The Extended and Asymmetric Extended Krylov Subspace in Moment-Matching-Based Order Reduction of Large Circuit Models

Pavlos Stoikos, Dimitrios Garyfallou, George Floros, Nestor Evmorfopoulos, and George Stamoulis
Department of Electrical and Computer Engineering, University of Thessaly, Volos, Greece
{pastoikos, digaryfa, gefloros, nestevmo, georges}@e-ce.uth.gr

Abstract—The rapid growth of circuit complexity has rendered Model Order Reduction (MOR) a key enabler for the efficient simulation of large circuit models. MOR techniques based on moment-matching are well established due to their simplicity and computational performance in the reduction process. However, moment-matching methods based on the ordinary Krylov subspace are usually inadequate to accurately approximate the original circuit behavior, and at the same time do not produce reduced-order models as compact as needed. In this paper, we present a moment-matching method which utilizes the extended and the asymmetric extended Krylov subspace (EKS and AEKS), while it allows the parallel computation of the transfer function in order to deal with circuits that have many terminals. The proposed method can handle large-scale regular and singular circuits and generate accurate and efficient reduced-order models for circuit simulation. Experimental results on industrial IBM power grids demonstrate that the EKS method can achieve an error reduction up to 85.28% over a standard Krylov subspace method, while the AEKS method greatly reduces the runtime of EKS, introducing a negligible overhead in the reduction error.

Index Terms—Model Order Reduction, Moment-Matching, Krylov Methods, Circuit Simulation

I. INTRODUCTION

The ongoing miniaturization of modern IC devices has led to extremely complex circuits. This results in the increase of the problems associated with the analysis and simulation of their physical models. In particular, the performance and reliable operation of ICs are largely determined by several critical subsystems such as the power distribution network, multi-conductor interconnections, and the semiconductor substrate. The electrical models of the above subsystems are very large, consisting of hundreds of millions or billions of electrical elements (mostly resistors $R$, capacitors $C$, and inductors $L$), and their simulation is becoming a challenging numerical problem. Although their individual simulation is feasible, it is completely impossible to combine them and simulate the entire IC in many time-steps or frequencies. However, for the above subsystems it is often not necessary to fully simulate all internal state variables (node voltages and branch currents), as we only need to calculate the responses in the time or frequency domain for a small subset of output terminals (ports) and given excitations at some input ports. In these cases, the very large electrical model can be replaced by a much smaller model whose behavior at the input/output ports is similar to the behavior of the original model. This process is called Model Order Reduction (MOR).

MOR methods are divided into two main categories. System theoretic techniques, such as Balanced Truncation (BT) [1], provide very satisfactory and reliable bounds for the approximation error. However, BT techniques require the solution of Lyapunov matrix equations which are very computationally expensive, and also involve storage of dense matrices, even if the system matrices are sparse. On the other hand, moment-matching (MM) techniques [2] are well established due to their computational efficiency in producing reduced-order models. Their drawback is that the reduced-order model depends only on the quality of the Krylov subspace.

The majority of MM methods exploit the standard or the rational Krylov subspace in order to approximate the original model. Authors in [3], [4] employ rational Krylov MM methods to reduce power delivery networks. Using this projection subspace requires a heuristic and expensive parameter selection procedure, while the approximation quality is usually very sensitive to an inaccurate selection of these parameters. Moreover, in [2], [5] a standard Krylov subspace is employed for the reduction of regular and singular systems, respectively. Generally, established MM methods construct the subspace only for positive directions, usually leading to a large approximated subspace to obtain a satisfactory error. Recent developments in a wide range of applications have shown that the approximation quality of the Extended Krylov Subspace (EKS) outperforms the one of the standard Krylov subspace [6]. However, the application of EKS in the context of circuit simulation is not trivial. In several problems, EKS computation involves singular circuit models and dense matrix manipulations, which can hinder the applicability of this subspace.

In this paper, we introduce the EKS Moment-Matching (EKS-MM) and the alternate Asymmetric EKS-MM (AEKS-MM) methods that greatly decrease the error induced by MM methods by approximating both ends of the spectrum. Moreover, the proposed methods enable the parallel approximation of the transfer function, by splitting the original model with respect to each input port and employing the corresponding...
II. RELATED WORK

In this section, we briefly describe some previous works in the area of MOR techniques developed for circuit simulation problems. As mentioned before, mainly MOR methods have so far relied on MM and system theoretic techniques. MM methods for producing reduced-order models are well studied in the area of mathematics and easily applied in circuit simulation problems. Many circuit simulators apply the passive reduced-order interconnect macromodeling (PRIMA) method that is one of the most successful MM reduction algorithms, which preserves the passivity of the reduced model through a congruence transformation. Finally, in certain circuit simulation problems, the energy storage elements matrix might be singular. Applying a reduction process directly in such a model will lead to wrong results. Authors in [5] split the problematic part of the system using a set of projection matrices and then perform a MOR process through the PRIMA algorithm.

Moreover, many approaches try to produce reduced-order models in specific frequency ranges by applying rational MM methods using several expansion points. Authors in [9] propose a method for multi-point expansion to determine which poles are accurate and should be included in the final transfer function at each expansion point. Moreover, in [6], a guaranteed stable and parallel algorithm is proposed, which can select the frequency points in order to match the output at multiple frequencies with a reduced-order function. Similarly, in [9], the authors propose a method to obtain superior accuracy of the reduced-order model in the frequency range of interest, where the reduced models are calculated on different expansion points, implementing an adaptive multi-point version of the PRIMA algorithm exploiting the potential of modern multi-core processors. However, these type of methods require a huge number of candidate frequency selection points. Finally, authors in [10] and [11] focus on power delivery networks due to the large power grid sizes that require great computational resources. In the first approach, a multinode MOR technique to assist time domain simulations based on the superposition property is used. In the second case, the authors propose a parallel methodology based on the binary search algorithm for finding the optimal location points for the selected frequency range along with the superposition principle to reduce power delivery networks through a congruence transformation. However, a suboptimal selection of the expansion points renders MM methods very inaccurate in the computation of the reduced-order model.

