

Fast Balanced Stochastic Truncation Via A Quadratic Extension of the Alternating Direction Implicit Iteration

Ngai Wong

Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
Email: nwong@eee.hku.hk

Venkataramanan Balakrishnan

School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
Email: ragu@ecn.purdue.edu

Abstract—Balanced truncation (BT) model order reduction (MOR) is known for its superior accuracy and computable error bounds. Balanced stochastic truncation (BST) is a particular BT procedure that provides a general, structure-independent MOR framework to preserve both passivity and stability of original models. Its application toward large scale systems, however, has been limited by the complexity of solving large size continuous time algebraic Riccati equations (CAREs). This paper introduces a novel quadratic extension of the alternating direction implicit (ADI) iteration, called *QADI*, that efficiently solves a CARE. A Cholesky factor variant of QADI, called *CFQADI*, further exploits low rank matrices and produces solution in factor form that greatly accelerates BST. Remarkable efficiency of the proposed BST/(CF)QADI integration is demonstrated with numerical examples.

I. INTRODUCTION

Simulation of interconnect models after parasitic extraction, despite being a big time sink, is a critical step to ensure functionality of a chip. The importance of this post-layout verification is ever-increasing as wire thinning and crosstalk have pushed the interconnect delay beyond gate delay. A major difficulty in this modeling exercise is the huge data volume and model sizes that forbid direct computer handling. Model order reduction (MOR) has subsequently become an integral operation in which high order models are considerably reduced to smaller ones without much sacrifice in accuracy [1]. Generally, there are two classes of MOR algorithms, namely, the traditional projection-based algorithms [2]–[4] and the recently promoted balanced truncation (BT) schemes [5]–[14]. Starting with a state space formulation of the initial model, the former approach projects the original system onto low rank subspaces that capture most state activities. The resultant reduced order models often show similar responses to the original systems, but there is neither direct error connection between the two nor optimality in the final models. Representative schemes in this class are PRIMA [2], Padé approximation via Lanczos (PVL) [3], and pole analysis via congruence transformations (PACT) [4]. PRIMA and PACT preserve passivity¹, if any, of the original systems. However, both algorithms assume special structures in the internal (passive) state space and these restrictions are not always feasible or practical [8].

Balanced truncation (BT) represents a distinct paradigm arising from rich theories in control. Instead of relying on projection, the “balancing” step aligns the states in descending relevance based on their involvement in input/output activities or energy transfer. The “truncation” step then throws away the least important states, resulting in a usually much smaller, (near-)optimal model of superior accuracy [8]. A bonus is that such schemes often come with a deterministic bound for the approximation error (e.g. [5], [7]).

¹Passivity means a system is dissipative. Non-passive reduced order models (of passive original systems) can result in erroneous global simulation [1], [2].

Algorithms among this class are optimal Hankel-norm approximation, standard BT, and the passivity-preserving balanced stochastic truncation (BST) [5]–[8]. An obstacle in these BT realizations is the expensive solution of large size matrix equations followed by large size matrix factorizations. For example, standard BT calls for the solution of a pair of dual Lyapunov equations (linear matrix equations), while BST comes with a pair of dual continuous time algebraic Riccati equations (CAREs) (quadratic matrix equations) whose solution can be computationally prohibitive [12]. To lower the cost of standard BT, recent results took advantage of the low rank input/output matrices pertinent to interconnect models and came up with the Cholesky factor (CF) standard BT variants of speed comparable to the projection-based methods [9]–[11], [14]. Standard BT, however, does not necessarily preserve passivity. BST guarantees stability and passivity and poses no special structural requirements on the internal state space [8], but is highly restricted by large size CAREs. Conventional techniques of solving a CARE include forming a Hamiltonian matrix and identifying its stable invariant subspace, or by Newton method that solves a Lyapunov equation in each iteration step [12], [15], [16]. The former way does not explicitly exploit low rank/sparse matrices and is generally slow, while the latter requires an appropriate initial condition which may not be readily available.

