

Time Domain Model Order Reduction by Wavelet Collocation Method

Xuanzeng¹ Lihong Feng¹ Yangfeng Su² Wei Cai³ Dian Zhou⁴ Charles Chiang⁵

¹ASIC & System State-key Laboratory, Microelectronics Department, Fudan University, Shanghai, China, xzeng@fudan.edu.cn, lhfeng@fudan.edu.cn.

²Mathemaitics Department, Fudan University, Shanghai, China, yfsu@fudan.edu.cn.

³Department of Mathematics, University of North Carolina at Charlotte, Charlotte, NC 28269 USA, wcai@uncc.edu.

⁴Departemnt of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75083, USA, zhoud@utdallas.edu.

⁵Synopsys Inc., Mountain View, CA 94043, USA, Charles.Chiang@synopsys.com.

Abstract

In this paper, a wavelet based approach is proposed for the model order reduction of linear circuits in time domain. Compared with Chebyshev reduction method, the wavelet reduction approach can achieve smaller reduced order circuits with very high accuracy, especially for those circuits with strong singularities. Furthermore, to compute the basis function coefficient vectors, a fast Sylvester equation solver is proposed, which works more than one or two orders faster than the vector equation solver employed by Chebyshev reduction method. The proposed wavelet method is also compared with the frequency domain model reduction method, which may loose accuracy in time domain. Both theoretical analysis and experiment results have demonstrated the high speed and high accuracy of the proposed method.

1. Introduction

As ULSI technology steps into the nanometer era, the interconnect networks have dominated the performance of the whole systems. During the last decade, model order reduction (MOR) techniques have become the main stream approaches for fast simulation of interconnect networks with huge dimensions. Model order reduction can be performed either in frequency domain or in time domain. For frequency domain approaches, the most classical order reduction methods are AWE, PVL and some of the Krylov subspace based methods [1,2], which try to get a reduced

order system by approximating a certain number of moments of the original transfer function. However by these methods, the accuracy of the output response of the reduced circuits in time domain cannot always be guaranteed even if the reduced transfer function can be very accurate in frequency domain.

As operating frequency is continuously increasing, signal integrity phenomenon due to magnetic coupling effects of interconnects, makes the impulse response of interconnects very complicated in waveform [8]. Especially, the fast changing nature of the signal waveforms poses challenges to obtain the accurate time-domain response from the reduced model which based on the accurate frequency-domain response. To tackle with this problem, it is more desirable to do model-order reduction directly in the time domain. The time domain reduction methods such as Chebyshev [8] and orthonormal basis projection [8] directly approximate the state variables in time domain by proper basis functions. By computing the coefficients of the basis functions, a projection matrix is derived by orthonormalization of the coefficient vectors and a reduced model thus can be obtained by projection of the original circuit. It is shown in [8] that Chebyshev reduction method in time domain is very efficient and more accurate than the frequency domain reduction method [2] when the original system is reduced to the same order.

The accuracy and efficiency of time domain model order reduction strongly depends on the approximation capability of the basis functions. As mentioned above, the fast changing waveform in high speed interconnects indicates the singularity of the circuit. To approximate the waveforms with strong singularities, the local support basis functions such as wavelets will be more efficient than the global support functions like Chebyshev functions. Motivated by this idea, we propose to use wavelet functions to perform model order reduction in time domain.

This research is supported by NSF grants CCR-0306298, partly by NSFC research project 90307017, 90207002, partly by the National Basic Research Program of China under the grant 2005CB321701, partly by Synopsys Inc., partly by Cross-Century Outstanding Scholar's fund of Ministry of Education of China, partly by National 863 plan projects 2004AA1Z105, Shanghai Science and Technology committee Key project 04JC14015 and Shanghai AM R&D fund 0418, Shanghai

The wavelet collocation method has been applied in transient simulation of linear and nonlinear circuits in [3,4,6]. Due to the local support and multi-resolution property of wavelets, there exists powerful adaptive scheme to automatically select higher-level wavelets around singularities to meet the required accuracy. Moreover the wavelets in [3,4] have h^4 convergence rate which can use much less number of wavelet basis functions to meet the given reduction accuracy requirement. Therefore, wavelet reduction method is capable of reducing the circuit into much smaller size than the Chebyshev approach especially when dealing with circuits with strong singularity properties.

