Scalable Symbolic Model Order Reduction [*]

Yiyu Shi[†] Lei He[†]
[†]Electrical Engineering Dept., UCLA
[†] Los Angeles, California, 90024
[†]{yshi, lhe}@ee.ucla.edu

C.-J. Richard Shi[*]
[*]Electrical Engineering Dept., UW
[*]Seattle, WA, 98105
[*]{cjshi}@u.washington.edu

ABSTRACT

Symbolic model order reduction (SMOR) is to reduce the complexity of a model with symbolic parameters. It is an important problem in analog circuit synthesis and digital circuit modeling with process variations. However, existing symbolic model order reduction (SMOR) methods do not scale well with the number of symbols or with the model order. This paper presents a scalable SMOR algorithm, namely S^2MOR. We first separate the original multi-port multi-symbol system into a set of single-port systems by superposition theorem, and then integrate them together to form a lower-bordered block diagonal (LBBD) structured system. Each block is reduced independently, with a stochastic programming to distribute the given overall model order between blocks for best accuracy. The entire system is efficiently solved by low-rank update. Compared with existing SMOR algorithms, given the same memory space, S^2MOR improves accuracy by up to 78% at a similar reduction time. In addition, the factorization and simulation of the reduced model by S^2MOR is up to 17× faster.

1. INTRODUCTION

Symbolic circuit techniques are playing an increasingly important role with the advance of design technology, especially when we have entered the nano regime. Plenty of algorithms exist in literature discussing how to analyze and simulate those symbolic circuits [1–3]. However, all those methods are practical only if the circuit has a moderate size. Unfortunately, to guarantee reasonable model accuracy, the circuits from physical extraction usually contain millions of nodes, thus rendering those methods inefficient. Towards this end, numerous model order reduction (MOR) techniques have been successfully applied to the reduction of linear large scale circuits over the past decade ([4, 5],etc). There were also efforts to extend those methods to nonlinear circuits ([6, 7], etc) and parametrized circuits ([8, 9], etc). However, despite their wide application, unsolved problems do exist when directly extending them to symbolic circuits.

The idea of symbolic model order reduction (SMOR) was first introduced in [10], which contains three different methods: symbolic isolation, nominal projection and first order expansion. The symbol isolation method first removes all the symbols from the circuits, and the nodes to which the symbols are connected are modeled as ports. As such, the symbolic circuit becomes a symbol-free circuit with massive ports, and can be reduced by any traditional methods [4,5,11–13]. However, the size of the reduced circuit is proportional to the port number multiplying the number of moments to be matched. Accordingly, the time and space complexity for the reduced model increases cubically with the number of ports and thus the number of symbols, which renders those methods of less practical value for circuits with many symbols. The nominal projection method uses the nominal values of the symbols to compute the projection matrix and is accurate only when the symbol values slightly deviate from the nominal value. The first order expansion method uses the first order expansion of the matrix inversion and multiplication to find the projection matrix, which is first order matrix polynomial w.r.t. all the symbols. Again, no large change is allowed for the symbols in order for the method to be accurate.

In addition, all the above three symbolic MOR methods suffer from the following three problems: First, they don't scale well with the number of symbols or with the reduced order. In literature only the experimental results for circuits with less than ten symbols are reported. Second, if the original circuit has no special pattern to be explored, the reduced model is dense and therefore is still time consuming for simulation. Third, when the values of the symbols change, it is difficult to directly update the factorization of the reduced model instead of redo it from scratch, especially for the nominal projection and first order expansion methods. A more detailed analysis of those symbolic MOR methods are provided in Section 2.

The main contribution of our paper is as follows: A scalable SMOR algorithm, namely S^2MOR is presented. We first separate the original multi-port multi-symbol system into a set of single-port systems by superposition theorem, then integrate them together to form a lower-bordered block diagonal (LBBD) structure. Each block is reduced independently, with a stochastic programming to distribute the given overall model order between blocks for best accuracy. And the entire system is efficiently solved by low-rank update. Experiments using analog designs show that S^2MOR handles circuits with up to 49965 nodes and 661 symbols in 50 minutes. Compared with existing SMOR algorithms, S^2MOR improves accuracy by up to 78% at a similar reduction time and a same memory space. In addition, the factorization and simulation of the reduced model by S^2MOR is up to 17× faster.

The remaining of the paper is organized as follows: Section 2 reviews the existing approach for handling symbolic circuits and its disadvantages. Section 3 presents our LBBD transformation and system projection algorithm. We also discuss how to fully utilize the special sparsity pattern of the reduced model for the efficient simulation and update of the reduced model. Experimental results are presented in Section 4 and concluding remarks are given in Section 5.

2. PRELIMINARIES

In this section, we briefly review the three SMOR methods proposed in [10], and address their disadvantages by complexity analysis w.r.t. the reduced order q. For later comparison with our method, the complexity for the three different methods are summarized in the first three rows of Table 1.

2.1 Symbol Isolation

The main idea of symbol isolation is to remove all the symbols from

[*]This work is supported in part by NSF CAREER Award CCR-0306682, Defense Advanced Projects Agency under contract $W31P4\theta-05-O-R164$ and in part by Defense Threat Reduction Agency under contract $\#FA9451-06-C-0384$. This work was performed while Yiyu Shi was an intern at Orora Design Technologies, Inc., Redmond, WA.

the circuit, and model the nodes to which the symbols are connected as ports. Then any traditional MOR methods can be applied.

