Symbolic Model Order Reduction*

Bo P. Hu, Guoyong Shi, and C.-J. Richard Shi
Mixed-Signal CAD Research Laboratory
Department of Electrical Engineering
University of Washington, Seattle, WA 98195
{hubo, gshi, cjshi}@ee.washington.edu

Abstract – Symbolic Model Order Reduction (SMOR) is the problem of reducing a large circuit that contains symbolic circuit parameters to smaller low order models at its ports. Several methods, including symbol isolation, single frequency point reduction, and multiple frequency point reduction, are described and compared. Test circuits with simulation results are presented to demonstrate the accuracy and efficiency of SMOR.

I. Introduction

Model Order Reduction (MOR) is an efficient technique for fast and accurate simulation of RLC circuits. Various types of algorithm such as AWE [1], PVL [3], and PRIMA [4] have been developed to numerically reduce a large RLC network to smaller models, which can be used to replace the original large network in the subsequent simulation.

The drawback of those numeric MOR algorithms is the lack of flexibility. Whenever some element or parameter values in the large circuit change, the reduction has to be repeated.

Progress has been made to construct parameterized reduced model for interconnect [7], and to include variation analysis in RLC interconnect modeling [9]. Each method is efficient for some special cases.

In this paper, we present Symbolic Model Order Reduction (SMOR) as an attempt to overcome the limit of numeric model order reduction methods; several approaches are developed to handle symbolic elements inside a large circuit. Ideally, with SMOR, we will obtain a small model with symbols inside, and those symbols represent circuit elements (such as R, L, or C) or/and design parameters of interest (such as width, length of interconnect). The advantage of SMOR is its flexibility: when those symbolic elements or parameters change values, we can quickly update the symbolically reduced model, without performing the reduction again.

In this paper, first the PRIMA algorithm, on which SMOR is build, is briefly reviewed, followed by symbol isolation method, single frequency point reduction and multiple frequency point reduction method. Examples are given after the description of each method. Finally, the difficulties of SMOR are discussed.

* This research was partly supported by DARPA NeoCAD Program under Grant No. 66001-01-1-8920 and in part by SRC under Contract No. 2001-TJ-921.

II. Review on PRIMA

Let us first examine the PRIMA algorithm briefly. Consider a SISO system that can be described by the following equation:

$$C\frac{dx}{dt} = -G\mathrm{x} + bu \qquad (1)$$
$$y = \ell x$$

where u is the external stimulus to the system, b is the input vector, ℓ is the output vector, and y is the output of the model under the stimulus u. C and G are the system matrices which describe the dynamic behavior.

The key issue in model order reduction is to find a transformation matrix V. There are many ways to compute V, and in PRIMA it is given by:

$$V = K_q\left(G^{-\mathbf{1}}C, G^{-\mathbf{1}}b\right) \qquad (2)$$

The notation $K_q(A,b)$ denotes the Krylov subspace spanned by the vectors $[b, Ab, \cdots, A^{q-1}b]$. The original system can be reduced by congruence transformation [2] as follows:

$$C_r = V^{\mathbf{T}}CV,\ \ G_r = V^{\mathbf{T}}GV, b_r = V^{\mathbf{T}}b,\ \ell_r = \ell V \qquad (3)$$

And after the transformation, we have a smaller model, which is described as follows:

$$C_r\frac{dz}{dt} = -G_{\mathrm{r}}z + b_r\mathrm{u} \qquad (4)$$
$$y = \ell_r z$$

The sizes of the system matrices C_r and G_r are much smaller than those of C and G. This reduced model can be used to achieve faster simulation speed, and in the same time reserve the required accuracy of the original system at its input and output ports [2] [4].

III. SMOR by Symbol Isolation

In the simplest but quite frequently happening scenario, the sweeping analysis of some circuit elements is desired, and it will be convenient to construct a compact model with those *symbolized* elements retained. The symbol isolation method is for such a purpose.

The basic idea is to isolate and replace each element of interest with a symbol. For each symbol, we model its interaction with the circuit as a port. For a typical two terminal element, its terminal voltage (or current flowing through it) would be added as the input stimulus to the circuit, and the current flowing through (or voltage across) the element would be the output under such a stimulus. After adding such extra inputs and outputs to the circuit, we carry out reduction on the circuit (which is purely numeric after isolating those symbolic elements from it), and then combine those symbolic elements to construct a symbolic model. The symbolic model can be reused whenever the value of any symbolic element changes.

