Title	Efficient matrix exponential method based on extended Krylov subspace for transient simulation of large-scale linear circuits
Author(s)	Chen, Q; Zhao, W; Wong, N
Citation	The 19th Asia and South Pacific Design Automation Conference (ASP-DAC 2014), Suntec, Singapore, 20-23 January 2014. In Conference Proceedings, 2014, p. 262-266
Issued Date	2014
URL	http://hdl.handle.net/10722/201220
Rights	©2014 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Efficient Matrix Exponential Method Based on Extended Krylov Subspace for Transient Simulation of Large-Scale Linear Circuits

Quan Chen, Wenhui Zhao and Ngai Wong

Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong
Emails: quanchen@eee.hku.hk, whzhao@eee.hku.hk, nwong@eee.hku.hk

Abstract— Matrix exponential (MEXP) method has been demonstrated to be a competitive candidate for transient simulation of very large-scale integrated circuits. Nevertheless, the performance of MEXP based on ordinary Krylov subspace is unsatisfactory for stiff circuits, wherein the underlying Arnoldi process tends to oversample the high magnitude part of the system spectrum while undersampling the low magnitude part that is important to the final accuracy. In this work we explore the use of extended Krylov subspace to generate more accurate and efficient approximation for MEXP. We also develop a formulation that allows unequal positive and negative dimensions in the generated Krylov subspace for better performance. Numerical results demonstrate the efficacy of the proposed method.

I. Introduction

Accurate yet fast transient simulation capability of large-scale integrated circuits has long been a challenge in electronic design automation. The work-horse approach so far is the direct method utilizing one-off sparse matrix factorization plus forward-backward substitution at each time step, adopted in most of the mainstream circuit simulators such as SPICE. Thanks to the recent developments in sparse matrix factorization technique, such as KLU [2], the direct methods have been demonstrated to be able to handle some millions-scale circuits.

However, there are still several shortcomings associated with the direct methods. First, the existing direct method is established on top of the traditional linear multi-step (LMS) schemes (trapezoidal, Gear's method, etc.) based on truncated polynomial expansion [7]. The allowable step size is usually small due to the limitation from the local truncation error (LTE). Second, the time and memory storage required in matrix factorization is highly sensitive to the sparsity pattern of the matrix; when parasitic coupling is included, i.e., the nonzero bandwidth increases and sparsity pattern becomes irregular, dealing with the excessive fill-ins generated during the factorization process remains a challenging task for direct solvers. Parallelization of matrix decomposition, though possible, is also more complicated. Thirdly, adaptive time-stepping is not favored in direct methods since in general the system matrix has to be re-factorized once the step size is changed.

As an alternative to the direct methods, the matrix exponential method (MEXP) receives considerable attention in recent years [13–15]. The MEXP method differs from the traditional approaches in mainly two aspects: 1) it is based on the analytical solution of ordinary differential equation (ODE) using exponential operator (LMS methods can be viewed as polynomial approximation of the exponential operator); 2) the core computation is changed from matrix factorization to the approximation of the product of MEXP with a vector in the Krylov subspace.

$$\mathcal{K}_m = span\left\{v, Av, A^2v, ...A^{m-1}v\right\} \tag{1}$$

where m is the subspace dimension and v the starting vector.

The first difference implies that the MEXP method is exact if the matrix exponential is calculated exactly, whereas the LMS methods remain approximative even when the matrix inversion is exact. Therefore, the step size in MEXP depends only on the quality of the Krylov subspace approximation other than the LTE. In other words, the "order of accuracy" of MEXP can be made very high by increasing the dimension of the Krylov subspace, in contrast to that in the LMS context such order is fixed once a particular scheme is chosen. Therefore, the step size in MEXP can be much larger than that allowed in LMS schemes even with a moderate subspace dimension [13]. The second aspect renders the computational cost less sensitive to the sparsity pattern of the matrices and allows convenient parallelization. The shift-invariant property of the Krylov subspace also greatly facilitates the adaptive time-stepping, enabling further computational advantages compared against tradition methods.