Generally speaking, a circuit with a total number of a symbols and p ports is converted to an equivalent symbol-free circuit with $p+a$ ports, assuming all the symbols are two-terminal. A general reduction method such as PRIMA [4] states that in order to match the first q moments, the reduced model will have a size of $q(p+a)$ with a cost of $O((p+a)^2q^2)$ for the reduction (the orthonormalization of $q(p+a)$ vectors plus a constant matrix factorization cost). With no special structure, the time complexity to factorize the reduced model is $O(q^3(p+a)^3)$ which increases cubically with a, and the space complexity to store the reduced model is $O(q^2(p+a)^2)$. Moreover, to update the reduced model with a new set of symbol values, the cost is $a(p+a)^2q^2$ by using the method of low rank update. Therefore, such a method is useful when the circuits contain only a few symbols.

2.2 Nominal Projection

The main idea of nominal projection method is to use the nominal values of the symbols to compute the projection matrix. The modified nodal analysis (MNA) equation symbolic circuit can be written as

$$(\mathbf{G}_0+\Delta\mathbf{G})+s(\mathbf{C}_0+\Delta\mathbf{C})=\mathbf{B}u, \tag{1}$$

where $\mathbf{G}_0$ and $\mathbf{C}_0$ are the nominal matrices, and $\Delta\mathbf{G}$, $\Delta\mathbf{C}$ contain the stamping of the symbols. Then the nominal projection method computes the projection matrix $\mathbf{V}$ such that

$$\mathbf{V}\subseteq\kappa_q(\mathbf{A}_0,\mathbf{R}_0), \tag{2}$$

where $\mathbf{A}=\mathbf{G}_0{}^{-1}\mathbf{C}_0$ and $\mathbf{R}_0+\mathbf{G}_0{}^{-1}\mathbf{B}$, and $\kappa_q(\mathbf{A}_0,\mathbf{R}_0)$ is the q^{th} order Krylov subspace spanned by $\mathbf{A}_0$ and $\mathbf{R}_0$. This method is very efficient as the projection matrix does not depend on the number of symbols. In order to *approximately* match q moments, the reduced circuit will have a size of pq with the cost of p^2q^2 for the reduction (the orthonormalization of pq vectors). Accordingly, the time complexity for analyzing the reduced model is $O(q^3p^3)$, and the space complexity for the reduced model including the coefficient matrices for all the symbols is $O(ap^2q^2)$. Moreover, to update the reduced model with a new set of symbol values costs $O(p^3q^3)$ because it requires new factorization of the reduced model.

As the values of the symbols deviate from the nominal values, the projection matrix no longer contains the Krylov subspace spanned by the circuit matrix. Accordingly, the model accuracy decreases. [10] showed that the error of the reduced model increases dramatically even if the value of the symbols deviates only a few percent from the nominal value.

2.3 First Order Approximation

Similar problem exists for the first order approximation method. The method computes the projection matrix using first order expansion, i.e., when computing the projection matrix, the following approximation is used

$$\begin{aligned}&[(\mathbf{G}_0+\Delta\mathbf{G})^{-1}(\mathbf{C}_0+\Delta\mathbf{C})]^k(\mathbf{G}_0+\Delta\mathbf{G})^{-1}\mathbf{B}\\ \approx\;&\mathbf{A}_0{}^k\mathbf{R}_0-\{\mathbf{A}_0{}^k\mathbf{G}_0{}^{-1}\Delta\mathbf{G}-\sum_{i=1}^{k-1}\mathbf{A}_0{}^i\mathbf{G}_0{}^{-1}\Delta\mathbf{G}\mathbf{A}_0{}^{k-i}\\ &+\sum_{j=0}^{k}\mathbf{A}_0{}^j\mathbf{G}_0{}^{-1}\Delta\mathbf{C}\mathbf{A}_0{}^{k-j}\}\mathbf{R}_0.\end{aligned} \tag{3}$$

After first order expansion, the Krylov subspace is directly used without orthonormalization to avoid the expensive cost of symbolic computation which may render the reduced model singular. Such a method can only work when the values of the symbols do not deviate much from the nominal values. Similar to the nominal projection method, the reduced circuit will have a size of pq if we *approximately* match q moments. Note that the reduction cost is $O(pq)$ because no orthonormalization is employed. The reduce model has the same space complexity and time complexity as the nominal projection method.

3. S²MOR ALGORITHM

3.1 Port Separation and Model Reduction

The key idea of the S^{2}MOR algorithm is to separate the original multi-port multi-symbol system into a set of single-port systems by superposition theorem, and then integrate them together to form a lower-bordered block diagonal (LBBD) structure. We further show that by using the projection matrices from each of the single-port systems, we can reduced such LBBD system with the structure preserved. The whole procedure is detailed below.