The isolation method is discussed for the case of R, L or C element separately. In the case of multiple symbolic elements, there will be an extra step to combine them together into one compact symbolic model.

For simplicity, we will consider the SISO dynamic system as discussed in Section II. In the first case, we consider that a capacitor in this SISO system between nodes *i* and *j* is going to be isolated and symbolized. As illustrated in the following diagram, first, we isolate the capacitor and add one port to the large network, then reduce the large network, and finally put the capacitor back to make a compact symbolic model.

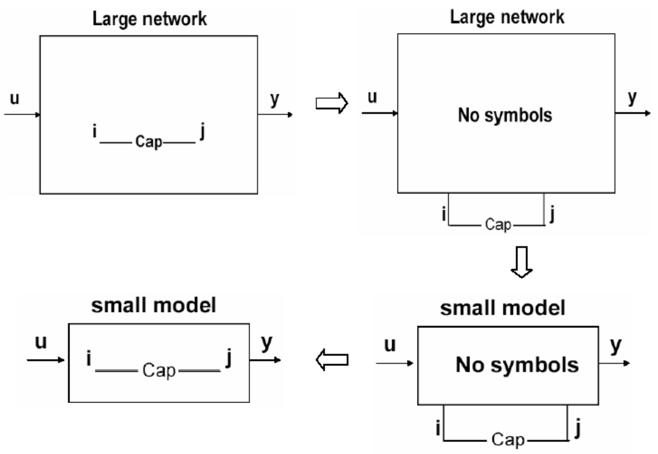


Fig. 1. Flow graph of isolation method when a capacitor is treated as a symbolic element.

Let $\hat{c}$ be the capacitance of the isolated capacitor, which is treated as a symbol. Let $\hat{I}$ and $\hat{v}$ be the current through (from *i* to *j*) and voltage across the capacitor respectively. We can describe this new system as follows:

$$\begin{bmatrix} \mathbf{C} & 0 \\ 0 & 0 \end{bmatrix} \frac{d}{dt}\begin{bmatrix} x \\ \hat{I}\end{bmatrix} = -\begin{bmatrix} G & E_{ij} \\ -E_{ij}^T & 0\end{bmatrix}\begin{bmatrix} x \\ \hat{I}\end{bmatrix} + \begin{bmatrix} 0 \\ -1\end{bmatrix}\hat{v} + \begin{bmatrix} b \\ 0\end{bmatrix} u \quad (5)$$

$$\hat{c}\frac{d\hat{v}}{dt} = \hat{I}$$

$$\begin{bmatrix} y \\ \hat{I}\end{bmatrix} = \begin{bmatrix} \ell & 0 \\ 0 & 1\end{bmatrix}\begin{bmatrix} x \\ \hat{I}\end{bmatrix}$$

where

$$E_{ij} = [\cdots 1 \cdots -1 \cdots]^T$$

is a column vector containing all zeros but 1 and –1 at the locations corresponding to v_i and v_j in x, respectively.

After the isolation of the capacitor, the original SISO system becomes a MIMO system (there are two inputs and two outputs). If we apply the congruence transformation [2], we have

$$\begin{bmatrix} x \\ \hat{I}\end{bmatrix} = Vz \qquad \text{(V is the transformation matrix)}$$

$$b_v = V^{\mathrm{T}}\begin{bmatrix} 0 \\ -1\end{bmatrix}, \qquad b_r = V^{\mathrm{T}}\begin{bmatrix} b \\ 0\end{bmatrix}$$

$$\ell_r = [\ell \quad 0]V, \qquad \ell_v = [0 \ 1]V$$

$$C_r = V^T\begin{bmatrix} C & 0 \\ 0 & 0\end{bmatrix}V, \qquad G_r = V^{\mathrm{T}}\begin{bmatrix} G & E_{ij} \\ -E_{ij}^T & 0\end{bmatrix}V$$

Then the MIMO system is reduced to:

$$C_r\frac{d\mathbf{z}}{dt} = -G_r\mathbf{z} + b_r u + b_v \hat{u} \quad (6)$$

$$\begin{bmatrix} y \\ \hat{I}\end{bmatrix} = \begin{bmatrix} \ell_r \\ \ell_v\end{bmatrix} z$$

$$\hat{c}\frac{d\hat{v}}{dt} = \hat{I}$$

If we introduce $\hat{v}$ as a state variable, the MIMO system becomes an SISO system again with $\hat{c}$ as a symbol.

