# Sparse Implicit Projection (SIP) for Reduction of General Many-Terminal Networks

Zuochang Ye, [†]Dmitry Vasilyev, Zhenhai Zhu, Joel R. Phillips
Cadence Research Laboratories
[†]Massachusetts Institute of Technology

## ABSTRACT

This paper is concerned with model order reduction of large scale dynamic systems that have sparse matrix representations, particularly systems with large numbers of input/output "ports." We present an algorithm that combines the advantages of widely-used approaches such as PRIMA and TICER but avoids many of the drawbacks of both. The resulting algorithm is capable of high-order rational approximation, exploits network sparsity, preserves passivity, can be extended to general non-symmetric systems, and can be applied to networks with hundreds or thousands of ports. We develop a common mathematical framework that can encompass all three algorithms, show mathematical relations between them, and point out certain special cases where they are equivalent. We show examples from analysis of industrial on-chip RC/RLC networks that demonstrate performance advantages of more than three orders of magnitude.

## 1. INTRODUCTION

For some time the standard technique for analysis involving on-chip parasitic networks, particularly resistor-capacitor (RC) networks has been to apply model-order-reduction (MOR) techniques to the network to obtain a macromodel of manageable size, then to follow with a more sophisticated analysis, at which point nonlinear transistor or cell models are incorporated. An ideal reduction technique is fast to execute, results in compact models suitable for further analysis, and preserves electrical quantities of interest such as passivity.

Two classes of algorithms have been successful in industrial applications, which with some abuse of terminology we will refer to as projection based approaches[1, 2, 3], in which we include the balanced truncation class[4, 5], and elimination based approaches [6, 7, 8]. The most well-known of the projection based approaches is PRIMA[2]. The primary advantages of this class of methods is that, from a practical perspective, they can be made more or less arbitrarily accurate, and they are very general, applicable to system of arbitrary internal structure. In industry they have been successful for modeling networks with small numbers of bi-directional input/output ports.[1] The primary drawback of these methods is that they do not scale well with the number of ports. For a network described by $n$ nodes and $m$ externally connected terminals, if the approximation order is $q$, then in terms of reduction cost, projection-based methods require $O(qmn)$ memory, and $O(q^2m^2n)$ CPU time, both of which are prohibitive when the size of the system and the number of ports become large. For example, cases with $m = O(10^3)$ and $n = O(10^8)$ can be encountered in modern designs. In addition, the reduced model with size $mq \times mq$ is generally dense, and thus it is still too large in terms of number of non-zeros. Compression-based techniques have been proposed[9, 10] to apply projection techniques on RC networks with large numbers of ports, but these techniques require special system structure and are still not computationally feasible for the $m, n, q$ in the range cited above.

An alternative class of methods are based on direct manipulations of the parasitic network itself, in a manner similar to sparse Gaussian elimination[6, 7, 8]. The earlier approaches, which, again with some abuse of notation, we will refer to as symbolic graph elimination (SGE) are first-order methods that can be shown to generate macromodels that match the first two moments of the transfer function of the starting system. The TICER[8] algorithm gives up the moment matching property in order to guarantee all positive capacitances.[2] The elimination based approaches have three main drawbacks. First, as first-order methods, the achievable accuracy is very limited. We will show industrial examples where the user has the choice between infeasibly large errors (50% relative) and infeasibly large macromodels (20,000 states). Second, they are more complex to program efficiently, as opposed to projection based methods which can benefit from off-the-shelf high performance computational cores (BLAS3 etc.). Third, the implementation of many of these methods is tied closely to system structure. While they have been extended to some limited cases beyond simple RC circuits[11], in general they are difficult to generalize in a way that preserves accuracy, sparsity, and passivity, does not introduce a large number of additional case-specific processing rules, and is very general (e.g. includes arbitrary mutual inductance coupling).

In this paper we propose an approach to the reduction problem that we term Sparse Implicit Projection (SIP). Mathematically, SIP falls in the framework of projection methods, and inherits their accuracy (through matching moments at multiple expansion points), generality, and passivity-preserving properties. However, the SIP implementation is based on sparse matrix manipulations, similar to the elimination approaches, and so is much more efficient and can deal with problems that are orders of magnitude larger than those that can be solved with existing projection methods. In fact a SIP implementation can be based directly on highly optimized off-the-

[1] The methods efficiently handle large numbers of outputs, or large numbers of inputs, but not both at once.

[2] As a side effect of our analysis, we will show that forcing positive capacitances in the context of an SGE approach is in fact not necessary to guarantee a passive macromodel. At one point, it was desirable that a reduction method produce models consisting of positive R/C elements only. However, this is increasingly a minor consideration for many industrial EDA tool flows, as many downstream analysis tools (e.g. delay calculators and circuit simulators) now support state-space models directly. As the loss of the moment matching property can have a deleterious effect on accuracy and robustness of the procedure in some cases, it is desirable to retain it if possible.

shelf sparse matrix codes – it is in a certain sense a sparse-matrix reduction method, whereas the core computations in PRIMA etc. are dense-matrix based. This general viewpoint means that the SIP approach can be extended to more complex problems, for example systems with general, asymmetric matrices, while preserving its essential simplicity and efficiency. Analysis of the SIP algorithm also reveals deep connections between the projection approaches, elimination approaches, and sparse-matrix Gaussian elimination. In certain cases SIP reduces to PRIMA, in certain cases SIP reduces to SGE, and in in certain cases SIP, SGE, TICER, and PRIMA produce mathematically equivalent models.

SIP can also be effective in combination with other algorithms. The methods in [12] can be used for more efficient representation of the LU factors inside the SIP algorithm itself. Also, any of a number of algorithms developed for problems with large numbers of ports [9, 10, 13] can be applied after an initial SIP reduction. In many cases these algorithms cannot be applied to the flat unreduced problem due to its scale, but can be applied after SIP processing.