We start with the following modified nodal analysis (MNA) equation, which represents an RLC circuit with a symbols and p ports

$$(\mathbf{G}+s\mathbf{C})x+\sum_{i=1}^{a}P_is_i\circ(P_i^Tx)=\mathbf{B}u \tag{4}$$

$$y=\mathbf{L}^Tx, \tag{5}$$

where $\mathbf{G}$ and $\mathbf{C}$ ($\in R^{N\times N}$) are the inductance and capacitance matrices, $\mathbf{B}$ and $\mathbf{L}$ ($\in R^{N\times p}$) are the incidence matrices for input and output, u ($\in R^p$) is the input current vector. The vector P_i ($\in R^N$) is the incidence vector for symbol i and it takes the form[1]

$$P_i=\begin{pmatrix}0 & \ldots & 1 & \ldots & -1 & \ldots 0\end{pmatrix}^T \tag{6}$$

with 1 at the j^{th} element (positive node) at -1 at the k^{th} element (negative node). $s_i\circ$ is the operator corresponding to the $i-v$ relation of the i^{th} symbol. Typically, for an resistive symbol, $s_i\circ=1/R$ is a constant operator. For capacitive symbol, we have $s_i\circ=sC$ and for inductance, $s_i\circ=1/sL$. Accordingly, the current from symbol i can be computed as

$$w_i=s_i\circ(P_i^Tx),\quad 1\le i\le a \tag{7}$$

.

Denote B_i as the i^{th} column of B and u_i as the i^{th} element of u. Then according to superposition theorem, Eq. (4) becomes

$$(\mathbf{G}+s\mathbf{C})x=\sum_{i=1}^{p}B_iu_i-\sum_{i=1}^{a}P_iw_i, \tag{8}$$

Eq. (8) can be further divided into a set of $p+a$ equations, with each equation in the form of

$$(\mathbf{G}+s\mathbf{C})x^{(i)}=\begin{cases}B_iu_i, & 1\le i\le p\\ P_{i-p}w_{i-p}, & p+1\le i\le p+a\end{cases}, \tag{9}$$

$$x=\sum_{i=0}^{p+a}x^{(i)}, \tag{10}$$

where $x^{(i)}$ is the state variable for the i^{th} equation, which corresponds to a system with single port, and w_i is the current through symbol i.

On the other hand, from Eq. (10) and Eq. (7), we have

$$\begin{aligned}w_i &= s_i\circ(\sum_{j=1}^{p+a}P_i^Tx^{(j)})\\ &= \sum_{j=1}^{p+a}s_i\circ(P_i^Tx^{(j)}),\end{aligned} \tag{11}$$

[1] For simplicity of presentation, we assume that the symbol is a two-terminal device. However, it is understood that the algorithm is readily to be applied to multi-terminal devices as well.

Table 1: Space and time complexity comparison between four different methods w.r.t. the reduced order q.

Method	Mom Matched	Reduced Size	Space Complexity	Time Complexity		
				Reduction	Factorization	Update
Symbol Isolation	q	$(p+a)q$	$(p+a)^2q^2$	$(p+a)^2q^2$	$(p+a)^3q^3$	$a(p+a)^2q^2$
Nominal Projection	Approx. q	pq	ap^2q^2	p^2q^2	p^3q^3	p^3q^3
1^{st} Order Expansion	Approx, q	pq	ap^2q^2	$\mathbf{a}pq$	p^3q^3	p^3q^3
S^{2}MOR	q	$(p+a)q$	$(\mathbf{p}+\mathbf{a})\mathbf{q}^\mathbf{2}$	$(p+a)q^2$	$(p+a)q^3$	$\mathbf{c}onst$

where we have used the fact that $s_i \circ$ is a scalar for RLC elements. We introduce

$$z = \begin{pmatrix} x^{(1)T} & x^{(2)T} & \dots & x^{(p+a)T} \end{pmatrix}^T \tag{12}$$

and Eq. (7) becomes

$$w_i = \begin{pmatrix} s_i \circ P_i^T & s_i \circ P_i^T \dots & s_i \circ P_i^T \end{pmatrix} z, \quad 1 \le i \le a. \tag{13}$$

Therefore, Eq. (9)-Eq. (10) can be cast into a compact matrix form as

$$(\hat{\mathbf{G}} + s\hat{\mathbf{C}})z = \hat{\mathbf{B}}u, \tag{14}$$

where $\hat{\mathcal{G}}$ is a lower bordered block diagonal (LBBD) matrix and $\hat{\mathcal{C}}$ is a block diagonal matrix as shown in Eq. (15) at the top of next page. In addition, we have

$$\hat{\mathbf{B}} = \begin{pmatrix} \mathbf{B}_1 & & \\ & \ddots & \\ & & \mathbf{B}_p \\ 0 & & 0 \\ \vdots & \vdots & \vdots \end{pmatrix}. \tag{16}$$

With the augmented system, the output can be computed as

$$y = \hat{\mathbf{L}}z, \tag{17}$$

where

$$\hat{\mathbf{L}} = \begin{pmatrix} \mathbf{L} & \mathbf{L} & \dots & \mathbf{L} \end{pmatrix}^T. \tag{18}$$

One of the advantages of the augmented system is that a symbol-free projection matrix can be found, which can preserve the special sparse structure of the $\hat{G}$ and $\hat{C}$ matrices as well. This is detailed in the following theorem.