$$\begin{bmatrix} C_r & 0 \\ 0 & \hat{c}\end{bmatrix}\frac{d}{dt}\begin{bmatrix} z \\ \hat{v}\end{bmatrix} = -\begin{bmatrix} G_r & -b_v \\ -\ell_v & 0\end{bmatrix}\begin{bmatrix} z \\ \hat{v}\end{bmatrix} + b_r u \quad (7)$$

$$y = [\ell_r \quad 0]\begin{bmatrix} z \\ \hat{v}\end{bmatrix}$$

After the combination step, we obtain a compact symbolic model, which is an SISO system as the original circuit, but has the extra flexibility to quickly update the symbolic capacitance.

For the resistor case, we follow the same procedure as for capacitor, and the system could be reduced to:

$$\begin{bmatrix} C_r & 0 \\ 0 & 0\end{bmatrix}\frac{d}{dt}\begin{bmatrix} z \\ \hat{v}\end{bmatrix} = -\begin{bmatrix} G & -b_v \\ -\ell_v & 1/\hat{r}\end{bmatrix}\begin{bmatrix} z \\ \hat{v}\end{bmatrix} + b_r u \quad (8)$$

$$y = [\ell_r \quad 0]\begin{bmatrix} z \\ \hat{v}\end{bmatrix}$$

For inductor, we introduce $\hat{I}$ as additional variable and the system could be reduced to:

$$\begin{bmatrix} C_r & 0 \\ 0 & \hat{L} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} z \\ \hat{I} \end{bmatrix} = -\begin{bmatrix} G_r & -b_v \\ -\ell_r & 0 \end{bmatrix} \begin{bmatrix} z \\ \hat{I} \end{bmatrix} + b_r u \quad (9)$$

$$y = \begin{bmatrix} \ell_r & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \hat{I} \end{bmatrix}$$

In the case of multiple symbolic elements, we first need to identify those symbolic elements as ports in the circuit, introduce auxiliary variables as needed, reduce the remaining numerical model, then combine the symbolic ports with the numerically reduced model. The main processing overhead involved in this process is in the isolation and combination part.

We use the circuit shown in Fig. 2 to illustrate the effectiveness of the symbol isolation method. The circuit is an RLC ladder network with 301 elements (101 nodes). The size of the system matrices describing the original network is 103 by 103, and the size of the symbolic model is 12 by 12. The resistor, capacitor, or inductor placed between nodes 11 and 12 is treated as a symbolic element.

Simulation results are shown in Figs. 3 – 5. The results demonstrate that the model is accurate compared to the full order system, and has the flexibility to handle symbolic elements.

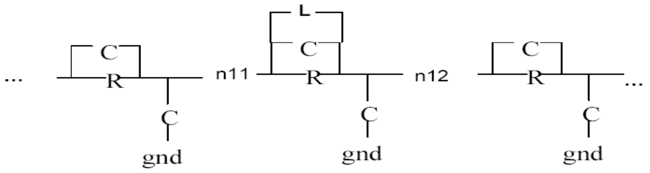


Fig. 2. A ladder circuit (301 elements).

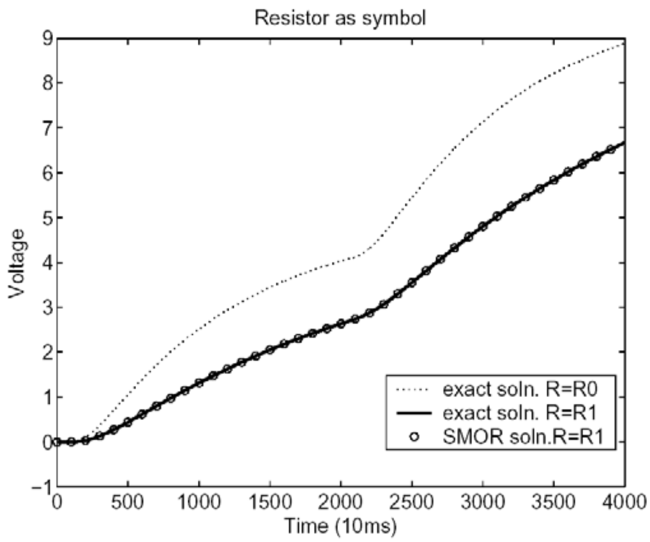


Fig. 3. The resistor between node 11 and node12 is a symbolic element. The time domain response to an external voltage stimulus is compared to the result given by the unreduced system, at both the nominal value and changed value of the resistor. The reduced symbolic model size is 12; the original network's size is 103.

