$$H(s) = L(sC+G)^{-1}B, \tag{1}$$

with $C, G \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m_{in}}$, and $L \in \mathbb{R}^{m_{out}\times n}$.

The number of inputs m_{in} is not necessarily equal to the number of outputs, here m_{out}. Consider the i-th block moment of (1) defined as

$$\mathbf{m_i} = L(-G^{-1}C)^i G^{-1}B, \tag{2}$$

in terms of $\mathbf{m_i}$ as an $m_{out} \times m_{in}$ matrix

$$\mathbf{m_i} = \begin{bmatrix} m^i_{1,1} & m^i_{1,2} & \dots & m^i_{1,m_{in}} \\ m^i_{2,1} & m^i_{2,2} & \dots & m^i_{2,m_{in}} \\ \vdots & \vdots & \ddots & \vdots \\ m^i_{m_{out},1} & m^i_{m_{out},2} & \dots & m^i_{m_{out},m_{in}} \end{bmatrix}. \tag{3}$$

Note that the moments in (2) are equal to the coefficients of the Taylor series expansion of (1) in $s = 0$. The expansion in $s = s_0$ leads to frequency shifted moments defined as

$$\mathbf{m_i}(s_0) = L(-(s_0C+G)^{-1}C)^i(s_0C+G)^{-1}B. \tag{4}$$

The ESVDMOR approach uses the information of a combination of these moments to create a decomposition of (1) in the following way. To allow terminal reduction for inputs and outputs separately, r different block moments forming two moment matrices are used: the input response matrix M_I and the output response matrix M_O defined as

$$M_I = \begin{bmatrix} \mathbf{m_0} \\ \mathbf{m_1} \\ \vdots \\ \mathbf{m_{r-1}} \end{bmatrix}, \qquad M_O = \begin{bmatrix} \mathbf{m_0}^T \\ \mathbf{m_1}^T \\ \vdots \\ \mathbf{m_{r-1}}^T \end{bmatrix}, \tag{5}$$

where column k of M_I represents the coefficients (moments) of the series expansion of (1) at all outputs due to input k. Similarly, each column k of M_O represents the coefficients of output k due to all inputs. Note, that we expect the number of rows in each matrix to be larger than the number of columns so that the rank is determined by the column vectors. If not, r has to be increased.

Applying the SVD to these matrices, we can obtain a low rank approximation

$$M_I = U_I\Sigma_I V_I^T \approx U_{I_{r_i}}\Sigma_{I_{r_i}}V_{I_{r_i}}^T, \qquad M_O = U_O\Sigma_O V_O^T \approx U_{O_{r_o}}\Sigma_{O_{r_o}}V_{O_{r_o}}^T, \tag{6}$$

where

- $\Sigma_{I_{r_i}}$ is an $r_i \times r_i$ diagonal matrix,
- $\Sigma_{O_{r_o}}$ is an $r_o \times r_o$ diagonal matrix,
- $V_{I_{r_i}}^T$ and $V_{O_{r_o}}^T$ are orthogonal $r_i \times m_{in}$ and $r_o \times m_{out}$ matrices that contain the dominant column subspaces of M_I and M_O
- $U_{I_{r_i}}$ and $U_{O_{r_o}}$ are $rm_{out} \times r_i$ and $rm_{in} \times r_o$ matrices that are not used any further,
- r_i and r_o are the numbers of significant singular values as well as the numbers of the reduced virtual input and output terminals.

Equations (6) are the crucial points for our improvements described in Section 2.2.

Due to the fact that the important information about the dependencies of the I/O-ports is hidden in the matrices $V_{I_{r_i}}^T$ and $V_{O_{r_o}}^T$, approximations of B and L using the results of (6) lead to

$$B \approx B_r V_{I_{r_i}}^T \text{ and } L \approx V_{O_{r_o}} L_r, \tag{7}$$

where $B_r \in \mathbb{R}^{n\times r_i}$ and $L_r \in \mathbb{R}^{r_o \times n}$ are consequences of applying the Moore-Penrose pseudoinverse (denoted by $(\cdot)^+$) of $V_{I_{r_i}}^T$ and $V_{O_{r_o}}$ (which are isometric) to B and L, respectively. In detail, we have

$$B_r = BV_{I_{r_i}}(V_{I_{r_i}}^T V_{I_{r_i}})^{-1} = BV_{I_{r_i}}^{T+} = BV_{I_{r_i}} \tag{8}$$

and

$$L_r = (V_{O_{ro}}^T V_{O_{ro}})^{-1} V_{O_{ro}}^T L = V_{O_{ro}}^+ L = V_{O_{ro}}^T L, \tag{9}$$

where $B_r \in \mathbb{R}^{n \times r_i}$ and $L_r \in \mathbb{R}^{r_o \times n}$. Consequently, we get a new internal transfer function $H_r(s)$,

$$H(s) \approx \widehat{H}(s) = V_{O_{ro}} \underbrace{L_r (G + sC)^{-1} B_r}_{:=H_r(s)} V_{I_{ri}}^T. \tag{10}$$