Another problem encountered in the existing time domain model order reduction of large-scale circuit is the huge amount of computations involved in coefficient calculation, if the number of the coefficients is not small. For example, in Chebyshev projection order reduction method, a number of $N \times (K+1)$ Chebyshev coefficients are calculated by solving a vector equation with the size of $N \times (K+1)$ by $N \times (K+1)$, where N is the number of state variables and each state variable has a number of $K+1$ coefficients. Solving such a vector equation will be very time consuming. Therefore Chebyshev order reduction will lose efficiency when applied for very large scale circuits.

In this research, we find that the coefficient equations no matter employed in wavelet approach or in Chebyshev method can be formulated in a matrix equation called Sylvester equation. There exist fast Sylvester equation solvers like complete Schur decomposition [5,6] and Hessenberg Schur decomposition [7], which have proven to be much faster than the vector solver. In this paper, by making use of the matrix sparsity of real circuit equations derived from RLC models of interconnect network, we further propose a Partial Schur decomposition algorithm which is even faster than the traditional algorithms in [5,6]. Therefore order reduction in time domain can be achieved with very cheap computation.

In section 2, we first briefly introduce the time domain Fast Wavelet Collocation Method [3,4], then propose the new wavelet order reduction approach. In section 3, we analyze the limitations of the direct vector equation solver and the complete Schur algorithm in [5,6]. Then we propose a fast Partial Schur algorithm to efficiently solve the function coefficients. In section 4, numerical experiment results of the fast Partial Schur algorithm and wavelet order reduction with comparison of Chebyshev reduction method and frequency domain reduction method are all presented. Finally we draw conclusions.

2. Order Reduction by Wavelets

2.1 FWCM in time domain

Generally a linear circuit system can be described by

$$\frac{dx}{dt} = Ax + Bu(t) \quad (2.1)$$

$$y(t) = C^T x(t) \quad (2.2)$$

where $x(t)=[x_1(t), x_2(t) \dots x_N(t)]^T$ is the unknown N dimensional state vector and $u(t)$ is the input function. $y(t)$ is the unknown output. A, B, C are the system matrices.

In FWCM [3, 4], the simulation interval $[0, T]$ in time domain is first mapped to the wavelet interval $[0, L]$ by conversion of $t = T \times l/L$, system (2.1) is then transformed to (2.3), where $\hat{x}(l) = x(l \times T/L)$, $\hat{u}(l) = u(l \times T/L)$, $\hat{A} = A \times T/L$, $\hat{B} = B \times T/L$.

$$\frac{d\hat{x}(l)}{dl} = \hat{A}\hat{x}(l) + \hat{B}\hat{u}(l) \quad (2.3)$$

In order to find the solution of the state variables $\hat{x}(l)$, for a given wavelet order $J \geq 0$, we expand the state variables $\hat{x}(l)$ by (2.4),

$$\hat{x}(l) = \begin{bmatrix} \hat{x}_1(l) \\ \hat{x}_2(l) \\ \vdots \\ \hat{x}_N(l) \end{bmatrix} = \begin{bmatrix} h_1^1 & h_2^1 & \dots & h_M^1 \\ h_1^2 & h_2^2 & \dots & h_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1^N & h_2^N & \dots & h_M^N \end{bmatrix} \bullet \begin{bmatrix} \theta_1(l) \\ \theta_2(l) \\ \vdots \\ \theta_M(l) \end{bmatrix} \quad (2.4)$$

$$= [h_1 \ h_2 \ \dots \ h_M] \theta(l) = H \theta(l)$$

where H is the $N \times M$ unknown coefficients matrix, $\theta(l) = [\theta_1(l), \theta_2(l), \dots, \theta_M(l)]^T$ represents the wavelet basis functions, $\hat{x}_i(l) = x_i(T \times l/L)$ ($i=1, 2, \dots, N$) and M is the total number of the basis functions employed for a given order J , $M = 2^J \times L + 3$. By (2.4), if matrix H is solved, the solutions of each state variable are also obtained. To solve H , substituting $\hat{x}(l)$ in equation (2.3) with equation (2.4), we get equation (2.5) as below

$$H \frac{d\theta(l)}{dl} = \hat{A}H\theta(l) + \hat{B}\hat{u}(l) \quad (2.5)$$

Discretizing equation (2.5) with a number of M collocation points [3,4], we derive equation (2.6).