The contribution of this paper is the formulation of a quadratic alternating direction implicit (ADI) algorithm [17], called *QADI*, that efficiently solves a (large size) CARE. A CF variant of QADI, called *CFQADI*, further exploits low rank matrices and produces factor-form solution that accelerates BST. The BST/(CF)QADI approach constitutes a general framework for high speed, large scale, passivity-preserving MOR. This paper is organized as follows. Section II revises the BST settings and the ADI method for solving Lyapunov equations. Section III presents QADI, proves its convergence, and derives CFQADI which makes use of low rank matrices to provide speedup and memory savings. Remarkable efficiency of adapting (CF)QADI for BST realization is seen from the numerical examples in Section IV. Finally, Section VI draws the conclusion.

II. PRELIMINARIES

A. Basics of BST

Modeling of VLSI interconnects and pin packages generally involve passive RLC components. We consider a large scale RLC circuit represented in a state space format

$$\dot{x} = A_0x + B_0u \quad (1a)$$

$$y = C_0x + D_0u \quad (1b)$$

where $A_0 \in \mathbb{R}^{n \times n}$, $B_0 \in \mathbb{R}^{n \times m}$, $C_0 \in \mathbb{R}^{m \times n}$, $D_0 \in \mathbb{R}^{m \times m}$, B_0 , C_0 are generally of low ranks (i.e., $m \ll n$) and u , y are power-

conjugate². A_0 is stable or equivalently its spectrum is in the open left half plane, denoted by $\text{spec}(A_0) \subset \mathbb{C}^-$. Let $M > 0$ ($M \geq 0$) denote a positive definite (positive semidefinite) matrix M , we assume $D_0 + D_0^T > 0$. For RLC models in modified nodal analysis (MNA) format [18], we have $A_0 + A_0^T \leq 0$, $B_0 = C_0^T$, and $D_0 = 0$. Such MNA system can then be transformed into an equivalent form with $D_0 + D_0^T > 0$ [13]. Moreover, an RLC system in the *descriptor* format [1] with a singular E_0 before \dot{x} can be put into the standard form in (1) [11] which is therefore assumed without loss of generality. The positive real lemma [1] states that system (1) is passive if and only if there exists a $P (\in \mathbb{R}^{n \times n}) \geq 0$ satisfying the linear matrix inequality

$$\begin{bmatrix} A_0^T P + P A_0 & P B_0 - C_0^T \\ B_0^T P - C_0 & -(D_0 + D_0^T) \end{bmatrix} \leq 0. \quad (2)$$

Using Schur complement, (2) is equivalent to

$$A_0^T P + P A_0 + (P B_0 - C_0^T)(D_0 + D_0^T)^{-1}(B_0^T P - C_0) \leq 0. \quad (3)$$

The solution of (3) being an equality is a CARE. Taking the matrix root $LL^T = (D_0 + D_0^T)^{-1}$ and defining $B = B_0 L$, $C = L^T C_0$, and $A = A_0 - BC$, the CARE is rewritten as

$$A^T P + P A + P B B^T P + C^T C = 0. \quad (4)$$

By assumption we have the unique *stabilizing solution* $P \geq 0$ to (4) which satisfies $\text{spec}(A + B B^T P) \subset \mathbb{C}^-$. In BST, the stabilizing solutions P_{min} and Q_{min} respectively of two CAREs

$$A^T P + P A + P B B^T P + C^T C = 0 \quad (5a)$$

$$A Q + Q A^T + Q C^T C Q + B B^T = 0 \quad (5b)$$

are first computed. Let $Q_{min} = Y Y^T$, $P_{min} = Z Z^T$ be any Cholesky factorizations. Now perform the singular value decomposition (SVD)

$$Y^T Z = U \Sigma V^T \quad (6)$$

where $\Sigma \geq 0$ is an ‘‘economic’’ k -by- k ($k \leq n$) diagonal matrix with singular values in descending order. Suppose the singular values of Σ are

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \gg \sigma_{r+1} \geq \dots \geq \sigma_k. \quad (7)$$

Define I_m to be the identity matrix of dimension m , $0_{m \times n}$ be an $m \times n$ zero matrix, and

$$T_L = [I_r \quad 0_{r \times (k-r)}] \Sigma^{-\frac{1}{2}} V^T Z^T \quad (8a)$$

$$T_R = Y U \Sigma^{-\frac{1}{2}} [I_r \quad 0_{r \times (k-r)}]^T. \quad (8b)$$

The system $(T_L A_0 T_R, T_L B_0, C_0 T_R, D_0)$ is then the stochastically balanced and reduced model whose states are aligned in descending order of importance in the energy transfer [6], [8].

