Multi-Shift Quadratic Alternating Direction Implicit Iteration for High-Speed Positive-Real Balanced Truncation

N. Wong
Department of Electrical and Electronic Engg
The University of Hong Kong, Hong Kong
nwong@eee.hku.hk

V. Balakrishnan
School of Electrical and Computer Engg
Purdue University, IN, USA
ragu@ecn.purdue.edu

ABSTRACT

This paper presents a multi-shift generalization of the recently proposed quadratic alternating direction implicit (QADI) iteration. QADI and its Cholesky Factor (CF) variant, CFQADI, have been shown to be efficient ways of solving the large-scale algebraic Riccati equations (AREs) required in positive-real balanced truncation (PRBT). However, only their single-shift implementations have been considered so far. Using linear fractional transformation (LFT), we present elegant multi-shift extensions of both QADI and CFQADI, thereby enabling even faster and more accurate PRBT.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development—*modeling methodologies*; J.6 [**Computer-Aided Engineering**]: —*computer-aided design (CAD)*

General Terms

Algorithms

Keywords

Model order reduction, positive real, balanced truncation, ADI

1. INTRODUCTION

Balanced truncation (BT) model order reduction (MOR) schemes offer superior accuracy in reduced-order models [1], but are largely restricted by the complexity of solving high-order matrix equations. A particular BT approach known as positive-real BT (PRBT)[1] preserves passivity, exhibits high modeling accuracy, and poses no special structural constraint on the initial state space. However, PRBT requires solving a pair of algebraic Riccati equations (AREs). These are quadratic matrix equations whose solution involves intensive computation, thereby forming the bottleneck in large-scale PRBT. Various methods have been proposed to alleviate the workload in solving AREs, e.g., [3, 4]. They are mainly based on the Newton's method or alike for iteratively and progressively solving an ARE, wherein a linear matrix equation, called the Lyapunov equation, is solved in each Newton step. In this context, the alternating direction implicit (ADI) iteration [5] provides a versatile way of solving Lyapunov equations. Its major merit comes from a Cholesky Factor variant, called CF-ADI [5], that directly iterates on the matrix square roots, thus taking full advantage of the low-rank input/output matrices pertinent to circuit models. Nonetheless, these methods are *indirect* in the sense that each iteration still targets at solving a Lyapunov equation, rather than directly at the ARE. Recently, a quadratic extension of ADI, called QADI, together with its CF variant, called CFQADI, have been proposed to solve an ARE iteratively without setting up Lyapunov intermediates [2]. Both algorithms enjoy simple coding, and the same numerical properties and complexities as in ADI and CF-ADI. Like CF-ADI, CFQADI directly computes factor-iterates of an ARE solution, thus exploiting the low-rank matrices in VLSI models and enabling better scalability and further memory savings. However, only single-shift QADI and CFQADI have been considered so far. This paper generalizes single-shift (CF)QADI to its faster converging multi-shift version (CF)QADI(l) where l stands for the number of shifts. Linear fractional transformation (LFT) has been used to achieve the otherwise intractable derivations. Numerical examples confirm the efficiency of the practically attractive CFQADI(l).

[1]PRBT is also called positive-real truncated balanced realization (PR-TBR) [1]. In [2], balanced stochastic truncation (BST) is used to denote PRBT. In any case, these refer to the passivity-preserving BT procedure wherein a pair of dual AREs are solved.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

2. PRBT AND (CF)QADI

Interconnect and package modelings generally make use of passive RLC components. Consider a large-scale RLC network cast into a state space

$$\dot{x} = A_0 x + B_0 u \ , \ y = C_0 x + D_0 u \tag{1}$$

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-conjugate [2]. A_0 is stable, i.e., its spectrum is in the open left half plane, denoted by $spec(A_0) \subset \mathbb{C}_-$. Let $M > 0$ ($M \geq 0$) denote a positive definite (positive semidefinite) matrix M, we assume without loss of generality that $D_0 + D_0^T > 0$, otherwise the reduction techniques in [6] is used to achieve this. Next, define the matrix root $LL^T = (D_0 + D_0^T)^{-1}$, $B = B_0 L$, $C = L^T C_0$, and $A = A_0 - BC$. In PRBT, the unique stabilizing solutions, $X \geq 0$ and $Q \geq 0$, respectively, to two AREs

$$A^T X + XA + XBB^T X + C^T C = 0 \tag{2a}$$

$$AQ + QA^T + QC^T CQ + BB^T = 0 \tag{2b}$$

are solved such that $spec(A + BB^T X) \subset \mathbb{C}_-$ and $spec(A^T + C^T CQ) \subset \mathbb{C}_-$. The matrix square roots (or Cholesky factors)