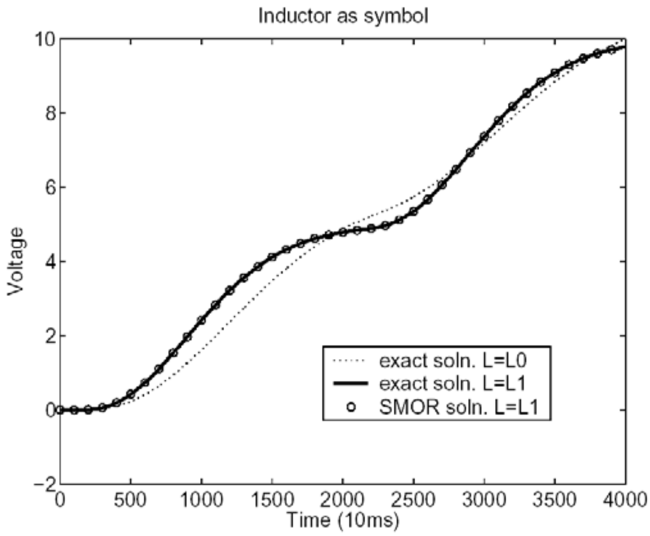


Fig. 4. Inductor between node 11 and node 12 is the symbol.

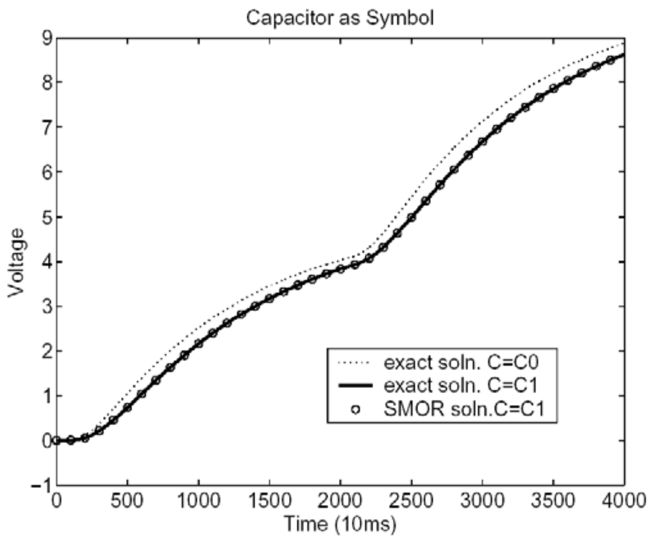


Fig. 5. Capacitor between node 11 and node 12 is the symbol.

In summary, the isolation method described above is useful in fast sweep analysis on some variables of a large circuit. However, this method is only applicable to a few symbolic elements. New approaches become necessary for more general problems.

IV. SMOR at Single Frequency Point

In the general case, it is not practical to isolate symbols one by one as in the previous example. Further the number of ports can grow rapidly, and thus decreases the value of model order reduction. In this section we propose a single frequency point based method for symbolic model order reduction.

Let $X(s) = T(s)U(s)$, where $T(s) = (Cs+G)^{-1}b$. The frequency point method arises from the expansion of $T(s)$, i.e.

$$
\begin{aligned}
T(s) &= (Cs+G)^{-1}b \\
&= [C(s-\sigma)+(C\sigma+G)]^{-1}b \\
&= \sum_{i=0}^{\infty}[-(C\sigma+G)^{-1}C]^{i}(C\sigma+G)^{-1}b(s-\sigma)^{i}
\end{aligned}
$$

σ is called a frequency point. In this section we consider $\sigma = 0$. In the next section, we consider multiple σ's, so called multiple frequency points.

We will restrict our discussion to the SISO network that can be described by the equation (1), where C and G contain symbols. The single frequency point method constructs transformation matrix V, from the Krylov subspace $K_q\left(G^{-1}C, G^{-1}b\right)$, then performs congruence transformation symbolically. The following algorithm constructs a symbolic transformation matrix V:

Algorithm 1

1). Symbolically inverse G (expensive operation)

$$G^{-1} = \text{inverse}(G)$$

2). Perform matrix-vector multiplication

$$v_1 = G^{-1}b$$

3). For *k=2* to *q* (*q* is the size of reduced model)

$$v_k = G^{-1}Cv_{k-1}$$

4). For *k=1* to *q*, construct transformation matrix V

$$V(:,k) = v_k$$

V is an *N* by *q* matrix, *N* is the original size of the network, and *q* is the size of reduced model.

