

Approximate Implicit Subspace Iteration with Alternating Directions for LTI system Model Reduction[†]

Yunkai Zhou¹, D. C. Sorensen²

¹ *Department of Mathematics, Southern Methodist University, Dallas, TX 75275-0156, USA.*
yzhou@smu.edu.

² *Department of Computational and Applied Mathematics; Rice University; Houston, Texas, 77005-1892.*
USA. sorensen@caam.rice.edu.

SUMMARY

We propose an Approximate Implicit Subspace Iteration with Alternating Directions (AISAD) framework for Linear Time Invariant (LTI) system model reduction. Within this framework, dominant eigensubspaces of the product of the system Gramians are approximated directly. This has advantage over approaches that consider the system Gramians separately. We construct two methods within the framework, one uses the QR updates, the other uses the SVD updates. Numerical results show the efficiency of the proposed methods. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: Implicit subspace iteration; dominant eigensubspace; Lyapunov equations; projected matrix equation; balanced model reduction

1. Introduction

We consider the following linear time invariant (LTI) system

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t), \end{cases} \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{q \times n}$; A is stable, i.e., eigenvalues of A all lie in the left half plane. LTI systems of this form arise frequently in different branches of engineering ([1, 36] and references therein). In many applications, such as circuit simulation, or time dependent PDE control problems, the state dimension n is quite large, while the number of inputs p and the number of outputs q usually satisfy $p, q \ll n$. Closely related to the LTI system (1) are two Lyapunov equations:

$$AP + PA^T + BB^T = 0, \quad (2)$$

$$A^T Q + QA + C^T C = 0. \quad (3)$$

[†]This work was supported in part by the NSF through Grants CCR-9988393, CCR-0306503 and ACI-0325081.

The solutions P and Q are called system *controllability Gramian* and *observability Gramian* respectively. It is well known that P , Q are symmetric positive semi-definite [20].

The important but also common feature $p, q \ll n$ of the LTI system (1) usually leads to the rapid decay rate for the eigenvalues of P and Q and also the rapid decay rate of the system Hankel singular values [2, 26, 33]. This implies that the original system can be approximated well by a reduced system of possibly much lower order.

This paper is concerned with providing a model order reduction technique for balanced truncation originally proposed by Moore [24]. The approach given here provides an approximate balancing transformation that is suitable for large scale problems. There are several approaches to model reduction of LTI systems including moment matching via Krylov techniques and Gramian based balanced reduction. Two major advantages of balanced truncation over moment matching methods (see for example [10, 1]) are the stability of the reduced system and the global error bound [11, 7, 8]. Most of these approaches approximate the dominant Cholesky factors (equivalently, dominant eigensubspaces) of P and Q separately [15, 18, 19, 22, 23, 25, 27], then combining the two Cholesky factors based on the algorithm proposed in [21] to compute the reduced model.

It is shown in [29] that the essential projection bases for LTI system model reduction are the bases of the dominant eigensubspaces of PQ . For dense problems, this reduction can be computed through Cholesky factors of P and Q . In the large scale setting, approximate dominant low rank Cholesky factor are used. One drawback is that one of the two Gramians may have a much slower eigenvalue decay rate than the other while the product PQ actually has a rapid decay rate. Another problem associated with the dominant Cholesky factor approaches is that when only partial Cholesky factors are used, the reduced model is not guaranteed to be balanced (note that the Algorithm in [21] requires full Cholesky factors to obtain the final truncated balanced transformation. When only dominant Cholesky factors are available, the reduced model may not be balanced), hence the important global error bound may not be rigorously attained. The approach presented here produces orthogonal basis sets for the dominant subspaces of PQ and QP , but as with the approximate Cholesky methods, it can only produce approximate balancing.

Methods based upon the *Cross Gramian* have been developed. The Cross Gramian approach [9, 32] has a considerable advantage over other approaches for symmetric systems since it reduces two Lyapunov equations to one Sylvester equation. Hence it almost halves the computational cost for model reduction. However, the Cross Gramian cannot always be defined for a general non-square system (i.e., $p \neq q$); moreover, one crucial property of the Cross Gramian, namely $\mathcal{X}^2 = PQ$ where \mathcal{X} is the solution of the Sylvester equation $A\mathcal{X} + \mathcal{X}A + BC = 0$, may not hold for square nonsymmetric LTI systems. The computational cost of the currently available methods needed to symmetrize* the system is more than solving one Lyapunov equation, which means for nonsymmetric systems the Cross Gramian approach does not offer advantages over approaches that handle two Lyapunov equations.

In [13], a very nice way to approximate the dominant eigensubspaces of PQ directly is proposed. But at each iteration step the method requires integrating four differential systems associated with the original LTI system (two for each Gramian), which may not be efficient

* Equivalent to computing a nonsingular symmetrizer J s.t. $J = J^T$, $AJ = JA^T$.

when n is large, especially when the differential systems are stiff.

Inspired by the ideas in [13], we found ways to approximate the dominant eigensubspace of PQ via projected matrix equations instead of numerical integration. This makes our approaches more viable for large scale LTI systems.

The organization of this paper is as follows: we first propose the general AISIAD (Approximate Implicit Subspace Iteration with Alternating Directions) framework for model reduction. Then we proceed to construct two new approaches within the AISIAD framework which approximate the principal eigensubspaces of PQ directly.