$$H \frac{d[\theta(l_1) \ \theta(l_2) \ \dots \ \theta(l_M)]}{dl} = \hat{A}H[\theta(l_1) \ \dots \ \theta(l_M)] + \hat{B}[\hat{u}(l_1) \ \dots \ \hat{u}(l_M)] \quad (2.6)$$

where l_i ($i=1, 2, \dots, M$) are the M collocation points. The coefficient matrix H is thus obtained by solving linear equation (2.6).

One of the main advantages of the wavelet collocation method is that there exists an adaptive scheme for the choice of wavelet basis function in (2.4) based on the multi-resolution theory. Using adaptive techniques, those wavelet basis functions, which are needed for approximating state variables with high accuracy, can be employed automatically. The detailed adaptive scheme is explained in [4], we here give a simple introduction. The B-spline wavelets [4] consist of a closed subspace which belongs to the second order integrable Sobolev subspace. The approximation accuracy depends on the wavelet space level. The higher the space level is, the less the error will be. The magnitudes of the wavelet coefficients will indicate whether a refinement, by increasing the wavelet space level, is needed or not. For example, denote h_{ij} the coefficient of the wavelet function F_{ij} in the j -th level

subspace W_j . For a given error bound of ε , if $|h_{ij}| > \varepsilon$, it means that in the region covered by F_{ij} , the approximation is not accurate enough and wavelet space level needs to be increased from j to $j+1$. Otherwise, it means that the current subspace is accurate enough to approximate the original system in the region covered by F_{ij} . So we don't need to increase j any more.

2.2 Order reduction by wavelets

From Equation (2.4), at each point l_i of the M collocation points $\{l_1, l_2, \dots, l_M\}$, vector $\hat{x}(l_i)$ can be approximately represented by the columns $\{h_1, h_2, h_3, \dots, h_M\}$ in matrix H .

$$\hat{x}(l_i) = [h_1 \ h_2 \ \dots \ h_M] \theta(l_i) \quad (2.7)$$

Equation (2.7) means that the vector $\hat{x}(l_i)$ is in the subspace spanned by $\{h_1, h_2, h_3, \dots, h_M\}$. For large scale circuits, $\hat{x}(l_i)$ can be approximated accurately only by a small number of wavelet basis functions, i.e. $M \ll N$, due to the h^4 high convergence rate of wavelets [3]. Therefore, if $M \ll N$, we can do order reduction on the original system (1) by using the subspace spanned by the columns of H . The detail is as follows.

Firstly the columns $\{h_1, h_2, h_3, \dots, h_M\}$ are orthonormalized to get an orthonormal matrix $V = [V_1, V_2, V_3, \dots, V_q]$, $q \leq M$. Then using V , we make projection $x(t) \approx Vz(t)$, $\tilde{B} = V^T B$ and do congruence transformation $\tilde{A} = V^T A V$, a reduced system of dimension $q \ll N$ in (2.8) can be obtained from the original system with order N in (2.1) and (2.2).

$$\begin{aligned} \frac{dz(t)}{dt} &= \tilde{A}z(t) + \tilde{B}u(t) \\ y &= C^T Vz(t) \end{aligned} \quad (2.8)$$

3. Fast Sylvester Equation Solver

3.1 Proposed partial Schur method

From the above analysis, we see that the key step for wavelet order reduction is how to calculate matrix H in (2.6) efficiently.

