On the Compressibility of Power Grid Models

João M. S. Silva
Technical University of Lisbon
Instituto Superior Técnico / INESC ID
R. Alves Redol, 9, Sala 133
1000-029 Lisboa, Portugal
jmss@algos.inesc-id.pt

L. Miguel Silveira
Technical University of Lisbon
Instituto Superior Técnico / INESC ID
Cadence Laboratories
R. Alves Redol, 9
1000-029 Lisboa, Portugal
lms@inesc-id.pt

Abstract

The simulation of power distribution networks is a difficult task owing to the large number of elements and ports in such circuits. In this work, we elaborate on the compressibility of power grid models. For this purpose, two main options are available, namely sparse or hierarchical model representations of such systems and equivalent reduced order models. A proxy for comparison is the number of nonzero entries in system representation. The problem with model order reduction methods is the large number of ports of these networks, since the number of nonzero entries of the reduced model is, in general, proportional to the square of the number of ports. In this paper, we propose for the first time the utilization of a specific hierarchical model representation, in which a Cholesky decomposition of the system matrix can be efficiently computed and later used in the simulation phase. Results show that for higher problem sizes the hierarchical representation is more compact than the sparse representation, while the reduced order models are of no use.

1 Introduction

The simulation of power distribution networks is a difficult task owing to the large number of elements and ports of such circuits. Power grid models are generally represented in sparse matrices, where the number of nonzero entries is $\mathcal{O}(n)$, being n the number of nodes in the circuit. In a realistic power grid with several layers of metal, the number of nodes can ascend to several millions, as the power distribution network must cover the whole area of the circuit. As for the ports, these are either C4 bumps (if, for instance, a flip-chip technology is used) and connections to transistor drains (VDD network) and sources (GND network). In this way, the number of ports can also reach the order of millions. The computational cost associated with the time simulation of power grids is proportional to the number of nonzero entries in the models, so this number is usually used as a proxy for measuring the efficiency of the models.

A common way to deal with the dimension of the problem is to reduce the large models resulting from the discretization of the three-dimensional structures that represent the physical description of these networks [8, 3, 4, 13, 10]. Unfortunately, these reduced models consist in dense matrices whose size is lower-bounded by the number of ports of the circuit. Moreover, the number of entries in such models is, in general, proportional to the square of the number of ports, thus compromising the usability of the reduced order models.

Alternatively, we can pursue a hierarchical matrix representation [2] which takes advantage of the sparsity structure of the original matrices. With this representation, we can efficiently compute a Cholesky decomposition of the system matrix which can be directly used in the simulation phase. The increase in matrix density that results from the factorization can be managed, according to the desired accuracy, to obtain a reduced storage data structure. It is demonstrated that this kind of representation is of almost linear complexity. Results show that for higher problem sizes, which reflect the dimension of real power grid models, the hierarchical representation of the Cholesky factors is more compact than using a sparse matrix representation or reduced order models.

The remaining of this paper is organized in the following manner: in Section 2 we present the methods used to deal with the compression of power grid models. Then, in Section 3, we present the comparison between compression of the system resulting from the formulation of the problem with model order reduction methods and the proposed hierarchical representation. In Section 4 some conclusions are drawn.

IEEE Computer Society Annual Symposium on VLSI(ISVLSI'07)
0-7695-2896-1/07 $20.00 © 2007 IEEE

2 Methods

In this section we briefly overview some of the most relevant model order reduction (MOR) methods as well as the technique for hierarchical matrix representation.

We will use the following power grid model formulation:

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

where $C, G \in \mathbb{R}^{n\times n}$ and $M, N \in \mathbb{R}^{n\times p}$, being n the number of nodes and p the number of ports. For the sake of argument, we assume the number of ports in a power grid to be $\mathcal{O}(n^{1/2})$.

We will be interested in a time simulation of the model, so Backward Euler could be used, for instance. Applying Backward Euler, we can write Equation (1) in the time domain like:

$$\underbrace{\left(\frac{C}{h} + G\right)}_{A} v(t) = Bu(t) - \frac{C}{h}v(t-h) \tag{2}$$

where h is the time step and A is a constant matrix.

2.1 Model Order Reduction Methods

2.1.1 PRIMA

Projection-based reduction methods such as PRIMA [9] have been shown to produce excellent compression in many scenarios involving on- and off-chip interconnect and packaging structures.