The model reduction steps within the AISIAD framework is based on the dominant eigensubspaces Algorithm in [29]. The essential difference between our framework and [29] is that we do not compute either P or Q , and no product PQ is computed explicitly. Hence our approaches are more suitable for large scale LTI systems. They also inherit the advantages of dominant eigensubspaces approaches over the dominant Cholesky factors approaches.

In this paper, we adopt the MATLAB notations: $qr(M,0)$ and $svd(M,0)$ to denote respectively the thin QR and thin SVD decompositions of any matrix M .

2. Algorithm derivation: the general AISIAD framework

In [2, 33], it is shown that the eigenvalues of P and Q usually decay rapidly, as do the system Hankel singular values, i.e., P and Q can be well approximated by low rank matrices. We expect the subspace iteration (also called block power method) to be efficient in computing the dominant eigenspaces. But since P and Q are not directly available, we propose *implicit subspace iteration* to approximate the dominant eigensubspaces of P and Q . For the model reduction of LTI system (1), as proved in [29], the essential projection bases are the bases of the dominant left and right eigensubspaces of PQ . We exploit the *Alternating Direction* technique similar to the one in [13] to achieve the matrix multiplication PQ implicitly and approximate the dominant left and right eigensubspaces of PQ .

The Lyapunov equations (2) and (3) are treated simultaneously. Given an orthogonal basis $V_i \in \mathbb{R}^{n \times k}$, applying subspace iteration to compute the dimension k dominant eigensubspace of P requires one to compute PV_i at the next step. Since P is unknown, this multiplication can not be performed directly. Note that if we obtain a matrix equation which contains PV_i as the solution, then solving the equation we actually perform the multiplication *implicitly*.

To get this desired equation, we multiply equation (2) from the right by V_i :

$$APV_i + PA^T V_i + BB^T V_i = 0, \quad (4)$$

let $M_i = P(I - V_i V_i^T)A^T V_i$, then (4) becomes

$$APV_i + PV_i V_i^T A^T V_i + M_i + BB^T V_i = 0.$$

Now, let $H_i = V_i^T A V_i$, if there is a way to estimate $\tilde{M}_i \approx M_i$ without using P , then we can solve for X_i from

$$AX_i + X_i H_i^T + BB^T V_i + \tilde{M}_i = 0,$$

in this case $X_i \approx PV_i$.

The *alternating direction* may be introduced as follows to compute the left and right dominant eigensubspaces of PQ simultaneously: Perform QR decomposition of X_i ,

$$W_i S_i = qr(X_i, 0), \quad (5)$$

then we get another orthogonal projection basis W_i . Project the other Lyapunov equation (3) from the right by W_i , we get

$$A^T QW_i + QW_i F_i + C^T CW_i + N_i = 0,$$

where $N_i = Q(I - W_i W_i^T)AW_i$, $F_i = W_i^T AW_i$. Again estimate N_i by \tilde{N}_i , solve for Y_i from

$$A^T Y_i + Y_i F_i + C^T CW_i + \tilde{N}_i = 0,$$

we get $Y_i \approx QW_i$. Perform QR decomposition of Y_i

$$V_{i+1} R_{i+1} = qr(Y_i, 0), \quad (6)$$

we get the next projection basis V_{i+1} . From (5) and (6) we see that in the convergent case (here the subscripts are omitted),

$$\begin{aligned} PV &= X = WS, \\ QW &= Y = VR. \end{aligned}$$

The above equalities lead to

$$\begin{aligned} PQW &= PVR = WSR, \\ QPV &= QWS = VRS. \end{aligned}$$

Hence $\text{Range}(W)$ contains the right dominant eigensubspace of PQ and $\text{Range}(V)$ contains the right dominant eigensubspace of QP . Note $(QP)^T = PQ$ since P and Q are symmetric and hence the columns of V also span the left dominant eigensubspace of PQ .

The above derivation may be summarized in Framework 2.1. Once the dominant eigensubspaces of PQ are available, the reduced model can be obtained in the same way as proposed in [29]; this is implemented as the steps 4–6 in Framework 2.1.

The convergence criterion is based on canonical angles [12] between the previous basis and the current basis, this corresponds to computing the SVD of $V_{i-1}^T V_i$ and $W_{i-1}^T W_i$. The user will specify the maximum number of iterations allowed.

For the choice of the initial V_1 , in practice one often uses the orthonormal basis of $\mathcal{K}(A, B, k_0) = [B, AB, \dots, A^{k_0} B]$ (with k_0 a given integer). This is influenced by the explicit form of the solution P as in [20]. Other initial basis can also be used, for example, if solving with A is inexpensive, then one could use the basis of $\mathcal{K}(A^{-1}, B, k_0)$ as the initial V_1 . One could also combine the basis of certain Krylov and rational Krylov subspaces. Random initial vectors may also be used, but, as expected for subspace iteration, better initial vectors lead to faster convergence.

The most expensive computation for Framework 2.1 is in solving Sylvester equations of form

$$AX + XH + M = 0. \quad (7)$$

Framework 2.1. AISIAD approach for LTI model reduction

1. Select k ; generate an initial orthogonal matrix $V_1 \in \mathbb{R}^{n \times k}$, $i \leftarrow 1$
2. Iterate until convergence
 - (a) Solve $AX_i + X_i H_i^T + \hat{M}_i = 0$,
where $H_i = V_i^T A V_i$, $\hat{M}_i = \tilde{M}_i + B B^T V_i$
 - (b) $[W_i, S_i] = qr(X_i, 0)$
 - (c) Solve $A^T Y_i + Y_i F_i + \hat{N}_i = 0$,
where $F_i = W_i^T A W_i$, $\hat{N}_i = \tilde{N}_i + C^T C W_i$
 - (d) $[V_{i+1}, R_{i+1}] = qr(Y_i, 0)$
 - (e) $i \leftarrow i + 1$
3. $V_L \leftarrow V_{i+1}$, $W_R \leftarrow W_i$
4. $[U_e, \Sigma, V_e] = svd(V_L^T * W_R)$
5. $S_L \leftarrow V_L U_e \Sigma^{-1/2}$, $S_R \leftarrow W_R V_e \Sigma^{-1/2}$
6. The reduced system is: $A_b = S_L^T A S_R$, $B_b = S_L^T B$, $C_b = C S_R$.

