Model Reduction of Large Linear Systems via Low Rank System Gramians

by

Jing-Rebecca Li

B.Sc., Mathematics
University of Michigan, 1995

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mathematics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2000

Author .....
Department of Mathematics
August 4, 2000

Certified by .....
Jacob K. White
Professor
Thesis Supervisor

Accepted by .....
Daniel Kleitman
Chairman, Applied Mathematics Committee

Accepted by .....
Tomasz S. Mrowka
Chairman, Department Committee on Graduate Students

Model Reduction of Large Linear Systems via Low Rank System Gramians

by

Jing-Rebecca Li

Submitted to the Department of Mathematics
on August 4, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mathematics

Abstract

This dissertation concerns the model reduction of large, linear, time-invariant systems. A new method called the Dominant Gramian Eigenspaces method, which utilizes low rank approximations to the exact system gramians, is proposed for such system.

The Cholesky Factor ADI algorithm is developed to generate low rank approximations to the system gramians. Cholesky Factor ADI requires only matrix-vector products and linear solves, hence it enables one to take advantage of sparsity or structure in the system matrix.

A connection is made between approximating the dominant eigenspaces of the system gramians and the generation of various low order Krylov and rational Krylov subspaces.

The Cholesky Factor ADI algorithm is then used in conjunction with the Dominant Gramian Eigenspaces method in the model reduction of large, linear, time-invariant systems. It is demonstrated numerically that this approach often produces globally accurate reduced models, even when the low rank approximations to the system gramians have not converged to the exact gramians.

In addition, it is shown that, in a model reduction method for symmetric systems based on moment matching, the problem of choosing moment matching points can be approached by solving the rational min-max problem associated with CF–ADI parameter selection.

Thesis Supervisor: Jacob K. White
Title: Professor

Acknowle g ents

I would like to thank my advisor Jacob White for his four years of guidance, and for introducing me to the area of interconnect modeling and to the model reduction problem, as well as teaching me the importance of applicability of one's research.

I also have much for which to thank Joel Phillips, who has talked with me extensively about the technical aspects of my research over the years, and provided many useful suggestions for this dissertation.

Among my group-mates, I would like to thank Matt Kamon and Nuno Marques for providing me with the examples which I used over and over again.

In the Applied Mathematics department, I would like to thank Alan Edelman for suggestions on my dissertation, and Boris Schlittgen for proof-reading.

Finally, Thilo Penzl had shown me an entire aspect of the model reduction problem which I had not considered before. I have certainly missed him both as a colleague and a friend.

Contents

Lis f Figur s

is Tabl s

h p r 1

I r d c i

1.1 Dissertation outline

This dissertation has two parts. A self-contained part concerns the solution of the Lyapunov equation whose right hand side has low rank. The other part utilizes the Lyapunov results in the model reduction of large, linear, time-invariant systems.

A main contribution of this dissertation is the formulation of the Cholesky Factor ADI algorithm [33], which solves the following Lyapunov equation whose right hand side has low rank,

$$AX + XA^T = -BB^T, \quad A \in \mathbb{R}^{n\times n},\ X \in \mathbb{R}^{n\times n},\ \lambda_i(A) < 0,\ \forall i,\ rank(B) \ll n. \tag{1.1}$$

The unknown is the matrix X. The coefficient matrix A is stable, and the right hand side, $-BB^T$, has rank much lower than n. Such Lyapunov equations occur in the analysis and model reduction of large, linear, time-invariant systems, where the system size is much larger than the number of inputs and the number of outputs.

The Cholesky Factor ADI algorithm is a reformulation of the Alternate Direction Implicit algorithm [2, 57, 58], and gives exactly the same approximation. However, Cholesky Factor ADI requires only matrix-vector products and linear solves by A, hence it enables one to take advantage of sparsity or structure in the matrix A. The Cholesky Factor ADI algorithm can be used to generate a low rank approximation to the solution of (1.1).