PRIMA reduces a state-space model in the form of (1) by use of a projection matrix V through the following operations:

$$\begin{array}{rcl} \tilde{C} & = & V^T C V \\ \tilde{G} & = & V^T G V \\ \tilde{M} & = & V^T M \\ \tilde{N} & = & V^T N \end{array} \tag{3}$$

to obtain a reduced model in the form of:

$$\begin{array}{rcl} \tilde{C}\dot{z} + \tilde{G}z & = & \tilde{M}u \\ y & = & \tilde{N}^T z \end{array} \tag{4}$$

where $z = V^T v$. The projection matrix V is chosen as an orthogonal basis of a block Krylov subspace:

$$\mathcal{K}_q(A, p) = span\{p, Ap, \ldots, A^{q-1}p\} \tag{5}$$

When the projection matrix is obtained in this way, the moments of the reduced model match the moments of the original model at least to order q. If, for accuracy reasons, we need to match q moments, the size of the reduced model is then $q \times p = q \times n^{1/2}$. This model is however full, which means that the number of nonzero elements to work with is around $q^2 n$. Therefore, the cost of factoring this matrix is around $q^3 n^{3/2}$. As expected, this immediately raises some concerns over using PRIMA at all. For this many ports, $\mathcal{O}(n^{1/2})$, even a small size model will lead to a model with more nonzeros than using the original model.

2.1.2 SVDMOR

The SVDMOR [3] 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 (SVD) analysis in order to truncate the system to a smaller order.

The main idea behind SVDMOR is to assume that there is a large degree of correlation between the circuit ports. SVDMOR further assumes that such a correlation can be captured quite easily from observation of some system property, involving matrices M and N. This correlation matrix can be, for instance, the DC moment matrix $N^T G^{-1} M$, which contains only DC information, or more complicated response correlations such as k-order moment matrices

$$N^T (G^{-1}C)^k G^{-1} M$$

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:

$$B = U\Sigma W^T \tag{6}$$

where U, W are orthogonal matrices and Σ is the diagonal matrix containing the singular values ordered by magnitude. Assuming correlation, there will be only a small number, $r \ll p$, of dominant singular values. Therefore,

$$B \approx U_r \Sigma_r V_r^T \tag{7}$$

where truncation is performed leaving the r most significant singular values. The method then approximates:

$$\begin{array}{c} M \approx b_m V_r^T = M V_r (V_r^T V_r)^{-1} V_r^T \\ N \approx b_n U_r^T = N U_r (U_r^T U_r)^{-1} U_r^T \end{array} \tag{8}$$

where b_m and b_n are obtained using the Moore-Penrose pseudo-inverse, resulting in:

$$H(s) \approx U_r \underbrace{b_n^T (G + sC)^{-1} b_m}_{H_r(s)} V_r^T \tag{9}$$

Standard MOR methods, like PRIMA, can now be applied to $H_r(s)$, resulting in the final reduced model:

$$H(s) \approx H_k(s) = U_r \tilde{H}_r(s) V_r^T \tag{10}$$

We observe that in most cases the assumption of highly correlated ports is not valid. In this case, $r \approx p$, which leads to the same problems faced by PRIMA.

2.1.3 RecMOR

In order to further improve the theoretical sparsifying capabilities of SVDMOR, RecMOR [4] was introduced. This algorithm can recursively sparsify 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:

$$H(s) = \begin{bmatrix} M_1^T(G+sC)^{-1}M_1 & M_1^T(G+sC)^{-1}M_2 \\ M_2^T(G+sC)^{-1}M_1 & M_2^T(G+sC)^{-1}M_2 \end{bmatrix} \tag{11}$$

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

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

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 sparsified, meaning 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.

Suppose that clusters of inputs can be found for each of which reduction is performed. Assume there are d such clusters, thus leading to d^2 blocks in the system transfer matrix. For each of these sub-blocks we can compute a reduced model. Let us assume that for the diagonal blocks we generate a reduced model doing q PRIMA iterations. Then we have, just for the diagonal blocks, d models of size $q\frac{p}{d} = q\frac{n^{1/2}}{d}$, each of which is full. Therefore, the number of nonzeros of the total of such models is around $q^2\frac{n}{d}$.

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 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 and, as we will see in Section 3, this in general will not lead to useful reductions.