The rest of this paper is organized as follows. Section 2 formulates the problem mathematically. Section 3 introduces the proposed algorithm SIP, which for simplicity is developed for the simple case of single-point moment matching, and compares it to other approaches. Section 4 contains the description of the general multipoint SIP algorithm. Section 5 extends the proposed method to asymmetric problems, and discusses the extra cost one needs to pay. Section 6 illustrates the new algorithm on realistic examples and compares the algorithm with existing methods. Section 7 summarizes the paper.

## 2. BACKGROUND

Consider for the moment an RC network model. In impedance form, the equations for the network can be written in the frequency domain as

$$(G + sC)x = Eu \tag{1}$$

$$y = E^T x \tag{2}$$

where $u$ is the input vector (e.g. port currents), $x$ is the voltage on internal nodes of the network; $y$ is the output vector (e.g. voltages on the ports). $G, C \in \mathcal{R}^{n\times n}$ represent the conductance matrix and capacitance matrix and $E \in \mathcal{R}^{n\times m}$ is a topological matrix that maps the ports to the nodes. $E$ can be written as

$$E = \left[ \begin{array}{c} 0 \\ I \end{array} \right] \tag{3}$$

where we assume that the nodes are permuted such that all the internal nodes are placed at first and the port nodes are placed at last, and this permutation applies through this paper. In the usual formulation, $C$ and $G$ are positive-semidefinite, an important detail in the passivity-preserving properties of our algorithm. In the Z-form above, $C$ and $G$ are symmetric. $G$ may be singular if there is no resistive path to ground. The model may also be written in Y-form (admittance, voltage inputs, current outputs) as in [2], with non-singular, but non-symmetric $G$.

Consider reduction via an orthogonal projection method that proceeds by constructing a projector $M \in \mathcal{R}^{n\times mq}$, then performing a congruence transformation to the matrices to produce the model

$$\hat{G} = M^T GM, \;\; \hat{C} = M^T CM, \;\; \hat{E} = M^T E. \tag{4}$$

The transfer functions of the original system and the reduced system are written as

$$H(s) \;\; = \;\; E^T (G + sC)^{-1} E \tag{5}$$

$$\hat{H}(s) \;\; = \;\; \hat{E}^T (\hat{G} + s\hat{C})^{-1} \hat{E}. \tag{6}$$

It is desired that $mq \ll n$ so that the size of the model is much smaller than the size of the original network.

By construction[2], for models in this form, the reduced model is passive if the original system is. The accuracy of the method is determined by the choice of projector. The most common choice[2] is to match moments around DC. In the general non-symmetric case $q$ moments are matched for an order-$q$ model. For a symmetric formulation, $2q$ moments can be matched. Depending on formulation, treating the case of singular $G$ can require some special techniques that are outside the scope of this paper.

The PRIMA algorithm constructs the matrix $M$ explicitly, and as a dense matrix (it is typically structurally dense). Likewise dense matrix representations are used for the final models ($\hat{G}$ etc.) as well as intermediate quantities (for example $GM$) computed during the projection. In addition, the algorithms usually conduct orthogonalization procedures, with an additional $O(q^2 m^2 n)$ cost, but also lead to a reduced model that is fundamentally dense and unstructured. This inevitably leads to algorithms whose complexity is quadratic in the order $q$, and, more detrimentally, in the number of ports $m$.

We will now show that these dense manipulations are often not necessary.

## 3. SINGLE POINT SIP ALGORITHM

### 3.1 Derivation

In this section we develop the core SIP approach that matches moments at a single frequency point. From a conceptual viewpoint, the development is within the standard framework of projection methods. The contribution of this section is to show how moment-matching-via-projection can be achieved within the framework of sparse matrix calculations, in particular, sparse matrix factorization codes. There are two concepts that are key to the derivation of this algorithm: the equivalence between a moment-matching projection method and the computation of a Schur complement, and the implicit computation of Schur complements by sparse matrix factorization.

Assume we are given a system described by the equation (1) and (2). Here matrices $G$, $C$, $E$ are sparse and matrices $G, C$ are symmetric. Consider constructing a projection matrix to match moments at DC ($s = 0$). Let $H^{(0)}$ and $H^{(1)}$ denote the order-zero and order-one moments, respectively, of the original system, and $\hat{H}^{(0)}$ and $\hat{H}^{(1)}$ the moments of the reduced order model. The theory of projection methods connects the column span of the projection matrix $M$, the DC solution $V$, and moment matching.

LEMMA 3.1. *If $colsp(M) \supset V$, where $GV = E$, then $\hat{H}^{(0)}(0) = H^{(0)}(0), \hat{H}^{(1)}(0) = H^{(1)}(0)$.*

PROOF. See [1]. □

Now consider ordering and partitioning the matrix, RHS, and state variables to separate the "port" nodes. The equation to solve for the DC moment is

$$GV = \left[ \begin{array}{cc} A & B \\ B^T & D \end{array} \right] \left[ \begin{array}{c} V_1 \\ V_2 \end{array} \right] = E = \left[ \begin{array}{c} 0 \\ I \end{array} \right] \tag{7}$$

where $V_1$ denotes the voltages on internal nodes, and $V_2$ denotes the voltages on port nodes. Applying block-wise Gauss elimination the above equation leads to the following reduced size equation

$$SV_2 = (D - B^T A^{-1} B)V_2 = I. \tag{8}$$

Matrix $S$ is known as the Schur complement.

The Schur complement is intimately connected with the the DC solution. The DC solution is obtained by first solving the reduced equation set (8) involving the Schur complement to obtain $V_2$, and then obtaining $V_1$ as in the following lemma.

LEMMA 3.2. *With $M$ defined as*

$$M = \left[ \begin{array}{c} -A^{-1}B \\ I \end{array} \right] \tag{9}$$

*$V$ can be computed as*

$$V = MS^{-1}$$

PROOF. Follows from simple algebra. □

That the $M$ introduced in Lemma 3.2 is in fact the projection matrix we seek will become clear shortly.

Computing the matrix multiplication in the definition of Schur complement directly is inefficient, especially for large scale sparse matrices. The following theorem gives a more efficient way to calculate it.