Theorem 1. *If orthonormalized matrices $\mathbf{V}_i$ satisfies*

$$\mathbf{V}_i \subseteq \begin{cases} \kappa_q\{\mathbf{G}, \mathbf{C}, B_i\} & 1 \le i \le p \\ \kappa_q\{\mathbf{G}, \mathbf{C}, P_{i-p}\} & p+1 \le i \le p+a \end{cases}, \tag{19}$$

where $\kappa_q\{\mathbf{G}, \mathbf{C}, B_i\}$ is the q^{th} order Krylov subspace spanned by G, C and B_i and $\kappa_q\{\mathbf{G}, \mathbf{C}, P_{i-p}\}$ is the q^{th} order Krylov subspace spanned by $\mathbf{G}$, $\mathbf{C}$ and P_{i-p}. Then with the projection matrix

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_1 & & & \\ & \mathbf{V}_2 & & \\ & & \ddots & \\ & & & \mathbf{V}_{p+a} \end{pmatrix}, \tag{20}$$

the first q moments of the reduced system

$$(\hat{\mathbf{G}}_r + s\hat{\mathbf{C}}_r)z_r = \hat{\mathbf{B}}_r u \tag{21}$$

$$y_r = \hat{\mathbf{L}_r} z_r \tag{22}$$

and the original system are matched, where $\hat{G}_r = \mathbf{V}^T\hat{G}\mathbf{V}$, $\hat{\mathbf{C}}_r = \mathbf{V}^T\hat{\mathbf{C}}\mathbf{V}$, $\hat{\mathbf{B}}_r = \mathbf{V}^T\mathbf{B}_r$ and $\hat{\mathbf{L}}_r = \mathbf{V}^T\hat{\mathbf{L}}$.

Due to the space limit, the proof is omitted here. Although the projection matrix in Eq. (19) can guarantee the matching of the first q moments, it does not always provide the best accuracy depending on the symbol values. Later we will show how to find an optimal projection matrix in terms of accuracy, under the given reduced size, by changing the order of the Krylov subspaces spanning each $\mathbf{V}_i$ in Eq. (20).

To match q moments, the reduced matrices $\hat{\mathbf{G}}_r$ and $\hat{\mathbf{C}}_r$ would have a dimension of $(p+a)q$, which is the same as the that from PRIMA. However, different from the dense reduced matrices from PRIMA, it is easy to see that the $\hat{\mathbf{G}}_r$ of the reduced system still keeps the LBBD structure, and the $\hat{\mathbf{C}}_r$ still keeps the block diagonal structure as shown in Eq. (15). Note that for efficiency, the matrix $\mathbf{G}_r$ should never be formed explicitly. Instead, it can be cast as a block diagonal matrix plus with a rank q updates, i.e,

$$\hat{\mathbf{G}_r} = \mathcal{D} + \mathcal{L}\mathcal{H}^T \tag{23}$$

where $\mathcal{D}$ ($\in R^{(p+a)q \times (p+a)q}$) is a block diagonal matrix

$$\mathcal{D} = \begin{pmatrix} \mathbf{G}_{r,1} & & \\ & \ddots & \\ & & \mathbf{G}_{r,p+a} \end{pmatrix}, \tag{24}$$

and $\mathcal{L}$, $\mathcal{H}$ ($\in R^{(p+a)q \times a}$) can be written as

$$\mathcal{L} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -P_{r,1}^{p+1} & 0 & \dots & 0 \\ 0 & -P_{r,2}^{p+2} & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & -P_{r,a}^{p+a} \end{pmatrix}, \tag{25}$$

$$\mathcal{H} = \begin{pmatrix} s_1 \circ P_{r,1}^{1T} & s_2 \circ P_{r,2}^{1T} & \dots & s_a \circ P_{r,a}^{1T} \\ s_1 \circ P_{r,1}^{2T} & s_2 \circ P_{r,2}^{2T} & \dots & s_a \circ P_{r,a}^{2T} \\ \vdots & \vdots & \vdots & \vdots \\ s_1 \circ P_{r,1}^{(p+a)T} & s_2 \circ P_{r,2}^{(p+a)T} & \dots & s_a \circ P_{r,a}^{(p+a)T} \end{pmatrix}, \tag{26}$$

where only $\mathcal{D}$ and $\mathcal{L}$, $\mathcal{H}$ are stored and explicitly formed. This sparsity structure facilitates the efficient simulation as well as update of the reduced model, as will be discuss shortly.

The time complexity of the proposed algorithm can be decomposed into two portions: the time to compute $p+a$ Krylov subspaces, each with order q; Note that the factorization of of $\mathbf{G}^{-1}\mathbf{C}$ can be shared between different subspaces, and we need to do $p+a$ times of vector orthonormalization, each with q vectors. Accordingly, the time complexity is $O((p+a)q^2)$.

Assuming no special structure for $\mathbf{G}$ and $\mathbf{C}$ matrices in (4), the space needed to store the reduced matrices $\hat{\mathbf{C}}_r$ is $O(pq^2)$ since it is block diagonal. To store $\hat{\mathbf{G}}_r$, we need to store $\mathcal{D}$, $\mathcal{L}$ and $\mathcal{H}$, which cost $O((p+a)q^3)$, $O(aq)$ and $O(a(p+a)q)$ respectively. Keeping the dominant terms in q, and we can see that the total space complexity is $O((p+a)q^2)$.