This terminal reduced transfer function is now reduced to

$$\tilde{H}_r(s) = \tilde{L}_r (\tilde{G} + s\tilde{C})^{-1} \tilde{B}_r \approx H_r(s) = L_r (G + sC)^{-1} B_r \tag{11}$$

by some well known established MOR method, e. g., balanced truncation or a Krylov subspace method. At the end we get a very compact terminal and reduced-order model

$$H(s) \approx V_{O_{ro}} \tilde{H}_r(s) V_{I_{ri}}^T. \tag{12}$$

Note that SVDMOR can be considered as a special case of ESVDMOR, using only one moment and one SVD, e. g. $r = 1$, and using $\mathbf{m_0}$ as moment.

2.2 Drawbacks and Solutions

For very large subcircuits the (E)SVDMOR methods are not suitable due to the use of the SVD. Suppose we have a matrix with dimension $n = 10^6$ and a modern CPU with 3 GHz. The computation of an SVD needs about $22n^3$ flops. This would mean $22 \cdot 10^{18}$ flops and therefore a total CPU time of approximately 230 years. Obviously, this is computationally too expensive. Hence, we combine the (E)SVDMOR approaches with cheaper matrix decomposition methods, like the truncated SVD (TSVD), which computes the needed singular values and the corresponding singular vectors only, see (6). Also other ideas to compute a truncated SVD-like decomposition cheaply can be used [9–11].

Furthermore, an explicit computation of the moments in (2) would be numerically unstable and too expensive. Without loss of generality we explain the decomposition of M_I, so that

$$M_I \approx U_{I_{ri}} \Sigma_{I_{ri}} V_{I_{ri}}^T = \sum_{j=1}^{r_i} \sigma_j u_j v_j^T. \tag{13}$$

Recall that $r_i \ll m_{in}$ denotes the number of significant singular values and vectors. We do not know that number so we specify it depending on the error tolerance of the approximation. Unfortunately, there is no global error bound for the whole reduction yet (this is the topic of current research). We therefore simply use $\sigma_{r+1} < tol\,\sigma_1$ for a user-defined tolerance. Naturally, it is helpful to have a rapid decrease of the singular

values σ_j, that means a lot of dependencies within the ports and enables a gainful terminal reduction, see the examples in Section 3.

The TSVD can be computed in several ways [9, 10, 12]. Consider the augmented matrix $A \in \mathbb{R}^{r\cdot m_{out}+m_{in} \times r\cdot m_{out}+m_{in}}$ of the form

$$A = \begin{pmatrix} 0 & M_I \\ M_I^T & 0 \end{pmatrix}. \tag{14}$$

One possibility is to compute the eigenvalues of matrix A by the implicitly restarted Arnoldi method [13, 14]. It can be shown that the positive eigenvalues of A are equal to the square roots of the eigenvalues of $M_I^T M_I$, and those square roots are equal to the singular values of M_I. Using an established algorithm we only need to provide a function applying the matrix A to a vector $\mathbf{x}$ to build the needed Krylov subspace in order to determine the eigenvalues. This functions input arguments are a vector $\mathbf{x} \in \mathbb{R}^{r\cdot m_{out}+m_{in}}$ and a scalar r, which is equal to the number of used moments r, see (5). Output argument is a vector $\mathbf{y} \in \mathbb{R}^{r\cdot m_{out}+m_{in}}$,

$$A\mathbf{x} =: \mathbf{y} = ((y^1)^T, (y^2)^T, \ldots, (y^{r+1})^T)^T, \tag{15}$$

where for $i = 1, \ldots, r$

$$y^i = \begin{pmatrix} y_{(i-1)\cdot m_{out}+1} \\ \vdots \\ y_{i\cdot m_{out}} \end{pmatrix} \quad \text{and} \quad y^{r+1} = \begin{pmatrix} y_{r\cdot m_{out}+1} \\ \vdots \\ y_{r\cdot m_{out}+m_{in}} \end{pmatrix}. \tag{16}$$

Please note that we use the analog notation for the components x^j, $j = 1, \ldots, r+1$ of vector $\mathbf{x}$. If we insert (14) and (5) into (15) we get

$$\mathbf{y} = \begin{pmatrix} 0 & \begin{bmatrix} \mathbf{m_0} \\ \mathbf{m_1} \\ \vdots \\ \mathbf{m_{r-1}} \end{bmatrix} \\ \begin{bmatrix} \mathbf{m_0}^T & \mathbf{m_1}^T & \cdots & \mathbf{m_{r-1}^T} \end{bmatrix} & 0 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \\ \vdots \\ x^{r+1} \end{pmatrix}. \tag{17}$$