After V is constructed, the second step is to perform congruence transformation as described in (3), except that the matrix multiplication involves symbolic computation.

From our implementation with Maple [10], the two steps described above are quite complex, and the complexity grows exponentially with the sizes of *N*, *q* and the number of symbols inside the network to be reduced.

Note that in Algorithm 1, V is generated without orthonormalization. This reduces the symbolic computation complexity, but causes numerical instability problem. Since V is not orthonormalized, it could be extremely ill conditioned once every symbol is substituted by a numerical value. If we use such a V to do the congruence transformation, the reduced model may also become ill conditioned and likely useless.

To ease the condition number problem, we introduce a pseudo orthonormalization method.

The pseudo orthonormalization method is based on the assumption that the value of each symbol in matrices C and G will only change slightly from its nominal value. If this assumption holds, the transformation matrix V will not change much from its nominal value. The idea of pseudo orthonormalization method is to construct another numerical transformation matrix V_n (labeled as V_n to distinguish from V, which is a symbolic matrix; and if every symbol inside V takes its nominal value, then V_n equals V) along with the construction of the symbolic transformation matrix V; and use V_n as a tool to reduce the condition number of V.

The algorithm is described in pseudo code as follows:

(Notation: *vn(i)* is the *i*th column vector of the numerical transformation matrix V_n, and *v(i)* is the *i*th column vector of the symbolic transformation matrix V.)

Algorithm 2.

1). Generate vn(1), normalize vn(1) by coefficient *c11*

$$vn(1) \Leftarrow vn(1) \times c11$$

Generate *v(1),* and pseudo normalize v(1) with *c11*

$$v(1) \Leftarrow v(1) \times c11$$

2). Generate *vn(2),* orthonormalize it with respect to *vn(1)*

$$vn(2) \Leftarrow vn(2) \times c22 + vn(1) \times c21$$

Generate *v(2),* pseudo orthonormalize it with respect to *v(1),* using *c22* and *c21*

$$v(2) \Leftarrow v(2) \times c22 + v(1) \times c21$$

3). Continue until we get all vectors pseudo orthonormalized.

However, during the implementation we observed that the coefficients *cii* are very sensitive to the variation of each symbol, e.g. even when a symbol just changes very little (far less than 1%) from its nominal value, with every other symbol remaining unchanged at all, the corresponding *cii* will change a lot and cause the pseudo orthonormalization fail to produce a better conditioned transformation matrix V for most of the cases.

As a result, the reduced model must have low order (about 10) by using the single frequency point method for symbolic model order reduction. For some systems, low order models are sufficiently accurate, especially in overly damped systems, where oscillation is weak. However, in general the low order approximation is not sufficient, especially when the inductance is in presence.

Figure 7 shows an example of the quick increase of the condition number of matrix V. The comparison is made between the Algorithm 1 (no orthonormalization) and Algorithm 2 (pseudo orthonormalization method). The example is based on the circuit presented in Section III, and the resistor between nodes 10 and 11 is changed from its nominal value by 1%.

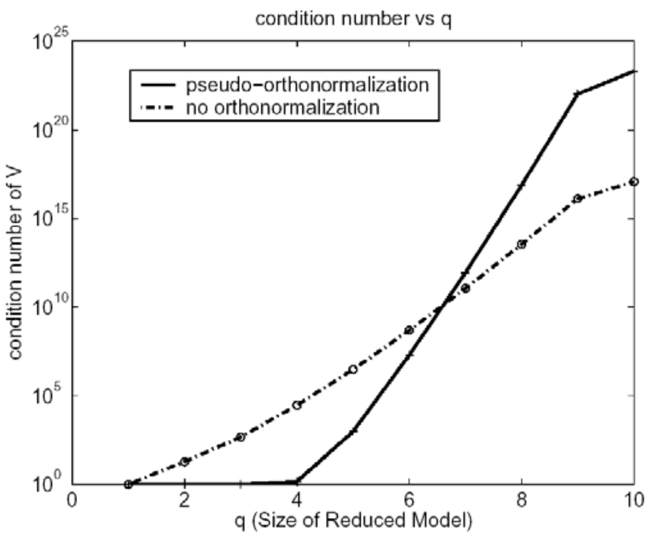


Fig 7. Condition number of V versus the size of reduced model q. It shows that the pseudo orthonormalization method is effective only when q is small.

