On the Effectiveness of Reducing Large Linear Networks with Many Ports

João M. S. Silva
INESC ID / IST
Tech. Univ. Lisbon, Portugal
E-mail: jmss@algos.inesc-id.pt

L. Miguel Silveira
Cadence Labs / INESC ID / IST
Tech. Univ. Lisbon, Portugal
E-mail: lms@inesc-id.pt

***Abstract*— Reduced order modeling is a well-known methodology for linear system modeling. In the past decade it has risen to prominence in the VLSI electronic design area as the de facto standard set of techniques for interconnect and package modeling. With shrinking technologies and faster operating frequencies, such previously ignored structures can have a first order influence in the behavior of many electronic systems. Reduced order modeling techniques can provide accurate, robust, accuracy-controlled models of linear networks. Unfortunately, most of these techniques have difficulty reducing networks with a large number of ports, such as power grids, substrate models and coupled data buses. In this paper we provide a characterization of this problem and discuss the complexity of several previously proposed techniques for handling this problem. We show that for most of these techniques there is little hope to expect that considerable reduction can be achieved. We also show that a simple, perhaps not obvious, approach can theoretically provide better reduction than most of the other techniques.**

I. Introduction

In the process of designing and verifying state-of-the-art electronic circuits, one is often required to work with linear network models which represent structures, such as interconnect, that have a first-order impact on system behavior. Model order reduction (MOR) has become the established methodology for this task, as it enables efficient modeling and analysis of such large networks. Numerous algorithms have been proposed for this task, covering multiple requirements and ensuring appropriate properties of the underlying models. These algorithms take as input a linear network model, and produce as output a smaller model that is suitable for simulation in conjunction with nonlinear circuit elements. The effectiveness of the model reduction algorithm is sometimes measured in terms of the state-space reduction. If a large linear system can be reduced to a system with much fewer states, then the reduction is deemed successful as analysis of the smaller model is expected to be performed at lower computational cost. This reasoning is however not entirely correct. Sometimes a larger model can be analyzed at lower cost, if it exhibits some special structure that one can take advantage of. A more appropriate measure is to judge the effectiveness of the reduction process on the basis of the decrease in the reduced circuit simulation time, compared to simulation with the full model, assuming an acceptable error is incurred in the modeling process. MOR algorithms rely on the fact that on a variety of contexts, only an accurate approximation to the input-output behavior of a dynamic linear system is necessary. It is quite typical for MOR techniques to be able to reduce large interconnect networks with just a few ports, to models with very few states, and still produce very accurate approximations of frequency- and time-domain behavior. Of course, it is reasonable to expect that when the number of ports increases, the number of states to be retained must also increase. Ultimately, however, if the number of retained states keeps increasing, this appears to leave little room for compression as the size of the matrix transfer function that characterizes all port interactions also increases and may approach the complexity of working with the original network equations. In this paper we will verify this relation in a precise manner and discuss its implications. We will discuss the reasons behind this loss of efficiency and relate it to the mechanics of certain algorithms as we believe that knowledge of the specific scenarios where each method may produce better results is an important asset when determining how to perform the reduction. We will work with power grids as a proxy for the type of networks that we are concerned with. Power grids are highly regular, but also very large networks that constitute a real problem of practical importance but also provide for a good characterization for the type of structured linear networks often encountered.

Recently, the efficient reduction of systems with a large number of ports has been addressed and several methods have been proposed [1]–[3]. In this paper we show that for most of these techniques there is little hope to expect that considerable reduction can be achieved. We also show that a simple, perhaps not obvious approach, can theoretically provide better reduction than most other techniques. We start in Section II to summarize the main features of the standard model-order reduction methods that are now in widespread use in many applications in several fields, including electronic design automation. In Section III we discuss the newly proposed methods for handling massively-coupled linear dynamic systems and, in Section IV, we characterize and compare them in order to determine their effectiveness in terms of the reduction of power grid networks. The measuring stick used for this comparison is not the cost of model generation but instead the subsequent cost of working with the model in a simulation environment. In Section V we show results from applying the various methods in a variety of settings to the power grid problem. Finally, conclusions are drawn in Section VI.