A second contribution of this dissertation consists of making the connection between approximating the dominant eigenspace of the solution to (1.1) and the generation of various low order Krylov and rational Krylov subspaces.

The second part of this dissertation concerns low rank model reduction methods for large, linear, time-invariant systems. A low rank model reduction method uses low rank approximations to the exact system gramians. A new method, the Dominant Gramian Eigenspaces method, is proposed here. Numerical comparison of the Dominant Gramian

Eigenspaces method is made with another low rank model reduction approach, the Low Rank Square Root method [41, 46]. It is shown that the Dominant Gramian Eigenspaces method often produces a better reduced model than the Low Rank Square Root method when the low rank approximations to the system gramians have not converged to the exact gramians. The Cholesky Factor ADI algorithm can be used to generate low rank approximations to the system gramians for either low rank model reduction method.

A further contribution of this dissertation is showing that, for symmetric systems, the problem of picking points where moments are to be matched in a moment matching via rational Krylov subspaces method can be approached by solving the rational min-max problem associated with CF–ADI parameter selection.

This dissertation is organized in the following way.

Chapter 1 covers the basics of linear, time-invariant systems theory, including controllability, observability, and the system gramians as the solutions to two Lyapunov equations.

Chapter 2 introduces the idea of model reduction via projection. Chapter 3 describes moment matching via projection onto a rational Krylov subspace. Chapter 4 describes the Truncated Balanced Realization method of model reduction, which requires exact system gramians.

Chapter 5 motivates the need to approximate Truncated Balanced Realization by low rank methods which use only low rank approximations to the system gramians. It is shown that this is achievable for symmetric systems, but is in general not possible for non-symmetric systems. For the model reduction of non-symmetric systems, the Dominant Gramian Eigenspaces method is proposed and shown to produce better reduced models than the existing Low Rank Square Root method.

Chapter 6 characterizes the different bases for the range of the solution to (1.1) as order n Krylov and rational Krylov subspaces with different starting vectors.

Chapter 7 turns to the solution of (1.1) and provides background on existing approaches. Chapter 8 develops the Cholesky Factor ADI method.

Chapter 9 motivates the low rank approximation of the solution to (1.1). It is shown that, various low rank approximations, including Cholesky Factor ADI, consist of finding a low order Krylov or rational Krylov subspace. These low rank methods, when run to n steps, yield the range of the solution to (1.1). This chapter includes numerical results on how well the low rank Cholesky Factor ADI approximation captures the dominant eigenspace of the exact solution to (1.1).

Chapter 10 uses Cholesky Factor ADI to generate low rank approximate gramians for the Dominant Gramian Eigenspaces method. It is shown that, for symmetric systems, the problem of picking points where moments are to be matched in a moment matching via rational Krylov subspaces method can be approached by solving the rational min-max problem associated with CF–ADI parameter selection. This chapter also includes numerical results on the model reduction of symmetric and non-symmetric systems, and on CF–ADI parameter

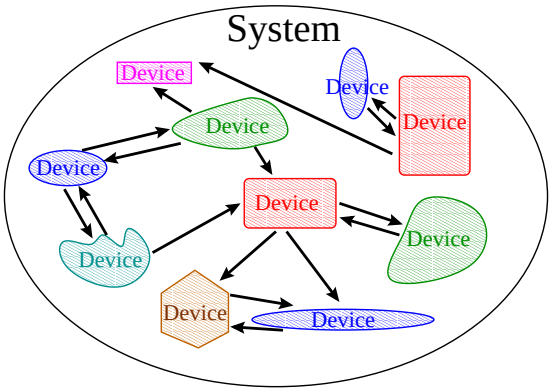


Figure 1-1: System with a large number of devices to be simulated

selection.

Chapter 11 contains conclusions and future work.

1.2 otivation

The design of complicated systems (figure 1-1) which are composed of a large number of disparate devices occurs in many engineering applications. In order to optimize a system for best performance, one needs to simulate it repeatedly, each time design parameters are varied. Devices that can belong to a system include circuits, sensors, and micro-machined devices. The devices couple to one another via inputs and outputs. The input-output behavior of the devices determines how the overall system performs.