V. SMOR by Multiple Frequency Point Mapping

To improve accuracy without greatly increasing the condition number of transformation matrix V, in this section we propose a multiple frequency point based reduction method. Using multiple frequency points (imaginary and/or real) in numerical model order reduction often generates more compact and accurate models [5] [6] [3], and here we use it for symbolic model order reduction.

The basic idea of multiple frequency point based method is to use the Krylov subspace:

$$K_q\left((C\sigma+G)^{-1}C,(C\sigma+G)^{-1}b\right)$$

For each choice of σ, we generate a few vectors in the Krylov subspace, and put them together to form the transformation matrix V. Specifically, we have the following algorithm.

Algorithm 3.

1). Symbolically inverse $(G+\sigma C)$

$$(G+\sigma C)^{-1} = \text{inverse}(G+\sigma C)$$

2). Perform matrix-vector multiplication

$$v_1=(G+\sigma C)^{-1}b$$

3). For $k=2$ to q (q is the size of reduced model)

$$v_k=(G+\sigma C)^{-1}Cv_{k-1}$$

4). For $k=1$ to q, construct transformation matrix V_σ

$$V_\sigma(:,k)=v_k$$

The procedure is similar to that described in Section IV, except that we replace G with $(G+\sigma C)$. The transformation matrix is denoted as V_σ correspondingly.

For the general case, when we expand the system at multiple frequency points $\{\sigma_1,\sigma_2,\cdots,\sigma_n\}$ the final transformation matrix V is formed by grouping each subspace together as follows:

$$V=colsp\{V_{\sigma_1},V_{\sigma_2},...,V_{\sigma_n}\} \qquad (10)$$

If we use the above V to transform the original system using Equation (3), the resulting model will be a good approximation to the original system in a broader frequency range than single frequency point reduction.

It is straightforward to construct V if all the frequency points chosen are real; if σ_i is complex frequency point, an approach to obtaining a real V_{σ_i} was developed in [6][8]. Putting in a simple way, if one frequency point σ_i, is complex, the corresponding V_{σ_i} will be the combination of the real part and the imaginary part, rather than a complex subspace.

The important problem in multiple frequency point reduction is the selection of frequency points set $\{\sigma_1,\sigma_2,\cdots,\sigma_n\}$. Two factors are important in this selection. First, the frequency points should cover the interested frequency range to satisfy certain accuracy requirement; second, the frequency points should be carefully chosen to minimize the condition number of the union transformation matrix V (which is the union of the transformation matrix at each frequency point).

One approach in numerical multiple frequency point reduction is to choose evenly spaced frequency points [6], and use orthonormalization procedure to make sure the union transformation matrix V is well conditioned. However, in symbolic model order reduction, this approach is not feasible, which means we have to rely on the careful selection of the set of frequency points to lower the condition number of V.

To minimize the condition number of union transformation matrix V, the set of frequency points $\{\sigma_1,\sigma_2,\cdots,\sigma_n\}$ should be well separated from each other, so that their corresponding transformation matrices do not overlap. If they overlap, the union transformation matrix V will have many vectors that are almost dependent on each other, and become ill conditioned.

Our first selection scheme is a very simple one: First, choose 0 as the first frequency point; then choose $\sigma_{\max}$ as the highest frequency point; and then choose σ_{mid} as the middle frequency point, based on the estimation of the dominant pole of the original circuit. Using the 3-point scheme, the resulting model would often be fairly good.

The second method is more computationally expensive, but compared with the saving of SMOR, such a cost is still justifiable. This selection scheme is based on the evaluation of the frequency domain behavior of the original system at each symbol's nominal value, using numerically reduced model by any reduction method. First, we perform numerical reduction (PRIMA, PVL etc) on the original system, and quickly evaluate the frequency response on the reduced model, then choose frequency points based on this evaluation.

One fundamental assumption in symbolic model order reduction is that the symbol values will not change drastically from their nominal values, hence the set of frequency points should still work, even if the symbols take different values from their nominal values.

There is a tradeoff between the number of frequency points and the condition number of the transformation matrix V. More frequency points will produce better accuracy in the reduced system, at the cost of high condition number of V. The general requirement for the condition number of V is by the order of 10^{10}, beyond that, the possibility of numeric breakdown will be very high. And in the practice of SMOR, we choose as many frequency points as possible under the constraint of the condition number of V.