A special structure of equation (7) is that H is much smaller in size than A , this structure can be exploited. Actually (7) may be solved by partial Schur decomposition via the *implicit restarted Arnoldi method* [31, 17, 32],

$$\begin{bmatrix} A & M \\ 0 & -H \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} R.$$

It can be shown that V_2 is nonsingular if and only if the eigenvalues of R and $-H$ are the same. One may choose the implicit restarting criteria to guarantee this condition. In this case, it is straightforward to show that the solution of (7) may be obtained from the partial Schur vectors: $X = V_1 V_2^{-1}$. Since only the orthogonal basis of the subspace spanned by X is required, one only needs to use V_1 and avoids V_2^{-1} .

Alternatively, methods that mainly exploit matrix-vector products for solving Sylvester equations can be used to solve (7), e.g., [16, 5, 4]. Methods based on Bartels-Stewart [3] but tailored for (7) can also be used, see e.g., [34]. This approach has the advantage that both iterative methods and sparse factorization methods may be applied to solve the final equations of form $(A + \mu_i I)x = b$, where μ_i 's are the eigenvalues of H .

A more general framework for applying AISIAD approach for model reduction is as follows. One only needs to change respectively the 2(a) and 2(c) steps in Framework 2.1 into:

- 2(a) Compute $X_i = P V_i$ implicitly,
- 2(c) Compute $Y_i = Q W_i$ implicitly.

Hence any approach that achieves the above multiplications implicitly can be adapted into the AISIAD framework for model reduction. This includes the numerical integration approach in [13].

One fundamental difficulty associated with Framework 2.1 is in estimating \tilde{M}_i and \tilde{N}_i of the two projection error terms

$$M_i = P(I - V_i V_i^T) A^T V_i, \quad N_i = Q(I - W_i W_i^T) A W_i, \quad (8)$$

since P and Q are unknown.

In [34], low rank solutions of the Lyapunov equations (2) and (3) are used to replace the unknown P and Q in (8). This results in a highly accurate model reduction scheme that can effectively reduce dimension of a range of difficult realistic models. Numerical results in [34] show that these models are not as effectively reduced by other model reduction schemes. The approximate solutions to (2) and (3) can be computed using sparse Lyapunov solvers as in [27, 22, 23, 14, 28].

In the following, we present two approaches that do not include the error terms in (8). The first one is based on direct QR updates, the second one is based on SVD updates. Both approaches usually provide satisfactory numerical results. However, in [34] examples are given where ignoring the error terms is problematic. A nice modification is proposed in [34] to approximate these terms and it is demonstrated that the inclusion of these approximate error terms gives superior results on these difficult examples. In the modifications proposed in [34], the Lyapunov equations are first solved approximately, and the approximate solutions are used to update the error terms in (8). Thus additional computational cost is required to attain the higher accuracy.

3. AISIAD with direct QR and SVD updates

3.1. AISIAD with direct QR updates

The approach proposed in this subsection is rather straightforward. Because the projection errors (8) are difficult to estimate, we neglect these error terms and proceed the algorithm using direct QR updates. This is described in Algorithm 3.1.

The underlying idea of Algorithm 3.1 is based on the following observations. If P and Q are in fact low rank so that $P = V_i \hat{P} V_i^T$ and $Q = W_i \hat{Q} W_i^T$ can be obtained as the solutions \hat{P} and \hat{Q} from the two-side projected equations:

$$\begin{aligned} H_i \hat{P} + \hat{P} H_i^T + V_i^T B B^T V_i &= 0, & \text{where } H_i &= V_i^T A V_i, \\ F_i^T \hat{Q} + \hat{Q} F_i + W_i^T C^T C W_i &= 0, & \text{where } F_i &= W_i^T A W_i, \end{aligned}$$

then the M_i and N_i in (8) satisfy $M_i = N_i = 0$.

If there is rapid decay in the eigenvalues of P and Q or in the Hankel singular values (square roots of the eigenvalues of PQ), then one expects M_i and N_i to be small in norm. However, this is predicated upon having determined a subspace dimension large enough to capture all of the dominant modes, leaving only the small Hankel singular values. Noting that the solutions