Denote $\Phi = [\theta(l_1) \ \theta(l_2) \ \dots \ \theta(l_M)]$, $U = [\hat{u}(l_1) \hat{u}(l_2) \ \dots \ \hat{u}(l_M)]$, $Z = \frac{d\Phi}{dl} \Phi^{-1}$ and $G = \hat{B}U\Phi^{-1}$, we finally derive the

Sylvester equation in (3.1) from equation (2.6),

$$HZ - \hat{A}H = G \quad (3.1)$$

where $Z \in R^{M \times M}$, $\hat{A} \in R^{N \times N}$, $H \in R^{N \times M}$, $G \in R^{N \times M}$.

The Sylvester equation in (3.1) is actually a matrix equation where the unknown variables are represented by matrix H . A direct approach for solving (3.1) is to transfer the matrix equation into a vector equation (3.2) by reshaping H into a $MN \times 1$ vector $\tilde{h} = [h_1^T \ h_2^T \ \dots \ h_M^T]^T$,

$$Q\tilde{h} = r \quad (3.2)$$

Vector equation (3.2) can be solved by either direct LU factorization or iteration methods like GMRES. Unfortunately, (3.2) is too large to be solved efficiently, as $\tilde{h} \in R^{MN}$. For example, the operation count of Gaussian elimination or LU factorization is $O((M \times N)^3)$. This could be prohibitive as N or M increases.

More efficient method to solve (3.2) is employing matrix equation solver, for instance the complete Schur decomposition algorithm proposed in [5, 6], which first decomposes the two matrices.

$$\bar{A} = P^T \cdot \hat{A} \cdot P \quad (3.3)$$

$$\bar{Z} = Q^T \cdot Z \cdot Q \quad (3.4)$$

Both P and Q are orthogonal and \bar{A} , \bar{Z} are quasi-triangular matrices with possible nonzero 2×2 blocks along the diagonal only if \hat{A} or Z has complex eigenvalues. Substituting (3.3) (3.4) into (3.1), and denoting $\bar{B} = P^T G Q$, $\bar{H} = P^T H Q$, we get:

$$\bar{H}\bar{Z} - \bar{A}\bar{H} = \bar{B} \quad (3.5)$$

As \bar{A} and \bar{Z} are quasi-triangular, \bar{H} can be computed from the upper-left corner to the bottom-right corner successively, solving no more than four elements each time. \bar{H} could easily be obtained from \bar{H} , that is $H = P\bar{H}Q^T$.

The total computation account of Complete Schur method [5] is $10(N^3 + M^3) + 5/2(N^2M + M^2M)$. Obviously, the complete Schur method takes less work than the director vector equation solver for (3.2). However, as the circuit scale grows, the Schur decomposition will become very time consuming. When $N \gg M$, work count of the complete Schur method can be approximated as $10N^3$, which increases quite fast with N . In section 4.1, experimental results prove the above analysis.

In the following, we propose a much faster algorithm for solving (3.1). Since the number of wavelet basis functions M is usually much less than the scale N of the original system, then Schur decomposition (3.4) is ready to be obtained, and the Schur decomposition of \hat{A} in (3.3) is the one really time consuming. However, for practical interconnect circuits, A is usually quite sparse, and the same is true for \hat{A} , because \hat{A} is only a scaling of A . As we know, sparsity can often be utilized to achieve great savings of computation, but it is spoiled in the above complete Schur decomposition method by the decomposition of \hat{A} in (3.3), because \bar{A} as a result is usually a dense matrix.

In order to make use of sparse of \hat{A} , we do not decompose \hat{A} , and only make a decomposition of Z by (3.4). Furthermore if \bar{Z} is triangular rather than quasi-triangular, \bar{H} could be solved column by column (or row by row). A triangular decomposition of Z can be easily obtained by the complex Schur decomposition listed in the following algorithm.

Step 1. Complex Schur decomposition for Z , while leaving \hat{A} intact, i.e. $\tilde{Z} = \bar{U}^T Z U$, \tilde{Z} is a complex upper triangular matrix whose elements are complex numbers i.e. $\tilde{Z}_{ij} \in C$,

U is an orthogonal matrix, \bar{U} means the conjugate matrix of U .