To test the idea of multiple frequency point reduction, we consider the circuit shown below. This circuit is composed of 300 similar blocks of RLC elements, and the value of each element are different from block to block, however within a certain range (*R: 10-100 Ohm, L: 1-10 nH, C: 1-10 pF*).

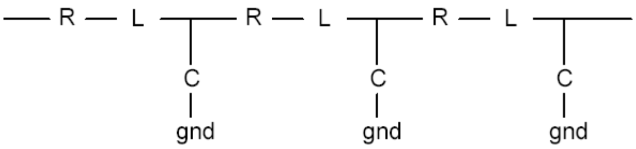
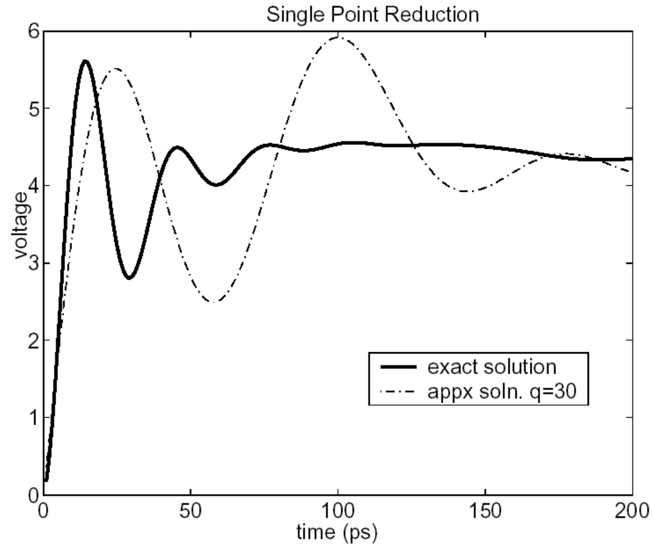
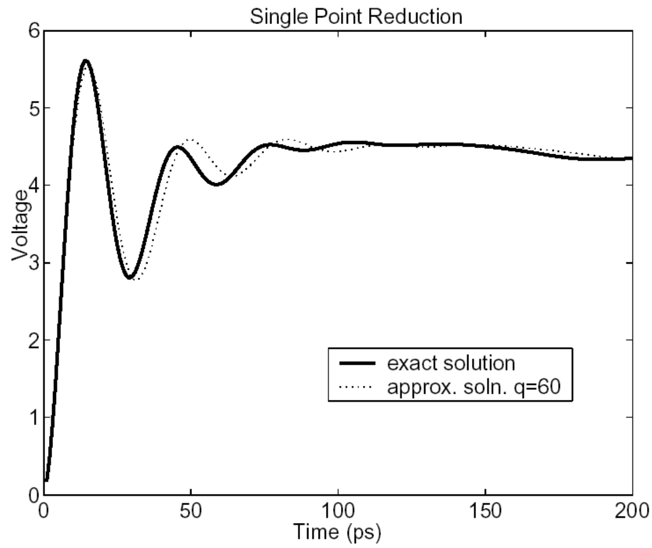


Fig. 6. The diagram of the test circuit (3 out of 300 blocks are plotted here)

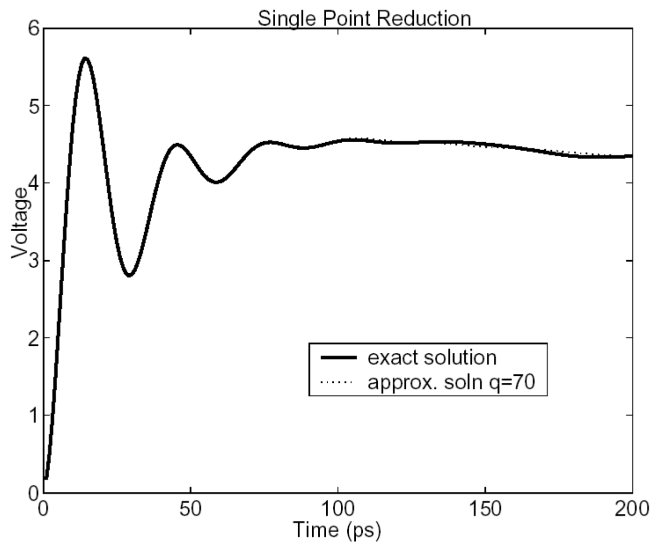
First, the efficiency of multiple frequency point reduction is demonstrated compared with single point reduction. Here the stimulus is a step voltage input at the left most node (node 1); the output is the 3rd node's voltage. The size of the original network is 902, and the size of the reduced model is increased from q=30 up to q=70 for the PRIMA algorithm; and for the multiple frequency point symbolic model, the size is just 9.