Often, the devices are initially described by mathematical models which are large. This can happen if the models were generated without the idea in mind that they will be a part of a much larger system which needs to be repeatedly simulated.

If a system has a large number of devices, and the devices themselves are described by large models, simulation of the entire system in figure 1-1 may be unacceptably time-consuming and expensive. The idea of model reduction is that the large models should be replaced by smaller models which are amenable to fast and efficient simulation and which still capture the devices' input-output behavior to an acceptable accuracy.

Henceforth leaving the large picture of the overall system, the rest of this dissertation focuses on the mathematical models which describe the devices. Model reduction is the simplification or reduction of a mathematical model, under the constraint that the input-output behavior of the device is well approximated over the relevant range of inputs. Usually, there are also constraints placed upon the reduced model size and the approximation error.

The mathematical model for a device may be a set of discretized integral equations, semi-discretized PDEs, or simply a large system of ODEs. Often, when a model comes from discretization, the resulting system of equations can be very large. It is not rare to encounter

a circuit model of interconnect with $O(100,000)$ elements. It is also not rare for the large initial model to contain a vast amount of redundant information and to be amenable to significant reduction in model size with little loss in accuracy.

Of course, with knowledge of the nature of the physical device a engineer can frequently reduce the model size by lumping together elements, or removing parts of the problem which are of little importance in the relevant input range. This is an extremely useful approach and can produce very good, application-specific, results. However, it is far from automatic, and, at times, the intuition of the engineer can fail when subtle high order effects come into play.

This dissertation is not concerned with reduction methods which are specific to a particular engineering application. Rather, it is concerned with numerical model reduction, meaning that very little knowledge of the physical device is assumed. The object of the reduction is the original large numerical model, which is assumed to be sufficiently accurate in modeling the input-output behavior of the physical device for the relevant range of inputs. It, rather than the underlying physical device, is the object by which the quality of approximation is measured. In fact, a reduced model from numerical model reduction frequently does not have a physical counterpart.

One benefit of numerical model reduction is that for linear, time-invariant systems, there are theoretical results regarding optimality and approximation error.

This dissertation is restricted to the numerical reduction of models described by linear, time-invariant systems which have large, sparse or structured, system matrices. Such systems occur in interconnect modeling, solution of PDEs, and other applications.

1.3 Syste s theory

This section contains basic known results on linear, time-invariant systems, some of which are taken from [4, 19, 50].

A linear, time-invariant system with realization, (A, B, C), is characterized by the equations,

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t), \tag{1.2}$$

$$y(t) = Cx(t). \tag{1.3}$$

The vector valued function, $x(t) : \mathbb{R} \mapsto \mathbb{R}^n$, gives the state at time t, and has n components. The input $u(t) : \mathbb{R} \mapsto \mathbb{R}^p$, and output $y(t) : \mathbb{R} \mapsto \mathbb{R}^q$, have p and q components, respectively. The matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{q\times n}$ are the system matrix, the input coefficient matrix, and the output coefficient matrix, respectively.

For single-input single-output (SISO) systems, $p = 1, q = 1$. Even for multiple-input, multiple-output (MIMO) systems, p and q are usually both very small compared to n.

The components in $u(t)$ and $y(t)$ have physical meaning as the inputs and outputs of the device being modeled. Often, the components in $x(t)$, as originally discretized, also have physical meaning, such as being the nodal voltages and branch currents of a circuit. The matrices B and C describe how the components in $x(t)$ are connected to the device inputs and outputs. The original matrix A usually comes from discretizing the governing equations. However, after model reduction, $x(t)$, A, B, and C may not have simple physical interpretations.