2.1.4 BSMOR

BSMOR [13] is a generalization of SPRIM [5] where a 2×2 partitioning of the state matrices was proposed. BSMOR consists in partitioning the complete domain in d blocks, and applies an accordingly partitioned PRIMA projection matrix to reduce the system, preserving its block structure. If the projection matrix from PRIMA is given by:

$$V = \begin{bmatrix} V_1^{(n_1\times p)} \\ V_2^{(n_2\times p)} \\ \vdots \\ V_d^{(n_d\times p)} \end{bmatrix} \tag{13}$$

then the matrix used in BSMOR obtained in the following way:

$$\tilde{V} = \begin{bmatrix} V_1^{(n_1\times p)} & & & \\ & V_2^{(n_2\times p)} & & \\ & & \ddots & \\ & & & V_d^{(n_d\times p)} \end{bmatrix} \tag{14}$$

which is a $\mathbb{R}^{n\times dp}$ matrix. While a q order PRIMA projection matches q moments, BSMOR matches $q\times d$ moments, although the model is also d times larger. While BSMOR clearly needs a smaller order for the same accuracy as PRIMA, in terms of matrix entries it behaves much like PRIMA, as we shall see in the results section.

2.1.5 PMTBR

The PMTBR algorithm [10, 12] was motivated by a connection between frequency-domain projection methods and approximation to truncated balanced realizations (TBR [7]). The method is less expensive in terms of computation, but tends to TBR when the order of the approximation increases. The actual mechanics of the algorithm are akin to multi-point projection. In a multi-point rational approximation, the projection matrix columns are computed by sampling in several frequency points along a desired frequency interval:

$$z_i = (G + s_iC)^{-1}M \tag{15}$$

where s_i, with $i = 1, 2, \ldots, N$, are N frequency sample points. The frequency-sampled matrix thus obtained can then be used to project the original system in order to obtain a reduced model.

In the PMTBR algorithm, a similar procedure is used. The connection to TBR methods is made by noting that an approximation $\hat{X}$ to the Gramian X can be can be computed as:

$$\hat{X} = \sum_i w_i z_i z_i^H \tag{16}$$

where $s_i = j\omega_i$ and the ω_i and w_i can be interpreted as nodes and weights of a quadrature scheme applied to a frequency-domain interpretation of the Gramian matrix (see [10] for details). If we let Z be a matrix whose columns are the z_i, and W is now the diagonal matrix of the square root of the weights, Eqn. (16) can be written more compactly as:

$$\hat{X} = ZW^2Z^H \tag{17}$$

If the quadrature rule applied is accurate, $\hat{X}$ will converge to X, which implies the dominant eigenspace of $\hat{X}$ converges to the dominant eigenspace of X. If we compute the singular value decomposition of ZW,

$$ZW = V_Z S_Z U_Z \tag{18}$$

with S_Z real diagonal, V_Z and U_Z unitary matrices, it is easy to see that V_Z converges to the eigenspaces of X, and the Hankel singular values are obtained directly from the entries of S_Z. V_Z can then be used as the projection matrix in a model order reduction scheme. The method was shown to perform quite well in a wide variety of settings [11].

An interesting additional interpretation, and quite relevant for our purposes, was recently presented (ICTBR [12]). It has been shown that if further information revealing time-domain correlation between the ports is available, this variant of PMTBR can be used leading to significant efficiency improvement. The idea is akin to the basic assumptions in SVDMOR and related to exploiting correlation between the inputs. Unlike SVDMOR, however, in ICTBR it is assumed that the correlation information is not contained in the circuit directly, but rather in its inputs. In this variant of PMTBR, a correlation matrix K is formed by columns which are samples of port values along the time-steps of some interval. Those samples, should characterize as well as possible the values expected at the inputs of the system, i.e. K should be a suitably representative model of the possible inputs. An SVD is then performed over K in order to retain only the most significant components of the input correlation information:

$$K \approx U_K \Sigma_K V_K^T \tag{19}$$

With this additional correlation information, the samples relative to multi-point approximation become:

$$z_i = (G + s_i C)^{-1} M U_K \Sigma_K \tag{20}$$

Using the z_i above as columns of the Z matrix in Equation (17) leads to the input-correlation truncated balanced realization algorithm, ICTBR. See [11] for details and a more thorough description of the probabilistic interpretation of both PMTBR and ICTBR.