Despite the above merits, the MEXP method becomes less efficient when dealing with stiff circuits which have time constants differing by orders of magnitude. From spectral analysis perspective, large magnitude, well-separated, eigenvalues will appear in the spectrum of the numerical systems arising from stiff circuits. The underlying Arnoldi process, when used to generate bases of Krylov subspace, tends to capture these dominant eigenvalues first to minimize interpolation error, while leaving the spectrum in the vicinity of zero undersampled. For passive systems, all eigenvalues are negative, thus the large

978-1-4799-2816-3/14/$31.00 ©2014 IEEE

(magnitude) eigenvalues actually have negligible contribution to the computation of MEXP. The undersampling of small (magnitude) eigenvalues, on the other hand, introduces large error in the results, and accounts for the inefficiency of MEXP method for stiff systems.

The difficulty in finding small eigenvalues by ordinary Krylov subspace is a known issue in the context of eigenvalue solvers and iterative methods [4]. A natural remedy is to use the shift-invert strategy to transform the small eigenvalues into the dominant ones so that they can be more easily captured. This motivates to use, instead of the ordinary Krylov subspace (1), the extended Krylov subspace

$$\mathcal{K}_{l,m} = span\left\{A^{-l+1}v, ...A^{-1}v, v, Av, ...A^{m-1}v\right\} \quad (2)$$

to generate the MEXP approximation. Using the extended Krylov subspace for evaluating matrix functions was first proposed in [3], which proved that, when A is symmetric, the approximation quality of the exponential function in $K^{2m}(A, v)$ is the same as in $K^{\sqrt{2m}}(A, A^{-m}v)$. In [3], the negative dimension l has to be prespecified, so the subspace is augmented only in the positive direction. Simoncini [10] later developed a more flexible scheme to add two vectors at a time, one multiplied by A and one by A^{-1}, into the basis set to gradually expand the subspace in both the negative and positive directions, i.e.

$$\mathcal{K}_{m,m} = span\left\{v, A^{-1}v, Av, ..., A^{-m+1}v, A^{m-1}v\right\}. \quad (3)$$

However, since the computation of $A^{-1}v$ is usually more expensive than the computation of Av in circuit simulation, an equal number of positive and negative dimensions may not always be the best strategy in terms of total runtime when it comes to individual problems.

In this paper, we apply the extended Krylov subspace concept to improve the performance of MEXP method in transient simulation of stiff circuits. In particular, we develop an approach to generate the extended Krylov subspace with unequal positive and negative dimensions. In this way, one can enjoy the benefit of the extended Krylov subspace while avoiding unnecessary $A^{-1}v$ for faster simulation. Numerical experiments are conducted to demonstrate the effectiveness of the proposed approach.

II. MEXP based on Extended Krylov Subspace

A. Formulation of MEXP Method

We first give a brief revisit to the MEXP method based on ordinary Krylov subspace method. For simplicity, we only consider linear circuits in this work. The numerical system to be solved in transient circuit analysis is a set of differential algebraic equations (DAE)

$$C\dot{x}(t) = Gx(t) + Bu(t). \quad (4)$$

where C, G and B denote the susceptance, conductance and input matrix, respectively, and $u(t)$ collects the voltage and current sources. The essence of MEXP lies in transforming (4) to an ODE

$$\dot{x}(t) = Ax(t) + b(t). \quad (5)$$

where $A = C^{-1}G \in \mathbb{R}^{N\times N}$ and $b(t) = C^{-1}Bu(t)$. Here we assume the C matrix is invertible. If C is singular, systematic regularization techniques [1] can be applied to produce a nonsingular system. With (5) and the common piece-wise linear (PWL) approximation of $u(t)$, the transient response of (4) can be analytically determined by

$$\begin{aligned} x(t+h) &= e^{Ah}x(t) \\ &+ (e^{Ah} - I)A^{-1}b(t) \\ &+ (e^{Ah} - (Ah + I))A^{-2}\frac{b(t+h) - b(t)}{h}, \end{aligned} \quad (6)$$

where h is the step size. With some transformation (see [12], eq. 20), the three MEXP functions in (6) can be merged into one MEXP with a slightly bigger matrix, i.e.,

$$\mathbf{x}(t+h) = e^{Ah}x(t), \quad (7)$$

in which all vectors/matrices have been augmented by 2 dimensions.