An example of a linear, time-invariant system comes from the semi-discretization of the 1-D diffusion equation,

$$\frac{\partial f(w,t)}{\partial t} = -\frac{\partial^2 f(w,t)}{\partial w^2}, \tag{1.4}$$

where $f(w,t)$ may be the temperature of a metal rod at time t and position w. If (1.4) is discretized in the space variable w only,

$$x_i(t) := f(w_i, t), \tag{1.5}$$

then it becomes a system of ODEs as in (1.2). The semi-discretized values of f are the components of the state vector $x(t)$. The system matrix A comes from discretizing the second order differentiation operator. The boundary condition determines the input. Indicating at which positions temperature is measured gives the output equation (1.3).

Equation (1.2) is a simple system of linear, time-invariant, non-homogeneous, first order ODEs. Equation (1.3) is an algebraic equation which produces the output $y(t)$, each of whose components is a linear combination of the components of the solution $x(t)$ to (1.2).

The solution of (1.2) is,

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\nu)}Bu(\nu)d\nu, \quad x_0 = x(t_0), \tag{1.6}$$

which gives the output as,

$$y(t) = Ce^{A(t-t_0)}x_0 + C\int_{t_0}^{t} e^{A(t-\nu)}Bu(\nu)d\nu. \tag{1.7}$$

1.3.1 Transfer function

The Laplace transform of a function $f(t)$ in the time domain is the function $F(s)$ in the frequency domain,

$$\mathcal{L}\{f(t)\} = F(s) := \int_0^\infty e^{-st} f(t)dt. \tag{1.8}$$

By taking the Laplace transforms of the quantities in (1.2-1.3) one obtains,

$$\begin{aligned} sX(s) &= AX(s) + BU(s), & (1.9)\\ Y(x) &= CX(s), & (1.10) \end{aligned}$$

where $U(s)$, $Y(s)$, $X(s)$ are the Laplace transforms of the input $u(t)$, the output $y(t)$, and the state vector $x(t)$, respectively.

The transfer function $G(s)$ of the system (1.2-1.3) is

$$G(s) = C(sI - A)^{-1}B. \tag{1.11}$$

It relates input to output in the Laplace or frequency domain according to,

$$Y(s) = G(s)U(s). \tag{1.12}$$

The following definition deals with the equivalence of different realizations in terms of the transfer function.

Definition 1. *A realization* $(\tilde{A}, \tilde{B}, \tilde{C})$,

$$\begin{aligned} \frac{d\tilde{x}(t)}{dt} &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t), & (1.13)\\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t), & (1.14) \end{aligned}$$

is equivalent to (1.2-1.3) if

$$\tilde{G}(s) = \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} = C(sI - A)^{-1}B = G(s), \quad \forall s. \tag{1.15}$$

A realization $(T^{-1}AT, T^{-1}B, CT)$, where $T \in \mathbb{R}^{n\times n}$ is an invertible matrix, is equivalent to (A, B, C). It corresponds to a change of variable, $x(t) = T\tilde{x}(t)$, in (1.2-1.3). There are infinitely many equivalent realizations of the same linear time-invariant system.

1.3.2 eachability and controllability

This section reviews the basics of controllability.

Definition 2. *The state $z \in \mathbb{R}^n$ can be* **reached** *from the state $w \in \mathbb{R}^n$, and equivalently, w can be* **controlled** *to z, if there exist $t_0, t, u(t)$ such that equation (1.6) is satisfied with $x_0 = w$ and $x(t) = z$.*

Definition 3. *A system is* **controllable** *if for any pair of states w and z, w can be controlled to z, or equivalently, z can be reached from w.*

Proposition 1. *The system in (1.2-1.3), because it is linear, is controllable if and only if every state z can be reached from the zero state [50].*