Notwithstanding, and as we shall see in Section 3, while for the same accuracy PMTBR yields a smaller order q than the other methods, the reductions thus obtained are still not advantageous over the original model.

2.2 Direct Matrix Representations

2.2.1 Sparse Representation

Sparse representations are generally used in iterative methods, but they can also be used in a Cholesky decomposition $A = LL^T$ which requires only a backward substitution for solving each time-step. The sparse model representation complexity is $\mathcal{O}(n)$ and the Cholesky factor L takes $\mathcal{O}(n^{3/2})$ space.

2.2.2 Hierarchical Representation

Hierarchical matrix representation [2] was used in [6, 14] and as an inspiration to RecMOR. In this approach, the aim is not to reduce the model size, but to obtain an efficient factorization of the system matrix which can be directly applied when solving the system. The size of the hierarchical representation of the model is the same and the number of nonzero entries increases, but this increase is uncorrelated to the number of ports of the circuit, contrarily to what happened with MOR methods. Moreover, for the larger sized models the number of nonzero entries in the hierarchical Cholesky factors is smaller than the number of nonzero entries in a sparse factorization. The space complexity of the hierarchical representation of the Cholesky factors is $\mathcal{O}(n \times log(n))$.

In the next section, we shall see the results of applying the described methods and techniques for the compact representation of power grid models.

3 Results

We conducted experiments on a bi-dimensional RC mesh consisting on $\sqrt{n} \times \sqrt{n}$ nodes. Conductances and capacitances were randomly generated in $[0, 1)$. All capacitances are grounded. The number of ports is around $\sqrt{n}$ and they are randomly positioned (cf. Figure 1).

In Table 1 we show results from applying PRIMA, SVDMOR and PMTBR to the grid. As we can see, SVDMOR cannot take advantage of any port correlation, so it falls back to PRIMA. PRIMA needs an order of $q = 5$ for accuracy, while for PMTBR $q = 4$ is enough. The number of nonzeros in all models is much larger than the number of entries of the original sparse model.

We can try to sparsify the models taking advantage of the correlation between sub-blocks of the original matrix.

Figure 1. Port positioning in the experimental 32×32 RC mesh.

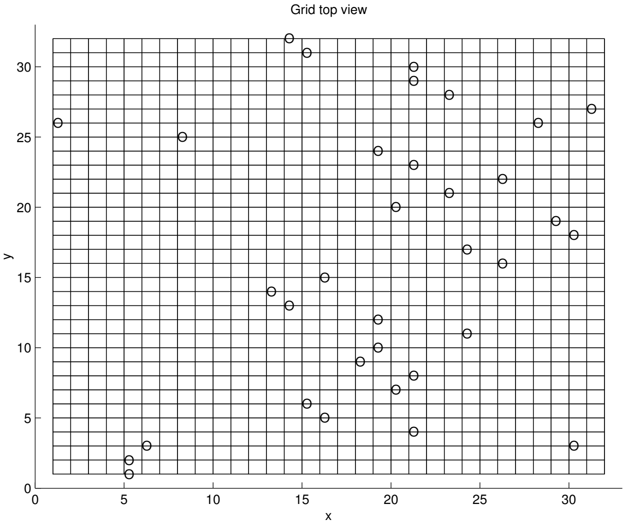

Table 1. Reduction of 32×32 RC mesh with 34 ports with PRIMA, SVDMOR and PMTBR.

	size	nnz	error
original	1024×1024	4992	
PRIMA	170×170	28900	4.576e-04
SVDMOR	170×170	28900	4.576e-04
PMTBR	136×136	18496	2.464e-05

Table 2. Reduction of 32×32 RC mesh with 34 ports with RecMOR and BSMOR (d is the number of sub-blocks used).

	d	q	size	nnz	error
original			1024×1024	4992	
RecMOR	2	8		49792	3.546e-04
RecMOR	4	8		50176	1.186e-03
BSMOR	2	5	340×340	115600	1.412e-04
BSMOR	4	4	544×544	184960	3.335e-04
BSMOR	8	4	1088×1088	406912	9.213e-04

In Table 2 we show the results of applying RecMOR and BSMOR to the grid. We observe that with more sub-blocks, BSMOR needs a smaller order to guarantee accuracy, although that does not mean the model is more compact. On the contrary, the sparsity of the models decreases as we use more sub-blocks. This means the assumption of port correlation is not valid in a densely port populated grid.