II. Background

Modeling a power grid as an RC network and using the nodal analysis formulation leads to

$$\begin{aligned} C\dot{v} + Gv &= Mu \\ y &= N^T v \end{aligned} \tag{1}$$

where $C, G \in \mathbb{R}^{n\times n}$ are the capacitance and conductance matrices, respectively, $M \in \mathbb{R}^{n\times p}$ is a matrix that relates the inputs $u \in \mathbb{R}^p$ to the states $v \in \mathbb{R}^n$ that describe the node voltages, $N \in \mathbb{R}^{n\times q}$ its counterpart with respect to the outputs $y \in \mathbb{R}^q$, n is the number of states, p the number of inputs and q the number of outputs. The $p\times q$ matrix transfer function of the network is then given by $H(s) = N^T(G + sC)^{-1}M$. Typically, matrices C and G are sparse but also very large. For a typical power grid, the number of nodes will be in

the order of millions. Solving Eqn. (1) directly or using it inside a circuit simulator is therefore too expensive.

The goal of model-order reduction is, generically, to determine a reduced model,

$$H_k(s) = \hat{N}^T(\hat{G} + s\hat{C})^{-1}\hat{M} \tag{2}$$

that closely matches the input-output behavior of the original model, and where the state description is given by $z = V^T v \in \mathbb{R}^k$. However, even if $k \ll n$, the reduced-order model may fail to provide relevant compression. This may happen because, for large networks, the matrices C and G are sparse, having a number of non-zeros entries of order $\mathcal{O}(n)$. If the number of non-zero entries in the reduced-order model increases with the number of ports, the benefits of reduction may vanish with increasingly large p and q.

A. Projection-based framework

Projection-based Krylov subspace algorithms, such as PRIMA [4], provide a general-purpose, rigorous framework for deriving interconnect modeling algorithms and have been shown to produce excellent compression in many scenarios involving on- and off-chip interconnect and packaging structures. In its simplest form, they can be used to compute individual approximations to each of the $p \times q$ matrix transfer function entries. However, more commonly, they are used to generate a single approximation to the full system transfer function. The PRIMA algorithm [4], for instance, reduces a state-space model in the form of (1) by use of a projection matrix V, through the operations $\hat{G} = V^T G V$, $\hat{M} = V^T M$, $\hat{C} = V^T C V$ and $\hat{N} = V^T N$ to obtain a reduced model in the form of (2). In the standard approach, the projection matrix V is chosen as an orthogonal basis of a block Krylov subspace, $\mathcal{K}_m(A, p) = span\{p, Ap, \ldots, A^{m-1}p\}$, a typical choice being $A = G^{-1}C$ and $p = G^{-1}M$. The construction of the projection matrix V is done iteratively by blocks, with each block being generated through a back-orthogonalizing procedure. When the projection matrix is constructed in this way, the moments of the reduced model can be shown to match the moments of the original model to some order. There are two difficulties associated with applying these algorithms for reduction. The first is that the model size is proportional to the number of matched moments multiplied by the number of ports. Furthermore, the reduced system matrices will be dense. Therefore such methods are almost impractical for networks with large numbers of ports as occur in substrate and package modeling, as well as power grids.

B. Truncated Balanced Realization

An alternative class of reduction algorithms are based on Truncated Balanced Realization (TBR). The TBR algorithm first computes the observability and controllability Gramians, X and Y, by solving the Lyapunov equations, and then reduces the model by projection onto the space associated with the dominant eigenvalues of the product XY [5]. Model size selection and error control in TBR is based on the eigenvalues of XY, also known as the the Hankel singular values. In the proper case, there is an *a-posteriori* theoretical bound on the frequency-domain error for the TBR model. The existence of such an error bound is an important advantage of the TBR-like class of algorithms as there is no counterpart in the projection-based class of algorithms. Theoretically, the model selection criteria, and therefore the size of the generated model, can be done independently of the number of inputs. However, there is an indirect dependence in most problems and in particular for networks such as power grids, that exhibit a large number of inputs, useful reductions are not achievable (see [6] for a simple example that shows that for constant accuracy the order needed grows with the number of inputs). Furthermore, the solution of the Lyapunov equations required to obtain X and Y is computationally intensive for large systems and as such the technique is only of theoretical interest in this context. A variety of approximate methods have been proposed that attempt to circumvent this problem [6].