After a simple step of matrix multiplication we get the components y^i for $i = 1, \ldots, r$ and y^{r+1} of vector $\mathbf{y}$ as

$$y^i = \mathbf{m_{i-1}} x^{r+1} \qquad \text{and} \qquad y^{r+1} = \mathbf{m_0}^T x^1 + \cdots + \mathbf{m_{r-1}}^T x^r. \tag{18}$$

To compute these components efficiently we replace the block moments by their factors. In fact, we compute the $r+1$ parts of $\mathbf{y}$ by repeatedly applying the same factors to parts of $\mathbf{x}$, depending on whether it is a part of (18a) or (18b). We want to emphasize that we use the same factors each time. According to (2) the computation for (18a) follows Algorithm 6. The computation of (18b) is more involved, but follows the same recursive principle laid out in Algorithm 6. The computation of the decomposition of M_O works analogously. These methods become numerically

Algorithm 6 Computation of the components y^i

$a = Bx^{r+1}$
$a = G^{-1}a$
for $i = 1$ to r **do**
  $y^i = La$
  $a = Ca$
  $a = -G^{-1}a$
end for

unstable for large r but in practice r often is small. For linear circuits with the same number of inputs and outputs, mostly one moment of the transfer function in (5), i. e., $r = 1$ so that we use the SVDMOR approach, is sufficient. Summarizing, this is a quite easy way which allows us to apply the SVD to large scale systems in a truncated way.

3 Numerical Results and Conclusions

The decay of the singular values of the moment used for computing the SVD is essential for (E)SVDMOR, so we firstly concentrate on this issue. Figure 1 shows the decrease of the 500 largest singular values of a circuit provided by the NEC Laboratories Europe, IT Research Division, NEC Europe Ltd. in St. Augustin, Germany. The circuit is called *circuit3* and consist of 3916 nodes, 1905 of them are terminals. We choose about 130 singular values to be significant based on the tolerance $\sigma_{r+1} < 10^{-2}\sigma_1$. That means, after the reduction we have 130 virtual input and output pins instead of 1905 terminals originally. Figure 2 shows the range of the 30 largest singular values of another circuit. It was provided by the Qimonda AG,

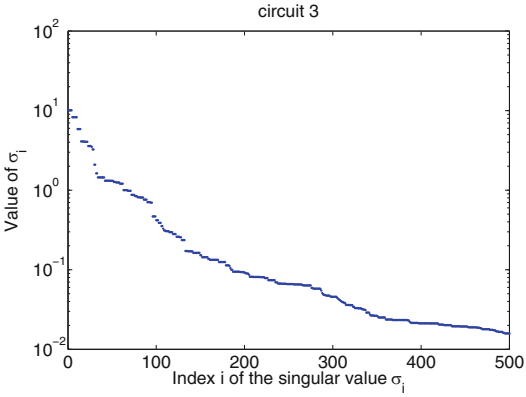

Fig. 1: Range of the largest 500 singular values of $\mathbf{m_0}$ of circuit *circuit3*

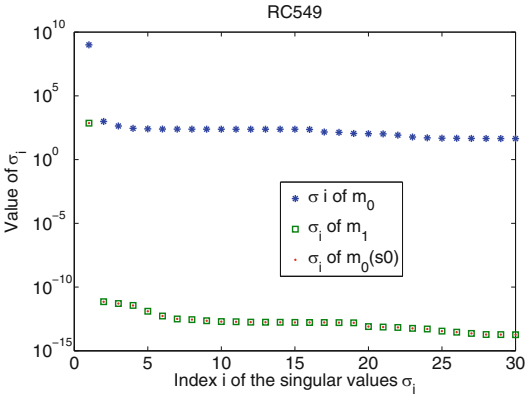

Fig. 2: Range of the largest 30 singular values of $\mathbf{m_0}$, $\mathbf{m_1}$ and $\mathbf{m_0}(s_0 = 10^8)$ of circuit *RC549*