The proposed alternative is to use a hierarchical matrix representation of the model matrices [1]. In order to efficiently solve (2) we can use a Cholesky factorization. In Table 3 we show results for the sparse and hierarchical representations of matrix G as well as for the Cholesky $G = LL^T$ factorization. We see that while using a Cholesky factorization does not help in reducing the number of nonzeros in the matrices, the advantage in the simulation phase is obvious. Nevertheless, the hierarchical representation for larger sized problems, which are the ones of real interest, has fewer nonzeros entries than the sparse representation. This is due to some controlled accuracy lost. Overall, the hierarchical representation is a good alternative to MOR methods, since in most cases these are not capable of producing useful reductions, yielding models more computationally intensive than the hierarchical Cholesky factorization.

4 Conclusions

Power distribution networks are large linear circuits with many ports. Model order reduction methods have an extreme difficulty in dealing with this kind of circuits, due to the fact that the reduced models are full and its size proportional to multiples of the number of ports. Even the recent methods that claim to take advantage of a low rank representation of sub-blocks of the system matrices fail to do so unless in unrealistic cases. The proposed alternative is to use a hierarchical representation of the Cholesky factors of the system matrices. While this does not help reducing the number of entries in the model matrices, results show this hierarchical representation of the Cholesky factors is more compact than the sparse representation with obvious benefits in the simulation phase.

5 Acknowledgements

We would like to thank Steffen Börm and Lars Grasedyck for the kind support and insights on the questions related to the hierarchical matrix representation.

This work was partly supported by the Portuguese Foundation for Science and Technology under the grant SFRH/-BD/10586/2002.

References

[1] Hlib package. `http://www.hlib.org/`.

[2] S. Börm, L. Grasedyck, and W. Hackbusch. *Hierarchical matrices*. Max Planck Institute for Mathematics in the Sciences, June 2006.

Table 3. Sparse and hierarchical representations of the conductance matrix.

	size(G)	nnz(G)	nnz(L)	error
sparse	1024 × 1024	4992	31624	
hierarchical	1024 × 1024	11776	44968	1.451e-05
sparse	4096 × 4096	20224	246721	
hierarchical	4096 × 4096	48128	270664	7.406e-05
sparse	16384 × 16384	81408	1960188	
hierarchical	16384 × 16384	194560	**1500424**	4.585e-04
sparse	65536 × 65536	326656	15669551	
hierarchical	65536 × 65536	782336	**8012296**	1.311e-03

[3] 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*, volume 2, pages 944–947, Paris, France, February 2004.

[4] 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.

[5] R. W. Freund. Sprim: structure-preserving reduced-order interconnect macromodeling. In *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 80–87, Washington, DC, USA, 2004. IEEE Computer Society.

[6] S. Kapur and D. E. Long. IES^3: A fast integral equation solver for efficient 3-dimensional extraction. In *Proceedings of the Int. Conf. on Computer-Aided Design*, pages 448–455, November 1997.

[7] B. Moore. Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction. *IEEE Transactions on Automatic Control*, AC-26(1):17–32, February 1981.

[8] A. Odabasioglu, M. Celik, and L. Pileggi. PRIMA: Passive reduced-order interconnect macromodeling algorithm. In *International Conference on Computer Aided-Design*, pages 58–65, San Jose, California, November 1997.

[9] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design*, 17(8):645–654, August 1998.

[10] J. R. Phillips and L. M. Silveira. Poor man's TBR: A simple model reduction scheme. In *DATE'2004 - Design, Automation and Test in Europe, Exhibition and Conference*, Paris, France, February 2004.

[11] J. R. Phillips and L. M. Silveira. Poor Man's TBR: A simple model reduction scheme. *IEEE Trans. Computer-Aided Design*, 24(1):43–55, Jan. 2005.

[12] 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*, pages 385–388, San Diego, CA, USA, June 2004.

[13] H. Yu, L. He, and S. X.-D. Tan. BSMOR: Block structure preserving model order reduction. In *IEEE International Behavioral Modeling and Simulation Conference (BMAS)*, September 2005.

[14] Z. Zhu and J. White. Fastsies: a fast stochastic integral equation solver for modeling the rough surface effect. In *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*, pages 675–682, Washington, DC, USA, 2005. IEEE Computer Society.