3.2 Simulation and Update of the Reduced Model

To simulate the reduced model, either in time domain for transient analysis or in frequency domain for AC analysis, we would face the problem of efficiently factorizing a LBBD matrix. We will concentrate on the QR factorization as it is shown to be the most stable factorization algorithm [14]. We will show that instead of the general factorization cost $O((p+a)^3q^3)$ for non-structuralized matrices with dimension $(p+a)q$, our special LBBD matrix can be factorized at a much lower cost. For

$$\hat{\mathbf{G}} = \begin{pmatrix} \mathbf{G} & & & & & \\ & \ddots & & & & \\ & & \mathbf{G} & & & \\ -P_1 s_1 \circ P_1^T & \dots & \dots & \mathbf{G} - P_1 s_1 \circ P_1^T & \dots & -P_1 s_1 \circ P_1^T \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -P_a s_a \circ P_a^T & \dots & \dots & \dots & \dots & \mathbf{G} - P_a s_a \circ P_a^T \end{pmatrix} \quad \hat{\mathbf{C}} = \begin{pmatrix} \mathbf{C} & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & \dots & \dots & \ddots & \dots & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ & \dots & \dots & \dots & \dots & \mathbf{C} \end{pmatrix} \tag{15}$$

the simplicity of presentation, we will simply discuss the factorization of $\hat{\mathbf{G}}_r + s\hat{\mathbf{C}}_r$ matrix for frequency domain analysis. Similar algorithm can be applied for the time domain analysis as the matrix structure will remain the same.

Suppose we want to solve

$$(\hat{\mathbf{G}}_r + s_0\hat{\mathbf{C}}_r)x = \hat{\mathbf{B}}_r u \tag{27}$$

for any input u and frequency s_0. To start with, from the matrix inversion lemma we know that

$$\begin{aligned} & (\hat{G}_r + s_0\hat{C}_r)^{-1} = (\mathcal{D} + s_0\hat{C}_r + \mathcal{L}\mathcal{H}^T)^{-1} \\ = \quad & (\mathcal{D} + s_0\hat{C}_r)^{-1} - (\mathcal{D} + s_0\hat{C}_r)^{-1}\mathcal{L} \\ & \left(\mathbf{I} + \mathcal{H}^T(\mathcal{D} + s_0\hat{C}_r)^{-1}\mathcal{L}\right)^{-1} \mathcal{H}^T(\mathcal{D} + s_0\hat{C}_r)^{-1} \end{aligned} \tag{28}$$

Accordingly, an efficient factorization method can be obtained by first factorizing $\mathcal{E} = \mathcal{D} + s_0\hat{C}_r$, which is block diagonal, and then factorizing $\mathcal{M} = \mathbf{I} + \mathcal{H}^T(\mathcal{D} + s_0\hat{C}_r)\mathcal{L}$ ($\mathcal{M} \in R^{a\times a}$). Then for any input u, we can first solve

$$\mathcal{E}x' = \mathbf{B}_r u, \tag{29}$$

and then solve

$$\mathcal{M}x'' = x'. \tag{30}$$

Finally, solve

$$\mathcal{E}x''' = \mathcal{L}x'' \tag{31}$$

and the solution can be obtained as

$$x = x' - x''' \tag{32}$$

The main cost for such an algorithm lies in the factorization of $\mathcal{E}$ and $\mathcal{M}$, which is $O((p+a)q^3)$ and $O(a^3)$. Keeping the dominant term in q, we get that the overall time complexity for factorization is $O((p+a)q^3)$. Moreover, each time the values of the symbols are changed, we only need to re-factorize $\mathcal{M}$, the cost of which is $O(a^3)$ and does not depend on q.

The comparison of complexity between the symbol isolation method, the nominal projection method, the first order expansion method as well as the S^{2}MOR method is presented in Table 1. The reduced circuits from the four methods all match or approximately match the first q moments of the original circuit. The best of the four method is shown in bold. From the table we can see that the S^{2}MOR method outperforms the other three methods in almost all the complexity comparisons. For the same number of moments to be matched, the S^{2}MOR algorithm has the lowest space and time complexity.

3.3 Min-max Programming based Projection Order Decision

Theorem 1 provides a method of computing the projection matrix which can match the first q moments of the original circuit with a reduced size of $d = (p+a)q$. It would also be interesting to note the following problem: If we are given the overall reduced size d, how to distribute it between each V_i in (20), such that the reduced model has the best accuracy. We will show that uniform distribution between the blocks as in Theorem 1 does not necessarily provide the best accuracy.

Specifically, the freedom of choosing the projection matrix lies in the order of the Krylov subspace for each $\mathbf{V}_i$ in (19). If we choose a Krylov subspace with order q_i, then the error introduced due to the mismatching of the $(q_i+1)^{th}$ moment can be computed from Eq. (4). If $1 \le i \le p$, then

$$\begin{aligned} \Delta_i &= |\sum_m L_m^T(\mathbf{G}^{-1}\mathbf{C})^{q_i+1}\mathbf{G}^{-1}B_i| \\ &= |\sum_m L_m^T\mathbf{A}_0{}^{q_i+1}R_i|, \end{aligned} \tag{33}$$

where $\mathbf{A}_0 = \mathbf{G}^{-1}\mathbf{C}$, $R_i = \mathbf{G}^{-1}B_i$ and L_m is the m^{th} column of $\mathbf{L}$. If $p+1 \le i \le p+a$, then

$$\begin{aligned} \Delta_i = & \left|\sum_m L_m^T((\mathbf{G} + P_{i-p}s_{i-p} \circ P_{i-p}^T)^{-1}\mathbf{C})^{q_i+1}\right. \\ & \left.(\mathbf{G} + P_{i-p}s_{i-p} \circ P_{i-p}^T)^{-1}P_{i-p}\right|. \end{aligned} \tag{34}$$