The main computation in each time step is the application of MEXP to a vector $e^{Ah}v$ in (7). It is computed by the Krylov subspace method [5, 9] using the Krylov subspace $\mathcal{K}_m(A, v)$ in (1) constructed by the Arnoldi process

$$AV_m = V_{m+1}\hat{T}_m \quad (8)$$

Here $V_m \in \mathbb{R}^{N\times m}$ is an orthonormal basis of $\mathcal{K}_m(A, v)$ and $\hat{T}_m \in \mathbb{R}^{(m+1)\times m}$ contains the orthonormalization coefficients. Then the MEXP-vec product is approximated in the computed subspace

$$e^{Ah}v \approx \beta V_m e^{T_m h}e_1, \quad (9)$$

where $\beta = \|v\|_2$, V_m, T_m is the $m \times m$ leading part of $\hat{T}_m$, and e_1 the 1-st column of identity matrix. A convenient posterior error estimate can be given for the approximation in (9) by [9]

$$err = \beta t_{m+1,m}\|e_m^{\mathbf{T}}e^{T_m h}e_1\|, \quad (10)$$

where $t_{m+1,m}$ is the bottom right element of $\hat{T}_m$.

The Arnoldi process itself is an iterative process. Since $Av = C^{-1}(Gv)$, in each iteration the main computation involves one sparse matrix-vector product and one sparse system solve (with C). In general, the cost per time step for MEXP is higher than that for LMS methods (except in the cases where C is much sparser than G). The benefit of using MEXP comes mainly from the possibility of using larger step sizes and thus fewer time steps for the same time interval than the low-order LMS methods. In addition, the ease in parallelization and adaptive time-stepping also add to the speedup over traditional methods, see, e.g., [13, 14].

A moderate Krylov subspace dimension m for (9) to converge, e.g., < 100, is desired for the performance of MEXP. In million-scale problems, m is also better kept small to control the memory storage of the basis vectors. The necessary m generally depends on the eigenvalues distribution of A. It is known that the Arnoldi process for

the ordinary Krylov subspace will prioritize the approximation of the eigenvalues that are large in magnitude and well-separated. In circuit analysis it implies the circuit contains many distinct fast modes, or is stiff (provided a slow mode also exist). If many such distinct outliers are present in the matrix spectrum, the Arnoldi process will spend most of its "dimension resources" building subspace close to the eigenspace associated with these eigenvalues. As a consequence, the part of spectrum in the neighborhood of zero is left inadequately sampled. Recall that the eigenvalues of A are with negative real parts, the exponential of these small (magnitude) eigenvalues are nonnegligible and thus the undersampling will induce substantial error in the approximation of $e^A v$. Then a larger subspace or a smaller step size (to compress the spectrum) is required to maintain the approximation accuracy, which significantly degrades the efficiency of MEXP.

B. Generalized Extended Krylov Subspace

The idea of using extended Krylov subspace is to generate the subspace containing the information of A^{-1} so that the MEXP approximation (9) can converge with fewer iterations. The technique was first proposed in [3], in which the number of negative dimensions l is fixed *a-priori* (only m is allowed to increase). Simoncini later developed an approach to increase the positive and negative dimensions of the subspace simultaneously by adding two basis vectors at a time [10], and showed that the extended Krylov subspace outperformed the standard Krylov subspace in a range of applications [6, 10, 11]. The technique is also known as the Krylov-plus-inverted-Krylov (KPIK) method [8]. In particular, it is shown that an Arnoldi-type relation analogous to (8) can be recovered from the orthonomalization coefficients without extra matrix-vector products [10]

$$AV_m = V_{m+2}\hat{T}_m, \tag{11}$$

where $\hat{T}_m \in \mathbb{R}^{(m+2)\times m}$ is a block Heisenbeig matrix.

Nevertheless, an equal number of positive and negative dimensions in KPIK may not always be the optimal choice. Since $A = C^{-1}G$, basis generation with Av requires a linear system solve with C, while that with $A^{-1}v$ requires one with G, which is usually denser than C in the MNA formulation. Thus expanding the subspace along negative dimension is more expensive, and the improvement in approximation quality as well as the increase in step size thereby allowed may not be able to pay off the extra cost resulted from the computation of $A^{-1}v$. Therefore, it is reasonable to allow an unequal number of dimensions in the extended Krylov subspace, i.e., we add one vector with A^{-1} after every k vectors with A.

$$\mathcal{K}_{m,km} = span\left\{ \begin{array}{l} v, A^1 v, A^2 v, ...A^k v, A^{-1}v, A^{k+1}v, ..., \\ A^{2k}v, A^{-2}v, ..., A^{km-1}v, A^{-m+1}v \end{array} \right\}. \tag{12}$$

Apparently KPIK is a special case of (12) for $k = 1$.