$Q = YY^T$ and $X = ZZ^T$ are then computed to form the projection matrices for reducing the original model (1) while preserving passivity and stability. The reader is referred to [2] for details.

QADI and CFQADI aim at solving the stabilizing solution of an ARE of the form (2a) [2] (application to (2b) is trivial). A stable A-matrix is assumed which is the case for passive circuits. QADI consists of two half-steps:

$$(A^T + X_{j-1}^T BB^T + p_j I)X_{j-\frac{1}{2}}^T = -C^T C - X_{j-1}^T (A - p_j I) \quad (3a)$$

$$(A^T + X_{j-\frac{1}{2}} BB^T + p_j I)X_j = -C^T C - X_{j-\frac{1}{2}} (A - p_j I) \quad (3b)$$

where $X_0 = 0$ and $p_j \in \mathbb{C}_-$ ($j = 1, 2, \cdots$) are either real or conjugate pairs. A key insight that greatly simplifies analysis of QADI is by recognizing (3) as linear fractional transformations (LFTs) [7]. Denoting $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$, we define the (lower) LFT, denoted by $F_l(P, \Delta)$, as the transfer matrix from w to z, i.e., $z = F_l(P, \Delta)w$, which is also a mapping of Δ:

$$F_l(P, \Delta) = P_{11} + P_{12}\Delta \left(I - P_{22}\Delta\right)^{-1} P_{21}$$

where the matrix dimensions are implicitly assumed compatible and the matrix inverse valid. Defining $S_j = (A + p_j I)^{-1}$ and $T_j = (A - p_j I)$, it can be shown that

$$X_{j-\frac{1}{2}} = F_l(P, X_{j-1}) \quad \text{and} \quad X_j = F_l(P^T, X_{j-\frac{1}{2}}), \qquad (4)$$

$$P = \begin{bmatrix} -C^T C S_j & -T_j^T + C^T C S_j BB^T \\ S_j & -S_j BB^T \end{bmatrix}.$$

The chief property of LFTs is that their interconnection is still an LFT, e.g., the nested connection $X_j = F_l(P^T, F_l(P, X_{j-1}))$, called the *Redheffer Star Product* [7], is an LFT. This enables the combination of the two half-steps in (3) into one, namely,

$$X_j = M_{11} + M_{12} X_{j-1} (I - M_{22} X_{j-1})^{-1} M_{12}^T \qquad (5a)$$

$$\text{where } M_{11} = -2p_j S_j^T C^T (I - CS_j BB^T S_j^T C^T)^{-1} CS_j, \quad (5b)$$

$$M_{12} = I - 2p_j S_j^T (I - C^T CS_j BB^T S_j^T)^{-1}, \qquad (5c)$$

$$M_{22} = -2p_j S_j B (I - B^T S_j^T C^T CS_j B)^{-1} B^T S_j^T. \quad (5d)$$

As illustrated in [2], well-posedness of the matrix inverses in (5) can be verified by the bounded real lemma. Starting with $X_0 = 0$, all X_js are symmetric and their convergence to the stabilizing X is superlinear. Define $\tilde{A} = A + BB^T X$, similar to ADI [3, 5, 8], fast convergence in QADI can be achieved by choosing the l shifts, p_js, according to the minimax problem

$$\min_{\{p_1, p_2, \cdots, p_l\}} \left(\max_{\lambda_i \in spec(\tilde{A})} \left| \prod_{j=1}^{l} \frac{p_j - \lambda_i}{p_j + \lambda_i} \right| \right) \qquad (6)$$

where the spectrum of $\tilde{A}$ is known prior to solving X [7]:

$$spec(\tilde{A}) = spec\left(\begin{bmatrix} A & BB^T \\ -C^T C & -A^T \end{bmatrix} \right) \cap \mathbb{C}_-. \qquad (7)$$