B. Basics of ADI

This section briefly revises the traditional *alternating direction implicit* (ADI) iteration [14], [17]. Key results necessary for understanding later sections are presented. In general, ADI solves the Lyapunov equation of the form

$$A^T W + W A + C^T C = 0 \quad (9)$$

²For every entry of u that is a node voltage (branch current), the corresponding component of y is a branch current (node voltage) so that $u^T y$ gives the instantaneous power delivered to the system.

where the matrix dimensions are consistent with those in (4). It is assumed A is stable so there exists a $W (\in \mathbb{R}^{n \times n}) \geq 0$ that solves (9). The basic ADI consists of two iterative steps:

$$(A^T + p_j I) W_{j-\frac{1}{2}}^T = -C^T C - W_{j-1}^T (A - p_j I) \quad (10a)$$

$$(A^T + p_j I) W_j = -C^T C - W_{j-\frac{1}{2}} (A - p_j I) \quad (10b)$$

where $W_0 = 0$ and the shift parameters $p_j \in \mathbb{C}^-$ ($j = 1, 2, \dots$) appear as real numbers or conjugate pairs. For compactness we define $S_j = (A + p_j I)^{-1}$ and $T_j = (A - p_j I)$. One useful fact is that for any integers m and n , the multiplication among S_m , T_n , and A are commutative, and similarly for S_m^T , T_n^T , and A^T . It can be verified that

$$W_j = - \sum_{i=1}^j 2p_i \left(\prod_{k=1}^{i-1} S_k^T T_k^T \right) S_i^T C^T C S_i \left(\prod_{k=1}^{i-1} T_k S_k \right). \quad (11)$$

In [14] it is shown that the ordering of p_j s in (11) is immaterial. Combining (9) and (10), we have

$$W - W_j = \left(\prod_{k=1}^j S_k^T T_k^T \right) W \left(\prod_{k=1}^j T_k S_k \right). \quad (12)$$

To achieve the best convergence in, say, N runs of (10), p_j s are chosen (or approximately chosen) according to the minimax problem

$$\min_{\{p_1, p_2, \dots, p_N\}} \left(\max_{\lambda_i \in \text{spec}(A)} \left| \prod_{j=1}^N \frac{p_j - \lambda_i}{p_j + \lambda_i} \right| \right). \quad (13)$$

III. QUADRATIC ADI

This section focuses on the following CARE

$$A^T X + X A + X B B^T X + C^T C = 0 \quad (14)$$

whose dimensions are consistent with those in (4). We assume a stabilizing solution $X \geq 0$ exists such that $\text{spec}(A + B B^T X) \subset \mathbb{C}^-$. Standard control theory also states that existence of this stabilizing X is equivalent to the condition $\sup \bar{\sigma}(C(j\omega - A)^{-1} B) < 1$, $\forall \omega \in \mathbb{R}$, where $\bar{\sigma}(\circ)$ denotes the maximum singular value [16]. The following second order generalization of ADI, called *quadratic ADI* or *QADI*, is proposed for solving (14):

$$(A^T + X_{j-1}^T B B^T + p_j I) X_{j-\frac{1}{2}}^T = -C^T C - X_{j-1}^T (A - p_j I) \quad (15a)$$

$$(A^T + X_{j-\frac{1}{2}} B B^T + p_j I) X_j = -C^T C - X_{j-\frac{1}{2}} (A - p_j I) \quad (15b)$$

where $X_0 = 0$ and $p_j \in \mathbb{C}^-$, $j = 1, 2, \dots$, are either real or conjugate pairs. Apparently, (15) reduces to (10) when $B = 0$. For ease of illustration we will assume, for the rest of the paper, all p_j s are negative real. However, all qualitative results hold for conjugate pairs if we combine two runs of (15) into one such that all quantities remain real. It will be shown that, as in ADI, X_j converges to X when j tends to infinity. One may also merge the two half-steps of (15) into one. Again, using the definitions $S_j = (A + p_j I)^{-1}$ and $T_j = (A - p_j I)$, it can be carefully shown that