Algorithm 3.1. *AISIAD-QR*

1. Select k ; generate an initial orthogonal matrix: $V_1 \in \mathbb{R}^{n \times k}$, $i \leftarrow 1$
2. Iterate until convergence
 - (a) Solve $AX_i + X_i H_i^T + \hat{M}_i = 0$,
where $H_i = V_i^T A V_i$, $\hat{M}_i = B B^T V_i$
 - (b) $[W_i, S_i] = qr(X_i, 0)$
 - (c) Solve $A^T Y_i + Y_i F_i + \hat{N}_i = 0$,
where $F_i = W_i^T A W_i$, $\hat{N}_i = C^T C W_i$
 - (d) $[V_{i+1}, R_{i+1}] = qr(Y_i, 0)$
 - (e) $i \leftarrow i + 1$
3. $V_L \leftarrow V_{i+1}$, $W_R \leftarrow W_i$
4. $[U_e, \Sigma, V_e] = svd(V_L^T * W_R)$
5. $S_L \leftarrow V_L U_e \Sigma^{-1/2}$, $S_R \leftarrow W_R V_e \Sigma^{-1/2}$
6. The reduced system is: $A_b = S_L^T A S_R$, $B_b = S_L^T B$, $C_b = C S_R$

of the one-sided projected equations in step 2(a) and 2(c) satisfy the following relation:

$$\|X_i - P V_i\| = \left\| \int_0^\infty e^{A\tau} M_i e^{H_i^T \tau} d\tau \right\|,$$

$$\|Y_i - Q W_i\| = \left\| \int_0^\infty e^{A^T \tau} N_i e^{F_i \tau} d\tau \right\|.$$

one might find the norm difference $\|X_i - P V_i\|$ and $\|Y_i - Q W_i\|$ to be small even though the norms of M_i , N_i are not small. However, no explicit error bounds based upon this observation have been derived.

Algorithm 3.1 works well for all the models we have, including models not satisfying the passivity condition $A + A^T < 0$. For these non-passive models, the algorithm proposed in [15] is reported to work poorly. This may be attributed to the fact that combining the two Gramians via the alternating direction technique has advantage over approaches that consider the Gramians separately.

3.2. AISIAD with SVD updates

The approach proposed in this subsection does not include the projection error terms either, but we utilize SVD deflation techniques instead of the direct QR updates. Although SVD is more expensive than QR, it leads to two advantages. The first one is that the dimension of projection basis need not be fixed, it can be determined by a given tolerance. Moreover, the maximum subspace dimension can be easily fixed by providing an integer k_{max} .

The second advantage, which is more interesting, is the existence of abundant choices in augmenting the projection basis by “preconditioned vectors”. The essential idea closely resembles the “preconditioning” techniques used in subspace methods for solving linear equations or eigenvalue problems, e.g., [6, 30, 35].

The method is described in Algorithm 3.2.

Algorithm 3.2. *AISIAD-SVD*

1. Given a tolerance tol and two integers k and k_{max} , generate an orthogonal matrix $V_1 \in \mathbb{R}^{n \times k}$, $i \leftarrow 1$
2. Iterate until convergence
 - (a) Solve $AX_i + X_i H_i^T + BB^T V_i = 0$, where $H_i = V_i^T A V_i$
 - (b) Compute “preconditioned augmentation” T_{ix}
 - (c) $[U_{tmp}, S_i, V_{tmp}] = svd([X_i, T_{ix}], 0)$
 - (d) Let k be the largest number of diagonal elements of S_i s.t. $\frac{S_i(k,k)}{S_i(1,1)} \geq tol$; if $k > k_{max}$, set $k = k_{max}$
 - (e) $W_i \leftarrow U_{tmp}(:, 1 : k)$
 - (f) Solve $A^T Y_i + Y_i F_i + C^T C W_i = 0$, where $F_i = W_i^T A W_i$
 - (g) Compute “preconditioned augmentation” T_{iy}
 - (h) $[U_{tmp}, R_{i+1}, V_{tmp}] = svd([Y_i, T_{iy}], 0)$
 - (i) Let k be the largest number of diagonal elements of R_{i+1} s.t. $\frac{R_{i+1}(k,k)}{R_{i+1}(1,1)} \geq tol$; if $k > k_{max}$, set $k = k_{max}$
 - (j) $V_{i+1} \leftarrow U_{tmp}(:, 1 : k)$
 - (k) $i \leftarrow i + 1$
3. $V_L \leftarrow V_{i+1}$, $W_R \leftarrow W_i$
4. $[U_e, \Sigma, V_e] = svd(V_L^T * W_R)$
5. $S_L \leftarrow V_L U_e \Sigma^{-1/2}$, $S_R \leftarrow W_R V_e \Sigma^{-1/2}$
6. The reduced system is: $A_b = S_L^T A S_R$, $B_b = S_L^T B$, $C_b = C S_R$

Several choices for the augmentation vectors T_{ix} and T_{iy} at the i -th iteration are as follows:

C1: Set $T_{1x} = A \setminus B$, $T_{ix} = A \setminus T_{(i-1)x}$; $T_{1y} = A^T \setminus C^T$, $T_{iy} = A^T \setminus T_{(i-1)y}$.

C2: If a shift μ_i is available at the i -th iteration, then set
 $T_{ix} = (A - \mu_i I) \setminus X_i$, $T_{iy} = (A^T - \mu_i I) \setminus Y_i$.

C3: Set $T_{ix} = A \setminus X_i$, $T_{iy} = A^T \setminus Y_i$; this corresponds to using 0 as the shift, and it leads to very high accuracy for the lower frequency region.

Note that the backslash “\” can be solved using either iterative methods or sparse factorizations. Note also that the system solves represented by “\” above may be replaced by multiplications “*”. The system solves can be considered as adding useful vectors from certain (implicit) *rational Krylov* subspaces to the projection basis; while the “*” may be regarded as enhancing certain *Krylov* components.

Many other alternatives can be applied to compute the augmentation vectors, e.g., using a combination of “\” and “*” such as $T_{ix} = [A \setminus X_i(:, 1 : p), A * X_i(:, 1 : p)]$; or, using different right hand sides than the ones in C1–C3; or, replacing A by certain preconditioners for the system solves.