Clearly, the concept of the EKS and AEKS has not yet been explored in the context of circuit simulation. The proposed methodologies alleviate the computational cost by applying state-of-the-art sparse numerical techniques in order to compute the reduced-order model, enabling the analysis of large-scale circuits. Moreover, singular descriptor models are manipulated with sparse matrix operations, without introducing significant computational cost to the proposed methods.

III. BACKGROUND OF MOR BY MOMENT-MATCHING

Consider the Modified Nodal Analysis (MNA) description of an \( n \)-node, \( m \)-branch (inductive), \( p \)-input, and \( q \)-output RLC circuit in the time domain:

\[
\begin{pmatrix}
G & W \\
-W^T & 0
\end{pmatrix}
\begin{pmatrix}
v(t) \\
i(t)
\end{pmatrix}
+
\begin{pmatrix}
C & 0 \\
0 & M
\end{pmatrix}
\begin{pmatrix}
v(t) \\
i(t)
\end{pmatrix}
=
\begin{pmatrix}
B_1 \\
0
\end{pmatrix}
\begin{pmatrix}
u(t) \\
i(t)
\end{pmatrix}
+
Du(t)
\]

where \( G \in \mathbb{R}^{n \times n} \) (node conductance matrix), \( C \in \mathbb{R}^{n \times n} \) (node capacitance matrix), \( M \in \mathbb{R}^{m \times m} \) (branch inductance matrix), \( W \in \mathbb{R}^{n \times m} \) (node-to-branch incidence matrix), \( N \in \mathbb{R}^{n \times n} \) (vector of node voltages), \( i \in \mathbb{R}^{m} \) (vector of current sources). \( B_1 \in \mathbb{R}^{n \times p} \) (input-to-node connectivity matrix), \( y \in \mathbb{R}^{q} \) (vector of output measurements), \( L_1 \in \mathbb{R}^{q \times n} \) (node-to-output connectivity matrix), and \( D \in \mathbb{R}^{q \times p} \) (input-to-output connectivity matrix). Without loss of generality, in the above, we assume that any voltage sources have been transformed to Norton-equivalent current sources and that all outputs are obtained at the nodes as node voltages. Furthermore, \( v(t) \equiv \frac{dv(t)}{dt} \) and \( i(t) \equiv \frac{di(t)}{dt} \).

If we now denote the model order as \( N = n + m \), the state vector as \( x(t) \equiv \begin{pmatrix} v(t) \\ i(t) \end{pmatrix} \), and also:

\[
A \equiv -\begin{pmatrix}
G & W \\
-W^T & 0
\end{pmatrix},
\quad E \equiv \begin{pmatrix}
C & 0 \\
0 & M
\end{pmatrix},
\quad B \equiv \begin{pmatrix}
B_1 \\
0
\end{pmatrix},
\quad L \equiv (L_1 0),
\]

then \( 1 \) can be written in the following generalized state-space form or so-called descriptor form:

\[
E \frac{dx(t)}{dt} = Ax(t) + Bu(t),
\quad y(t) = Lx(t) + Du(t)
\]
The objective of MOR is to produce a reduced-order model:

\[
\begin{align*}
\hat{E} \frac{d\hat{x}(t)}{dt} &= \hat{A}\hat{x}(t) + \hat{B}u(t), \\
\hat{y}(t) &= \hat{L}\hat{x}(t) + Du(t)
\end{align*}
\]

with \(\hat{A}, \hat{E} \in \mathbb{R}^{r \times r}, \hat{B} \in \mathbb{R}^{r \times p}, \hat{L} \in \mathbb{R}^{q \times r}\), where the order of the reduced model is \(r < < N\) and the output error \(\|\hat{y}(t) - y(t)\|_2\) is small. An equivalent metric of accuracy in the frequency domain (via Plancherel’s theorem [12]) is the distance \(\|\hat{H}(s) - H(s)\|_\infty\), where

\[
\begin{align*}
H(s) &= L(sE - A)^{-1}B + D \\
\hat{H}(s) &= \hat{L}(s\hat{E} - \hat{A})^{-1}\hat{B} + D
\end{align*}
\]

are the transfer functions of the original and the reduced-order model, and \(\|\cdot\|_\infty\) is the induced \(\mathcal{L}_2\) matrix norm (or the \(\mathcal{H}_\infty\) norm of a rational transfer function).

The most important and successful MOR methods for linear systems are based on MM. They are very efficient in circuit simulation problems and are formulated in a way that has a direct application to the linear model of (2).

By applying the Laplace transform to (2), we obtain the \(s\) domain equations as:

\[
\begin{align*}
sEx(s) - x(0) &= Ax(s) + Bu(s) \\
y(s) &= Lx(s) + Du(s)
\end{align*}
\]

Assuming that \(x(0) = 0\) and that a unit impulse is applied to \(u(s)\) (i.e., \(u(s) = 1\)), then the above system of equations can be written as follows:

\[
\begin{align*}
(sE - A)x(s) &= B \\
y(s) &= Lx(s) + D
\end{align*}
\]

and by expanding the Taylor series of \(x(s)\) around zero, we derive the following equation:

\[
(sE - A)(x_0 + x_1s + x_2s^2 + \ldots) = B
\]

The transfer function of (2) is a function of \(s\), and can be expanded into a moment expansion around \(s = 0\) as follows:

\[
H(s) = M_0 + M_1s + M_2s^2 + M_3s^3 \ldots
\]

where \(M_0, M_1, M_2, M_3, \ldots\) are the moments of the transfer function. Specifically, in circuit simulation problems, \(M_0\) is the DC solution of the linear system. This means that the inductors of the circuit are considered as short circuits and the capacitors as open circuits. Moreover, \(M_1\) is the Elmore delay of the linear model, which is defined as the time required for a signal at the input port to reach the output port. In general, \(M_i\) is related to the system matrices as:

\[
M_i = L(A^{-1}E)^iA^{-1}B
\]

The goal of MM reduction techniques is the derivation of a reduced-order model where some moments \(M_i\) of the reduced-order transfer function \(\hat{H}(s)\) match some moments of the original transfer function \(H(s)\).