Step 2. Substitute this decomposition into (3.1), denote $\tilde{B} = GU, \tilde{H} = HU$, we have $\tilde{H}\tilde{Z} - \hat{A}\tilde{H} = \tilde{B}$.

Step 3. Expand both sides of the above equation, we obtain,

$\sum_{j=1}^M \tilde{Z}_{jk} \tilde{h}_j - \hat{A} \tilde{h}_k = \tilde{b}_k$, for $k=1,2,\dots,M$. where \tilde{h}_j, \tilde{b}_k are the columns in \tilde{H}, \tilde{B} . Because \tilde{Z} is upper triangular, $\tilde{h}_1, \tilde{h}_2, \dots, \tilde{h}_M$ can be solved successively by

$$(\tilde{Z}_{kk} I_N - \hat{A}) \tilde{h}_k = \tilde{b}_k - \sum_{j=1}^{k-1} \tilde{Z}_{jk} \tilde{h}_j \quad (3.6)$$

Step 4. Get H from $H = \text{real}(\tilde{H}\bar{V}^T)$.

The function *real* (x) that extracts the real part of x is necessary, as round-off errors during operations might introduce a very small complex part in $\tilde{H}\bar{V}^T$, which should be neglected because all the known matrices in (3.1) are real. Note that the coefficient matrix $(\tilde{Z}_{kk} I_N - \hat{A})$ in (3.6) is nearly as sparse as \hat{A} . Equation (3.6) can be solved by many fast algorithms dealing with large-scale sparse matrix problems such as preconditioned *GMRES*. We call the proposed algorithm the Partial Schur decomposition algorithm since only one Schur decomposition is done.

For partial Schur decomposition method, Schur decomposition is done only on the small matrix $Z \in R^{M \times M}$, the computation account in step 1 is only $O(M^3)$, which is much less than the complete Schur decomposition in [5, 6].

3. 2 Comparison with Chebyshev order reduction method

Both wavelet order reduction method and Chebyshev projection order reduction method [8] are based on the same principle of approximating the state variable in time domain and making use of function coefficient vectors to construct the projection space. However, the wavelet reduction method outperforms the Chebyshev approach in the following aspects.

Firstly, let's compare the computation cost of the two methods. In Chebyshev order reduction, to calculate the Chebyshev coefficients, the original system was discretized into equation (4.35) in paper [8] at a number of K time points $\bar{t}_i (0 \leq i < K)$, which is rewritten in the following.

$$\tilde{M} \sum_{k=0}^K d_{ik} \tilde{x}(\bar{t}_k) + \tilde{N} \tilde{x}(\bar{t}_i) = 0 \quad (3.7)$$

where $\tilde{x}(\bar{t})$ is transformation of state variables $x(t)$ in the domain of Chebyshev functions. $\tilde{x}(\bar{t}_i)$ is the value of $\tilde{x}(\bar{t}_i)$ at time point $\bar{t}_i (0 \leq i < K)$. \tilde{M}, \tilde{N} are system matrices and d_{ik} is the k -th expansion coefficient of the derivative of $\tilde{x}(\bar{t})$ at time point \bar{t}_i . In order to solve $\tilde{x}(\bar{t}_i)$, equation (3.7) was formulated as a vector equation in equation (4.36) in [8] with dimension of $N \times (K+1)$ by $N \times (K+1)$. As illustrated in Section 3.1, solving large scale vector

equation will be very time consuming. Fortunately, we find that by following the same procedure as we use to develop wavelet coefficient equation in (2.6), we can reformulate (3.7) into a Sylvester equation in (3.8), where

$$D = [d_0, \dots, d_i, \dots, d_K], d_i = [d_{i0}, \dots, d_{i1}, \dots, d_{iK}]^T \cdot$$

$$[\tilde{x}(\bar{t}_1), \tilde{x}(\bar{t}_2), \dots, \tilde{x}(\bar{t}_K)] D + \tilde{M}^{-1} \tilde{N} [\tilde{x}(\bar{t}_1), \tilde{x}(\bar{t}_2), \dots, \tilde{x}(\bar{t}_K)] = 0 \quad (3.8)$$

We can see that the function coefficient equations (2.6) in wavelet method or (3.8) in Chebyshev method can all be formulated as a Sylvester equation and be solved by the proposed Partial Schur Decomposition method. However, in Chebyshev method, the coefficients are computed by expensive vector equation solver.