THEOREM 3.1. *If G is Cholesky factored as*

$$G = \left[ \begin{array}{cc} A & B \\ B^T & D \end{array} \right] = \left[ \begin{array}{cc} L_1 & \\ L_3 & L_2 \end{array} \right] \left[ \begin{array}{cc} L_1^T & L_3^T \\ & L_2^T \end{array} \right] \tag{10}$$

*then*

$$S = L_2 L_2^T \tag{11}$$

PROOF. Follows simple algebra. □

Since the size of $L_2$ is much smaller than the size of the system, the cost to evaluate (11) is small, and the total cost is determined by the Cholesky decomposition. A more sophisticated analysis will show that there is a slightly more clever way to compute the Schur complement $S$ without first finishing the Cholesky decomposition in (10) and then performing the matrix multiplication $L_2 L_2^T$ in (11).

The next theorem, key to the SIP approach, reveals the relation between the Schur complement and projection-via-congruence.

THEOREM 3.2. *The Schur complement can be written as a congruence transformation*

$$S = M^T G M \tag{12}$$

*with $M$ defined in (9).*

PROOF. The proof is trivial. □

We are now in a position to connect sparse matrix manipulation and the projection in the moment-matching reduction algorithm SIP.

THEOREM 3.3. *With $M$ defined in (9) as the projector, the reduced model of (4) has matching zero and first order moments, $\hat{H}^{(0)}(0) = H^{(0)}(0), \hat{H}^{(1)}(0) = H^{(1)}(0)$.*

PROOF. This follows directly from Lemma 3.1, Lemma 3.2 and Theorem 3.2. For an alternative proof please refer to the appendix. □

At this point we have already shown how to compute $\hat{G}$ in the SIP algorithm. It is simply $\hat{G} = S$ and can be read off directly from the Cholesky factorization of G.[3] Obtaining the reduced C-matrix $\hat{C}$ requires some further manipulation as will be discussed in the next section.

Before proceeding to implementation, we need to address one further detail, the column rank of $M$. In general projection-based MOR algorithms, orthogonalization is an important step to ensure the column independence of the projector. A projector with linearly dependent columns will lead to a singular model. In the SIP algorithm, no orthogonalization is performed, since orthogonalization will destroy the sparsity. But the column independence is guaranteed by the following theorem.

THEOREM 3.4. *$M$ has full column rank.*

PROOF. In view of (9), we have

$$M^T M = B^T A^{-1} A^{-1} B + I. \tag{13}$$

Since $B^T A^{-1} A^{-1} B$ is guaranteed to be non-negative definite, and $I$ is positive definite, their summation is also positive definite. Thus $M$ always has full-rank. □

In practice, $M$ could be somewhat ill-conditioned. However, it follows from the interlacing property of the eigenvalues of Schur complements [14] that $\hat{G}$ is always better conditioned than $G$.

## 3.2 Fast Implementation

The combination of Theorems 3.1-3.3 implies that we can use the Cholesky decomposition to perform the projection of the $G$ matrix with the matrix $M$ in (9) as the projector. In addition, since the projection is a congruence transformation, it preserves passivity. If the projection can also be done efficiently to matrices $C$ and $E$, then we have a reduced model matching the lowest two moments of the original system. This has important practical implications. In the highly developed area of sparse matrix factorization [15], efficient ordering techniques are available to reduce the fill-ins and efficient symbolic analysis is available to predict the non-zero pattern. More importantly, we never need to construct $M$ explicitly. This saves considerable memory and CPU time because matrix $M$ is generally much denser than matrix $L$ in (10). The combination of these two techniques provides a much more efficient way to calculate the projection, compared to the explicit and dense projection in PRIMA.

Now the question becomes: how do we project the $C$ and $E$ matrices without explicitly constructing matrix $M$? In order to answer this question, we need to take a closer look at the steps in the Cholesky decomposition. We will now need to assume, without loss of generality, that matrices $G$ and $C$ have the same non-zero structure. This is certainly true if matrix $C$ is diagonal, an important special case. For the cases where the two matrices have different non-zero structures, we can use the union of the non-zero structure in matrix $G$ and $C$ to guide the Cholesky decomposition.

A standard Cholesky decomposition is performed recursively. At the beginning we let $G^{(1)} = G$, and in each step, a variable is eliminated. At step $i$ the matrix $G^{(i)}$ has the following form

$$G^{(i)} = \left[ \begin{array}{ccc} I_{i-1} & & \\ & a_{i,i} & \mathbf{b}_i \\ & \mathbf{b}_i^T & B^{(i)} \end{array} \right] \tag{14}$$

where $I_{i-1}$ denotes the identity matrix of dimension $i-1$. To eliminate node $i$ we can define matrix $L^{(i)}$ as

$$L^{(i)} = \left[ \begin{array}{ccc} I & & \\ & \sqrt{a_{i,i}} & \\ & -\frac{1}{a_{i,i}}\mathbf{b}_i & I_{n-i} \end{array} \right] \tag{15}$$

and then write $G^{(i)}$ as

$$G^{(i)} = L^{(i)} G^{(i+1)} (L^{(i)})^T \tag{16}$$

[3] We should point out that even though G may be singular, it still has a Cholesky factorization. As it turns out, we do not need the full factorization. So this is not an issue. Matrix $A$ must be nonsingular, but this can always be achieved.

**Algorithm 1** [$\hat{G}$,$\hat{C}$,$\hat{E}$]=SIPcore($G$, $C$, $E$, ports)

```
Get the optimal ordering to reduce the fill-in of matrix $G$
move port nodes to the bottom
permute matrices $G$ and $C$
for $i = 1$ to $n - m$ do
    Construct matrix $M^{(i)}$
    $G^{(i+1)} \leftarrow {M^{(i)}}^T G^{(i)} M^{(i)}$
    $C^{(i+1)} \leftarrow {M^{(i)}}^T C^{(i)} M^{(i)}$
end for
$\hat{G} = G^{(n-m+1)}(n - m + 1 : n, n - m + 1 : n)$
$\hat{C} = C^{(n-m+1)}(n - m + 1 : n, n - m + 1 : n)$
```

where

$$G^{(i+1)} = \begin{bmatrix} I_{i-1} & & \\ & 1 & 0 \\ & 0 & B^{(i)} - \frac{1}{a_{i,i}}\mathbf{b}_i\mathbf{b}_i^T \end{bmatrix}. \qquad (17)$$

We repeat this for $i$ from 1 to $n$. After $n$ steps we have $G^{(n+1)} = I$.