Let us now denote the two projection matrices onto a lower dimensional subspace as \(V_\ell, V_r \in \mathbb{R}^{N \times r}\). These matrices can be derived from the associated moments using one or more expansion points. As a result, if we assume that \(s = 0\), then the matrices \(V_\ell\) and \(V_r\) are defined as follows:

\[
\text{range}(V_r) = \text{span}\{A^{-1}B, (A^{-1}E)A^{-1}B, \ldots, (A^{-1}E)^{-1}A^{-1}B\}
\]

\[
\text{range}(V_\ell) = \text{span}\{L^T, (E^TA^{-T})L^T, \ldots, (E^TA^{-T})r^{-1}L^T\}
\]

The computed reduced-order model matches the first \(2r\) moments and is obtained by the following matrices:

\[
\begin{align*}
\tilde{E} &= V_\ell^TEV_r, \\
\tilde{A} &= V_\ell^TAV_r, \\
\tilde{B} &= V_\ell^TB, \\
\tilde{L} &= LV_r
\end{align*}
\]

This reduced model provides a good approximation around the DC point. Finally, in case we employ a one-sided Krylov method, which is usually the case, the matrix \(V_r\) can be set equal to \(V_\ell\), an equality that also holds for symmetric systems.

IV. THE EXTENDED AND THE ASYMMETRIC EXTENDED KRYLOV SUBSPACES (EKS AND AEKS)

A. Specification of EKS and its Application to MM

The essence of MM methods is to iteratively compute a projection subspace, and then project the original system into this subspace in order to obtain the reduced-order model of (3). The dimension of the projection subspace is increased in every iteration, until an “a priori” selection of the moments is matched. More specifically, if \(r\) is the desired order for the reduced system and \(k = \frac{p}{r}\) is the number of moments (where \(p\) denotes the total number of input/output ports), then \(V \in \mathbb{R}^{N \times r}\) is a projection matrix whose columns span the \(r\)-dimensional Krylov subspace:

\[
\mathcal{K}_r(A_E, B_E) = \text{span}\{B_E, A_EB_E, A_E^2B_E, \ldots, A_E^{-1}B_E\}
\]

where

\[
A_E \equiv A^{-1}E, \quad B_E \equiv A^{-1}B
\]

Then, the reduced-order model is obtained through the following matrix transformations:

\[
\begin{align*}
\tilde{E} &= V^TEV_r, \\
\tilde{A} &= V^TAV_r, \\
\tilde{B} &= V^TB, \\
\tilde{L} &= LV_r
\end{align*}
\]

with \(\tilde{A}, \tilde{E} \in \mathbb{R}^{r \times r}, \tilde{B} \in \mathbb{R}^{r \times p}, \tilde{L} \in \mathbb{R}^{q \times r}\).

The effectiveness of the projection process can be enhanced by expanding around more points than \(s = 0\), leading to the rational Krylov subspace that has seen various implementations such as \(\mathcal{K}_r(A_E, B_E)\). However, there exists no universal procedure for the selection of the expansion points since this is highly problem-dependent. In order to address this issue, we propose the use of the extended Krylov subspace (EKS) for MOR by moment-matching. The EKS is effectively the combination of the standard Krylov subspace \(\mathcal{K}_{r/2}(A_E, B_E)\) and the subspace \(\mathcal{K}_{r/2}(A_E^{-1}, B_E)\) corresponding to the inverse matrix \(A_E^{-1}\), i.e.,

\[
k_r^E(A_E, B_E) = k_{r/2}(A_E, B_E) + k_{r/2}(A_E^{-1}, B_E) = \text{span}\{B_E, A_E^{-1}B_E, A_EB_E, A_E^{-2}B_E, A_E^2B_E, A_E^{-3}B_E, \ldots\}
\]
\[
{A_E^{(r/2)} - B_E, A_E^{-r/2}B_E}\]  \quad (14)

The EKS is an enriched subspace that expands around the extreme points of zero and infinity, and has been used successfully in the areas of matrix function approximation [15] and solution of large Lyapunov equations [16], [17], having proved to greatly enhance the performance of Krylov-subspace methods, while avoiding the ambiguity in the choice of expansion points (note that this is a different EKS than the one proposed in [18] which still requires expansion points). A projection matrix \( V \in \mathbb{R}^{N \times r} \) whose columns span the EKS \((14)\) can be computed by the Arnoldi procedure given in Algorithm 1 [19], which begins with the pair \( \{B_E, A_E^{-1}B_E\} \) and then iteratively increases the dimension of the generated EKS. Regarding the implementation of the algorithm, a modified Gram-Schmidt procedure is employed to implement the \( qr() \) steps of 3 and 9, while step 8 performs orthogonalization with respect to another matrix employing the procedure shown in Algorithm 2 [19].