Secondly, we compare the accuracy of the two order reduction methods. The key feature of the wavelet-balance method is the fact that wavelet basis has local support in time domain, whereas the Chebyshev function has a global support. Hence, the wavelet functions will be more powerful than Chebyshev functions to approximate the fast changing waveforms. Especially the wavelet bases employed here have a fourth-order convergence rate [4], resulting in low computational complexity. Furthermore, by making use of multi-resolution analysis in wavelet theory [4], an adaptive technique exists for automatically selecting proper wavelet basis functions needed at a given accuracy. Consequently, high order wavelet basis functions are only employed near fast changing parts of the waveform. However, there is no such an elegant adaptive scheme for choosing Chebyshev basis. Moreover, because of their global support, Chebyshev basis functions will lose efficiency to capture the fast changing waveforms, as demonstrated in experiment results. Actually, the response of high speed interconnect circuits with magnetic coupling will exhibit localized singularities, which can be easily captured by the wavelet approach through adaptive Scheme.

4. Numerical Experiments

In this section, we will examine the proposed wavelet model order reduction method by testing a clock tree circuit [6] and a 4-bit bus line circuit with 16 RLC segments. A step signal with 2ps rise time is applied as input. The simulation time interval is set as 0~1ns. We first demonstrate the efficiency of the Partial Schur algorithm for coefficient calculation by comparison with the traditional vector equation algorithm and complete Schur algorithm. Then we compare the wavelet order reduction with Chebyshev reduction and frequency domain order reduction.

4.1 Comparison between partial Schur algorithm and traditional algorithms

By testing clock tree circuits with different state variable number, we obtain in Tab. 1 the comparison results of the three Sylvester equation solvers. Simulation error of each algorithm is described by $\|y - \bar{y}\|_2 / \|y\|_2$, where y is the solution of (3.1) by the function in Matlab, which we assume as the exact solution. \bar{y} is the solution

of each of the three algorithms concerned. The tests are run on a PIV 1.7G PC with 256M RAM.

Numerical results in Tab.1 show that solving H by vector equation method (V.E.) for circuit with 56 state variables will cost more than 1000 seconds and it is impossible to obtain the results for the other large dimension circuits due to memory limitation. This confirms that solving vector equation is definitely time and memory consuming and cannot be applied to large circuits. For all of the testing examples, both partial Schur (P.S.) and complete Schur (C.S.) method in [5,6] can obtain very accurate results. Under the same simulation accuracy, the partial Schur method is almost one order faster than the complete Schur method. Moreover, as the circuit scale increases, the simulation time of the complete Schur method will increase in an ultra-linear way, while the simulation time of partial Schur increases almost linearly with the circuit size.

Tab. 1: Comparison of different Sylvester equation solvers.

State variable number	Method	Time(s)	Error
56	V. E.	>1e+003	----
	C. S.	2.156000e+000	2.27e-005
	P. S.	1.812000e+000	2.27e-005
504	V. E.	Out of Memory	----
	C. S.	1.515700e+001	5.65e-005
	P. S.	3.968000e+000	5.65e-005
2040	V. E.	Out of Memory	----
	C. S.	7.695000e+002	1.10e-004
	P. S.	1.279700e+001	1.10e-004

4.2 Comparison with Chebyshev order reduction and frequency domain order reduction

In this subsection, we compare wavelet order reduction method with the Chebyshev order reduction method. The error measurement is the mean square error defined by $\|y_r - y\|_2 / \|y\|_2$, where y is the exact output response of the original system (2.1) and y_r is the output response of the reduced system, as listed in Tab.2 and Tab.3.