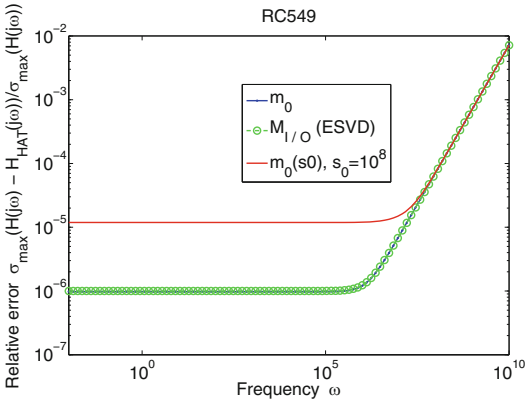

Fig. 3: Relative error ε_{rel} of *RC549* by using SVDMOR with $\mathbf{m_0}$ and $\mathbf{m_0}(s_0 = 10^8)$ and ESVDMOR with $M_{I/O}$ consisting the information of the first 3 moments $\mathbf{m_0}$, $\mathbf{m_1}$, and $\mathbf{m_2}$

Munich, Germany. It is a test circuit called *RC549* and consists of 141 nodes and therefrom 70 terminals. Figure 2 points out clearly one significant singular value. Consequently, we reduce the system to one virtual terminal. The relative approximation error for circuit *RC549* is shown in Figure 3. We can observe that the error is sufficiently small up to the Gigahertz range which is enough for the application behind this problem (subcircuit of a memory chip).

Finalizing we would like to draw a few conclusions. If the pencil $sC + G$ of (1) is stable and a stability preserving MOR methods is used in (11), then the whole MOR algorithm described is stability preserving. Also, for typical classes of RLC circuits, the procedure is passivity preserving if the inner MOR method in (11) is. Due to space limitation, we will elaborate on this aspect elsewhere. In the future we want to present a global error bound as well as other approaches to perform the decomposition in (6) and (13) efficiently.

Acknowledgements The work reported in this paper was supported by the German Federal Ministry of Education and Research (BMBF), grant no. 03BEPAE1. We also want to thank our industrial project partners Qimonda AG and NEC Laboratories Europe, IT Research Division, NEC Europe Ltd. Responsibility for the contents of this publication rests with the authors.

References

1. Feldmann, P., Liu, F.: Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals. In: ICCAD '04: Proceedings of the 2004 IEEE/ACM Intl. Conf. Computer-aided design, pp. 88–92. IEEE Computer Society, Washington, DC, USA (2004)
2. Liu, P., Tan, S.X.D., Li, H., Qi, Z., Kong, J., McGaughy, B., He, L.: An efficient method for terminal reduction of interconnect circuits considering delay variations. In: ICCAD '05: Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design, pp. 821–826. IEEE Computer Society, Washington, DC, USA (2005)
3. Liu, P., Tan, S.X.D., Yan, B., Mcgaughy, B.: An extended SVD-based terminal and model order reduction algorithm. In: Proceedings of the 2006 IEEE International, Behavioral Modeling and Simulation Workshop, pp. 44–49 (2006)
4. Liu, P., Tan, S.X.D., Yan, B., McGaughy, B.: An efficient terminal and model order reduction algorithm. Integr. VLSI J. **41**(2), 210–218 (2008)
5. Tan, S.X.D., He, L.: Advanced Model Order Reduction Techniques in VLSI Design. Cambridge University Press, New York, NY, USA (2007)
6. Tan, S.X.D., Shi, C.J.R., Lungeanu, D., Lee, J.C., Yuan, L.P.: Reliability-Constrained Area Optimization of VLSI Power/Ground Networks Via Sequence of Linear Programmings. In: in Proceedings of the ACM/IEEE Design Automation Conference, pp. 78–83 (1999)
7. Singh, J., Sapatnekar, S.: Congestion-aware topology optimization of structured power/ground networks. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on **24**(5), 683–695 (2005)
8. Lin, Z., Carpenter, A., Ciftcioglu, B., Garg, A., Huang, M., Hui, W.: Injection-locked clocking: A low-power clock distribution scheme for high-performance microprocessors. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **16**(9), 1251–1256 (2008)
9. Stoll, M.: A Krylov-Schur approach to the truncated SVD. Preprint, available at http://www.comlab.ox.ac.uk/files/721/NA-08-03.pdf (2008)
10. Hochstenbach, M.E.: A Jacobi–Davidson type SVD method. SIAM J. Sci. Comput. **23**(2), 606–628 (2001)
11. Berry, M.W., Pulatova, S.A., Stewart, G.W.: Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices. ACM Trans. Math. Softw. **31**(2), 252–269 (2005)
12. Chan, T.F., Hansen, P.C.: Computing truncated singular value decomposition least squares solutions by rank revealing QR-factorizations. SIAM J. Sci. Stat. Comput. **11**(3), 519–530 (1990)
13. Sorensen, D.C.: Implicit application of polynomial filters in a k-step Arnoldi method. SIAM J. Matrix Anal. Appl. **13**(1), 357–385 (1992)
14. Lehoucq, R., Sorensen, D., Yang, C.: ARPACK user's guide. Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods. Software - Environments - Tools, 6. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. 142 p. (1998)