We can expand (34) to the first order by letting $\Delta\mathbf{G} = P_i s_i \circ P_i^T$ and $k = q_i + 1$ in (35), i.e.,

$$\begin{aligned} \Delta_i = & \left|\sum_m L_m^T((\mathbf{G} + P_{i-p}s_{i-p} \circ P_{i-p}^T)^{-1}\mathbf{C})^{q_i+1}(\mathbf{G}\right. \\ & \left. + P_{i-p}s_{i-p} \circ P_{i-p}^T)^{-1}P_{i-p}\right| \\ \approx & \left|\sum_m L_m^T\left(\mathbf{A}_i{}^{q_i+1}\bar{R}_{i-p} - (\mathbf{A}_i{}^{q_i+1}F(s_i\circ)\right.\right. \\ & \left.\left. - \sum_{j=1}^{q_i}\mathbf{A}_i{}^{j}F(s_i\circ)\mathbf{A}_i{}^{q_i+1-j})\bar{R}_{i-p}\right)\right|, \end{aligned} \tag{35}$$

where $F(s_i\circ) = \mathbf{G}^{-1}P_{i-p}s_{i-p} \circ P_{i-p}^T$ and $\bar{R}_{i-p} = \mathbf{G}^{-1}P_{i-p}$. Accordingly, the total error is

$$\begin{aligned} f = & \sum_{i=1}^{p}\left|\sum_m L_m^T\mathbf{A}_0{}^{q_i+1}R_i\right| + \sum_{i=p+1}^{p+a}\left|\sum_m L_m^T\left(\mathbf{A}_i{}^{q_i+1}\bar{R}_{i-p} - \right.\right. \\ & \left.\left.(\mathbf{A}_i{}^{q_i+1}F(s_i\circ) - \sum_{j=1}^{q_i}\mathbf{A}_i{}^{j}F(s_i\circ)\mathbf{A}_i{}^{q_i+1-j})\bar{R}_{i-p}\right)\right|, \end{aligned} \tag{36}$$

which is a function of q_i ($1 \le i \le p+a$) and $s_i\circ$ ($1 \le i \le a$).

Since we know the reduced size is d, we have the constraint

$$\sum_{i=1}^{p+a} q_i = d \tag{37}$$

and they must be non-negative integer

$$q_i \in Z^+ \cup \{0\}, \quad 1 \le i \le p+a \tag{38}$$

Moreover, assume we have the statistical description for the symbols, i.e.,

$$s_i \circ \in \omega_i, \tag{39}$$

where ω_i is some statistical description for s_i. Then we need to solve the following statistical optimization problem

$$min_{q_1,\ldots q_{p+a}} \quad f(q_1 \ldots, q_{p+a}; s_1\circ, \ldots s_a\circ) \tag{40}$$

$$s.t. \quad \sum_{i=1}^{p+a} q_i = d \tag{41}$$

$$q_i \in Z^+ \cup \{0\}, \quad 1 \le i \le p+a, \tag{42}$$

$$s_i\circ \in \omega_i, \quad 1 \le i \le a \tag{43}$$

Note that the objective function in Eq. (43) is statistical, and we try to minimize its worst case value for all possible values of the symbols, which gives the following constrained min-max mixed integer optimization problem

$$\min_{q_1,\cdots q_{p+a}} \quad \max_{s_1\circ,\ldots s_a\circ} f(q_1, \ldots q_{p+a}; s_1\circ, \ldots s_a\circ) \tag{44}$$

$$s.t. \quad \sum_{i=1}^{p+a} q_i = d \tag{45}$$

$$q_i \in Z^+ \cup \{0\}, \quad 1 \le i \le p+a, \tag{46}$$

$$s_i\circ \in \omega_i, \quad 1 \le i \le a \tag{47}$$

Generally the min-max problem is very hard to solve, especially the above problem is mixed-integer based and is nonlinear. Below we propose an efficient heuristic algorithm to approximate solve it, based on the key observation that the constraints for $s_i\circ$ and q_i can be separated (i.e., there are no cross terms). Accordingly, we separate the min-max problems into the following two sub-problems. Similar algorithm has been used in [15] for decoupling capacitance budgeting problem.

$$(\mathbf{P}1:) \min_{q_1,\cdots q_{p+a}} \quad f(q_1, \ldots q_{p+a}; s_1\circ, \ldots s_a\circ) \tag{48}$$

$$s.t. \quad \sum_{i=1}^{p+a} q_i = d \tag{49}$$

$$q_i \in Z^+ \cup \{0\}, \quad 1 \le i \le p+a, \tag{50}$$

which is a integer-minimization problem for fixed $s_1\circ, \ldots, s_a\circ$, and

$$(\mathbf{P}2:) \max_{s_1\circ,\ldots,s_a\circ} \quad f(q_1, \ldots, q_{p+a}; s_1\circ, \ldots s_a\circ) \tag{51}$$

$$s.t. \quad s_i\circ \in \omega_i, \tag{52}$$

which is a maximization problem for fixed $q_1, \ldots, q_{p+a}$. Efficient algorithms exist to solve the above two sub-problems [16] and is beyond the scope of this paper.

We iteratively solve (**P1**) and (**P2**) until the solution converges. In certain cases, the solution may fail to converge, and we force the algorithm to exit after certain iterations to guarantee the convergence.