**Algorithm 1: EKS computation by Arnoldi procedure**

**Input:** \( A_E \equiv A^{-1}E, B_E \equiv A^{-1}B \), desired order \( r \), #ports \( p \)

**Output:** \( V \)

1. Function `compute_EKS(A_E,B_E,r)`:
   2. \( j = 1 \)
   3. \( V^{(j)} = qr([B_E,A_E^{-1}B_E]) \)
   4. \( k = \lfloor r/2 \rfloor \)
   5. while \( j < k \) do
      6. \( k_1 = 2p(j - 1), k_2 = k_1 + p, k_3 = 2pj \)
      7. \( V_1 = [A_EV^{(j)}(:, k_1 + 1 : k_2), A_E^{-1}V^{(j)}(:, k_2 + 1 : k_3)] \)
      8. \( V_2 = \text{orth}(V_1, V^{(j)}, p) \)
      9. \( V_3 = qr(V_2) \)
      10. \( V^{(j+1)} = [V^{(j)}, V_3] \)
   11. \( j = j + 1 \)
   12. end
   13. \( V = V^{(j)}(:, 1 : r) \)
   14. return \( V \)
15. End Function

**Algorithm 2: Orthogonalization w.r.t. another matrix**

**Input:** \( V_1, V^{(j)}, \# \text{ports} \) \( p \)

**Output:** \( V_2 \)

1. Function `orth(V_1, V^{(j)}, p)`:
   2. for \( k_1 = 1, \ldots, j \) do
      3. \( k_2 = 2p(k_1 - 1), k_3 = 2pk_1 \)
      4. \( V_2 = V_1 - V^{(j)}(:, k_2 + 1 : k_3)V^{(j)}(:, k_2 + 1 : k_3)V_1 \)
   5. end
   6. return \( V_2 \)
7. End Function

**B. Sparse Matrices and the Asymmetric EKS**

In practical implementations, the inputs to Algorithm [1] are not actually the matrices \( A_E = A^{-1}E \) and \( B_E = A^{-1}B \) but the original system matrices \( A \) and \( E \) which are very sparse in typical circuit problems. This is because the generally dense inverse matrices \( A^{-1} \) and \( E^{-1} \) are only needed in products with \( p \) vectors (initially in step 3) and \( 2pj \) vectors (in step 7) at every iteration, where the iteration count \( j \) is normally very small and thus \( 2pj \ll N \). These products can be implemented as sparse linear solves \( (EY = R \text{ and } AY = R) \) by employing any sparse direct [13] or iterative [14] algorithm.

However, it is also the case in common circuit problems that one of the matrices \( A \) or \( E \) is considerably sparser than the other. That is usually the matrix \( E \) containing the energy-storage elements (capacitances and inductances), which are typically much fewer than the conductances in matrix \( A \) (unless the inductance block \( M \) is dense - containing many mutual inductances - where the situation is reversed). In that case, the solves with \( E \) are much more efficient than the solves with \( A \), and it can be very computationally beneficial to deviate from symmetry in the construction of the EKS \((14)\).

Here, we introduce the Asymmetric EKS (AEKS) method which, after initially computing the sparsity (i.e., number of nonzeros) of the matrices \( A \) and \( E \), expands predominantly in the direction \( (A_E^{-1} \text{ or } A_E^{-1}) \) that will generate more sparse solves, by adding one moment block corresponding to the denser solve for every \( m \) moment blocks of the sparser solve. For example, in the usual case where \( E \) is sparser than \( A \), and for \( m = 3 \), the proposed AEKS method will generate one moment block of \( A_E = A^{-1}E \) after 3 moment blocks of \( A_E^{-1} = E^{-1}A \), and will create the following \( r \)-dimensional subspace:

\[
K_{t/4}^{AE}(A_E,B_E) = \mathcal{K}_t(A_E,B_E) + \mathcal{K}_{3r/4}(A_E^{-1},B_E) = span\{B_E, A_E^{-3}B_E, A_E^{-2}B_E, A_E^{-1}B_E, A_EB_E, A_E^{-4}B_E, \ldots, A_E^{(r/4)-1}B_E, A_E^{-(3r/4-2)}B_E, A_E^{-(3r/4-1)}B_E, A_E^{-(3r/4)}B_E\}
\]

(15)

The AEKS procedure is shown visually in Fig. 1 (compared with the symmetric EKS), and its computation is given in Algorithm 3.

**V. SINGULAR DESCRIPTOR CIRCUIT MODELS**

**A. General Handling**

The standard Krylov subspace [11] used in traditional moment-matching requires the system matrix \( A \) containing the conductances to be nonsingular, which is always the case for solvable and physically viable circuits. However, the EKS and AEKS procedures of Algorithms [1] and [3] also require the matrix \( E \) to be nonsingular, which is not actually the case for several circuit simulation problems. Such circuit models are referred to as singular descriptor models, and typically result when there are some nodes, say \( n_2 \), where no capacitance is connected, leading to corresponding all-zero rows and columns in the submatrix \( C \). Note that in case the circuit contains no voltage sources, the submatrix \( M \) of inductive branches...
Fig. 1: Visual representation of (a) the EKS sequence, (b) the AEKS sequence. Blue and purple blocks correspond to powers of matrices $A_E$ and $A_E^{-1}$, respectively.

is always nonsingular. If the $n_2$ nodes with no capacitance connection are enumerated last, and the remaining $n_1 = n - n_2$ nodes first, then $[L]$ can be partitioned as follows:

$$
\begin{pmatrix}
G_{11} & G_{12} & W_1 \\
G_{12}^T & G_{22} & W_2 \\
-W_1^T & -W_2^T & 0
\end{pmatrix}
\begin{pmatrix}
v_1(t) \\
v_2(t) \\
i(t)
\end{pmatrix}
+ 
\begin{pmatrix}
C_1 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
v_1(t) \\
v_3(t) \\
i(t)
\end{pmatrix} = 
\begin{pmatrix}
B_1 \\
B_2 \\
0
\end{pmatrix}u(t)  \tag{16}
$$

where $G_{11} \in \mathbb{R}^{n_1 \times n_1}$, $G_{12} \in \mathbb{R}^{n_1 \times n_2}$, $G_{22} \in \mathbb{R}^{n_2 \times n_2}$, $W_1 \in \mathbb{R}^{n \times m}$, $W_2 \in \mathbb{R}^{n_2 \times m}$, $C_1 \in \mathbb{R}^{n_1 \times n_1}$, $v_1 \in \mathbb{R}^{n_1}$, $v_2 \in \mathbb{R}^{n}$, $B_1 \in \mathbb{R}^{n_1 \times p}$, $B_2 \in \mathbb{R}^{n_2 \times p}$, $L_1 \in \mathbb{R}^{q \times n_1}$, and $L_2 \in \mathbb{R}^{q \times n_2}$.