$$X_j = M_{11} + M_{12} X_{j-1} (I - M_{22} X_{j-1})^{-1} M_{12}^T \quad (16)$$

where

$$M_{11} = -2p_j S_j^T C^T (I - C S_j B B^T S_j^T C^T)^{-1} C S_j \quad (17a)$$

$$M_{12} = I - 2p_j S_j^T (I - C^T C S_j B B^T S_j^T)^{-1} \quad (17b)$$

$$M_{22} = -2p_j S_j B (I - B^T S_j^T C^T C S_j B)^{-1} B^T S_j^T. \quad (17c)$$

The matrix inverses in (17a)-(17c) are well-defined because existence of a stabilizing solution guarantees $\bar{\sigma}(C S_j B) < 1$. Well-posedness

of the inverse in (16) will be shown in Section III-B. Also, it can be seen that a symmetric X_{j-1} implies a symmetric X_j . Since $X_0 = 0$, all X_j s are symmetric.

A. Cholesky Factor Variant

As in ADI, in the presence of low rank B and C , it is desirable for QADI to work with the Cholesky factor (CF) iterate Z_j where $X_j = Z_j Z_j^T$. Utilizing (16) and (17), we formulate a CF variant of QADI called *CFQADI*. In particular, setting $Z_0 = 0$, for $j = 1, 2, \dots$,

$$1. M_{11}^{\frac{1}{2}} = \sqrt{-2p_j S_j^T C^T (I - C S_j B B^T S_j^T C^T)^{-\frac{1}{2}}} \quad (18a)$$

$$2. M_{12} = I - 2p_j S_j^T (I - C^T C S_j B B^T S_j^T)^{-1} \quad (18b)$$

$$3. M_{22} = -2p_j S_j B (I - B^T S_j^T C^T C S_j B)^{-1} B^T S_j^T \quad (18c)$$

$$4. Z_j = [M_{11}^{\frac{1}{2}} \quad M_{12} Z_{j-1} (I - Z_{j-1}^T M_{22} Z_{j-1})^{-\frac{1}{2}}]. \quad (18d)$$

In each run of CFQADI (i.e., each j) the number of columns in Z_j grows by the number of rows in C . Low rank B and C also allow the use of matrix inversion lemma in (18a)-(18c) to reduce arithmetics. Iteration is terminated when the update in $\|Z_j Z_j^T\|$ is smaller than a preset tolerance. In summary, in case of low rank system matrices, CFQADI provides significant computational and memory savings (as only low rank factors are stored). Symmetry of X_j is perfectly preserved by reconstruction from Z_j . Moreover, the converged factor Z , where $X = Z Z^T$, can readily be adapted to the BST MOR process.

B. Well-Posedness and Convergence

Here we prove the well-posedness and convergence of the basic QADI in (16). The properties carry over to CFQADI since it is mathematically equivalent to QADI. Define $\tilde{A} = A + B B^T X$, and with the assumption of a stabilizing solution X to (14), we have $\text{spec}(\tilde{A}) \subset \mathbb{C}^-$. Let $\tilde{S}_j = (\tilde{A} + p_j I)^{-1}$ and $\tilde{T}_j = (\tilde{A} - p_j I)$. The key of the proof is to rewrite (15a) and (15b) by the knowledge of (14), namely,

$$X - X_{j-\frac{1}{2}} = -\tilde{T}_j^T (X - X_{j-1}) \left(I - \tilde{S}_j B B^T (X - X_{j-1}) \right)^{-1} \tilde{S}_j \quad (19a)$$

$$X - X_j = -\tilde{S}_j^T (X - X_{j-\frac{1}{2}}) \left(I - B B^T \tilde{S}_j^T (X - X_{j-\frac{1}{2}}) \right)^{-1} \tilde{T}_j. \quad (19b)$$