When low-rank B and C are present, it is desirable for QADI to work with the CF iterate Z_j where $X_j = Z_j Z_j^T$. Utilizing (5), the CF variant of QADI, called CFQADI, can be derived. Specifically, setting $Z_0 = 0$, for $j = 1, 2, \cdots$,

$$Z_j = \begin{bmatrix} M_{11}^{\frac{1}{2}} & M_{12} Z_{j-1} (I - Z_{j-1}^T M_{22} Z_{j-1})^{-\frac{1}{2}} \end{bmatrix}. \qquad (8)$$

In each iteration of CFQADI, the number of columns in Z_j grows by the number of rows in C. When low-rank system matrices are present, CFQADI is always preferred as it provides computational and memory savings since only low-rank factors are stored. Symmetry of X_j is perfectly preserved by reconstruction from Z_j. The converged factor Z, where $X = ZZ^T$, can then be readily adapted to PRBT to elude large-scale matrix factorizations [2].

3. MULTI-SHIFT QADI AND CFQADI

Referring to (6), so far only single-shift ($l = 1$) QADI and CFQADI have been considered [2]. This section presents the multi-shift QADI and CFQADI, denoted by QADI(l) and CFQADI(l) where l is the number of shifts (p_js). Analogous to ADI and for computational reasons, all arithmetics in QADI(l) and CFQADI(l) must be confined to the real domain. This entails a two-step QADI and CFQADI such that when the shifts, p_j and p_{j+1}, are conjugate pairs or both real, a real X_{j+1} can be obtained directly from a real X_{j-1} skipping the possibly complex X_j. To begin with, a further LFT breakdown of (5) shows that X_j, being the transfer matrix from w to z, namely $z = X_j w$, can be written as

$$\begin{array}{r} z \leftarrow \\ y_1 \leftarrow \\ y_2 \leftarrow \\ y_3 \leftarrow \end{array} \left[\begin{array}{cc|cc} 0 & 0 & -\sqrt{-2p_j}S_j^T & -S_j^T T_j^T \\ 0 & 0 & S_j^T & \sqrt{-2p_j}S_j^T \\ \hline -\sqrt{-2p_j}S_j & S_j & 0 & 0 \\ -S_j T_j & \sqrt{-2p_j}S_j & 0 & 0 \end{array}\right] \begin{array}{l} \leftarrow w \\ \leftarrow u_1 \\ \leftarrow u_2 \\ \leftarrow u_3 \end{array}$$

$$\begin{array}{r} y_1 \rightarrow \\ y_2 \rightarrow \\ y_3 \rightarrow \end{array} \left[\begin{array}{c|c|c} BB^T & 0 & 0 \\ \hline 0 & C^T C & 0 \\ \hline 0 & 0 & Z_{j-1} Z_{j-1}^T \end{array}\right] \begin{array}{l} \rightarrow u_1 \\ \rightarrow u_2 \\ \rightarrow u_3 \end{array} \qquad (9)$$

where we have used connector symbols to avoid cumbersome connections. Using (9), we now formulate the two-step QADI. Equations (10)-(13), which are equivalent LFTs, detail the flow in deriving the two-step QADI. In these equations, we use indices $\{0, 1, 2\}$ to replace $\{j-1, j, j+1\}$ for compactness. Specifically, it is obvious from (13) that a pair of conjugate shifts always give rise to a real X_{j+1} provided X_{j-1} is real. Nonetheless, direct computation of the LFT in (13) is numerically unfavorable. This is because the matrix inverse in (13), when compared to the original single-step single-shift QADI or QADI(1), has at least double the order and this increase in computation, especially in large-scale problems, almost always outweighs the possibly faster convergence. In fact, the power of QADI(l) comes from its low-rank CFQADI(l) formulation, which in turn relies on the availability of a two-step CFQADI. To derive this, from (13) we have

$$\begin{aligned} X_{j+1} &= S_1^T S_2^T F_l(F_l(M, \hat{C}^T \hat{C}), \hat{B}\hat{B}^T) S_2 S_1 \\ &= S_1^T S_2^T F_l(\hat{M}\hat{C}^T \hat{C}\hat{M}^T, \hat{B}\hat{B}^T) S_2 S_1. \end{aligned} \qquad (14)$$