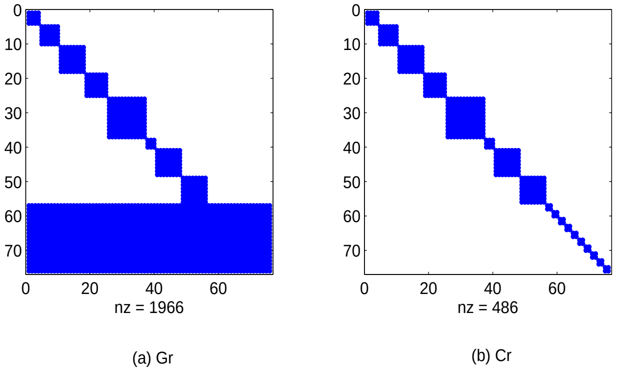
This heuristic algorithm can provide optimal solution only in the case that (**P1**) is convex and (**P2**) is concave. Otherwise, the optimality is not guaranteed. However, experimental results show that significant accuracy improvement is achieved by such heuristic.

4. EXPERIMENTAL RESULTS

In this section, we present experimental results for runtime and accuracy comparison between the S^{2}MOR method and the symbol isolation, the nominal projection and the first order expansion methods. All the methods are implemented in C++. We use the CSparse as the sparse matrix package [17]. We run experiments on a UNIX workstation with Pentium IV 2.66G CPU and 1G RAM. Finally, we use MOSEK as the linear/quadratic programming solver [18] for optimal projection order decision.

4.1 Algorithm Verification

Fig. 1 illustrates the sparsity of the reduced matrices from the S^{2}MOR method. The original circuit is a low-noise amplifier (LNA) design with parasitics, which contains 4920 nodes, 8 ports 10 symbols. The circuit is reduced to order 76 by the S^{2}NMOR method. From the figure we can see that although there is no special structure for the original matrices, the reduced $\hat{G}_r$ and $\hat{C}_r$ from the S^{2}MOR method has LBBD structure with sparsity 34% and block diagonal structure with sparsity 7.7% respecitvely.



(a) Gr

(b) Cr

Figure 1: Sparsity pattern for the matrices $\hat{G}_r$ and $\hat{C}_r$ after reduction from S^{2}MOR.

Next we verify the effectiveness of our optimal projection order decision. We perform Monte Carlo simulation on the symbol values for 10,000 runs on the same LNA circuit, and compare the error between uniform projection order decision, assuming the same projection order for all symbols, and our stochastic programming based projection order decision. The error is defined as the integral of the absolute difference between the original time domain waveform and the one of the reduced circuit. The error distributions for both methods are shown in Fig. 2. From the figure we can see that by our method the mean error is reduced by approximately 30% and the 3σ error by 50%, which verifies the effectiveness of the proposed stochastic programming based approach.

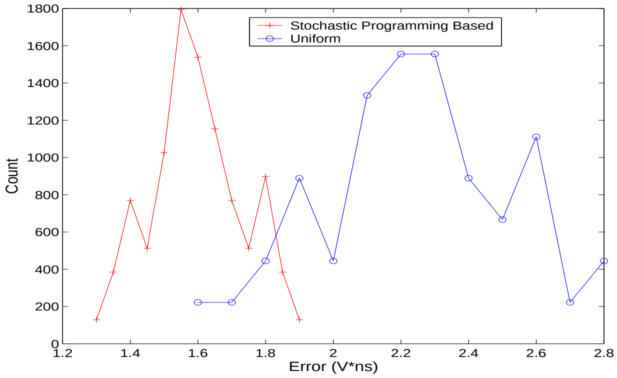


Figure 2: Accuracy Comparison between uniform projection order and our stochastic programming based approach based on the Monte Carlo simulation.

4.2 Accuracy Comparison

In this section, we compare the accuracy of the three existing methods and the S^{2}MOR method on some real industrial designs. For fair comparison, we reduce the circuit to different orders for different methods such that the reduced models require the same memory space. This

Table 2: Testbench information.

ckt name	node #	port #	symbol #
LNA1	1392	14	6
LNA2	5573	33	14
LNA3	11380	79	137
LNA4	49965	147	661

Table 3: Accuracy comparison between symbol isolation (S.I.), nominal projection (N.P.), first order expansion (F.E) and the S^{2}MOR with different variation amount (var) of symbol values. All the errors are in the unit of V*ns.

ckt name	var	S.I.	N.P.	F.E	S^{2}MOR
LNA1	10%	1.2	0.7	**0.9**	0.6 (-24%)
	30%	**1.2**	8.4	6.8	0.6 (-50%)
LNA2	10%	3.7	**1.1**	1.9	0.8 (-27%)
	30%	**3.6**	11.6	17.8	0.8 (-78%)
LNA3	10%	4.2	**3.7**	3.9	0.9 (-76%)
	30%	**4.2**	13.7	19.8	1.0 (-76%)
LNA4	10%	**5.2**	6.7	N.A.	1.6 (-69%)
	30%	**5.2**	28.4	N.A.	1.6 (-69%)

Table 4: Runtime comparison between symbol isolation (S.I.), nominal projection (N.P.), first order expansion (F.E.) and the S^{2}MOR method. The reduced sizes are also reported (size). All units are in seconds.

ckt name	method	size	reduce	factor	update
LNA1	S.I.	300	427	43.7	13.6
	N.P.	300	421	43.7	43.6
	F.E.	300	374	43.7	43.7
	S^{2}MOR	930	484	7.6	1.5
LNA2	S.I.	420	835	86.4	38.4
	N.P.	420	816	86.5	86.9
	F.E.	420	741	86.4	86.5
	S^{2}MOR	1340	975	11.3	2.2
LNA3	S.I.	480	1124	91.5	47.6
	N.P.	480	1190	91.4	91.2
	F.E.	480	1011	91.6	91.5
	S^{2}MOR	1440	1238	12.3	2.9
LNA4	S.I.	500	2977	123.6	61.2
	N.P.	500	2918	123.6	123.5
	F.E.	500	2715	123.6	123.6
	S^{2}MOR	1610	3020	13.1	3.6

implies that the three existing methods have the same reduced order because they are all dense, while the S^{2}MOR method can have a much higher reduced order due to its sparsity for the same memory space.