Substituting (19a) into (19b) we get

$$X - X_j = \tilde{S}_j^T \tilde{T}_j^T (X - X_{j-1}) \cdot \left(I - 2p_j \tilde{S}_j B B^T \tilde{S}_j^T (X - X_{j-1}) \right)^{-1} \tilde{T}_j \tilde{S}_j \quad (20)$$

which is equivalent to (16). Applying (20) recursively to itself and noting $X_0 = 0$, we get

$$X - X_j = \Pi_j^T X (I + \Omega_j X)^{-1} \Pi_j \quad (21)$$

where

$$\Pi_j = \left(\prod_{k=1}^j \tilde{T}_k \tilde{S}_k \right),$$

$$\Omega_j = - \sum_{i=1}^j 2p_i \left(\prod_{k=1}^{i-1} \tilde{S}_k \tilde{T}_k \right) \tilde{S}_i B B^T \tilde{S}_i^T \left(\prod_{k=1}^{i-1} \tilde{T}_k \tilde{S}_k \right).$$

Therefore, the QADI described by (16) is well-defined if and only if the inverse in (21) is well-defined for every j , $j = 1, 2, \dots$. An

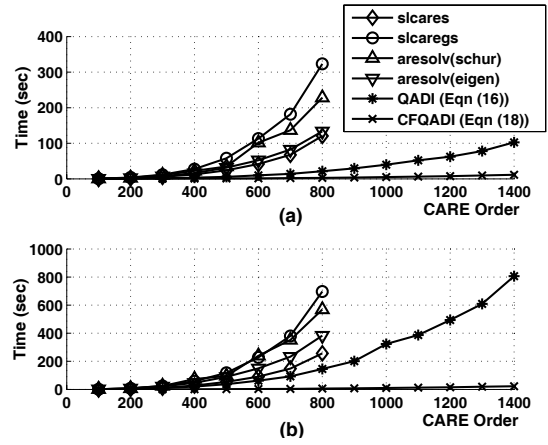


Fig. 1. CPU time for (a) solving one CARE; (b) doing BST with the solver.

important observation is that Ω_j is exactly the j th iterate of the ADI solution (c.f. (11)) to the Lyapunov equation

$$\tilde{A} \Omega + \Omega \tilde{A}^T + B B^T = 0 \quad (22)$$

so we have $\Omega \geq \Omega_j \geq 0$. The inverse $(I + \Omega_j X)^{-1}$ in (21) is therefore always valid as the product of two positive semidefinite matrices contains only non-negative eigenvalues. Lastly, convergence of QADI (and therefore that of CFQADI) can readily be deduced from (21), namely,

$$\|X - X_j\| \leq K_1 \|\Pi_j^T X \Pi_j\| \leq K_2 \left(\max_{\lambda_i \in \text{spec}(\tilde{A})} \left| \prod_j \frac{p_j - \lambda_i}{p_j + \lambda_i} \right| \right)^2 \quad (23)$$

where K_1 and K_2 are positive lumped constants. This shows that, as in ADI, QADI exhibits (super-)linear convergence. Also, the convergence rate depends on p_j s whose choice is analogous to the minimax problem in ADI (see (13)). The major difference is that the shift parameters, p_j s, should now be chosen with respect to the spectrum of \tilde{A} instead of that of A .

IV. NUMERICAL EXAMPLES

The QADI in (16) and CFQADI in (18) are coded in MATLAB m-files (text files) and executed, without compilation, in the MATLAB R14 environment. They are compared against the MATLAB subroutine *aresolv* with the *schur* and *eigen* options turned on successively. The former option implements the Schur vector method, while the latter uses the eigenvector method [19]. Two other solvers, *slcares* (Schur vector method) and *slcaregs* (generalized Schur vector method), are called from the SLICOT library [20]. These are prebuilt FORTRAN 77 subroutines written with numerically reliable and efficient algorithms. It should be noted that QADI and CFQADI are non-Hamiltonian solvers, while others, and in fact almost all industrial/commercial solvers, are Hamiltonian-based. The experiments were done on a 3GHz desktop with 1G RAM.