Partitioning $\hat{M}\hat{C}^T$ and denoting the first n rows as Θ_1 and the rest as Θ_2, it can be shown that

$$Z_{j+1} = S_1^T S_2^T \begin{bmatrix} \Theta_1 & \Theta_1 \Theta_2^T \hat{B} (I - \hat{B}^T \Theta_2 \Theta_2^T \hat{B})^{-1/2} \end{bmatrix} \qquad (15)$$

where $X_{j+1} = Z_{j+1} Z_{j+1}^T$. Noting the low-rank nature of $\hat{B}$ and $\hat{C}$, Z_{j+1} serves as a low-rank iterate through which a perfectly symmetric X_{j+1} can be reconstructed. The most expensive step in implementing CFQADI(l) is the matrix inversions in S_j corresponding to each shift p_j, which in general takes $O(n^3)$ work (so CFQADI(l) is in general an $O(n^3)$ algorithm). However, in the case that A possesses special structures, such as banded or sparse, CFQADI(l) can be reduced to $O(n^2)$ complexity. QADI(l) represents a theoretical tool and is not numerically favorable when compared to QADI(1). In practice, the speed gain of the multi-shift approach comes from CFQADI(l). Because CFQADI(l) is a superset of CFQADI(1) in [2], it possesses all merits of the latter. These include, in terms of PRBT, the avoidance of two large-size matrix factorizations and one large-size singular value decomposition [2].

Equations (10)-(13) are equivalent LFTs featuring a two-step QADI development. Quantities in (13) are real.

$$
\begin{array}{r}
z \leftarrow \\ z_1 \leftarrow \\ z_2 \leftarrow \\ z_3 \leftarrow
\end{array}
\left[\begin{array}{cc|cc}
0 & 0 & -\sqrt{-2p_2}S_2^T & -S_2^T T_2^T \\
0 & 0 & S_2^T & \sqrt{-2p_2}S_2^T \\
\hline
-\sqrt{-2p_2}S_2 & S_2 & 0 & 0 \\
-S_2T_2 & \sqrt{-2p_2}S_2 & 0 & 0
\end{array}\right]
\begin{array}{l}
\leftarrow w \\ \leftarrow w_1 \\ \leftarrow w_2 \\ \leftarrow w_3
\end{array}
\quad
\begin{array}{r}
w_3 \leftarrow \\ y_1 \leftarrow \\ y_2 \leftarrow \\ y_3 \leftarrow
\end{array}
\left[\begin{array}{cc|cc}
0 & 0 & -\sqrt{-2p_1}S_1^T & -S_1^T T_1^T \\
0 & 0 & S_1^T & \sqrt{-2p_1}S_1^T \\
\hline
-\sqrt{-2p_1}S_1 & S_1 & 0 & 0 \\
-S_1T_1 & \sqrt{-2p_1}S_1 & 0 & 0
\end{array}\right]
\begin{array}{l}
\leftarrow z_3 \\ \leftarrow u_1 \\ \leftarrow u_2 \\ \leftarrow u_3
\end{array}
$$

$$
\begin{array}{r}
z_1 \rightarrow \\ z_2 \rightarrow
\end{array}
\left[\begin{array}{c|c}
BB^T & 0 \\
\hline
0 & C^TC
\end{array}\right]
\begin{array}{l}
\rightarrow w_1 \\ \rightarrow w_2
\end{array}
\quad
\begin{array}{r}
y_1 \rightarrow \\ y_2 \rightarrow \\ y_3 \rightarrow
\end{array}
\left[\begin{array}{c|c|c}
BB^T & 0 & 0 \\
\hline
0 & C^TC & 0 \\
\hline
0 & 0 & Z_0Z_0^T
\end{array}\right]
\begin{array}{l}
\rightarrow u_1 \\ \rightarrow u_2 \\ \rightarrow u_3
\end{array}
\tag{10}
$$