Equation (16) can also be written as

$$G^{(i+1)} = (M^{(i)})^T G^{(i)} M^{(i)} \qquad (18)$$

where $M^{(i)} = \left(L^{(i)}\right)^{-T}$. This reveals that the Cholesky decomposition is actually composed of a series of congruence transformations. Every congruence transformation eliminates a node. Thus if we perform the Cholesky decomposition for $i$ from 1 to $m - n$ instead of doing it all the way, all the internal nodes will be eliminated and all the port nodes will be preserved. The result of this procedure is nothing but the Schur complement in (11). This explains why the Schur complement can be obtained without finishing the Cholesky decomposition. It is clear now that the projector defined in (9) is just the product of many individual projectors

$$M = M^{(1)} M^{(2)} \cdots M^{(n-m)} \qquad (19)$$

This suggests an easy way to perform the projection reduction of the matrix $C$ defined in (4). We just need to recursively project matrix $C$ the same way as in (18).

Further more, it is easy to verify that with matrix $E$ in (3) the projection reduction of matrix $E$ is

$$M^T E = I \qquad (20)$$

The final reduction algorithm is presented in Algorithm 1. We call it *Sparse Implicit Projection* (SIP) because it performs the projection in an implicit manner and it uses the sparse matrix techniques. In particular, only matrix factorizations are required, avoiding the $O(q^2m^2n)$ steps in explicit projection that make those approaches infeasible for large $m$.

## 3.3 Relationships between SIP, PRIMA, and TICER

In this section we discuss some relationships between the various reduction algorithms. For the moment we assume symmetric systems.

First, from Lemma 3.2, it is clear that single-point SIP is mathematically equivalent to a one-pole PRIMA approximation. Structurally they may be very different, which is where SIP gains its efficiency. Likewise, working through the detailed algorithm above (see also Section 3.4), it can be shown that single-point SIP is equivalent to a first-order symbolic graph elimination (SGE) algorithm as described in [6], which was derived from a single-variable Taylor series expansion. Again, the details may vary depending on implementation. The main advantages of SIP are (1) the ability to use existing sparse matrix codes, and (2) the passivity proof. In the TICER approach [8], closely related to SGE, the negative capacitors resulting from the Taylor expansion were dropped. This destroys the moment matching property, and we have found empirically that it can occasionally lead to large errors (see Section 6.2). However, the analysis above shows that dropping negative capacitors in SGE is not necessary to obtain a passive macromodel. The model from SIP is passive by construction. In the most restrictive case where TICER does not generate any negative capacitors that must be dropped, the TICER algorithm, the first-order SGE, single-point SIP and one-point PRIMA are all equivalent. Thus SIP provides a unified theoretical framework to analyze the SGE-based algorithms and the projection-based algorithms.

## 3.4 Node retention for error control

As has been discussed in [8, 16], it can be advantageous to "reserve" additional nodes in the macromodel, beyond the port nodes, to control either sparsity in the macromodel or to reduce the macromodel error. This is the case for SIP as well. To calculate the error introduced by each step of elimination, we start from an error-free formula for the $k$-th step elimination

$$\begin{bmatrix} G_{kk} + sC_{kk} & \mathbf{g}_k + s\mathbf{c}_k \\ (\mathbf{g}_k + s\mathbf{c}_k)^T & \tilde{G} + s\tilde{C} \end{bmatrix} \begin{bmatrix} v_k \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{i} \end{bmatrix} \qquad (21)$$

where $\tilde{G}$ and $\tilde{C}$ are the conductance and capacitance matrices for the sub-circuit excluding node $k$. $\mathbf{g}_k$ and $\mathbf{c}_k$ are vectors of size $n - k - 1$. Eliminating node $k$, we obtain

$$(\tilde{G} + s\tilde{C} - \mathcal{E})\mathbf{v} = \mathbf{i} \qquad (22)$$

where

$$\mathcal{E} \quad = \quad \frac{(\mathbf{g}_k + s\mathbf{c}_k)(\mathbf{g}_k + s\mathbf{c}_k)^T}{G_{kk} + sC_{kk}}. \qquad (23)$$

Matrix $\mathcal{E}$ is not an affine function of frequency $s$, hence the circuit described by (22) can not be realized by an RC circuit with $n-k-1$ nodes. We can use Taylor series to approximate matrix $\mathcal{E}$ such that the reduced network can be realized in the first-order affine form

$$\mathcal{E} \quad \approx \quad \frac{\mathbf{g}_k\mathbf{g}_k^T}{G_{kk}} + s\left(\frac{\mathbf{c}_k\mathbf{g}_k^T + \mathbf{g}_k\mathbf{c}_k^T}{G_{kk}} - \frac{C_{kk}}{G_{kk}}\frac{\mathbf{g}_k\mathbf{g}_k^T}{G_{kk}}\right). \qquad (24)$$

The error introduced in this step can be estimated to determine if the node should be reserved or not. Close inspection reveals these expressions are *exactly the same as* the models obtained by the SIP procedure of Algorithm 1. Therefore, the node retention strategy can be easily utilized in SIP.

Note that the last negative capacitive term in (24) leads to the fundamental difference between the single point SIP and TICER. TICER drops this negative term to ensure the positive capacitance value. In Section 3.1, we have shown that the congruence transformation preserves the passivity. Hence dropping the negative capacitor is unnecessary to ensure the passivity. And this actually destroys the moment matching property.

## 3.5 Sparsity Control

Sparsity control enters at two points in the algorithm. First, the sparsity of the matrix factorization itself needs to be optimized via appropriate matrix orderings. This is a well studied area. Any effective sparse matrix ordering strategy (nested dissection, minimum degree orderings and their variants) can be used in the SIP algorithm. Second, it may be desirable to retain some sparsity in the final model. This can be achieved by simply stopping the factorization process whenever large amounts of fill-ins would occur. The definition of "large" is a user-controlled quantity.