One may also use no augmentation vectors. This corresponds to setting $T_{ix} = T_{iy} = \emptyset$ and $k = k_{max}$. This choice is observed to have similar behavior as AISIAD-QR. We mention that the choice $T_{ix} = V_i, T_{iy} = W_i$ in [36] is often inferior to AISIAD-QR, hence other augmentation vectors as proposed above should be preferred for the AISIAD-SVD method.

The goal of the “preconditioning” is to bring better vectors into the projection subspace through T_{ix} and T_{iy} , so that the algorithm has better chance to approximate the principal eigenspaces of PQ . With the “preconditioned augmentation”, AISIAD-SVD has more freedom adjusting the projecting subspaces than AISIAD-QR.

The “preconditioned augmentations” C1–C3 work well in practice. However, we mention that rigorous analysis for these heuristics still needs further study. For simplicity of discussions, we will mainly present numerical results for the choice C3.

4. Numerical experiments

In this section we report numerical results of Algorithm 3.1 and Algorithm 3.2 using augmentation scheme C3.

The numerical simulations are performed on a Dell PC (Intel Xeon processor, 3.0GHz CPU, 2GB RAM) running Linux, the MATLAB used is of version 7.0.1.

We first briefly discuss how to read the sigma plots. The *frequency response plot* is obtained by calling the MATLAB function `sigma`; `sigma` computes the singular values of the transfer function $H(j\omega)$, which is the frequency response $H(s) = C(sI - A)^{-1}B$ evaluated on the imaginary axis. Hence the plot shows the singular values of the matrix $C(j\omega I - A)^{-1}B$ as a function of the frequency ω . The *error plot* is the frequency response plot of the error system; it is the plot of singular values of the matrix $H(j\omega) - H_k(j\omega) := C(j\omega I - A)^{-1}B - C_b(j\omega I - A_b)^{-1}B_b$ as a function of ω . The frequency ω may be specified by the user or determined automatically by the algorithm inside the function `sigma`. The measure (dB) is defined as the $20 * \log_{10}(\cdot)$ of the singular value, i.e., if $dB(\sigma) = l$, then $\sigma = 10^{l/20}$.

The purpose of the numerical experiments is to verify the accuracy of the algorithms. The models used are relatively small, but they provide quite realistic tests for iterative methods used in model reduction. For these models we can easily compute the full L and U factors of P and Q , so that the standard balanced truncation via $L U$ factors can be used for comparison. The legend `btr-1u` in each plot shows reduction by this approach. Note that the final order- k model obtained by `btr-1u` is the best balanced order- k reduction, that can be obtained via

this standard approach.

In the header of each figure, the dimensions of the original system and the reduced system are listed as n and k respectively. All the methods reduce the full system to a system of the same order- k .

Figure 1 shows reduction of Penzl's constructed model [25], where

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & A_3 & \\ & & & A_4 \end{bmatrix}, \quad A_1 = \begin{bmatrix} -1 & 100 \\ -100 & -1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -1 & 200 \\ -200 & -1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} -1 & 400 \\ -400 & -1 \end{bmatrix}, \quad A_4 = \text{diag}(-1, -2, \dots, -1000), \quad B^T = C = \underbrace{[10 \ \dots \ 10]}_6, \underbrace{[1 \ \dots \ 1]}_{1000}.$$

For this model, the low-rank Smith method [25] computes the dominant Cholesky factor with 300 columns; the modified low rank Smith method [14] achieves a significant rank reduction, the dominant Cholesky factor has only 19 columns. Both methods used the six eigenvalues of A_1, A_2 , and A_3 as shifts. A final order-11 model is generated from the Cholesky factors by standard balanced truncation. Comparing with the sigma plot of the error system in the frequency interval $[10^1, 10^4]$ in [14], we see that AISIAD-QR produces an order-11 model of roughly the same accuracy as the two low rank Smith methods, while AISIAD-SVD has higher accuracy in the lower frequency range than **btr-lu** and other methods compared.

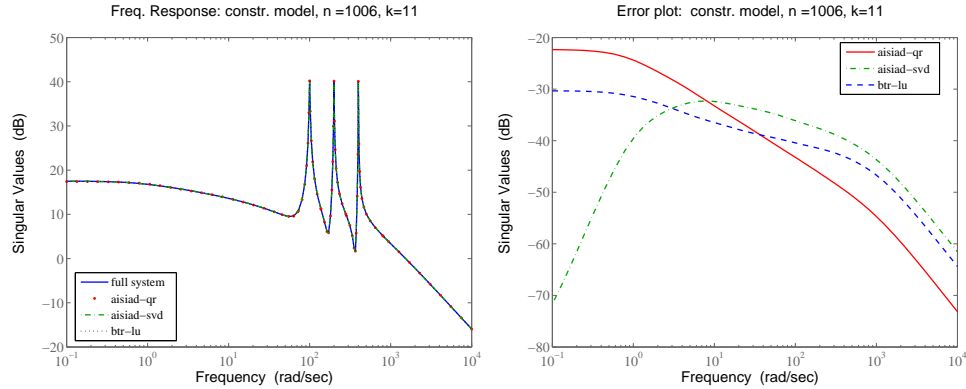


Figure 1. Penzl's constructed model [25]. ($A + A^T < 0$). **aisiad-qr** converged in 15 steps; **aisiad-svd** converged in 8 steps.