$$
\begin{array}{r}
z \leftarrow \\ z_1 \leftarrow \\ y_1 \leftarrow \\ z_2 \leftarrow \\ y_2 \leftarrow \\ y_3 \leftarrow
\end{array}
\left[\begin{array}{ccc|ccc}
0 & 0 & 0 & \sqrt{-2p_2}S_2^T & \sqrt{-2p_1}S_1^T S_2^T T_2^T & S_1^T T_1^T S_2^T T_2^T \\
0 & 0 & 0 & S_2^T & 2\sqrt{p_1p_2}S_1^T S_2^T & \sqrt{-2p_2}S_1^T T_1^T S_2^T \\
0 & 0 & 0 & 0 & S_1^T & \sqrt{-2p_1}S_1^T \\
\hline
\sqrt{-2p_2}S_2 & S_2 & 0 & 0 & 0 & 0 \\
\sqrt{-2p_1}S_1S_2T_2 & 2\sqrt{p_1p_2}S_1S_2 & S_1 & 0 & 0 & 0 \\
S_1T_1S_2T_2 & \sqrt{-2p_2}S_1T_1S_2 & \sqrt{-2p_1}S_1 & 0 & 0 & 0
\end{array}\right]
\begin{array}{l}
\leftarrow w \\ \leftarrow w_1 \\ \leftarrow u_1 \\ \leftarrow w_2 \\ \leftarrow u_2 \\ \leftarrow u_3
\end{array}
$$

$$
\begin{array}{r}
z_1 \rightarrow \\ y_1 \rightarrow
\end{array}
\left[\begin{array}{c|c}
BB^T & 0 \\
\hline
0 & BB^T
\end{array}\right]
\begin{array}{l}
\rightarrow w_1 \\ \rightarrow u_1
\end{array}
$$

$$
\begin{array}{r}
z_2 \rightarrow \\ y_2 \rightarrow \\ y_3 \rightarrow
\end{array}
\left[\begin{array}{c|c|c}
C^TC & 0 & 0 \\
\hline
0 & C^TC & 0 \\
\hline
0 & 0 & Z_0Z_0^T
\end{array}\right]
\begin{array}{l}
\rightarrow w_2 \\ \rightarrow u_2 \\ \rightarrow u_3
\end{array}
\tag{11}
$$

$$
\begin{array}{r}
z \leftarrow [S_1^T S_2^T] \leftarrow \\ z_1 \leftarrow \\ y_1 \leftarrow \\ z_2 \leftarrow \\ y_2 \leftarrow \\ y_3 \leftarrow
\end{array}
\left[\begin{array}{ccc|ccc}
0 & 0 & 0 & \sqrt{-2p_2}S_1^{-T} & \sqrt{-2p_1}T_2^T & T_1^T T_2^T \\
0 & 0 & 0 & S_1^{-T} & 2\sqrt{p_1p_2}I & \sqrt{-2p_2}T_1^T \\
0 & 0 & 0 & 0 & S_2^{-T} & \sqrt{-2p_1}S_2^{-T} \\
\hline
\sqrt{-2p_2}S_1^{-1} & S_1^{-1} & 0 & 0 & 0 & 0 \\
\sqrt{-2p_1}T_2 & 2\sqrt{p_1p_2}I & S_2^{-1} & 0 & 0 & 0 \\
T_2T_1 & \sqrt{-2p_2}T_1 & \sqrt{-2p_1}S_2^{-1} & 0 & 0 & 0
\end{array}\right]
\begin{array}{l}
\leftarrow [S_2S_1] \leftarrow w \\ \leftarrow w_1 \\ \leftarrow u_1 \\ \leftarrow w_2 \\ \leftarrow u_2 \\ \leftarrow u_3
\end{array}
$$

$$
\begin{array}{r}
z_1 \rightarrow \\ y_1 \rightarrow
\end{array}
\left[\begin{array}{c|c}
S_2S_1BB^TS_1^TS_2^T & 0 \\
\hline
0 & S_2S_1BB^TS_1^TS_2^T
\end{array}\right]
\begin{array}{l}
\rightarrow w_1 \\ \rightarrow u_1
\end{array}
$$

$$
\begin{array}{r}
z_2 \rightarrow \\ y_2 \rightarrow \\ y_3 \rightarrow
\end{array}
\left[\begin{array}{c|c|c}
C^TC & 0 & 0 \\
\hline
0 & C^TC & 0 \\
\hline
0 & 0 & Z_0Z_0^T
\end{array}\right]
\begin{array}{l}
\rightarrow w_2 \\ \rightarrow u_2 \\ \rightarrow u_3
\end{array}
\tag{12}
$$