An additional benefit of the SIP approach is that it preserves physical structure in the models. Since the states in the reduced model generated by SIP are part of the state variables of the original system, the original node voltages, the physical structure in the original model is inherited by the SIP reduced model. This is not necessarily true of the models generated by e.g. PRIMA. As a result, we find in many cases that the SIP reduced models, while structurally close to dense with aggressive reduction, are often numerically close to being sparse. Additional sparsity can be introduced by eliminating these small matrix entries, either as a post-processing step, or during the reduction itself, analogous to incomplete factorization. That is, we can delete small entries, corresponding to large resistors or small coupling capacitors, with negligible error. In general, such heuristics would put the passivity guarantee at risk. However, the matrices arising from RC circuits are M-matrices, with special structure. Note that the reduced G matrix, as the Schur complement of an M-matrix, is also an M-matrix. Deletion of off-diagonal entries will never affect model passivity. The reduced capacitance matrix is not an M-matrix, but as a congruence transform of an M-matrix, it has identifiable structure. A close analysis reveals that small numerical entries in both $G$ and $C$ can be deleted by a series of rank-one updates with positive semi-definite matrices, which preserves the semi-definiteness of the reduced $C, G$. Detailed discussion is out of the scope of this paper and will be presented in our future publications.

# 4. MULTI-POINT SIP REDUCTION

## 4.1 Motivation

A non-obvious drawback to the algorithms discussed in the previous section is that they are based on first-order approximations, and first-order approximations are not always sufficiently accurate, *even with the node reservation procedure previously discussed.* This is because not all useful macromodels can be obtained from successive first-order approximations. Examples exist where to obtain, for a given error criteria, sufficiently accurate models, an undesirably large number of nodes must be retained. If these nodes are eliminated, unacceptably large errors are incurred. The results section contains an industrial example where the choice is between 50% error on one hand, and a 20,000 state macromodel on the other. This motivates the question of whether the SIP algorithm can be extended to support higher-order approximations.

There are two choices to extend moment-matching approximations to higher order: we can match more moments at one frequency, as PRIMA does, analogous to keeping more terms in a Taylor series, or we can match moments at additional frequencies (multi-point moment matching), as the "rational Krylov" algorithms[1] do. While we have not conclusively ruled out a single-point high-order approach, exploiting network sparsity seems to be easier to do in a multipoint framework.

## 4.2 Multi-point Moment Matching

In this subsection we propose a multi-point reduction scheme[1]. The basic idea is to generate the projectors from different frequencies, and then merge the projectors together. It is generally appreciated that, for a given model order, multi-point approaches are at least as accurate as single-point, high-order expansions, and sometimes more efficient since they can better distributed the model error over the frequency range of interest. As before, the major difference between the multi-point projection in this paper and most previous work is that the SIP framework allows us to exploit the sparsity of the network.

The multi-point moment matching method can be stated as following. For a predetermined set of $q$ expansion frequencies $\{s_1, s_2, \cdots, s_q\}$, we can define a *shifted* system $\mathcal{S} = \{G_i, C, E\}$, where $G_i = G + s_i C$. The shifted system is governed by the following equation

$$(G_i + sC)x = Eu \tag{25}$$

$$y = E^T x \tag{26}$$

Thus if a reduced model matches the moments at DC for the $i$-th shifted system, it will equivalently match the moments at the finite frequency point $s = s_i$ for the original system, that is, $\hat{H}^{(0)}(s_i) = H^{(0)}(s_i), \hat{H}^{(1)}(s_i) = H^{(1)}(s_i)$.

To match the DC moments of the shifted system, we can do a SIP reduction on the shifted system. This results in the following model

$$\hat{G}_i = M_i^T G_i M_i, \quad \hat{C}_i = M_i^T C M_i. \tag{27}$$

During the process, the following projector will be constructed *implicitly*

$$M_i = \begin{bmatrix} -A_i^{-1} B_i \\ I \end{bmatrix}, \tag{28}$$

where $A_i = A + s_i C_{11}$, $B_i = B + s_i C_{12}$, and $C_{11}$ and $C_{12}$ are from the partitioning shown in (7). The union of all $M_i$ becomes the multi-point projector

$$M = \begin{bmatrix} M_1 & M_2 & \cdots & M_q \end{bmatrix}. \tag{29}$$

The final reduced model is obtained by

$$\hat{G} = M^T G M = \begin{bmatrix} M_1^T G M_1 & \cdots & M_1^T G M_q \\ \vdots & \ddots & \vdots \\ M_q^T G M_1 & \cdots & M_q^T G M_q \end{bmatrix} \tag{30}$$

$$\hat{C} = M^T C M = \begin{bmatrix} M_1^T C M_1 & \cdots & M_1^T C M_q \\ \vdots & \ddots & \vdots \\ M_q^T C M_1 & \cdots & M_q^T C M_q \end{bmatrix}. \tag{31}$$

The obvious way to perform the $q^2$ projections in (30) and (31) is to calculate each individual projection using Algorithm 1. It is easy to verify that the CPU time and memory usage of such a multi-point projection are respectively $q^2$ and $q$ times of those used by Algorithm 1. Fortunately, there is a much more elegant alternative for symmetric systems.