Figure 2 shows the reduction of a CD player model. Both AISIAD approaches generate an order-20 model comparable with the **btr-lu** method.

Figure 3 shows the reduction of an International Space Station 1r-c04 model. Again the AISIAD approaches generate an order-30 model of similar accuracy as **btr-lu** does.

Figure 4 shows the reduction of a clamped beam model. The AISIAD-SVD is much more accurate in the lower frequency range, but is less accurate in the higher frequency range.

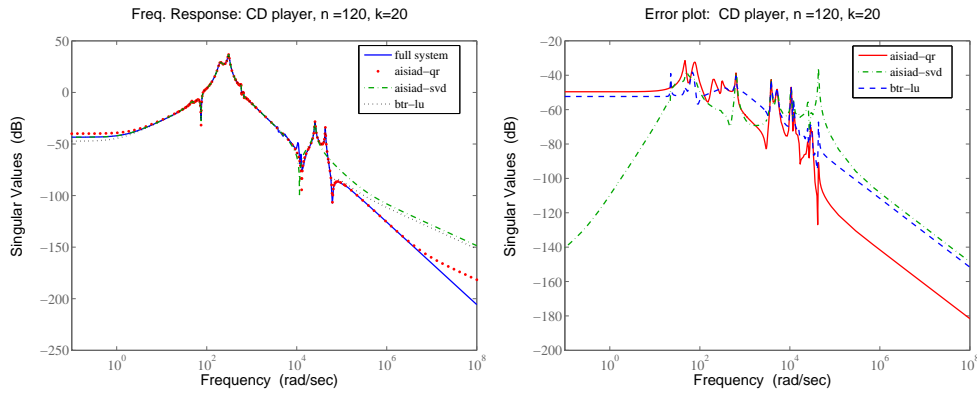


Figure 2. A simplified simulation model of CD player tracking mechanism. This model describes the dynamics between the lens actuator and the radial arm position of a portable compact disc player. ($A + A^T < 0$). `aisiad-qr` converged in 22 steps; `aisiad-svd` is run 35 steps.

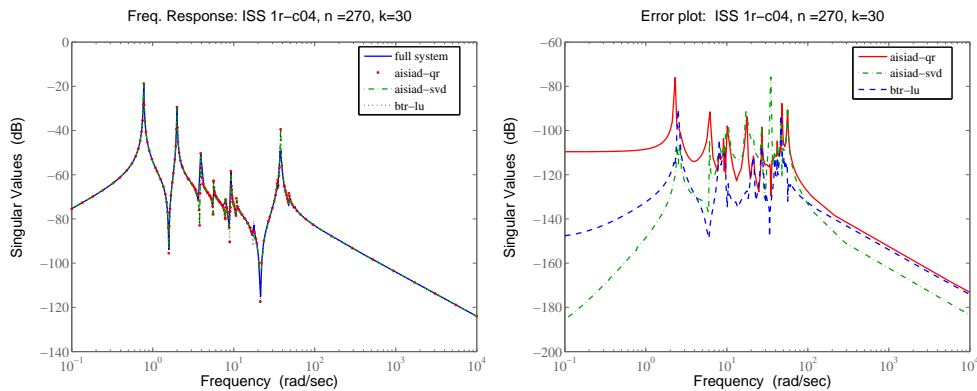


Figure 3. Finite element discretization of the flex modes of the Zvezda Service Module of the International Space Station 1r-c04. ($A + A^T \not< 0$). Both `aisiad-qr` and `aisiad-svd` are run 25 steps.

Figure 5 shows the reduction of a one-dimensional Laplacian. Figure 6 shows the reduction of a three-dimensional Laplacian. In both cases, the two AISIAD methods generate low order systems with high accuracy.

As seen from Figures 1—6, AISIAD-SVD with the augmentation scheme C3 often produces a much more accurate reduced model in the lower frequency region. This likely is related to the zero shift used in scheme C3. Further study will investigate how to generate or use better shifts in the “preconditioned augmentation” steps, so that the reduced model will have desired high accuracy in the frequency range that matters most.

Finally we make a comment on the memory requirement and complexity of the two AISIAD approaches. Because of the easily fixed maximum dimension, AISIAD is less memory demanding than the low-rank ADI or Smith approaches. However, since the one-side projected

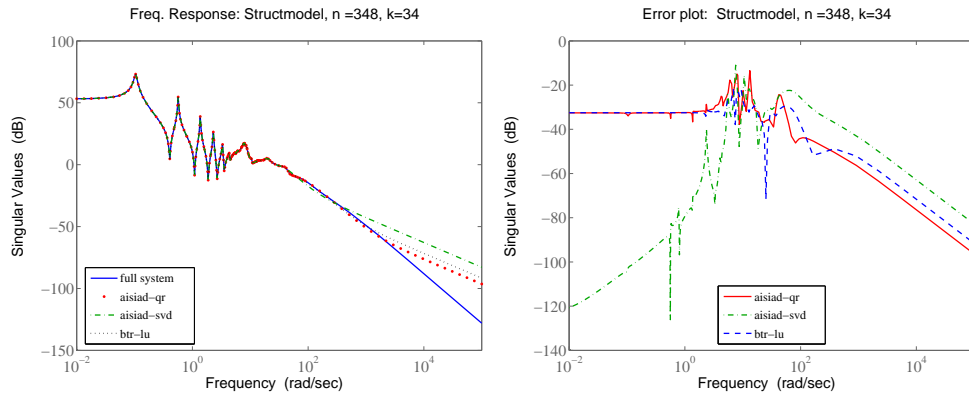


Figure 4. Finite element model of a clamped beam with a control force applied at the free end. ($A + A^T \not\prec 0$). `aisiad-qr` converged in 36 steps; `aisiad-svd` is run 45 steps.