$$
\begin{array}{r}
z \leftarrow [S_1^T S_2^T] \leftarrow \\ z_1 \leftarrow \\ y_1 \leftarrow \\ z_2 \leftarrow \\ y_2 \leftarrow \\ y_3 \leftarrow
\end{array}
\underbrace{\left[\begin{array}{ccc|ccc}
0 & 0 & 0 & \sqrt{-2p_s}A^T & \sqrt{-2p_sp_m}I & T_1^T T_2^T \\
0 & 0 & 0 & A^T & \sqrt{p_m}I & \sqrt{-2p_s}A^T \\
0 & 0 & 0 & \sqrt{p_m}I & -A^T - p_sI & \sqrt{-2p_sp_m}I \\
\hline
\sqrt{-2p_s}A & A & \sqrt{p_m}I & 0 & 0 & 0 \\
\sqrt{-2p_sp_m}I & \sqrt{p_m}I & -A - p_sI & 0 & 0 & 0 \\
T_2T_1 & \sqrt{-2p_s}A & \sqrt{-2p_sp_m}I & 0 & 0 & 0
\end{array}\right]}
\begin{array}{l}
\leftarrow [S_2S_1] \leftarrow w \\ \leftarrow w_1 \\ \leftarrow u_1 \\ \leftarrow w_2 \\ \leftarrow u_2 \\ \leftarrow u_3
\end{array}
$$

$$
:=M:=\left[\begin{array}{c|c}
0 & \hat{M} \\
\hline
\hat{M}^T & 0
\end{array}\right], \text{ and } \; {\scriptstyle p_s=p_1+p_2 \;,\; p_m=p_1p_2}
$$

$$
\begin{array}{r}
z_1 \rightarrow \\ y_1 \rightarrow
\end{array}
\underbrace{\left[\begin{array}{c|c}
S_2S_1BB^TS_1^TS_2^T & 0 \\
\hline
0 & S_2S_1BB^TS_1^TS_2^T
\end{array}\right]}_{:=\left[\hat{B}\hat{B}^T\right]}
\begin{array}{l}
\rightarrow w_1 \\ \rightarrow u_1
\end{array}
$$

$$
\begin{array}{r}
z_2 \rightarrow \\ y_2 \rightarrow \\ y_3 \rightarrow
\end{array}
\underbrace{\left[\begin{array}{c|c|c}
C^TC & 0 & 0 \\
\hline
0 & C^TC & 0 \\
\hline
0 & 0 & Z_0Z_0^T
\end{array}\right]}_{:=\left[\hat{C}^T\hat{C}\right]}
\begin{array}{l}
\rightarrow w_2 \\ \rightarrow u_2 \\ \rightarrow u_3
\end{array}
\tag{13}
$$

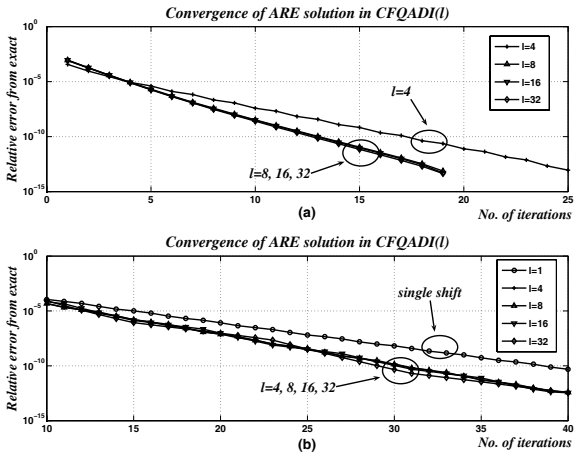

Figure 1: AREs from (a) transmission line; (b) RLC ladder.