The states that can be reached from $x_0 = 0$ are

$$\left\{ z \in \mathbb{R}^n \;\middle|\; z = z(t) = \int_{t_0}^{t} e^{A(t-\nu)} B u(\nu) d\nu \right\}, \tag{1.16}$$

which implies

$$z \in col sp \left[B, AB, \cdots, A^{n-1}B \right]. \tag{1.17}$$

The system (1.2-1.3) is controllable if and only if

$$rank \left(\left[B, AB, \cdots, A^{n-1}B \right] \right) = n. \tag{1.18}$$

Define $\mathcal{L}_p^2(\sigma, \tau)$ to be the set of square integrable functions $u : [\sigma, \tau) \mapsto \mathbb{R}^p$. The operator $L : \mathcal{L}_p^2(\sigma, \tau) \to \mathbb{R}^n$ which maps the input $u(\nu) \in \mathcal{L}_p^2(\sigma, \tau)$ to the state $x(\tau) \in \mathbb{R}^n$ at $t = \tau$, with zero initial state, $x_0 = x(\sigma) = 0$, is given by

$$L(u) = \int_{\sigma}^{\tau} k(\nu)^T u(\nu) d\nu, \tag{1.19}$$

where

$$k(\nu) = B^T (e^{A^T(\tau - \nu)}). \tag{1.20}$$

Define the inner product on $\mathcal{L}_p^2(\sigma, \tau)$ as

$$< u, v > := \int_{\sigma}^{\tau} u(\nu)^T v(\nu) d\nu, \tag{1.21}$$

and the inner product on $\mathbb{R}^n$ as

$$< w, z > := w^T z, \tag{1.22}$$

then the adjoint operator $L^* : \mathbb{R}^n \to \mathcal{L}_p^2(\sigma, \tau)$ of L, which must satisfy,

$$< Lu, x > = < u, L^* x >, \tag{1.23}$$

is

$$(L^* x)(\nu) = k(\nu) x = B^T (e^{A^T(\tau - \nu)}) x. \tag{1.24}$$

The system (1.2-1.3) is controllable if and only if L is onto, and if and only if LL^* is positive definite [50].

The controllability gramian, $W_c(\sigma,\tau) : \mathbb{R}^n \to \mathbb{R}^n$, is defined as

$$\begin{aligned} W_c(\sigma,\tau) = LL^* &= \int_\sigma^\tau k(\nu)^T k(\nu) d\nu, \\ &= \int_\sigma^\tau e^{A(\tau-\nu)} BB^T e^{A^T(\tau-\nu)} d\nu. \end{aligned} \tag{1.25}$$

It is a Hermitian, positive semidefinite matrix, and

$$< x, W_c(\sigma,\tau)x >= \|L^* x\|^2, \quad \forall x. \tag{1.26}$$

The generalized inverse of L, $L^\# : \mathbb{R}^n \to \mathcal{L}_p^2(\sigma,\tau)$, is

$$L^\# = L^*(LL^*)^{-1} = L^* W_c(\sigma,\tau)^{-1}. \tag{1.27}$$

$L^\# x$ is the unique solution to $Lu = x$ with the smallest norm,

$$L(L^\# x) = x, \quad \forall x \in \mathbb{R}^n, \tag{1.28}$$

and

$$\|L^\# x\| < \|u\|, \quad \forall u : Lu = x, \quad u \neq L^\# x. \tag{1.29}$$

Its norm is

$$\|L^\# x\|^2 = \|L^* W_c(\sigma,\tau)^{-1} x\|^2 =< x, W_c(\sigma,\tau)^{-1} x >= x^T W_c(\sigma,\tau)^{-1} x. \tag{1.30}$$

Thus, given any two states x and z the input

$$u(\nu) = B^T e^{A^T(\tau-\nu)} W_c(\sigma,\tau)^{-1} (z - e^{A(\tau-\sigma)} x) \tag{1.31}$$

is the unique input that minimizes $\|u\|$ among all inputs which take x to z in $[\sigma,\tau]$.

1.3.3 Observability

This section reviews the basics of observability.