With the arrangement in (12), the recursive relation to obtain a new basis vector is given below

$$\begin{array}{l} \text{If } n = 1 \text{ or } \mod(n, k+1) = 2, ..., k \\ h_{n+1,n} v_{n+1} = A v_n - V_n h_{1:n,n}, \\ \text{If } \mod(n, k+1) = 0 \\ h_{n+2,n+1} v_{n+2} = A v_n - V_{n+1} h_{1:n+1,n+1}, \\ \text{If } n > k \text{ and } \mod(n, k+1) = 1 \\ h_{n,n-1} v_n = A^{-1} v_{n-k-1} - V_{n-1} h_{1:n-1,n-1}. \end{array} \tag{13}$$

where $h_{1:n+1,n}$ collects the $n+1$ orthonomalization coefficients against the previous basis vectors.

To obtain the relation in (11), we follow a similar strategy in [10] to recover $\hat{T}_m = V_{m+2}^{\mathbf{T}} A V_m$ without really performing the matrix-vector products.

Proposition II.1 *Let* $\hat{T}_m = (t_{i,j}), i = 1, ..., 2m+2, j = 1, ..., m$. *Then*

$$\begin{array}{l} \text{If } n = 1 \text{ or } \mod(n, k+1) = 2, ..., k-1 \\ t_{:,n} = h_{:,n} \\ \text{If } \mod(n, k+1) = 0 \\ t_{:,n} = h_{:,n+1} \\ \text{If } n > k \text{ and } \mod(n, k+1) = 1 \\ t_{:,n} = \frac{1}{h_{n,n-1}} \left(e_{n-k-1} - \left[\begin{array}{c} \hat{T}_{n-1} h_{1:n-1,n-1} \\ 0 \end{array} \right] \right) \end{array} \tag{14}$$

Proof Using the first equality of (13) and the orthogonality of V, for the first situations in (13), we have

$$\begin{aligned} t_{:,n} = V_{n+2}^T A v_n &= V_{n+2}^T v_{n+1} h_{n+1,n} + V_{n+2}^T V_n h_{1:n,n} \\ &= h_{1:n+1,n}. \end{aligned}$$

Similarly, using the second equality of (13), we obtain

$$\begin{aligned} t_{:,n} &= V_{n+2}^T A v_n \\ &= V_{n+2}^T v_{n+2} h_{n+2,n+1} + V_{n+2}^T V_{n+1} h_{1:n+1,n+1} \\ &= h_{1:n+2,n+1}. \end{aligned}$$

For the third situation, one can obtain from the last row of (13) that

$$h_{n,n-1} A v_n = v_{n-k-1} - A V_{n-1} h_{1:n-1,n-1}.$$

Then we can get

$$\begin{aligned} t_{:,n} &= V_{n+2}^T A v_n \\ &= \frac{1}{h_{n,n-1}} \left(V_{n+2}^T v_{n-k-1} - V_{n+2}^T A V_{n-1} h_{1:n-1,n-1} \right) \\ &= \frac{1}{h_{n,n-1}} \left(e_{n-k-1} - \left[\begin{array}{c} \hat{T}_{n-1} h_{1:n-1,n-1} \\ 0 \end{array} \right] \right), \end{aligned}$$

which completes the proof. ∎

Note that in (15) the current column in $\hat{T}$ cannot be filled until the next iteration is performed to generate $h_{:,n+1}$, so the evaluation of MEXP has also to be postponed to the next run.

A posterior error estimate can be given as

$$err = \beta \tau_{m+1,m} \| e_m^{\mathbf{T}} e^{T_m h} e_1 \|, \tag{15}$$

where τ is the 2×2 bottom right block of $\hat{T}_m$.

III. Numerical Results

In this section, we compare the performance of the MEXP transient solver based on the standard Krylov subspace and the generalized extended Krylov subspace. The implementation was done in Matlab and the testing was performed with a 3.2GHz server with 32Gb memory.