III. Massively-Coupled Problems

In the previous section we briefly summarized the main techniques for model order reduction of linear interconnect networks. As discussed, the standard projection-based techniques, present a problem when dealing with networks with a large number of ports. In this section, we briefly review some techniques aimed at solving some of the issues related to reduction of such systems.

A. SVDMOR and RecMOR

The SVDMOR [1] algorithm was developed to address the reduction of systems with a large number of ports, like power grids. While the size of a reduced model produced via PRIMA is directly proportional to the number of ports in the circuit, SVDMOR theoretically overcomes this problem using singular value decomposition analysis in order to truncate the system to any desired order. The main idea behind SVDMOR is to assume that there is a large degree of correlation between the various inputs and outputs. SVDMOR further assumes that such input-output correlation can be captured quite easily from observation of some system property, involving matrices M and N. The method can, for instance, use an input-output correlation matrix, like the one given by the DC moment matrix $S_{DC} = N^T G^{-1} M$, for instance. If we let B be the appropriate correlation matrix, and if the basic correlation hypothesis holds true, then B can be approximated by a low-rank matrix. This low rank property can be revealed by computing the SVD of B, $B = U\Sigma W^T$ where U and W are orthogonal matrices and Σ is the diagonal matrix containing the ordered singular values. Assuming correlation, there will be only a small number, $r \ll m = p + q$, of dominant singular values. Therefore $B \approx U_r \Sigma_r V_r^T$, where truncation is performed leaving the r most significant singular values. The method then approximates $M \approx b_m W_r^T = M W_r (W_r^T W_r)^{-1} W_r^T$ and $N \approx b_n U_r^T = N U_r (U_r^T U_r)^{-1} U_r^T$ where b_m and b_n are obtained using the Moore-Penrose pseudo-inverse, resulting in $H(s) \approx U_r b_n^T (G + sC)^{-1} b_m W_r^T$. Standard MOR methods, like PRIMA, can now be applied to $H_r(s)$, resulting in the final reduced model $H_r(s) = U_r \tilde{H}_r(s) W_r^T$.

SVDMOR can achieve considerable reduction when the matrix transfer function is numerically low rank. Unfortunately this is not often the case when the transfer function of a large network with many ports is considered. A more realistic situation is to note that the matrix transfer function can indeed contain many low-rank sub-blocks. If such blocks are then identified, they can be reduced using SVDMOR. Therefore in [2], the authors introduce RecMOR, an algorithm for recursively detecting and sparsifying sub-blocks of the transfer function. The idea is quite simple. Assuming an appropriate partitioning of the network ports can be obtained, the matrix transfer function can likewise be partitioned into sub-blocks. To simplify the description assume that $M = N$ and that M is partitioned as $M = [M_1, M2]$. Then the matrix transfer function can be written as

$$H(s) = \begin{bmatrix} M_1^T (G + sC)^{-1} M_1 & M_1^T (G + sC)^{-1} M_2 \\ M_2^T (G + sC)^{-1} M_1 & M_2^T (G + sC)^{-1} M_2 \end{bmatrix} \tag{3}$$

At this point one could perform model order reduction technique separately on the four components of the transfer function. to obtain

a reduced system

$$H_k(s) = \begin{bmatrix} \hat{M}_1^T(\hat{G}_{11}+s\hat{C}_{11})^{-1}\hat{M}_1 & \hat{M}_1^T(\hat{G}_{12}+s\hat{C}_{12})^{-1}\hat{M}_2 \\ \hat{M}_2^T(\hat{G}_{21}+s\hat{C}_{21})^{-1}\hat{M}_1 & \hat{M}_2^T(\hat{G}_{22}+s\hat{C}_{22})^{-1}\hat{M}_2 \end{bmatrix} \quad (4)$$

Since the reductions are all done separately, one can construct a reduced state-space model for each of the components and evaluation of the full model can be performed by parallel evaluation of the component models. Furthermore, if any of the sub-matrices is low-rank, then it can be represented by a smaller model. However, if it is not low-rank, then one can recursively apply the same technique in order to split it into sub-blocks, some of which are likely to be low-rank. Obviously, the final model will consist of a large set of separate state-space representations for each of the individual sub-blocks, but hopefully enough reduction is done on them that the overall model will be less costly to manipulate. Exactly how much can be gained will be discussed later.