Definition 4. *The states $w \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$ are* **distinguishable** *if there exist $t_0, t, u(t)$ such that, $x_0 := w$ and $x_0 := z$ in (1.7) result in different $y(t)$'s.*

Definition 5. *A system is* **observable** *if any two distinct states are distinguishable.*

Proposition 2. *For linear systems, w and z are distinguishable if and only if $w - z$ is distinguishable from 0, and if and only if the zero input, $u(t) \equiv 0$, distinguishes them [50].*

Proposition 3. *The system (1.2-1.3), because it is linear, is observable if and only if the zero state is the only state which results in the zero output, $y(t) \equiv 0$, with zero input, $u(t) \equiv 0$ [50].*

According to (1.7), the states which result in the zero output, $y(t) \equiv 0$, with zero input, $u(t) \equiv 0$, are

$$\left\{ z \in \mathbb{R}^n \;\middle|\; Ce^{A(t-t_0)}z \equiv 0 \right\}, \tag{1.32}$$

which implies

$$z \in ker \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix}. \tag{1.33}$$

The system (1.2-1.3) is observable if and only if

$$ker \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = 0, \tag{1.34}$$

and if and only if

$$rank \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix} = n. \tag{1.35}$$

Define $L : \mathcal{L}_q^2(\sigma, \tau) \to \mathbb{R}^n$ as,

$$L(y) = \int_{\sigma}^{\tau} k(t)^T y(t) dt, \tag{1.36}$$

where

$$k(t) = Ce^{A(t-\tau)}. \tag{1.37}$$

Then the adjoint operator $L^* : \mathbb{R}^n \to \mathcal{L}_q^2(\sigma, \tau)$,

$$(L^*x)(t) = Ce^{A(t-\tau)}x = Ce^{A(t-\sigma)}e^{A(\sigma-\tau)}x, \tag{1.38}$$

maps x to the output $y(t)$ resulting from the initial state $x_0 = e^{A(\sigma-\tau)}x$ and zero input.

The system (1.2-1.3) is observable if and only if L^* is one-to-one, and if and only if LL^* is positive definite. The observability gramian, $W_o(\sigma, \tau) : \mathbb{R}^n \to \mathbb{R}^n$, is defined as

$$W_o(\sigma, \tau) = LL^* = \int_\sigma^\tau e^{A^T(t-\tau)}C^TCe^{A(t-\tau)}dt. \tag{1.39}$$

The adjoint of the pseudo-inverse $L^\#$, $(L^\#)^* : \mathcal{L}_q^2(\sigma, \tau) \to \mathbb{R}^n$, is

$$(L^\#)^* = W_o(\sigma, \tau)^{-1}L. \tag{1.40}$$

It gives the least-squares solution of $L^*z = y$. For each $y \in \mathcal{L}_q^2(\sigma, \tau)$, let $z = W_o(\sigma, \tau)^{-1}Ly$, then,

$$\|L^*z - y\| < \|L^*x - y\|, \quad x \neq z. \tag{1.41}$$

Thus, $e^{A(\sigma-\tau)}z$ is the initial state that results in an output that is closest to $y(t)$ in the least-squares sense.

1.4 ra ians an yapunov equations

If the particular choices of $\sigma = -\infty$ and $\tau = 0$ are made, and if the system matrix A is stable, i.e., all eigenvalues of A are in the open left half plane, the following definitions of the system controllability gramian P, and the system observability gramian Q, can be made,

$$P := \int_0^\infty e^{At}BB^Te^{A^Tt}dt = W_c(-\infty, 0), \tag{1.42}$$

$$Q := \int_0^\infty e^{A^Tt}C^TCe^{At}dt = W_o(-\infty, 0). \tag{1.43}$$

Proposition 4. *(See [19]) If $Re(\lambda_i(A)) < 0, \forall i$, then*
1. P is positive definite if and only if (A, B) is controllable.
2. Q is positive definite if and only if (A, C) is observable.