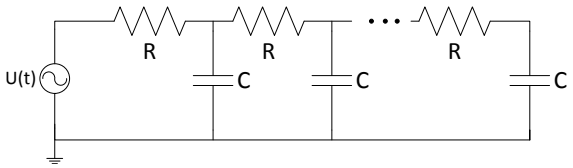

Fig. 1. RC ladder circuit

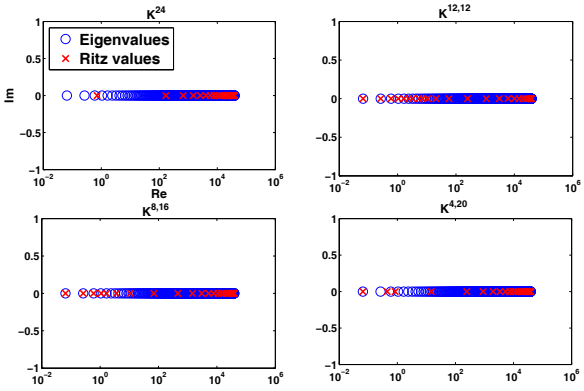

Fig. 2. Approximation of spectrum by standard Krylov subspace and extended Krylov subspace.

TABLE I
Error and runtime of different Krylov subspaces

Subspace	$\mathcal{K}_{24}$	$\mathcal{K}_{12,12}$	$\mathcal{K}_{8,16}$	$\mathcal{K}_{4,20}$
Error	4.2e-1	1.17e-6	3.6e-4	1.4e-2
Time (s)	0.09	0.37	0.17	0.11

We first use a simple RC ladder circuit (see the Fig. 1) to reveal the reason for the improvement led by the extended Krylov subspace. The order of the matrix is 1000, C is a diagonal matrix and G is simplified to the $[-1, 2, -1]$ pencil. The capacitor values are chosen uniformly distributed within $[1, 2] \times 10^{-15}F$ to produce a stiff circuit. We compute $e^{Ah}v$ by four Krylov subspaces with v being an all one vector and $h = 10^{-11}$. The dimension of the standard Krylov subspace is 24. The total dimensions of the three extended Krylov subspaces are also set to be 24, but with different negative-positive ratios $k = 1, 2, 5$, which correspond to the (negative, positive) dimensions of $(12, 12)$ (KPIK), $(8, 16)$ and $(4, 20)$. For notation convenience we denote the standard Krylov subspace as $k = 0$. Direct solution (backslash in Matlab) is applied in all sparse system solves. The eigenvalues of Ah and the Ritz values (the eigenvalues of $T_m h$) are plotted in Fig. 2. Note that the real parts are shown in log scale and the imaginary parts are in linear scale. The computation errors against the direct eigenvalue decomposition method and the runtime of the four cases are shown in Table I.

TABLE II
Specifications of test circuits

Circuits	Category	Nodes	Matrix size	Stiffness
C1	Power grid	39K	54K	medium
C2	Power grid	164K	165K	high
C3	Trans. lines	5.6K	8.8K	high

Due to the very small capacitance, many large (magnitude) eigenvalues ($\sim 10^4$) are present in the spectrum of A. The method based on the standard Krylov subspace generates a good approximation to the right-end part of the spectrum, but a poor one for the left-end spectrum. This is expected since the basis is generated using positive power of A only, which amplifies the dominant eigenvalues. Failure in sampling the eigenvalues near the origin, however, induces substantial error as shown in Table I. On the other hand, the extended Krylov subspace all improve the approximation to the near-origin spectrum by including the information of A^{-1}. Table I suggests that using more negative dimensions improves the accuracy since the meaningful parts of spectrum locates at the left end. However, the computation is also more costly when more negative dimensions are included due to requiring system solves with the denser matrix G.

Next we evaluate the performance of MEXP based on different Krylov subspace with real circuit examples. Three linear circuits specified in Table II are tested. For a better comparison among subspaces, we run 100 time steps with a constant step size in each case, and allow the subspace dimension to vary dynamically to satisfy a prescribed tolerance of 10^{-6} (per step) using the error estimate (15). The input is a ramp signal with the final amplitude of 1V.

The results are reported in Table III. Apparently, a large subspace dimension is required for MEXP based on the standard Krylov subspace to converge, which is highly undesirable in terms of runtime and memory. The solvers based on extended Krylov subspace all requires a significantly smaller subspace dimension to achieve the same level of accuracy. This shows the advantage of using the extended Krylov subspace in the simulation of stiff systems.