Sylvester equations need to be solved several times during the iteration, AISIAD is generally more expensive than low-rank ADI and low-rank Smith methods.

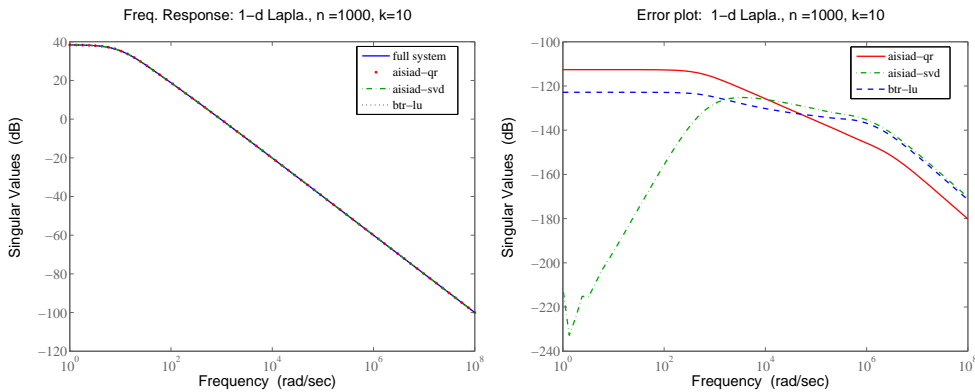


Figure 5. One-dimensional Laplacian discretized by finite difference, ($A + A^T \prec 0$). $B = C' = \text{ones}(n, 1)$. `aisiad-qr` converged in 21 steps; `aisiad-svd` converged in 7 steps.

5. Concluding remarks

A general AISIAD framework for LTI system model reduction was proposed. We constructed two approaches within this framework, one used direct QR updates, the other used SVD updates. By combining approximate projected-equation approach and the alternating direction technique, we constructed new ways in approximating the principal eigensubspaces of the product PQ of the two system Gramians. The combination has advantage over traditional

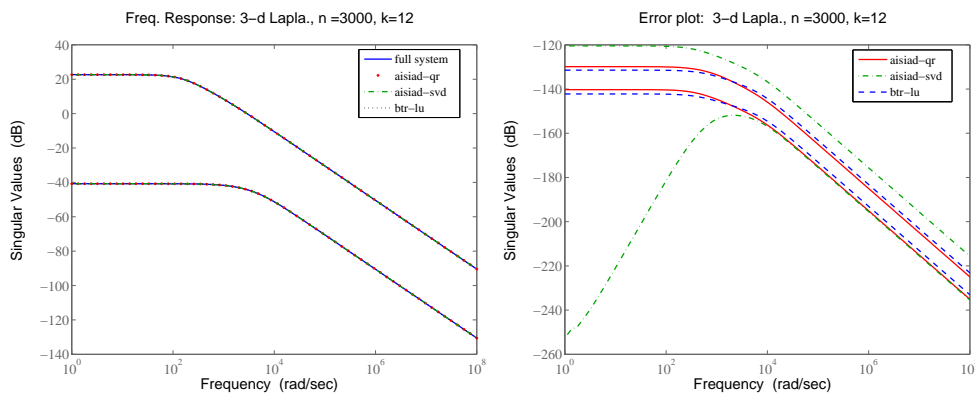


Figure 6. Three-dimensional Laplacian discretized by finite difference, $(A + A^T < 0)$. $B = C' = [\text{ones}(n, 1), [\text{ones}(30, 1); \text{zeros}(n - 30, 1)]]$. `aisiad-qr` converged in 8 steps; `aisiad-svd` converged in 7 steps. Note that for both `aisiad-svd` and `aisiad-qr`, the accuracy of approximation for the largest singular value is higher than that for the second largest singular value (This can be easily confirmed by plotting only the approximation error for the largest singular value).

methods that solve the two Lyapunov equations for P and Q separately.

The preliminary numerical results for both Algorithm 3.1 and 3.2 are satisfactory and encouraging. Even for some difficult models the algorithms can produce accurate reduced models. However, we have observed that Algorithm 3.1 and 3.2 may become less efficient when the passivity condition $A + A^T < 0$ does not hold; for the non-passive cases, even though the algorithms can generate a reduced model with required accuracy, the reduced model may be of larger dimension than necessary.

A rigorous convergence analysis of these algorithms is difficult and presently not available. Further research on the theoretical properties of the algorithms is desirable.

Acknowledgments

We thank Professor Athanasios Antoulas for stimulating discussions and for providing some of the test examples. We acknowledge the helpful and constructive comments from two anonymous referees.

REFERENCES

1. A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.
2. A. C. Antoulas, D. C. Sorensen, Y. Zhou, *On the decay rate of Hankel singular values and related issues*, *Systems and Control Letters*, 46 (2002) pp. 323–342.
3. R.H. Bartels and G.W Stewart, *Solution of the equation $AX + XB = C$* , *Communications in ACM* 15 (1972) pp. 820–826.
4. D. Calvetti, B. Lewis and L. Reichel, *On the solution of large Sylvester-observer equations*, *Numerical Linear Algebra with Applications*, 8 (2001) pp. 1–16.