First, We first study the BST MOR of randomly generated single input single output (SISO) RLC systems of different orders. Fig. 1(a) plots the CPU time for solving a CARE against its order and Fig. 1(b) plots the total time for BST utilizing the particular solver. For fairness, solutions from QADI and CFQADI are computed to the same or better accuracy than those by other solvers. It is immediately seen that QADI and CFQADI exhibit superior speed and scalability over conventional solvers. Table I normalizes the CPU time of CFQADI and compares it against QADI and the fastest SLICOT and MATLAB

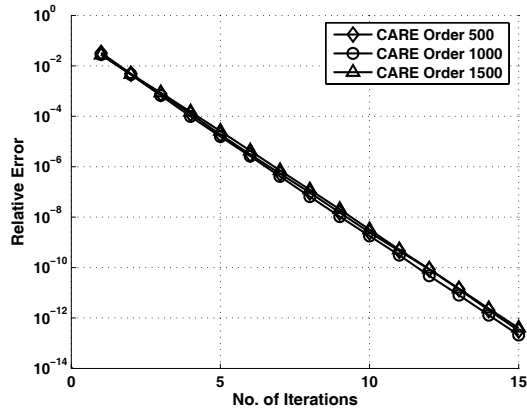


Fig. 2. Convergence of X_j to the stabilizing X at several CARE orders.

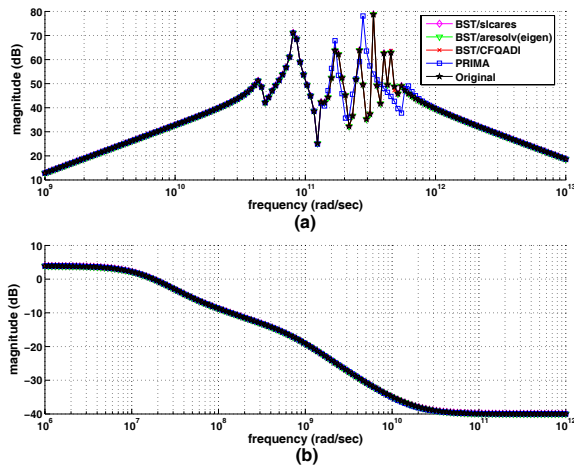


Fig. 3. Frequency responses of (a) the transmission line model (order=256) and the reduced models from various MOR schemes (order=35); (b) the spiral inductor model (order=500) and the reduced models from various MOR schemes (order=7).

subroutines, *slcares* and *aresolv(eigen)*, respectively, at several CARE orders. It is seen that QADI and CFQADI, as standalone CARE solvers, are much more efficient than others. In high order BST, BST/QADI is dominated by matrix factorizations and large scale SVD, while BST/CFQADI completely avoids these operations due to the readily available (low rank) factor solution. In [13], it is shown that BST time by Hamiltonian-based solvers can almost be halved by complete subspace separation, but with CFQADI it is possible to achieve more than double the efficiency (e.g., $> 40X$ at order=800 in this case). In our experiments, when the ranks of B and C are between 1 and 1/50 of the CARE order, the corresponding curves will go between those of QADI and CFQADI in Fig. 1. Specifically, the original QADI in (16) does not explicitly rely on low rank matrices, and is suitable when B and C are of high/full ranks wherein CFQADI does not bring about savings. However, when low rank B and C are present, CFQADI should always be used to achieve memory savings and faster computation. To see how QADI converges, Fig. 2 plots the metric $\|X_j - X\|_F / \|X\|_F$ ($\|\cdot\|_F$ being the Frobenius norm of a matrix) against the number of iterations at several CARE

TABLE I
CPU TIMES IN RANDOM SISO EXAMPLES WITH THAT OF CFQADI
NORMALIZED TO 1.

CARE Order	400	500	600	700	800
Solving one CARE (Fig. 1(a))					
CFQADI	1	1	1	1	1
QADI	5.12	5.75	6.47	6.82	7.39
slcares	19.12	24.32	27.42	33.01	41.76
aresolv(eigen)	25.03	31.98	34.63	40.26	46.23
Doing BST (Fig. 1(b))					
CFQADI	1	1	1	1	1
QADI	15.98	19.24	21.03	22.69	25.27
slcares	21.56	27.59	30.76	36.33	44.44
aresolv(eigen)	37.09	46.27	50.53	57.16	66.36

TABLE II
CPU TIMES FOR DIFFERENT MOR SCHEMES WITH THAT OF PRIMA
NORMALIZED TO 1.

	PRIMA	BST/ CFQADI	BST/ slcares	BST/ aresolv(eigen)
Fig. 3(a)	1	4.96	73.04	108.99
Fig. 3(b)	1	6.79	73.83	118.23

orders. From these straight-line curves, (super-)linear convergence of QADI is obvious, which carries over to the mathematically equivalent CFQADI.