We ompare the accuracy between the four different methods on different benchmarks, and the results are shown in Table 3. The information of the testbenches we used is shown in Table 2. They are all LNA's from extraction. From Table 3 we can see that for different circuits and for different variation amount, the S^{2}MOR method always have the smallest error compared with the other three methods. Specifically, when the variation amount is 10%, compared with the best one of the other three methods shown in bold, the S^{2}MOR method reduces the error by up to 76%; When the variation amount is 30%, the S^{2}MOR method reduces the error by up to 78%. Also note that the first order expansion method cannot finish the testbench LNA4 due to its singularity. This is a direct result from the lack of orthonormalization procedure for the projection matrix.

4.3 Runtime Comparison

In Table 4, we report the runtime for reduction, factorization, and update for the reduced model on different testbenches. All the circuits are reduced to a different size such that the reduced circuits from different methods requires the same memory space. From the table we can see that the reduction time is almost the same for all the methods. The factorization time is reduced by up to 9.4× for the S^{2}MOR method on the largest testbench, although the reduced circuit size of the S^{2}MOR method is almost 3× larger. This speedup comes from the efficient factorization algorithm based on the LBBD structure of the reduced circuit. Finally, the factorization update time for symbol value changes is reduced by 17× for our method compared with the symbol isolation method, which uses low rank update. This is because we only need to refactorize a matrix with size equal to the number of symbols. All the above results further verify the correctness of our conclusion in Table 1.

5. CONCLUSIONS

Symbolic model order reduction (SMOR) is to reduce the complexity of a model with symbolic parameters. It is an important problem in analog circuit synthesis and digital circuit modeling with process variations. However, existing symbolic model order reduction (SMOR) methods do not scale well with the number of symbols or with the model order. The literature only reports experimental results for circuits with fewer than ten symbols. This paper presents a scalable SMOR algorithm, namely S^{2}MOR, which handles circuits with up to 49965 nodes and 661 symbols in 50 minutes from experimental results. Compared with existing SMOR algorithms, given the same memory space, S^2MOR improves accuracy by up to 78% at a similar reduction time. In addition, the factorization and simulation of the reduced model by S^2MOR is up to 17× faster.

6. REFERENCES

[1] T. Halfmann, R. Sommer, and T. Sichmann, "Application of symbolic circuit analysis: An overview and recent resutls in nonlinear model generation," in *International Conference on Electronics, Circuits and Systems*, pp. 957–960, 1999.

[2] X.-D. Tan and C.-J. Shi, "Symbolic circuit-noise analysis and modeling with determinant decision diagrams," in *ASPDAC*, pp. 283–288, 2000.

[3] G. G. Gielen and W. M. Sansen, *Symbolic analysis for automated design of analog integrated circuits: a tutorial overview*. Kluwer Academic Publisher, 1991.

[4] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macro-modeling algorithm," *IEEE Trans. on CAD*, pp. 645–654, 1998.

[5] Y. Su and et. al., "SAPOR: Second-order arnoldi method for passive order reduction of RCS circuits," in *IEEE/ACM ICCAD*, pp. 74–79, 2004.

[6] M. Rewienski and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Trans. on CAD*, pp. 155–170, 2003.

[7] B. N. Bond and L. Daniel, "A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems," *IEEE Trans. on CAD*, pp. 2116–2129, 2007.

[8] Y. Li, Z. Bai, Y. Su, and X. Zeng, "Parameterized model order reduction via a two-directional arnoldi process," in *IEEE/ACM ICCAD*, pp. 868–873, 2007.

[9] Y. Shi and L. He, "Empire: An efficient and compact multiple-parameterized model order reduction method for physical optimizaion," in *ISPD*, pp. 51–58, 2007.

[10] G. Shi, B. Hu, and C. J. R. Shi, "On symbolic model order reduction," *IEEE Trans. on CAD*, pp. 1257–1272, 2006.

[11] Y. Shi, H. Yu, and L. He, "Samson: A generalized second-order arnoldi method for reducing multiple source linear network with susceptance," in *ISPD*, pp. 25–32, 2006.

[12] P. Li and W.-P. Shi, "Model order reduction of linear networks with massive ports via frequency-dependent port packing," in *IEEEACM/ DAC*, pp. 267–272, 2006.

[13] B. Yan, X.-D. Tan, P. Liu, and B. McGaughy, "Sbpor:second-order balanced truncation for passive order reduction of rlc circuits," in *IEEEACM/ DAC*, pp. 158–161, 2007.

[14] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, 3 ed., 1989.

[15] Y. Shi, J. Xiong, C. C. Liu, and L. He, "Efficient decoupling capacitance budgeting considering operation and process variations," *IEEE Trans. on CAD*, pp. 1253–1263, 2008.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[17] T. Davis, *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[18] in *http://www.mosek.com*.