Assuming now that the submatrix $G_{22}$ is nonsingular (a sufficient condition for this is at least one resistive connection from any of the $n_2$ non-capacitive nodes to ground), the second row of $[L]$ can be solved for $v_2(t)$ as follows:

$$
\begin{pmatrix}
1 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
v_1(t) \\
v_2(t) \\
i(t)
\end{pmatrix} + Du(t)
$$

$$
v_2(t) = G_{22}^{-1}B_2u(t) - G_{22}^{-1}G_{12}^TV_1(t) - G_{22}^{-1}W_2i(t)  \tag{17}
$$

The above can be substituted to the first and third row of (16), as well as the output part of (16), to give:

$$
\begin{pmatrix}
(v_1(t)) \\
0 \\
0
\end{pmatrix} = 
\begin{pmatrix}
C_1 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
(v_1(t)) \\
i(t)
\end{pmatrix} - 
\begin{pmatrix}
G_{11} - G_{12}G_{22}^{-1}G_{12}^T \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
v_1(t) \\
W_1 - G_{12}G_{22}^{-1}W_2 \\
i(t)
\end{pmatrix} + C_1v_1(t) + (B_1 - G_{12}G_{22}^{-1}B_2)u(t)
$$

$$
W_2^TG_{22}^{-1}G_{12}^TV_1(t) + W_2^TG_{22}^{-1}W_2i(t) + Mi(t) = W_2^TG_{22}^{-1}B_2u(t)
$$

$$
y(t) = (L_1 - L_2G_{22}^{-1}G_{12}^T)v_1(t) - L_2G_{22}^{-1}W_2i(t) + (L_2G_{22}^{-1}B_2 + D)u(t)  \tag{18}
$$

This can be put together in the following descriptor form:

$$
\begin{pmatrix}
C_1 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
(v_1(t)) \\
i(t)
\end{pmatrix} = 
\begin{pmatrix}
G_{11} - G_{12}G_{22}^{-1}G_{12}^T \\
W_2^TG_{22}^{-1}G_{12}^T - W_1^T \\
B_1 - G_{12}G_{22}^{-1}B_2
\end{pmatrix}
\begin{pmatrix}
v_1(t) \\
i(t)
\end{pmatrix} + 
\begin{pmatrix}
0 \\
W_2^TG_{22}^{-1}B_2 \\
0
\end{pmatrix}u(t)
$$

$$
y(t) = (L_1 - L_2G_{22}^{-1}G_{12}^T)v_1(t) + L_2G_{22}^{-1}W_2i(t) + (L_2G_{22}^{-1}B_2 + D)u(t)  \tag{18}
$$

Algorithm 3: AEKS computation by Arnoldi procedure

**Input:** $A_E = A^{-1}E, B_E = A^{-1}B$, desired order $r$, #ports $p$, modulo $m$

**Output:** $V$

```plaintext
Function compute_AEKS(A_E, B_E, r, m):
    j = 1
    if nnz(E) ≤ nnz(A) then
        V(j) = qr([B_E, A_E^1B_E])
    else
        V(j) = qr([A_E^1B_E, B_E])
    end
    k = ⌈r/(m+1)⌉
    k_1 = 0, k_2 = k_1 + p, k_3 = 2p
    while j < k do
        if mod(j, m) ≠ 0 & nnz(E) ≤ nnz(A) then
            V_1 = [A_E^1V(j)(:, k_2 + 1 : k_3)]
            k_2 = k_3, k_3 = k_2 + p
        end
        if mod(j, m) ≠ 0 & nnz(A) ≤ nnz(E) then
            V_1 = [A_E^1V(j)(:, k_1 + 1 : k_2), A_E^1V(j)(:, k_2 + 1 : k_3)]
            k_1 = k_1 + (m + 1)p, k_2 = k_1 + p,
            k_3 = k_2 + p
        end
        if mod(j, m) = 0 & nnz(A) ≤ nnz(E) then
            V_1 = [A_E^1V(j)(:, k_1 + 1 : k_2), A_E^1V(j)(:, k_2 + 1 : k_3)]
            k_1 = k_1 + (m + 1)p, k_2 = k_1 + p,
            k_3 = k_2 + p
        end
        V_2 = orth(V_1, V(j), p)
        V_3 = qr(V_2)
        V(j+1) = [V(j), V_3]
    end
    j = j + 1
return V
```

End Function
The above is a nonsingular (i.e., regular) state-space model which can be reduced normally by the EKS and AEKS procedures of Algorithms 1 and 3.

B. Sparse Implementation of EKS and AEKS

The Algorithms 1 and 3 are now applicable to the nonsingular descriptor model (18), but their execution is computationally inefficient because the inversion of $G_{22}$ renders the matrices dense and hinders the solution procedure. In this subsection, we present efficient ways to implement Algorithm 1 (and likewise Algorithm 3) by preserving the original sparse form of the system matrices.

1) Construction of RHS: The input-to-state and state-to-output connectivity matrices

$$
B = \begin{pmatrix} B_1 & B_2 \end{pmatrix}, \quad L^T = \begin{pmatrix} L_1^T & L_2^T \end{pmatrix}
$$

are explicitly constructed to compute the input matrix $B_E$ of Algorithm 1 and to obtain the reduced-order model through (13). The products $G_{22}^{-1}B_2$ and $G_{22}^{-1}L_2^T$ are computed by $p$ and $q$ sparse linear solves, respectively.