Next, we try out BST/CFQADI in two real-life benchmarks. The first one is a discretized transmission line model of order 256 and the second one is an extracted on-chip spiral inductor model of order 500 [10], [14]. Fig. 3 shows the frequency responses of the original systems and reduced-order models from various MOR schemes. While the curves in Fig. 3(b) virtually overlap, the PRIMA curve in Fig. 3(a) exhibits a relatively larger error. This is not surprising as models produced by BST tend to have better global accuracy [13]. Table II displays the normalized CPU times in respective schemes. For BST/CFQADI, an adaptive stopping criterion similar to that in the standard BT counterpart [10] is used. From the table, the computational effort of BST/CFQADI is about 5 to 7 times higher than that of PRIMA, but is considerably less than those of standard implementations using conventional solvers. Moreover, BST/CFQADI avoids the selection of expansion points and final model order as in PRIMA (both of which require *a priori* knowledge of the original response).

Consequently, these results demonstrate the effectiveness of the new QADI and CFQADI schemes, and verify them to be powerful candidates in performing large scale BST.

V. REMARKS

- 1) QADI and CFQADI represent new ways of solving (large scale) CAREs and are simple to code. Both algorithms converge with $X_0 = 0$. CFQADI, to the authors' knowledge, is the first algorithm that directly computes the factor-form solution of a CARE through CF iterates (instead of the "indirect" CF solution from Newton method [9], [12]). The CF solution thus obtained is usually of low rank, say, in the order of tens, thereby avoiding two large size matrix factorizations and one large size SVD in the original BST process. The low rank factor also permits reduced memory requirement and scalable implementation.
- 2) Runtime of QADI or CFQADI is mainly determined by the number of shifts. The most expensive step is the matrix

inversion in finding S_j for each p_j , which takes roughly $3n^3$ flops in the general case when A is dense. If the number of shifts is N , the work of both algorithms is proportional $3Nn^3$. QADI has another $O(n^3)$ component proportional to the number of iterations, while other operations in CFQADI are all of $O(n^2)$ due to exploitation of low rank matrices. In short, work of QADI or CFQADI increases in a cubic manner, but much more slowly compared to conventional solvers. If inversion can be done in $O(n^2)$ work, e.g., when A is sparse or banded, then both algorithms will reduce to $O(n^2)$ algorithms.

- 3) For simplicity and demonstrative purpose, only a single shift is used in the numerical experiments, which is analogous to the *Smith method* as a special case of ADI [9], [12]. Specifically, using the results in Section III-B, we have chosen the shift $p = -\sqrt{|\lambda_{max}(\tilde{A})\lambda_{min}(\tilde{A})|}$ [12] where $\lambda_{max}(\circ)$ and $\lambda_{min}(\circ)$ denote the maximum- and minimum-magnitude eigenvalues. Although $\tilde{A} = A + BB^T X$ is self-referential to the stabilizing solution X , from standard CARE theory [16] it can be easily verified that $p = -\sqrt{|\lambda_{max}(H)\lambda_{min}(H)|}$ where

$$H = \begin{bmatrix} A & BB^T \\ -C^T C & -A^T \end{bmatrix}$$

is the Hamiltonian matrix associated with (14). The shift is then estimated through simple power iterations whose impact on the overall cost is negligible.

- 4) It is possible to incorporate (CF)QADI into two-level passive reduction algorithms such as PRIMA+BST/CFQADI (e.g., [8]). Referring to table II, the runtime, and in most cases also the accuracy of reduced models, of this hybrid approach would fall between that of PRIMA and BST/CFQADI. As a general guideline, projection-type MOR algorithms like PRIMA can be used to reduce model order from millions down to thousands, which is then followed by BST to compress the model order to hundreds or tens while guaranteeing passivity and global accuracy. It can be noted from table II that such approach is computationally prohibitive with conventional CARE solvers.