The equal assignment of positive and negative dimensions ($k = 1$) results in the lowest total dimension, which, however, does not necessarily leads to the smallest runtime due to the extra cost in generating basis involving A^{-1}. For the relatively less stiff case (C1), the performance is the best for $k = 4$, i.e., one negative dimension is inserted after every four positive dimensions. For $k = 1$ it is even slower than the standard Krylov subspace method, since Av takes only $0.02s$ while $A^{-1}v$ requires $0.14s$. For

TABLE III
PERFORMANCE OF MEXP BASED ON DIFFERENT KRYLOV SUBSPACE

Circuits	Step size (s)	Subspace dimensions				Total runtime (s)			
		k=0	1	3	4	k=0	1	3	4
C1	1e-12	(0,191)	(35,35)	(29,59)	(16,67)	591.8	641.5	595.1	**390.2**
C2	1e-11	(0,308)	(15,15)	(12,37)	(12,49)	5001.2	1459.1	**1002.7**	1256.3
C3	1e-12	(0,169)	(11,11)	(8,25)	(8,33))	6364.6	**1439.4**	1730.6	2014.7

the systems with stronger stiffness (C2 and C3), a larger portion of negative dimensions is desired. Therefore, the solver performs better with smaller values of k. The above experiments indicate that the best breakdown of positive and negative dimensions is generally problem dependent, and thus for maximum performance it is beneficial to allow a flexible basis generation along the two directions using the scheme developed in this work.

IV. Conclusion

We have investigated the use of extended Krylov subspace to enhance the accuracy of numerical approximation of MEXP-vector product, which in turn benefits the MEXP-based transient circuit simulation. The improvement is analyzed in terms of the quality of the approximation of the targeted part of matrix spectrum. In addition, we generalize the extended Krylov subspace to allow unequal positive/negative dimensions to maximize the overall performance in circuit simulation. Numerical results have confirmed the efficiency of the proposed method.

References

[1] Q. Chen, S.-H. Weng, and C.-K. Cheng, "A practical regularization technique for modified nodal analysis in large-scale time-domain circuit simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 1031–1040, 2012.

[2] T. A. Davis and E. Palamadai Natarajan, "Algorithm 907: KLU, a direct sparse solver for circuit simulation problems," *ACM Transactions on Mathematical Software*, vol. 37, no. 3, p. 36, 2010.

[3] V. Druskin and L. Knizhnerman, "Extended Krylov subspaces: Approximation of the matrix square root and related functions," *SIAM J. Matrix Anal. Appl.*, vol. 19, no. 3, pp. 755–771, Jul 1998.

[4] M. Freitag and A. Spence, "Shift-invert Arnoldi's method with preconditioned iterative solves," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 942–969, 2010.

[5] M. Hochbruck and C. Lubich, "On Krylov subspace approximations to the matrix exponential operator," *SIAM J. Numer. Anal.*, vol. 34, no. 5, pp. 1911–1925, Oct 1997.

[6] L. Knizhnerman and V. Simoncini, "A new investigation of the extended Krylov subspace method for matrix function evaluations," *Numerical Linear Algebra with Applications*, vol. 17, no. 4, pp. 615–638, 2010.

[7] F. Najm, *Circuit Simulation*. Wiley, 2010.

[8] M. Popolizio and V. Simoncini, "Acceleration techniques for approximating the matrix exponential operator," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 657–683, 2008.

[9] Y. Saad, "Analysis of some Krylov subspace approximations to the matrix exponential operator," *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 209–228, 1992.

[10] V. Simoncini, "A new iterative method for solving large-scale Lyapunov matrix equations," *SIAM J. Sci. Comput.*, vol. 29, no. 3, pp. 1268–1288, May 2007.

[11] ——, "Extended Krylov subspace for parameter dependent systems," *Appl. Numer. Math.*, vol. 60, no. 5, pp. 550–560, May 2010.

[12] S.-H. Weng, Q. Chen, and C.-K. Cheng, "Circuit simulation using matrix exponential method," in *2011 IEEE 9th International Conference on ASIC (ASICON)*, 2011, pp. 369–372.

[13] ——, "Time-domain analysis of large-scale circuits by matrix exponential method with adaptive control," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 8, pp. 1180–1193, 2012.

[14] S.-H. Weng, Q. Chen, N. Wong, and C.-K. Cheng, "Circuit simulation via matrix exponential method for stiffness handling and parallel processing," in *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2012, pp. 407–414.

[15] S.-H. Weng, P. Du, and C.-K. Cheng, "A fast and stable explicit integration method by matrix exponential operator for large scale circuit simulation," in *2011 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2011, pp. 1467–1470.