B. Domain Decomposition MOR

Recently, a domain-decomposition technique [7] was proposed for analysis of power grids. Based on that, we experimented a different reduction algorithm. Consider the power grid equations written as in (1). Now consider applying a domain-decomposition type algorithm to the power grid. To simplify the notation assume that the power grid is divided into two domains and that the matrix is reordered such that nodes in each domain or partition, are clustered together (obviously the number of partitions can be increased). We then denote as v_k, where k is the partition identifier, the nodes that are internal to each partition, and v_F, the nodes that are in the boundary of all partitions. Then, the same equations can be written as:

$$\begin{bmatrix} C_1 & & \\ & C_2 & \\ & & C_F \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{v}_F \end{bmatrix} + \begin{bmatrix} G_1 & & E_1 \\ & G_2 & E_2 \\ F_1 & F_2 & G_F \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_F \end{bmatrix} = \begin{bmatrix} B_1 & & \\ & B_2 & \\ & & B_F \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_F \end{bmatrix} \quad (5)$$

Consider now partition k and let us rewrite the equations for that partition.

$$C_k\dot{v}_k + G_k v_k = B_k u_k - E_k v_F = [B_k, -E_k] \begin{bmatrix} u_k \\ v_F \end{bmatrix} = \tilde{B}_k \tilde{u}_k \quad (6)$$

Nodes v_k are all internal states to partition k. Eqn. (6) tells us that if we consider that partition in isolation, the possible ways to excite those nodes would come from setting sources at all ports inside that partition, plus at any node in the boundary (leaving these nodes out leads to accuracy loss, even at DC). Consider now applying some model order reduction technique separately to each of the k partitions. The resulting system can be written as:

$$\begin{bmatrix} \hat{C}_1 & & \\ & \hat{C}_2 & \\ & & C_F \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{v}_F \end{bmatrix} + \begin{bmatrix} \hat{G}_1 & & \hat{E}_1 \\ & \hat{G}_2 & \hat{E}_2 \\ \hat{F}_1 & \hat{F}_2 & G_F \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ v_F \end{bmatrix} = \begin{bmatrix} \hat{B}_1 & & \\ & \hat{B}_2 & \\ & & B_F \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_F \end{bmatrix} \quad (7)$$

where generically $\hat{C}_k = V_k^T C V_k, \hat{G}_k = V_k^T G V_k, \hat{E}_k = V_k^T E, \hat{F}_k = F V_k$. The interesting feature of this domain decomposition MOR technique is that inherent parallelism is obtained due to the partitioning. A similar advantage was noted above in discussing the RecMOR algorithm. Furthermore, there is inherent sparsity in the structure of (7) which makes it quite attractive. On the other hand, the reduced matrices are dense and the number of interface nodes is incompressible and could be quite large. Exactly how much reduction could be achieved will be discussed later.

IV. Effectiveness of Reduction

In the previous sections we reviewed some of the techniques for handling the reduction of networks with a large number of ports. The actual reduction obtained with each technique is in general system-dependent. Nevertheless, under some general assumptions they can be characterized and compared.

Let us assume that we want to compute a reduced model of a regular power grid of size n with p inputs. Further assume that p is large, say $p = \mathcal{O}(\sqrt{n})$, which is in practice a conservative assumption. Let us start by characterizing the original full model and use it as a yardstick for comparison. The matrix for this system has $\mathcal{O}(n)$ nodes since it is basically a 2D (or 3D) discretization pencil. It is known that in this case the cost of factoring this matrix will be around $\mathcal{O}(n^{3/2})$ as fill-ins will generate a small block of size $n^{1/2}$ at the bottom, which takes cubic effort to factor.