If the system (1.2-1.3) is controllable, hence P is invertible, then the solution of the

minimum energy problem,

$$\min_{u \in L_p^2(-\infty,0),\, x(0)=z} J(u), \tag{1.44}$$

where

$$J(u) = \int_{-\infty}^{0} u^T(t)u(t)dt, \tag{1.45}$$

is given by,

$$u_{opt}(t) = B^T e^{-A^T t} P^{-1} z, \tag{1.46}$$

and the energy of $u_{opt}(t)$ is

$$J(u_{opt}) = z^T P^{-1} z. \tag{1.47}$$

Hence, if $x(0) = z$ lies along one of the eigenvectors of P^{-1} with large eigenvalues, then $x(0) = z$ can be reached only if a large input energy is used. Eigenvectors of P^{-1} with large eigenvalues are also eigenvectors of P with small eigenvalues, since

$$P = U\Sigma U^T \Longleftrightarrow P^{-1} = U\Sigma^{-1}U^T, \tag{1.48}$$

because P is real and symmetric.

If the system is released from $x(0) = z$, with $u(t) = 0, t \geq 0$, then

$$\int_{0}^{\infty} y^T(t)y(t)dt = z^T Q z. \tag{1.49}$$

If $x(0) = z$ lies along one of the eigenvectors of Q with small eigenvalues, then it will have little effect on the output.

It can be seen that the system gramians P and Q satisfy the following Lyapunov equations [19],

$$AP + PA^T = -BB^T, \tag{1.50}$$

$$A^T Q + QA = -C^T C. \tag{1.51}$$

The solutions to both are unique if A is stable [19].

If the number of inputs p is much smaller than the number of state components n, then $rank(BB^T) = rank(B) \leq p \ll n$, and the right hand side of (1.50) has low rank. Similarly, if the number of outputs q is much smaller than n, then the right hand side of (1.51) has low rank.

The gramians P and Q provide information about the controllability and observability of the system (1.2-1.3) in terms of past inputs ($t \leq 0$) and future outputs ($t \geq 0$).

In the rest of this dissertation, eigenvectors of P with large eigenvalues will be referred to as the dominant controllable modes, and eigenvectors of Q with large eigenvalues will be referred to as the dominant observable modes.

Definition 6. *Let $Re(\lambda_i(A)) < 0, \forall i$, then the Hankel singular values of the transfer function $G(s)$ (1.11) are*

$$\sigma_i(G(s)) := \{\lambda_i(PQ)\}^{\frac{1}{2}}. \tag{1.52}$$

Proposition 5. *The Hankel singular values of $G(s)$ are also the singular values of the Hankel operator, $\Gamma_G : \mathcal{L}_p^2(0,\infty) \to \mathcal{L}_q^2(0,\infty)$,*

$$(\Gamma_G v)(t) := \int_0^\infty Ce^{(A(t+\nu))}Bv(\nu)d\nu. \tag{1.53}$$

Proof. [19] To find the singular values of Γ_G, note that its adjoint is

$$(\Gamma_G^* y)(t) = \int_0^\infty B^T e^{(A^T(t+\nu))}C^T y(\nu)d\nu. \tag{1.54}$$

Suppose σ_i is a singular value of Γ_G, with v the corresponding eigenvector of $\Gamma_G^*\Gamma_G$,

$$\Gamma_G^*\Gamma_G v = \sigma_i^2 v, \tag{1.55}$$

and let

$$y := \Gamma_G v = Ce^{At}x_0, \tag{1.56}$$

where

$$x_0 = \int_0^\infty e^{A\nu}Bv(\nu)d\nu, \tag{1.57}$$

then

$$\begin{aligned}
\Gamma_G^*\Gamma_G v &= \Gamma_G^* y, && (1.58)\\
&= B^T e^{A^T t}\int_0^\infty e^{A^T\nu}C^T Ce^{A\nu}x_0 d\nu, && (1.59)\\
&= B^T e^{A^T t}Qx_0, && (1.60)\\
&= \sigma_i^2 v. && (1.61)
\end{aligned}$$

Hence

$$v(t) = B^T e^{A^T t} Q x_0 \sigma_i^{-2}. \tag{1.62}$$

Substituting (1.62) into (1.57) gives

$$PQx_0 = \sigma_i^2 x_0. \tag{1.63}$$

Therefore,

$$\Gamma_G^* \Gamma_G v = \sigma_i^2 v \Longleftrightarrow PQx_0 = \sigma_i^2 x_0. \tag{1.64}$$