The first example has a slow changing waveform (Fig. 1) at the output of a 120-order clock tree circuit. The waveforms of the 33th wavelet reduced model and the 30th Chebyshev reduced model are so close to each other and cannot be distinguished from the exact solution.

More comparison results of different size clock tree circuits are listed in Tab.2. We can see that when the size of the circuit is small, both the Chebyshev method and the wavelet method are accurate. However, when the circuit size goes up to 1016 and beyond, Chebyshev method cannot obtain the reduced model because the vector equation for the coefficients is too large to be solved. On the contrary, the wavelet method can treat circuits with size up to 24556 within several minutes with very high accuracy.

This experiment shows that for slow changing waveform, both the two reduction methods work well with

moderate size circuits. When the circuit size becomes large, Chebyshev reduction method will fail, whereas wavelet method still keeps enough efficiency and accuracy.

The second example has a fast changing waveform (the solid line in Fig.2) at the output of a 147th order coupling bus line circuit. For this fast changing circuit, Chebyshev method reach a 10th order reduced model with 16.18% error after using $k=30$ basis functions. This error has little change when the number of basis functions is increased to $k=160$. The corresponding results have been plotted respectively in dotted and dashed lines in Fig.2. This example shows that even by increasing the number of Chebyshev basis functions, the waveform computed from the reduced model still fails to catch the fast changing part of the waveform. The dashed line with triangles describes the result by wavelet order reduction. The error of the 13th reduced model is below 10^{-7} . This result demonstrates the superiority of wavelet method for model order reduction of circuit with strong singularities.

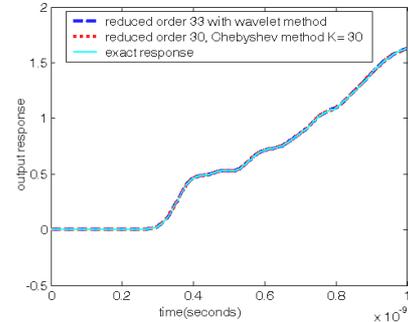


Fig.1: Comparison between wavelet method and Chebyshev method for the clock tree example.

For the second circuit example, Fig.3 depicts the distribution of the wavelets over the whole simulation interval from which we can see the local support feature of the wavelet. It is very interesting that more number of wavelets and higher order wavelets are employed to approximate the left part of the waveform which changes faster than the right part, where smaller number of wavelet and lower order wavelets are chosen. This example exactly demonstrates the capability of adaptive scheme to automatically choose wavelets to meet the required accuracy. It also shows that wavelet basis functions can easily catch the fast changing part of the waveform because of their local support.

Tab. 2: Comparison between wavelet method and Chebyshev method.

Original order	Wavelet order	Cheby. order	Wavelet Error	Cheby. Error
120	33	30	1.07e-14	3.57e-9
504	47	49	1.32e-7	4.65e-3
1016	56	-----	1.87e-7	-----
6134	54	-----	9.60e-4	-----
12278	58	-----	1.25e-2	-----
24556	61	-----	2.91e-2	-----

In the following, we compare the wavelet order reduction method with the frequency domain order reduction method PRIMA [2] by testing the clock tree circuits in [6]. The listed simulation results in Tab.3 represent the best reduction results of PRIMA, which means that the reduction errors can not be further reduced even if the order of the reduced system is further increased.

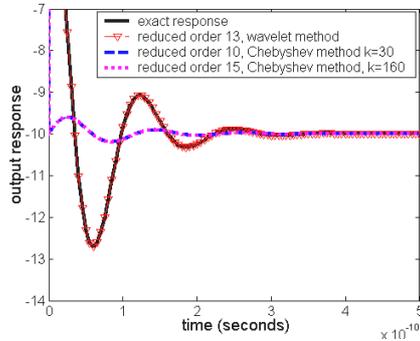


Fig. 2: Comparison between Wavelet method and Chebyshev method for the 4-bit bus line circuit example.