Consider now, as an initial alternative, that we decide to use PRIMA to generate a reduced order model. If we decide to match q_P moments, we perform q_P block orthogonalizations. The size of the reduced model is then $q_P p = q_P \times n^{1/2}$. This model is however full, which means that the number of nonzeros elements to work with is around $q_P^2 n$. Therefore, the cost of factoring this matrix is around $q_P^3 n^{3/2}$. As expected, this immediately raises some concerns over using PRIMA at all. For this many inputs, even a moderate size model (i.e. small q_P) will lead to a model with more nonzeros and higher flop count than the original full system.

Consider now using a technique such as RecMOR and suppose that clusters of inputs can be found for which reduction is performed separately. Assume there are d such clusters, thus leading to d^2 blocks in the system transfer matrix (actually, given the symmetry one really needs to compute and store $d(d-1)/2$ sub-blocks, but that is a minor issue). For each of these sub-blocks we can compute a reduced model. Let us assume that for the diagonal blocks (which display the effects of approximately p/d ports) we generate a reduced model doing q_R PRIMA iterations. Then we have, just for the diagonal blocks d models of size $q_R p/d = q_R n^{1/2}/d$, each of which is full. Therefore, the number of nonzeros of the total of such models is around $d(q_R p/d)^2 = q_R^2 n/d$. Of course RecMOR advocates doing the reduction using SVDMOR to take advantage of correlations and produce sparsified models. For the method to provide real sparsification, however, not only must q_R be small, but dependence on n has to be dropped fairly quickly as there are indeed $\mathcal{O}(d^2)$ functions to approximate. In practice, it is not clear under what conditions this will happen. Still, if enough correlation can be procured, perhaps such dependence can be broken and the final model be independent of p (i.e. n).

Suppose now that the domain-decomposition type algorithm is pursued and that again d partitions are considered. Each of the d blocks is reduced separately, perhaps using q_D PRIMA-type iterations. The ports to be considered for this partition are the ones that fall within that partition plus the interface nodes. Unfortunately the number of interface nodes is $\mathcal{O}(\sqrt{n})$ which is the grid size in each dimension. This implies that each of the individual models will be represented by a dense matrix of size $q_D\sqrt{n}$ and therefore the cost of working with this block alone is already $q_D^3 n^{3/2}$, already exceeding the cost of working with the full system. Furthermore, the sub-matrices related

to the interface nodes also fill up and therefore the method presents no advantage whatsoever.

As a final note consider an alternative technique going back to the roots of MOR techniques. Consider, as in RecMOR, that we try to approximate each entry of the transfer function individually (in essence this is equivalent to the recursive RecMOR taken to the extreme). Each individual transfer function entry relates the input current at some port to the voltage at some other port. Therefore, computing an approximate for each entry requires using only the input vector corresponding to that input port. Actually it is known that projections using that single vector can be used to generate models at all outputs simultaneously. What this implies is that a full column of the matrix transfer function can be generated from approximations using a single input vector. So, for each of the p ports consider computing an approximation to all the outputs relative to that particular input. This can be achieved by doing q_S iteration of a PRIMA-type algorithm, but now with a single input vector (corresponding to the column of M for that input port). This leads to a model with q_S states represented by a matrix that is full, i.e., with q_S^2 nonzeros. Such a matrix will require q_S^3 effort to factor and work with. Of course there are p such models, corresponding to the p inputs (thus requiring the generation of p projection spaces), so the full model of the power grid will be represented by p independent systems with a total of $pq_S^2 = q_S^2 \times n^{1/2}$. Evaluation of the whole model may therefore require an effort on the order of $pq_S^3 = q_S^3 \times n^{1/2}$. Interestingly enough, this is the lowest complexity of all models considered. Furthermore, it is the only one that even theoretically has a good shot at improving over using the full model directly. Of course, since now we want the approximation to be good not only with respect to nearby outputs but also for far-away outputs, it is possible that $q_S >> q_P$. In practice this is observed, and in fact it is observed that for similar accuracy models, $q_S > q_R > q_D > q_P$, but the differences between the required orders seems to increase very slowly with n. In the results section we refer to this technique as STFMOR (single transfer function MOR).