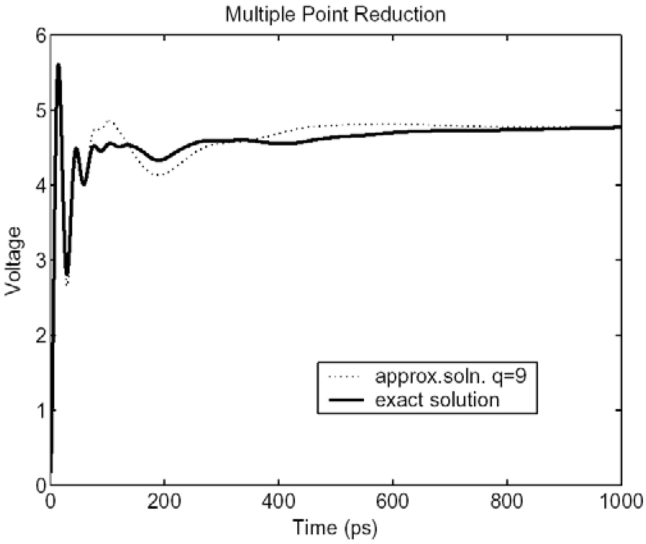
(a)



(b)



(c)



(d)

Fig 8. a),b),c): The step input response of reduced model (at single point $\sigma = 0$), with different model size: $q = 30$, $q = 60$, $q = 70$. d): step input response of 3-point reduced model: $q = 9$.

The test results show that the accuracy of single point reduction is improved by increasing the model size. And to obtain a reasonably good accuracy, the model size should be sufficiently large. However, using multiple frequency point reduction, we can achieve good accuracy with a small model size as shown above, in which case the model size is just $q = 9$, much smaller than the $q \geq 60$ in single point reduction.

From the comparison in the example above we conclude that the multiple frequency point based method is likely most effective for symbolic MOR, because it avoids computing high order Krylov subspace and in turn reduces the ill conditioning of the transformation matrix.

VI. Conclusion

In this paper, the problem of symbolic model order reduction (SMOR) is described. Some algorithms based on PRIMA are proposed and tested. Unlike numeric MOR, which is quite mature for linear dynamic system, symbolic MOR is still not well studied. The methods and algorithms proposed in this paper are only some preliminary results. Further research is needed for practical applications of symbolic model reduction techniques.

References

[1] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis", *IEEE Trans. Computer-Aided Design,* vol. 9, pp.352-366, April 1990.

[2] K. J. Kerns, I. Wemple, and A. Yang, "Stable and efficient reduction of substrate model networks using congruence transforms," in *Proc. IEEE Int. Conf. Computer-Aided Design (ICCAD),* 1995, pp.207-214.

[3] P. Feldman and R. W. Freund "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 5, pp. 639-649, May 1995.

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

[5] E. Chiprout and M. Nakhla, "Analysis of interconnect networks using complex frequency hopping (CFH)," in *IEEE Trans. Computer-Aided Design*, vol. 14, Feb. 1995, pp.186-200.

[6] Eric Grimme, *Krylov Projection Methods for Model Reduction*, Ph.D Thesis, Univ. of Illinois, 1997.

[7] Luca Daniel, C.S. Ong, S. C. Low, K. H. Lee, and J. White, "Geometrically parameterized interconnect performance models for interconnect synthesis", *Proc. Int'l Symposium on Physical Design,* 2002, pp. 202-207.

[8] A. Ruhe, "The rational Krylov algorithm for nonsymmetric eigenvalue problems III: Complex shifts for real matrices," *BIT*, vol.34, pp.165-176,1994.

[9] Y. Liu, L.T. Pileggi, and A.J. Strojwas, "Model order-reduction of RC(L) interconnect including variational analysis". in *Proc. 36th Design Automation Conference*, New Orleans, LA, 1999, pp. 201-206.

[10] Waterloo Maple Inc, *Maple 7,* 2001.