## 4.3 Efficient Multi-point Projections

THEOREM 4.1. *For arbitrary indices $1 < i, j < n$, it holds true that*

$$\hat{G}_i = M_j^T G_i M_i = M_i^T G_i M_j \tag{32}$$

PROOF. We first prove the first equality

$$M_j^T G_i M_i = \begin{bmatrix} -B_j^T A_j^{-1} & I \end{bmatrix} \begin{bmatrix} A_i & B_i \\ B_i^T & D_i \end{bmatrix} \begin{bmatrix} -B_i A_i^{-1} \\ I \end{bmatrix} \tag{33}$$

$$= D_i - B_i^T A_i^{-1} B_i = \hat{G}_i. \tag{34}$$

The second equality in (32) can be proved in a similar way. □

Thus we have

$$\hat{G}_i = M_j^T (G + s_i C) M_i \tag{35}$$

$$\hat{G}_j = M_j^T (G + s_j C) M_i \tag{36}$$

Subtracting (35) from (36) results in

$$\hat{C}_{ji} = M_j^T C M_i = \frac{1}{s_j - s_i} (\hat{G}_j - \hat{G}_i), \tag{37}$$

where $\hat{C}_{ji}$ is the $(j,i)$ sub-matrix of matrix $\hat{C}$ in (31). Substituting $\hat{C}_{ji}$ into (35) gives

$$\hat{G}_{ji} \quad = \quad M_i^T G M_j = \hat{G}_i - s_i \hat{C}_{ji}. \qquad (38)$$

It can be easily verified that $\hat{C}_{ji} = \hat{C}_{ij}$ and $\hat{G}_{ji} = \hat{G}_{ij}$.

Equations (37) and (38) clearly show that the multi-point reduced models in (30) and (31) can be directly constructed from multiple single-point reduced models for the shifted systems in (27). The CPU time is precisely $q$ times that of the single-point SIP reduction plus the cost to compose the reduced model. No additional memory is required since the reductions are done sequenctially. Thus we have the algorithm for the multi-point SIP reduction, as shown in Algorithm 2.

**Algorithm 2** [$\hat{G}$,$\hat{C}$,$\hat{E}$]=SIP($G$, $C$, $E$, ports, $s_1$,$s_2$,...,$s_q$)

```
for $i = 1$ to $q$ do
  [$\hat{G}_i$,$\hat{C}_i$]=SIPcore($G + s_i C$, $C$, ports)
end for
for $i = 1$ to $q$ do
  for $j = 1$ to $i$ do
    $\hat{C}_{ji} = \frac{1}{s_j - s_i}(\hat{G}_j - \hat{G}_i)$
    $\hat{G}_{ji} = \hat{G}_i - s_i \hat{C}_{ji}$
  end for
end for
```

## 4.4 Structural Singularities

Conventional projection methods rely on an orthogonalization step to ensure a well-conditioned basis for projection. For large $m$ and $q$, the orthogonalization is too expensive. We have shown in section 3.1 that the resulting reduced matrices in the single-point SIP approach, the reduced G matrix in particular, do not lose rank. The multi-point case is more involved. A sufficient but not necessary condition for $M$ being full-rank is that $C$ is full-rank. Intuitively, this makes the columns of the projectors generated at different expansion points different from each other. This can be ensured by requiring a capacitive path to the ground from every node, which is the case in all physical systems. If $C$ is not full-rank, the singularity problem becomes more involved and further work is required to resolve it. We have formulated a post-processing procedure to identify and remove these structural singularities. The details of this approach are quite involved and will be presented in a future publication.

# 5. ASYMMETRIC SIP ALGORITHM

So far, we have assumed that matrices $G$ and $C$ are symmetric. This is the case for RC circuit models. It is desirable to extend the SIP approach to encompass non-symmetric problems. The simplest examples are the RLC circuits. As shown in Section 3.1, there are two key relationships in the SIP algorithm: 1) the equivalence between the Schur complement and the Cholesky decomposition; 2) the equivalence between the Schur complement and the moment matching projection-via-congruence. In order to extend the SIP algorithm to the asymmetric problems, we just need to replace the Cholesky decomposition with the LU decomposition and check if the two relationships still hold.

Similar to (7), we can partition the asymmetric $G$ as

$$G = \begin{bmatrix} A & B \\ C^T & D \end{bmatrix}. \qquad (39)$$

The Schur complement now becomes

$$S = D - C^T A^{-1} B \qquad (40)$$

If we already have the LU decomposition

$$G = LU = \begin{bmatrix} L_1 & \\ L_3 & L_2 \end{bmatrix} \begin{bmatrix} U_1 & U_3 \\ & U_2 \end{bmatrix}, \qquad (41)$$

then it can be verified that $S = L_2 U_2$. This means that the relationship between the Schur complement and the LU decomposition still holds. Using $M$ defined in (9) and substituting (39) into (12), one can easily check that equation (40) can be reproduced. Hence the equivalence between the Schur complement and the congruence transformation also holds for the asymmetric systems. Essentially we have shown that Lemma 3.1, Lemma 3.2 and Theorem 3.2 hold true for the asymmetric systems if we use the corresponding equations (39), (40) and (41). Using the same reasoning in the proof of Theorem 3.3, one can show that the moment matching property indeed holds. Same as when the projection-based methods like PRIMA being applied to the asymmetric problems, each single-point SIP reduction matches one moment instead of two moments.

The validity of these two key relationships shows that at least the single point SIP can, mathematically speaking, be extended to the asymmetric problems. In fact it can be extended to truly general matrices. Furthermore, this includes the passivity preserving results if the starting matrices have the requisite structure. The detailed implementation in Algorithm 1 can also be extended to the asymmetric cases, except that the Cholesky factorization is replaced by the LU factorization. Hence the cost of the single-point asymmetric SIP reduction is similar to that of the symmetric SIP reduction.

Multi-point SIP also extends to the asymmetric cases, as should be clear at this point. Unfortunately, we have not been able to find a "fast" method like the one based on equations (37) and (38) to calculate the sub-matrices in $\hat{G}$ and $\hat{C}$. We have to obtain the reduced model by calculating the individual projections in (30) and (31). This requires $q^2$ projections and the memory usage is $q$ times that of the single-point SIP reduction. While this is somewhat disappointing, it is still far better than the $O(q^2 m^2 n)$ CPU time and $O(qmn)$ memory used by the traditional algorithms like PRIMA. And we are not aware of any other algorithms that are (1) high order, (2) passivity preserving, (3) applicable to the general matrix descriptions, (4) feasible for the problems with large $m$ and $n$.

# 6. EXPERIMENTAL RESULTS

In this section we give test results to show various aspects of the proposed algorithms. All algorithms mentioned are implemented in C++. We have normalized all performance figures.