2) Sparse linear system solutions: The system matrix

$$
A \equiv \begin{pmatrix} G_{11} & G_{12} & G_{12}^{-1}G_{12}T & W_1 & G_{12}^{-1}W_2 \ 
W_2 & G_{22}^{-1} & W_2T & G_{12}^{-1} & G_{22}^{-1}W_2 
\end{pmatrix}
$$

of the model given in (18) is rendered dense due to the inversion of $G_{22}$. The linear system solutions with $A$ in steps 3, 7 of Algorithm 1 can be handled by partitioning the right-hand-side of these systems conformally to $A$, i.e., $R = \begin{pmatrix} R_1 & R_2 \end{pmatrix}$ with $R_1 \in \mathbb{R}^{n_1 \times p}$, $R_2 \in \mathbb{R}^{m \times p}$, and implementing their solution efficiently by keeping all the sub-blocks in their original sparse form as follows:

$$
\begin{pmatrix} G_{11} & W_1 & G_{12} & W_2T & G_{22}^{-1} 
W_2 & G_{12}^{-1}G_{12}T & W_2 & G_{12}^{-1} & G_{22}^{-1} 
\end{pmatrix} \begin{pmatrix} V_1 
V_2 
\end{pmatrix} = \begin{pmatrix} R_1 
R_2 
\end{pmatrix}
$$

where $T \in \mathbb{R}^{n_2 \times p}$ is a temporary sub-matrix.

3) Sparse matrix-vector products: The matrix-vector products with $V^{(j)}$ in step 7 of Algorithm 1 can be implemented efficiently by observing that:

$$
A = \begin{pmatrix} G_{11} & W_1 
W_1 & G_{22}^{-1} 
\end{pmatrix} + \begin{pmatrix} G_{12} & W_2 
W_2 & G_{22}^{-1} 
\end{pmatrix} = \begin{pmatrix} G_{11} & W_1 
W_1 & G_{22}^{-1} 
\end{pmatrix} + \begin{pmatrix} G_{12} & W_2 
W_2 & G_{22}^{-1} 
\end{pmatrix}
$$

Therefore, the product $AV^{(j)}$ with $p$ vectors $V^{(j)}$ can be carried out by a sparse solve $G_{22}V = -G_{12}^{-1}W_2^T V^{(j)}$, followed by a sum of products $\begin{pmatrix} G_{11} & W_1 
W_1 & G_{22}^{-1} 
\end{pmatrix} + \begin{pmatrix} G_{12} & W_2 
W_2 & G_{22}^{-1} 
\end{pmatrix} V^{(j)}$.

4) Construction of system matrix: In order to construct and then reduce the dense system matrix of (20), we need to employ sparse solves with the submatrix $G_{22}$. Since usually $n_2 << n_1$, it is better to first compute the left-solves $G_{12}^{-1}G_{22}^{-1}$ and $W_2^T G_{22}^{-1}$, followed by products with $G_{12}^{-1}$ and $W_2$. The left-solves can be performed as $G_{22}V = G_{12}$ and $G_{22}V = W_2^T$, where $V$ contains the rows of each left-solve.

VI. EFFICIENT COMPUTATION OF THE REDUCED-ORDER RESPONSE AND TRANSFER FUNCTION

To efficiently compute the transfer function and the output response of a multi-input multi-output (MIMO) descriptor model like (2), we can consider the following single-input multi-output (SIMO) subsystems:

$$
E \frac{dx(t)}{dt} = Ax(t) + bu_i(t), \quad y_i(t) = Lx(t) + d_iu_i(t)
$$

where $b_i$ and $d_i$ are the $i$-th columns of matrices $B$ and $D$, respectively, and $u_i(t)$ is the $i$-th input ($i = 1, \ldots, p$). From these SIMO subsystems, the output of the MIMO descriptor system is $y(s) = \sum_{i=1}^{p} y_i(s) = \sum_{i=1}^{p} h_i(s)u_i(s)$, where

$$
h_i(s) = L(sE - A)^{-1} b_i + d_i
$$

This effectively represents the superposition property of linear and time-invariant (LTI) systems.

The above decomposition can be employed for the parallel computation of the reduced-order MIMO transfer function. In particular, for each SIMO subsystem of (23), a projection matrix $V_i \in \mathbb{R}^{N \times r}$ can be computed, whose columns span the $r$-dimensional EKS (or AEKS):

$$
K_i^E(A_E, b_i^E) = K_{r/2}(A_E, b_i^E) + K_{r/2}(A_E^{-1}, b_i^E) = span\{b_i^E, A_E^{-1}b_i^E, A_E b_i^E, A_E^{-2}b_i^E, A_E^2 b_i^E, A_E^{-3}b_i^E, \ldots, A_E^{(r/2)-1}b_i^E, A_E^{-r/2}b_i^E\}
$$

where $b_i^E \equiv A^{-1}b_i$. The computation of the projection matrices (by Algorithm 1 or 3) is independent from one another and can be performed in parallel. The reduced-order SISO subsystems can then be computed in parallel as:

$$
\tilde{E}_i = V_i^T E V_i, \quad \tilde{A}_i = V_i^T A V_i, \quad \tilde{b}_i = V_i^T b_i, \quad \tilde{L}_i = L V_i
$$