5. B. N. Datta and Y. Saad, *Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment*, Numerical Linear Algebra with Applications, 154–156 (1991) pp. 225–244.
6. E. R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real symmetric matrices*, Journal of Computational Physics, 17 (1975) pp. 87–94.
7. D. F. Enns, *Model reduction for control system design*, Ph.D. dissertation, Dept. of Aeronautics and Astronautics, Stanford University, 1984.
8. D. F. Enns, *Model reduction with balanced realizations: an error bound and frequency weighted generalizations*, Proceedings of the 23rd Conference on Decision and Control, Las Vegas, NV, (1984) pp. 127–132.
9. K. V. Fernando and H. Nicholson, *On the structure of balanced and other principal representations of SISO systems*, IEEE Transactions on Automatic Control, AC-28 (1983) pp. 228–231.
10. R. W. Freund, *Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation*, Applied and Computational Control, Signals, and Circuits, Birkhäuser, 1 (1999) pp. 435–498.
11. K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ -error bounds*, International Journal of Control, 39 (1984) pp. 1115–1193.
12. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, 3rd ed., 1996.
13. T. Gudmundsson, A. J. Laub, *Approximate Solution of Large Sparse Lyapunov Equations*, IEEE Transactions on Automatic Control, AC-39 (1994) pp. 1110–1114.
14. S. Gugercin, A. C. Antoulas, D. C. Sorensen, *A modified low-rank Smith method for large-scale Lyapunov equations*, Numerical Algorithms, 32 (2003) pp. 27–55.
15. A. Hodel, K. Poolla, B. Tenison, *Numerical solution of the Lyapunov equation by approximate power iteration*, Linear Algebra and Its Applications, 236 (1996) pp. 205–230.
16. D. Hu and L. Reichel, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra and Its Applications, 172 (1992) pp. 283–313.
17. R. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi methods*, SIAM, 1998.
<http://www.caam.rice.edu/software/ARPACK>
18. I. M. Jaimoukha and E. M. Kasenally, *Krylov Subspace methods for solving Large Lyapunov Equations*, SIAM Journal on Numerical Analysis, 31 (1994) pp. 227–251.
19. I. M. Jaimoukha and E. M. Kasenally, *Oblique projection methods for large scale model reduction*, SIAM Journal on Matrix Analysis and Applications, 16 (1995) pp. 602–627.
20. P. Lancaster, *Explicit Solution of Linear Matrix Equations*, SIAM Review, 12 (1970) pp. 544–566.
21. A. J. Laub, M. T. Heath, C. C. Paige, R. C. Ward, *Computation of System Balancing Transformations and other Applications of Simultaneous Diagonalization Algorithms*, IEEE Transactions on Automatic Control, AC-32 (1987) pp. 115–122.
22. J. Li, *Model Reduction of Large Linear Systems via Low Rank System Gramians*, PhD thesis, Dept. of Mathematics, MIT, 2000.
23. J. Li, J. White, *Low rank solution of Lyapunov equations*, SIAM Journal on Matrix Analysis and Applications, 24 (2002) pp. 260–280.
24. B. C. Moore, *Principal component analysis in linear systems: controllability, observability, and model reduction*, IEEE Transactions on Automatic Control, AC-26 (1981) pp. 17–32.
25. T. Penzl, *Algorithms for model reduction of large dynamical systems*, Technical Report SFB393/99-40. TU Chemnitz, 1999.
26. T. Penzl, *Eigenvalue Decay Bounds for Solutions of Lyapunov Equations: The Symmetric Case*, Systems and Control Letters, 40 (2000) pp. 139–144.
27. T. Penzl, *A Cyclic Low-Rank Smith Method for Large Sparse Lyapunov Equations*, SIAM Journal on Scientific Computing, 21 (2000) pp. 1401–1418.
28. Y. Saad, *Numerical Solution of Large Lyapunov Equations*, Proceedings of the Int. symposium MTNS-89, Vol. III, M. Kaashoek, J.V. Schuppen, A. Ran, eds., Birkhäuser, Boston, MA, (1990) pp. 503–511.
29. M. G. Safonov and R. Y. Chiang, *A Schur Method for Balanced-Truncation Model Reduction*, IEEE Transactions on Automatic Control, AC-34 (1989) pp. 729–733.
30. G. L. G. Sleijpen and H. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM Journal on Matrix Analysis and Applications, 17 (1996) pp. 401–425.
31. D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM Journal on Matrix Analysis and Applications, 13 (1992) pp. 357–385.
32. D. C. Sorensen and A. C. Antoulas, *The Sylvester equation and approximate balanced reduction*, Fourth Special Issue on Linear Systems and Control, Linear Algebra and Its Applications, 351–352 (2002) pp. 671–700.
33. D. C. Sorensen and Y. Zhou, *Bounds on Eigenvalue Decay Rates and Sensitivity of Solutions to Lyapunov Equations*, CAAM TR02-07, Rice University, 2002.

34. D. Vasilyev and J. White, *A more reliable reduction algorithm for behavioral model extraction*, Proceedings of International Conference on Computer Aided Design, 2005.
35. Y. Zhou, *Studies on Jacobi-Davidson, Rayleigh Quotient iteration, inverse iteration generalized Davidson and Newton updates*, Numerical Linear Algebra with Applications, 13 (2006) pp. 621–642.
36. Y. Zhou, *Numerical Methods for Large Scale Matrix Equations with Applications in LTI System Model Reduction*, PhD thesis, Dept. of Comp. & Appl. Mathematics, Rice University, 2002.