Table 1: CPU times (sec) for the examples in Fig. 1.

	l=4	l=8	l=16	l=32
Fig. 1(a) (One ARE†)	14.22	15.64	6.82	7.68
Fig. 1(a) (PRBT)	28.48	28.95	12.68	14.44
Fig. 1(b) (One ARE‡)	32.47	33.30	36.27	37.11

Computed to the † same accuracy ‡ best possible accuracy

4. NUMERICAL EXAMPLES

Performance of CFQADI(1) in real-life benchmarks and comparison of it against other MOR schemes have been discussed in [2]. Here, we focus on the effect of shifts on the behavior of CFQADI(l) in (15). The algorithm was coded in MATLAB m-files and run in the MATLAB R14(SP2) environment. A desktop with a 3GHz CPU and 3G RAM was used. In our examples, the heuristics in [3] for shift selection and their cyclic application (i.e., $p_j = p_{j+kl}$ for $k = 1, 2, \cdots$) are employed. However, depending on specific problem structures, other shift selection schemes (e.g., [5, 8]) may be used.

The first example is a discretized transmission line model of order 300 with some dominant oscillatory modes. Fig. 1 (a) shows the convergence of X_j to X in solving one of the AREs in PRBT. It is seen that an increase in the number of shifts accelerates the convergence significantly, but the gain drops off with further increase in shifts. The first row of Table 1 shows the times to obtain the same order of accuracy in solving one ARE. Obviously, despite the increased number of matrix inversions in computing S_js, a net speed gain is warranted at a larger l owing to faster convergence. The second row lists the times for the whole PRBT process, which are essentially double the times in the first row. This is so because low-rank iterates from CFQADI(l) renders negligible computation in the subsequent matrix factorizations in PRBT. These PRBT times, as noted in [2], are about $5 \sim 10$ times slower than PRIMA [9], but more than 10 times faster than traditional PRBT. The slower time than PRIMA is then again offset by the superior accuracy in the reduced models. The second example, shown in Fig. 1 (b), stems from an order-500 RLC ladder used in interconnect simulation. The same observations can be drawn. The third row in Table 1 shows the times spent under different number of shifts where we allow CFQADI(l) to run till there is no further improvement in solution accuracy. Though more shifts are being used, the increase in computation is relatively slight.

Several remarks are drawn. In many cases, a small number of shifts (e.g., $8 \sim 32$) are enough to accelerate convergence. Further increase in shifts may not be advantageous, as has been observed in the ADI case [3]. The effect of multiple shifts is highly dependent on the model nature. As a rule of thumb, in heavily damped systems, a small number of shifts will suffice, while more shifts are needed in lightly damped systems. Also, for smaller size problems (say, a few hundreds or less), CFQADI(1) has the advantage of lower algorithmic complexity. While for larger size problems, CFQADI(l) (say, $l = 32$) may be chosen for an overall speed gain.

5. CONCLUSION

This paper has presented a multi-shift extension to the quadratic ADI called QADI(l) where l is the number of shifts. For high-speed positive-real balanced truncation (PRBT), a Cholesky Factor variant, known as CFQADI(l), has been developed. The core in CFQADI(l), namely, a real-domain two-step CFQADI, have been elegantly derived using linear fractional transformation (LFT). Numerical examples have shown that CFQADI(l) exhibits faster convergence, better numerical accuracy, and better or comparable complexity to its single-shift counterpart.

6. ACKNOWLEDGMENT

This work is supported in part by the Hong Kong Research Grants Council under Project HKU 7173/04E, in part by 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.

7. REFERENCES

[1] 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.

[2] N. Wong and V. Balakrishnan, "Fast balanced stochastic truncation via a quadratic extension of the alternating direction implicit iteration," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 2005, pp. 801–805.

[3] 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.

[4] P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí, "Efficient numerical algorithms for balanced stochastic truncation," *Int. J. Appl. Math. Comp. Sci.*, vol. 11, pp. 1123–1150, 2001.

[5] J. Li and J. White, "Low-rank solution of Lyapunov equations," *SIAM Review*, vol. 46, no. 4, pp. 693–713, 2004.

[6] H. Weiss, Q. Wang, and J. L. Speyer, "System characterization of positive real conditions," *IEEE Trans. Automat. Contr.*, vol. 39, no. 3, pp. 540–544, Mar. 1994.

[7] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[8] E. Wachspress, "The ADI minimax problem for complex spectra," *Appl. Math Lett.*, vol. 1, pp. 311–314, 1988.

[9] 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.