The $i$-th column of the MIMO reduced-order transfer function would then be:

$$
\tilde{h}_i(s) = \tilde{L}_i(s\tilde{E}_i - \tilde{A}_i)^{-1} \tilde{b}_i + d_i
$$

and the whole MIMO reduced-order transfer function can be derived as the concatenation:

$$
\tilde{H}(s) = [\tilde{h}_1(s); \tilde{h}_2(s); \ldots; \tilde{h}_p(s)]
$$

It must be noted that this decomposition is better combined with direct solvers, since the computation of the Krylov subspace requires sparse solves with $A$ and $E$ as mentioned previously. Using direct solvers, one can pre-compute the proper decomposition of the above matrices and then re-use them in each parallel computation.
TABLE I: Reduction results of EKS-MM and AEKS-MM vs MM for industrial IBM power grid benchmarks

| Bench. | Dimension | #ports | ROM Order | MM Max Error | MM Runtime (s) | MM Error Red. Percentage | MM Runtime (s) | EKS-MM Max Error | EKS-MM Error Red. Percentage | EKS-MM Runtime (s) | AEKS-MM Max Error | AEKS-MM Error Red. Percentage | AEKS-MM Runtime (s) |
|--------|-----------|--------|-----------|--------------|---------------|--------------------------|---------------|------------------|--------------------------|-----------------|------------------|--------------------------|-----------------|
| ibmpg1 | 44946     | 500    | 2000      | 0.177        | 0.075         | 57.62%                   | 0.146         | 0.075            | 71.07%                   | 0.052           | 0.122            | 49.16%                   | 0.336           |
| ibmpg2 | 127568    | 500    | 2000      | 0.178        | 1.206         | 65.67%                   | 0.146         | 0.122            | 53.64%                   | 0.336           | 0.098            | 59.50%                   | 0.343           |
| ibmpg3 | 852539    | 800    | 3200      | 0.240        | 11.029        | 50.97%                   | 10.998        | 0.063            | 83.69%                   | 0.343           | 0.098            | 51.64%                   | 0.343           |
| ibmpg4 | 954545    | 600    | 2400      | 0.233        | 16.642        | 65.67%                   | 3.782         | 0.108            | 49.16%                   | 0.336           | 0.052            | 71.07%                   | 0.146           |
| ibmpg5 | 1618397   | 600    | 2400      | 0.242        | 10.228        | 57.62%                   | 0.146         | 0.122            | 31.07%                   | 0.052           | 0.122            | 49.16%                   | 0.336           |
| ibmpg6 | 2506733   | 1000   | 6000      | 0.161        | 19.155        | 19.74%                   | 0.310         | 0.108            | 53.64%                   | 0.336           | 0.098            | 59.50%                   | 0.343           |
| ibmpg1t| 54265     | 400    | 1600      | 1.616        | 0.241         | 49.16%                   | 1.414         | 1.297            | 12.50%                   | 3.187           | 1.414            | 12.50%                   | 3.187           |
| ibmpg2t| 164897    | 800    | 3200      | 0.910        | 1.268         | 31.07%                   | 0.052         | 0.398            | 23.84%                   | 0.646           | 0.098            | 59.50%                   | 0.343           |

Fig. 2: Comparison of transfer functions of ROMs obtained by EKS-MM, AEKS-MM, and MM in the range \([10^0, 10^{12}]\) for ibmpg1 and ibmpg2t benchmarks at ports (9,9) and (4,4), respectively.

Fig. 3: Comparison of transfer functions and absolute error magnitudes of ROMs obtained by EKS-MM, AEKS-MM, and MM in the range \([10^0, 10^{12}]\) for ibmpg6 at port (5,5).

VII. EXPERIMENTAL RESULTS

For the experimental evaluation of the proposed methodologies, we used the available IBM power grid benchmarks [21]. Their characteristics are shown in the first three columns of Table I. Note that for the transient analysis benchmarks, ibmpg1t and ibmpg2t, a matrix of energy storage elements (capacitances and inductances) is provided. However, in order to perform transient analysis for the DC analysis benchmarks, ibmpg1 to ibmpg6, we had to add a (typical for power grids) diagonal capacitance matrix with random values in the order of picofarad. In order to evaluate our methodology on singular benchmarks, we enforced the capacitance matrix of ibmpg2 and ibmpg4 to have at least one node that was missing a capacitance connection. These benchmarks along with ibmpg1t and ibmpg2t were represented as singular descriptor models of (16), and thus we applied the techniques described in Section V-B for their efficient sparse handling.

EKS-MM and AEKS-MM were implemented with the procedures described in Sections IV and VI and were compared with a standard MM method also implemented with the superposition property. The reduced-order models (ROMs) were evaluated in the frequency range \([10^0, 10^{12}]\) with respect to their accuracy for given ROM order. For our experiments, an appropriate number of matching moments was selected such that the ROM order for both EKS-MM, AEKS-MM, and MM is the same. All experiments were executed on a Linux workstation with a 3.6GHz Intel Core i7 CPU and 32GB memory using MATLAB R2015a.
Our results are reported in the remaining columns of Table 1, where ROM Order refers to the size of A and B. ROM matrices, Max Error refers to the error between the infinity norms of the transfer functions, \( \| \tilde{H}(s) - H(s) \|_{\infty} \), Runtime refers to the computational time (in seconds) needed to generate each submatrix \( \mathbf{H}_i(s) \) of (27), while Error Red. Percentage refers to the error reduction percentage achieved by EKS-MM and AEKS-MM over MM. It can be clearly verified that, compared to MM for similar ROM order, EKS-MM and AEKS-MM produce ROMs with significantly smaller error. As depicted in Table 1, the Error Red. Percentage ranges from 19.25% to 85.28% for EKS-MM, while for AEKS-MM ranges from 11.80% to 65.67%. The execution time of EKS-MM is negligibly larger than standard MM for each moment computation, due to the expansion in two points, however, the efficient implementation can effectively mask this overhead to a substantial extent and make the procedure applicable to very large circuit models. On the other hand, AEKS-MM exploits the sparse solve and dramatically reduces the runtime, while the error remains acceptable with respect to EKS-MM and it is still superior to MM.