## 6.1 Efficiency for symmetric cases

The first set of examples shows the efficiency compared to PRIMA. The cases under test are 5 RC large-scale circuit networks that were obtained from industrial designs. We emphasize these are not contrived examples – they demonstrate behavior that is observed in practice on modern digital ASIC designs. The examples are summarized in the first column of Table 1. The memory and CPU time used in both SIP and PRIMA are normalized and compared in Table 1, where the numbers in parentheses indicate the PRIMA reduction could not be successfully done within reasonable time or terminated because of memory exhaustion. SIP succesfull reduced all the networks studied. In in this case the memory usage and CPU time by PRIMA were estimated.

The table clearly shows SIP has significant performance advantages over that of PRIMA. In particular, SIP is much faster than PRIMA (possibly orders of magnitude faster) for problems with large numbers of ports. SIP also scales better as the number of ports increases. For example, examples #1 and #4 share the same underlying network, but with different port definitions. The time

SIP takes for the reduction of the 1001-port example was only 2X that for the 11-port example.

**Table 1: Normalized Resource Requirements of SIP compared to PRIMA**

| Problem (n/m) | order | Memory Reduction | Speedup Factor |
|---|---|---|---|
| #1 | 1 | 2X | 4.7X |
| n=419k | 2 | 4X | 7.1X |
| m=11 | 4 | 7X | 10X |
| #2 | 1 | 3X | 20X |
| n=550k | 2 | 5X | 24X |
| m=36 | 4 | 9X | 42X |
| #3 | 1 | 2X | 4.7X |
| n=6.3M | 2 | 4X | 8.6X |
| m=16 | 4 | 7X | 17X |
| #4 | 1 | 66X | 15000X |
| n=419k | 2 | (132X) | (24000X) |
| m=1001 | 4 | (263X) | (21000X) |
| #5 | 1 | (18X) | (1000X) |
| n=10M | 2 | (35X) | (2200X) |
| m=288 | 4 | (69X) | (4800X) |

## 6.2 Accuracy for symmetric cases

In exploring accuracy of the SIP method we will first validate the multi-point approximation capability, and then demonstrate its usefulness. Then we will briefly comment on a loss of robustness TICER can exhibit, due to the loss of moment-matching properties, than SIP does not possess.

To demonstrate multi-point rational approximation capabilites, the easiest means is to compare to the rational Krylov (R-Krylov) method [1], i.e. multi-point moment matching approximations computed with explicit projection. Theoretically, the SIP and R-Krylov models are equivalent, though they are computed differently. We take problem #3 as the test case, and plot the relative error of the transfer functions of both reduced models in Figure 1. It shows in the figure that the curves resulting from SIP and R-Krylov are on top of each other, as theory predicts.

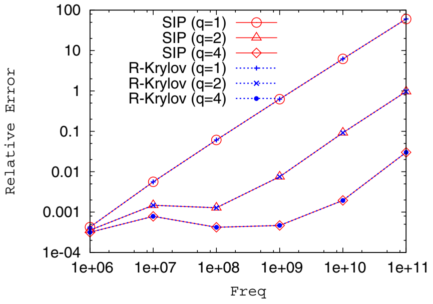

**Figure 1: Accuracy of SIP and Rational Krylov Method**

Next we show that multi-point approximation can in some cases be an essential technique on some industrial cases. Specifically, multi-point expansion is critical to treat networks with certain wide distributions of time constants. Such networks are not reducible by first-order techniques such as TICER [8]. Still taking problem #3 as the test case, we plot the size of the TICER-“reduced” model vs. the error control tolerance used in TICER. The tolerance is defined at 1 GHz. It shows that to achieve 10% error, the size of “reduced” model is about 20,000. That is far too large for further analysis. By comparison, for 1% accuracy (corresponding to size 100k in the figure) at the same frequency, SIP without node retention only needs 2 expansion points, which means a model of size $16 \times 2 = 32$. This means that the SIP model much smaller than TICER model, while at the mean time much more accurate.

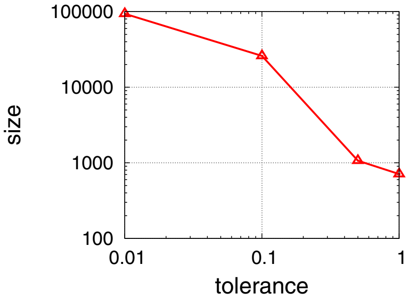

**Figure 2: Model size vs. error tolerance (at 1 GHz) for TICER Algorithm.**

Next we show that retaining complete moment matching can result in more robust reduction algorithms. In particular, single-point SIP retains two-moment matching in all cases, whereas TICER may not as it has different behavior when negative capacitors appear. Both algorithms used their error-controlled variants (nodes not eliminated when an error metric is violated). To show this difference we use two collections of RC circuits from industrial examples, denoted as *collection A* with 24792 small RC circuits, and *collection B*, with 35900 large RC circuits. A major concern in designing industrial-strength reduction codes is the elimination of “outliers”. It is easy to design codes that show good average error performance. It is harder to design codes that eliminate infrequent large errors. Therefore, much time is spent analyzing the “tail” of the error distribution. Table 2 shows the relative error point achieved for 98% of the circuits in each collection (basically the 98% point on the error CDF).

It should be noted that the compression ratio by both algorithms in this case was controlled to be the same, so we do not accidentally measure an error/compression tradeoff. From Table 2 it is clear that the number of outliers (cases with large error) for SIP is much smaller than that for TICER. Thus, for a fixed compression ratio, the error distribution for SIP is measurably “tighter.” We should be clear that we don’t believe this is necessarily due to the loss of moment matching, but with full moment matching, proper Taylor-series based error control is possible and this can have a significant impact on robustness.

**Table 2: 98% probability error point comparison**

| | collection A | collection B |
|---|---|---|
| TICER | 3.3e-2 | 2.3e-1 |
| SIP | 2.2e-3 | 6.5e-3 |

## 6.3 Accuracy for Asymmetric Systems

To demonstrate the application of SIP to more general systems, we apply it to an RLC network described by asymmetric system matrices. The starting system had 6.3M states and the number of ports is 16. Figure 3 shows error vs. frequency, indicating that multi-point SIP models can indeed be constructed and are effective.