The conclusion of this analysis are therefore not entirely satisfactory. It seems that for most methods proposed until now, there does indeed seem to be little hope of improving over using the original model. The main problem is that none of the proposed method seems able to break the connection between model order and number of inputs. In fact, recent results seem to indicate that such a connection is indeed hard to avoid [3].

V. Results

In the following, we show the results of applying PRIMA, RecMOR and STFMOR on two types of grid models. The first one, Grid A, is a 32×32 grid with 32 voltage ports on the left side and 32 current ports on the right side. The second one, Grid B, is a 32×32 grid with 9 flip-chip C4 bumps and around 10% of the number of nodes randomly distributed current sources, simulating transistors.

A. Grid A Results

This grid has 1024 nodes, so the full-model has 4992 non-zeros (2D stamp sparsity). In order to achieve a reduced model with as much as 10^{-3} maximum absolute error, we adjusted the order of the models obtained by each method. The results are presented in Table I. We can see that in order to stay below the maximum admissible error, we had to set the order q of the models to high values which yielded models with an increasing number of non-zeros. For the RecMOR algorithm we used a number of 8 blocks.

TABLE I

Results from reducing a 32×32 grid with 32 voltage ports on the left side and 32 current ports on the right side.

	q	nnz	max abs err
Full model	-	4992	-
PRIMA	4	65536	2.322e-03
RecMOR	8	73728	5.463e-03
STFMOR	12	9216	7.687e-03

TABLE II

Results from reducing a 32×32 grid with 9 voltage sources and 107 randomly distributed ports.

	q	nnz	max abs err
Full model	-	4992	-
PRIMA	3	103041	7.508e-04
RecMOR	8	329280	7.902e-03
STFMOR	12	15408	5.275e-03

B. Grid B Results

On this grid setup, we had 107 current sources. The reduction results are presented in Table II. We can observe that similarly to the previous case no relevant sparsity can be obtained. Note that for this case $p \approx 10\%$ of n, which is nevertheless lower than $\sqrt{n}$. Still, only STFMOR manages to come closer to the density of the original system.

VI. Conclusions and Acknowledgements

In this paper we provide a characterization of the problem of model order reduction of networks with a very large number of ports. We review the basic methods that have been presented to cope with this problem and show that for most of these techniques there is little hope to expect that considerable reduction can be achieved. We also show that a simple, perhaps not obvious approach, can theoretically provide better reduction than most of the other techniques. Still, the end result seems to be that a method that can provide a representation of such a network that is somehow independent of the number of ports is yet to be found.

This work was partly supported by the Portuguese Foundation for Science and Technology through the grant SFRH/BD/10586/2002 and the FP6/IST/027378 Chameleon-RF project.

References

[1] P. Feldmann, "Model order reduction techniques for linear systems with large number of terminals," in *DATE'2004 - Design, Automation and Test in Europe, Exhibition and Conference*, vol. 2, Paris, France, February 2004, pp. 944–947.

[2] P. Feldmann and F. Liu, "Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals," in *International Conference on Computer Aided-Design*, San Jose, California, U.S.A., November 2004.

[3] L. M. Silveira and J. Phillips, "Exploiting input information in a model reduction algorithm for massively coupled parasitic networks," in 41^{st} *ACM/IEEE Design Automation Conference*, San Diego, CA, USA, June 2004, pp. 385–388.

[4] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer-Aided Design*, vol. 17, no. 8, pp. 645–654, August 1998.

[5] B. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction," *IEEE Transactions on Automatic Control*, vol. AC-26, no. 1, pp. 17–32, February 1981.

[6] J. R. Phillips and L. M. Silveira, "Poor Man's TBR: A simple model reduction scheme," *IEEE Trans. Computer-Aided Design*, vol. 24, no. 1, pp. 43–55, Jan. 2005.

[7] Q. Zhou, K. Sun, K. Mohanram, and D. C. Sorensen, "Large power grid analysis using domain decomposition," in *Proceedings of the Design, Automation and Test in Europe conference (DATE)*, Munich, Germany, March 2006.