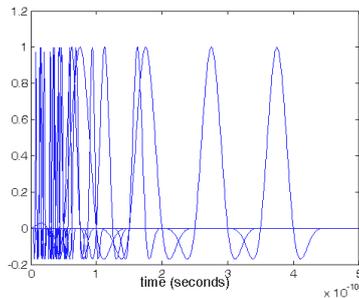


Fig. 3: Distribution of wavelets.

Tab. 3: Comparison between Wavelet method and frequency domain method PRIMA

Original order	Reduced order	Error of PRIMA	Error of Wavelet MOR
504	47	0.0039	1.2568e-007
2040	60	0.0090	4.3939e-006
6134	54	0.2487	9.6041e-004
12278	58	0.1565	0.0125

It shows that frequency domain method PRIMA results in very large error for time domain response. For instance, the maximum error of PRIMA can get to around 25%, which is not acceptable for timing analysis and circuit simulation. We notice that this large error comes from the problem of choosing proper expansion points for transfer function, which is still a relatively open problem and is not completely solved [1]. However, the proposed wavelet model order reduction depends on strong mathematic theory such as adaptive algorithm to automatically choose the wavelet functions to meet the reduction error requirement. We also notice that the wavelet method may

cost more computation time than the frequency domain method during the model reduction process. However this extra cost is deserved if accurate time domain circuit response is more desired than frequency domain response. Furthermore, the reduced order system by wavelet method is rather small, which can greatly save the simulation time after the reduced model is embedded in the whole circuit simulation.

5. Conclusions

We present an order reduction approach using wavelet basis function to approximate the state variables in time domain. By orthogonalizing the wavelet coefficient vectors, a projection space can be constructed and thus applied to project the original system to a lower order system. To fast calculate the wavelet coefficients, a Partial Schur decomposition algorithm is proposed to directly solve the Sylvester equation rather than solve the vector equation. The local support property of wavelets makes them capable of approximating fast changing waveforms very well. This is the main advantage of wavelets over Chebyshev functions when dealing with fast changing waveforms. Detailed theoretical analysis and experimental comparison with Chebyshev order reduction method and frequency domain order reduction method demonstrate that the time domain wavelet order reduction method is very efficient and accurate for time domain model order reduction, especially when dealing with very large scale interconnect circuits with singularities.

6. References

- [1] Roland W. Freund, Reduced-Order Modeling Techniques Based on Krylov Subspaces and Their Use in Circuit Simulation, Numerical Analysis Manuscript No. 98-3-02, Bell Laboratories, Murray Hill, New Jersey, Feb. 1998.
- [2] A. Odabasioglu, M.Celik, L. T. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm", in Tech. Dig. 1997 *IEEE/ACM ICCAD*, Nov. 1997..
- [3] D. Zhou and W. Cai, "A Fast Wavelet Collocation Method for High-Speed Circuit Simulation", *IEEE Trans. Circuits and Systems*, vol. 46, no. 8, pp. 920-930, Aug. 1999.
- [4] D. Zhou, W. Cai and W. Zhang "An Adaptive Wavelet Method For Nonlinear Circuit Simulation", *IEEE Trans. Circuits and Systems*, vol. 46, no. 8, pp. 931-938, Aug. 1999.
- [5] R. H. Bartels and G. W. Stewart, "Solution of the Matrix Equation $AX+XB=C$ ", *Commun. ACM*, Vol. 15, pp. 820-826, 1972.
- [6] Xuan Zeng, Sheng Huang, Yangfeng Su and Dian Zhou "An Efficient Sylvester Equation Solver for Time Domain Circuit Simulation By Wavelet Collocation Method", *IEEE ISCAS*, vol. III, pp. 168-171, May, 2003.
- [7] G. H. Golub, S. Nash and C. Van Loan, "A Hessenberg-Schur Method for the Problem $AX+XB=C$ ", *IEEE Tran. Automatic Control*, Vol. Ac-24, No.6, Dec. 1979.
- [8] Janet Meiling Wang, Chia-Chi Chu, Qingjian Yu and Ernest S. Kuh, "On Projection-Based Algorithms for Model-Order Reduction of Interconnects", *IEEE Tran on Circuit and Systems-I*, Vol.49, No. 11, Nov. 2002.