VI. CONCLUSION

This paper has presented a highly efficient implementation of balanced stochastic truncation (BST) based on a novel, quadratic extension of alternating direction implicit algorithm called QADI. Well-posedness and convergence of QADI have been analytically proven and characterized. In the context of BST, QADI facilitates a Cholesky factor variant called CFQADI. CFQADI exploits low rank matrices and avoids large scale matrix factorizations, thereby allowing significant memory savings and speed gain in handling large scale CAREs and BST. Numerical examples have demonstrated the remarkable efficiency of the BST/(CF)QADI approach over conventional realizations.

VII. ACKNOWLEDGMENT

This work was supported in part by the Hong Kong Research Grants Council, the University Research Committee of The University of Hong Kong, and the National Science Foundation of the United States of America under Grant No. ECS-0200320.

REFERENCES

- [1] Z. Bai, P. M. Dewilde, and R. W. Freund, "Reduced-order modeling," *Numerical Analysis Manuscript No. 02-4-13*, Bell Laboratories, Mar. 2002.

- [2] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer-Aided Design*, vol. 17, no. 8, pp. 645–654, Aug. 1998.
- [3] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, no. 5, pp. 639–649, May 1995.
- [4] K. J. Kerns and A. T. Yang, "Stable and efficient reduction of large, multiport RC networks by pole analysis via congruence transformations," *IEEE Trans. Computer-Aided Design*, vol. 16, no. 7, pp. 734–744, July 1997.
- [5] K. Glover, "All optimal Hankel-norm approximation of linear multivariable systems and their L^∞ -error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, June 1984.
- [6] M. Green, "Balanced stochastic realizations," *Linear Algebra Appl.*, vol. 98, pp. 211–247, 1988.
- [7] X. Chen and J. T. Wen, "Positive realness preserving model reduction with H_∞ norm error bounds," *IEEE Trans. Circuits Syst. I*, vol. 42, no. 1, pp. 23–29, Jan. 1995.
- [8] J. R. 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, Aug. 2003.
- [9] T. Penzl, "A cyclic low rank Smith method for large sparse Lyapunov equations with applications in model reduction and optimal control," *SIAM J. SCI. COMPUT.*, vol. 21, no. 4, pp. 1401–1418, 2000.
- [10] V. Balakrishnan, Q. Su, and C.-K. Koh, "Efficient balance-and-truncate model reduction for large scale systems," in *Proc. American Control Conf.*, June 2001, pp. 4746–4751.
- [11] Q. Su, V. Balakrishnan, and C.-K. Koh, "Efficient approximate balanced truncation of general large-scale RLC systems via Krylov methods," in *Proc. ASPDAC/Int. Conf. VLSI Design*, Jan. 2002, pp. 311–316.
- [12] N. Wong, V. Balakrishnan, C.-K. Koh, and T. S. Ng, "A fast Newton/Smith algorithm for solving algebraic Riccati equations and its application in model order reduction," in *Proc. IEEE Conf. Acoustics, Speech, and Signal Processing*, May 2004, pp. 53–56.
- [13] N. Wong, V. Balakrishnan, and C.-K. Koh, "Passivity-preserving model reduction via a computationally efficient project-and-balance scheme," in *Proc. IEEE Design, Automation Conf.*, June 2004, pp. 369–374.
- [14] J. Li and J. White, "Low-rank solution of Lyapunov equations," *SIAM Review*, vol. 46, no. 4, pp. 693–713, 2004.
- [15] V. Mehrmann, *The Autonomous Linear Quadratic Control Problem: Theory And Numerical Solution*, ser. No. 163 Lecture Notes in Control and Information Sciences. Berlin, Heidelberg: Springer-Verlag, Nov. 1991.
- [16] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.
- [17] A. Lu and E. L. Wachspress, "Solution of Lyapunov equations by alternating direction implicit iteration," *Computers Math. Appl.*, vol. 21, no. 9, pp. 43–58, 1991.
- [18] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Kluwer Academic Publishers, July 1993.
- [19] I. W. F. Arnold and A. J. Laub, "Generalized eigenproblem algorithms and software for algebraic Riccati equations," *Proc. IEEE*, vol. 72, no. 12, pp. 1746–1754, Dec. 1984.
- [20] P. Benner, V. Mehrmann, V. Sima, S. Van-Huffel, and A. Varga, "SLICOT—a subroutine library in systems and control theory," *Appl. and Comput. Contr., Signals, and Circuits*, vol. 1, pp. 499–539, 1999.