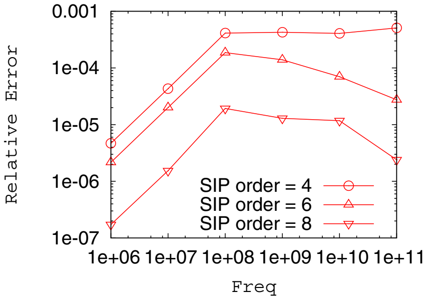

**Figure 3: Accuracy of SIP for Asymmetric Case.**

## 6.4 Numerical Sparsity of Reduced Model

Fig. 4 shows the sparsity pattern of the matrix resulting from the reduction of test case #4, after off-diagonal entries that are smaller than 1% of the corresponding diagonal terms were dropped. The overall relative error was also observed at about the 1% level. This plot shows that the models resulting from the SIP algorithm can be highly structured, and it is easy to do further sparsification. We examined PRIMA models for the same test cases here, and it was not possible to find exploitable sparsity structure in these models. While the SIP and PRIMA approaches are mathematically equivalent, from an input-output perspective, they operate in substantially different coordinate systems, which has practical implications for sparsity.

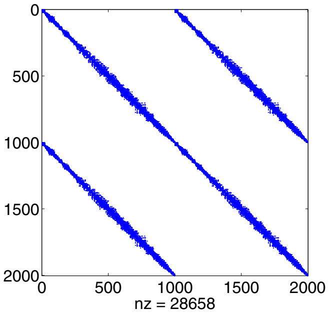

**Figure 4: Numerical Sparsity of SIP-Reduced Model.**

# 7. CONCLUSION

In this paper we propose a new model order reduction approach, sparse implicit projection (SIP). It possesses desirable features of existing projection methods, such as applicability to general systems, moment matching, high order approximation, passivity preservation via congruence, but it exploits the sparsity of the system matrices and thus gains considerable improvements over existing explicit projection algorithms. Our experiments demonstrate large effiency gains over existing approaches. The reduced models can also exhibit sparser structure compared to explicit projection. Furthermore, the implementation of SIP algorithm is straightforward and can be built directly on well developed codes for sparse matrix factorization. Analysis of the SIP algorithm leads to a unified mathematical framework that covers the theories of the PRIMA, rational Krylov method, SGE, and SIP.

# REFERENCES

[1] E. Grimme, “Krylov projection methods for model reduction,” Ph.D. dissertation, Coordinated-Science Laboratory, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1997.

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

[3] P. Feldmann and R. W. Freund, “Efficient linear circuit analysis by Padé approximation via the Lanczos process,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639–649, May 1995.

[4] B. Moore, “Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction,” *IEEE Transactions on Automatic Control*, vol. AC-26, no. 1, pp. 17–32, February 1981.

[5] J. Phillips, L. Daniel, and L. M. Silveira, “Guaranteed passive balancing transformations for model order reduction,” *IEEE Trans. Computer-Aided Design*, vol. 22, no. 8, pp. 1027–1041, August 2003.

[6] S. P. McCormick, “Modeling and simulation of VLSI interconnections with moments,” Ph.D. dissertation, Massachusetts Institute of Technology, 1989.

[7] A. van Genderen and N. van der Meijs, “Reduced RC models for IC interconnections with coupling capacitances,” in *IEEE European Conference on Design Automation*, Brussels, Belgium, March 1992, pp. 132–136.

[8] B. Sheehan, “TICER: Realizable reduction of extracted RC circuits,” *Proceedings of ICCAD*, pp. 200–203, 1999.

[9] P. Feldmann, “Model order reduction techniques for linear systems with large numbers of terminals,” *Proceedings of DATE*, pp. 944–947, 2004.

[10] P. Feldmann and F. Liu, “Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals,” *Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM International Conference on*, pp. 88–92, Nov. 2004.

[11] M. Chowdhury, C. Amin, Y. Ismail, C. Kashyap, and B. Krauter, “Realizable reduction of RLC circuits using node elimination,” *Circuits and Systems, 2003. ISCAS ’03. Proceedings of the 2003 International Symposium on*, vol. 3, pp. 494–497, 25-28 May 2003.

[12] J. M. S. Silva, J. R. Phillips, and L. M. Silveira, “Efficient model representation and analysis of power grids,” *Proceedings of DATE*, 2008.

[13] P. Li and W. Shi, “Model order reduction of linear networks with massive ports via frequency-dependent port packing,” in *DAC ’06: Proceedings of the 43rd annual conference on Design automation*. New York, NY, USA: ACM, 2006, pp. 267–272.

[14] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge: Cambridge University Press, 1985.

[15] T. A. Davis, *Direct Methods for Sparse Linear Systems*. SIAM, 2007.

[16] P. Elias and N. van der Meijs, “Extracting circuit models for large RC interconnections that are accurate up to a predefined signal frequency,” *Design Automation Conference Proceedings 1996, 33rd*, pp. 764–769, 3-7 Jun, 1996.

# APPENDIX

*Proof of Theorem 3.3.* Symbolic inversion of $G$ gives

$$G^{-1} = \begin{bmatrix} A^{-1} + A^{-1}BSB^TA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}B^TA^{-1} & S^{-1} \end{bmatrix}. \tag{42}$$

The zero-th order moment is

$$H^{(0)} = E^TG^{-1}E = S^{-1} = \hat{E}^T\hat{G}^{-1}\hat{E} = \hat{H}^{(0)}. \tag{43}$$

The first order moment

$$H^{(1)} = E^TG^{-1}CG^{-1}E. \tag{44}$$

Notice that $G^{-1}E = MS^{-1}$, we obtain

$$H^{(1)} = S^{-1}M^TCMS^{-1} = \hat{E}^T\hat{G}^{-1}\hat{C}\hat{G}^{-1}\hat{E} = \hat{H}^{(1)}. \tag{45}$$