To demonstrate the accuracy of our method, we compare the transfer functions of the original model and the ROMs generated by EKS-MM, AEKS-MM, and MM. The corresponding transfer functions for one regular (ibmpg1) and one singular (ibmpg2t) benchmark, in the band \([10^9, 10^{12}]\), are shown in Fig. 2. Fig. 3 presents the transfer functions of ROMs produced by EKS-MM, AEKS-MM, and MM along with the absolute errors induced over the original model for a selected benchmark in the same band. As can be seen, the response of EKS-MM and AEKS-MM ROMs is performing very close to the original model, while the response of MM ROM exhibits a clear deviation. In particular, responses of ROMs produced by MM do not capture effectively the dips and overshoots that arise in some frequencies.

**VIII. Conclusions**

In this paper, we proposed the use of EKS and AEKS to enhance the accuracy of MM methods for large-scale descriptor circuit models. Our methods provide clear improvements in reduced-order model accuracy compared to a standard Krylov subspace MM technique. For the implementation, we made efficient computational choices, as well as adaptations and modifications for large-scale singular models. On top of that, we have shown that AEKS can greatly reduce the runtime of EKS, inducing only a small overhead in the reduction error. As a result, both proposed methods are generally computationally efficient, allowing the parallel computation of the transfer function and can be straightforwardly implemented without requiring computation of expansion points.

**References**

[1] J. Phillips, L. Daniel and L. Miguel Silveira, “Guaranteed passive balancing transformations for model order reduction,” in Design Automation Conference, pp 52–57, 2002.

[2] A. Odabasioglu, M. Celik and L. T. Pileggi, “PRIMA: passive reduced-order interconnect macromodeling algorithm,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 17, no. 8, pp. 645–654, 1998.

[3] S. Mei and Y. I. Ismail, “Stable Parallelizable Model Order Reduction for Circuits With Frequency-Dependent Elements,” in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 56, no. 6, pp. 1214–1220, 2009.

[4] W. Zhao et al., “Automatic adaptive multi-point moment-matching for descriptor system model order reduction,” in International Symposium on VLSI Design, Automation, and Test, pp. 1-4, 2013.

[5] N. Banagaaya, G. Ali, W. H. A. Schilders and C. Tischendorf, “Implicit index-aware model order reduction for RLC/RC networks,” in Design, Automation & Test in Europe Conference & Exhibition, pp. 1–6, 2014.

[6] L. Knizhnerman and V. Simoncini, “Convergence analysis of the extended Krylov subspace method for the Lyapunov equation,” in Numerische Mathematik, vol. 118, no. 3, pp. 567–586, 2011.

[7] C. Chatziegeorgiou, D. Garyfallou, G. Floros, N. Evmenopoulos and G. Stamoulis, “Exploiting Extended Krylov Subspace for the Reduction of Regular and Singular Circuit Models,” in Asia and South Pacific Design Automation Conference, pp. 773–778, 2021.

[8] Zhenyu Qi, S. X. - T. Han, H. Yu and L. He, “Wideband modeling of RF/analog circuits via hierarchical multi-point model order reduction,” in Asia and South Pacific Design Automation Conference, pp. 224–229, 2005.

[9] G. De Luca, G. Antonini and P. Benner, “A parallel, adaptive multi-point model order reduction algorithm,” in IEEE 22nd Conference on Electrical Performance of Electronic Packaging and Systems, San Jose, CA, 2013, pp. 115-118.

[10] J. Wang and X. Xiong, “Scalable power grid transient analysis via MOR-assisted time-domain simulations,” in International Conference on Computer-Aided Design, pp. 548–552, 2013.

[11] M. T. Kassis, Y. R. Akaveeti, B. H. Meyer and R. Khazaka, “Parallel transient simulation of power delivery networks using model order reduction,” in Conference on Electrical Performance Of Electronic Packaging And Systems, pp. 211-214, 2016.

[12] K. Gröchenig, “Foundations of time-frequency analysis,” in Applied and Numerical Harmonic Analysis, 2001.

[13] T. A. Davis and E. P. Natarajan, “Algorithm 907: KLU: A Direct Sparse Solver for Circuit Simulation Problems,” in ACM Trans. Math. Softw., vol. 37, no. 3, 2010.

[14] D. Garyfallou et al., “A Combinatorial Multigrid Preconditioned Iterative Method for Large Scale Circuit Simulation on GPUs,” in International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, pp. 209–212, 2018.

[15] L. Knizhnerman and V. Simoncini, “A new investigation of the extended Krylov subspace method for matrix function evaluations,” in Numerical Linear Algebra Appl., vol. 17, pp. 615–638, 2010.

[16] V. Simoncini, “A new iterative method for solving large-scale Lyapunov matrix equations,” in SIAM Journal on Scientific Computing, vol. 29, no. 3, pp. 1268–1288, 2007.

[17] G. Floros, N. Evmenopoulos and G. Stamoulis, “Frequency-Limited Reduction of Regular and Singular Circuit Models Via Extended Krylov Subspace Method,” in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 7, pp. 1610–1620, 2020.

[18] J. M. Wang and T. V. Nguyen, “Extended Krylov Subspace Method for Reduced Order Analysis of Linear Circuits with Multiple Sources,” in Design Automation Conference, pp. 247–252, 2000.

[19] G. Golub and C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, 1996.

[20] S. G. Talocia and A. Ubolli, “A comparative study of passivity enforcement schemes for linear lumped macromodels,” in IEEE Transactions on Advanced Packaging, vol. 31, no. 4, pp. 673–683, 2008.

[21] S. R. Nassif, “Power grid analysis benchmarks,” in Asia and South Pacific Design Automation Conference, pp. 376–381, 2008.