□

The Hankel operator associated with the system (1.2-1.3) maps past inputs to future outputs. If the input $u(t) = v(-t)$ for $t < 0$, then the output for $t > 0$ is $y(t) = (\Gamma_G v)(t)$.

r 2

M l R i

2.1 Intro uction

The two competing approaches for generating reduced order models of linear, time-invariant systems have been moment-matching via orthogonalized Krylov-subspace methods [12, 13, 15, 17, 18, 21, 22, 28, 29, 31, 37, 38, 40] and Truncated Balanced Realization (TBR) [11, 39, 45, 49, 53]. TBR produces a reduced model with good global accuracy and a known frequency domain L^∞-error bound. However, because it requires the solutions to two Lyapunov equations as well as matrix factorizations and products, TBR is too expensive computationally to use on large problems. Although moment-matching methods are inexpensive to apply, they often produce unnecessarily high order models.

2.2 Proble for ulation

The linear, time-invariant system with realization (A, B, C),

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t), \tag{2.1}$$

$$y(t) = Cx(t), \tag{2.2}$$

$$A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{n\times p}, \quad C \in \mathbb{R}^{q\times n}, \tag{2.3}$$

has the transfer function $G(s)$,

$$G(s) = C(sI - A)^{-1}B, \quad G(s) \in \mathbb{C}^{q\times p}, \tag{2.4}$$

which relates input to output in the frequency domain according to,

$$Y(s) = G(s)U(s). \tag{2.5}$$

The transfer function $G(s)$ can be written as $G(s) = C\frac{(mc(sI-A))^T}{det(sI-A)}B$, where $det(sI-A)$ is the determinant of the matrix $sI-A$, and $mc(sI-A)$ denotes the matrix of cofactors of $sI-A$. Thus, $G(s)$ is a $q \times p$ matrix whose entries are rational functions in s. The numerator degree of each rational function is strictly smaller than its denominator degree, because the degree of each entry in $(mc(sI-A))^T$ is at most $n-1$ and degree of $det(sI-A)$ is n.

In the simple case of $p = q = 1$, the system (2.1-2.2) is controllable and observable if and only if the numerator and denominator of the rational function $G(s)$ have no common factors, or in other words, $G(s)$ is irreducible.

The problem of model reduction is to find a smaller system,

$$\frac{dx_k^r(t)}{dt} = A_k^r x_k^r(t) + B_k^r u(t), \tag{2.6}$$

$$y_k^r(t) = C_k^r x_k^r(t), \tag{2.7}$$

$$A_k^r \in \mathbb{R}^{k\times k}, \quad B_k^r \in \mathbb{R}^{k\times p}, \quad C_k^r \in \mathbb{R}^{q\times k}, \tag{2.8}$$

such that k, the number of components in $x_k^r(t)$, is much smaller than n, and the transfer function of the new system $G_k^r(s)$,

$$G_k^r(s) = C_k^r(sI - A_k^r)^{-1}B_k^r, \quad G(s) \in \mathbb{C}^{q\times p}, \quad Y_k^r(s) = G_k^r(s)U_k^r(s), \tag{2.9}$$

is close to the original transfer function $G(s)$.

If $p = q = 1$, then $G_k^r(s)$ is a rational function of degree $\leq k$, and the problem of model reduction can also be viewed as the approximation of a high degree rational function by one of much lower degree.

2.3 rojection

Almost all model reduction methods are projection methods. An exception may be explicit moment matching methods, which will not be considered here.

Before proceeding with projection methods, generalized state space form will be briefly described. This will help to create a more general framework which can include projection methods whose left and right projection matrices are not bi-orthogonal.

A system given by (2.1-2.2) is in standard state space form. A system in generalized state space form, with realization (E, A, B, C), is described by the equations,

$$E\dot{x}(t) = Ax(t) + Bu(t), \tag{2.10}$$

$$y(t) = Cx(t), \tag{2.11}$$

and has the transfer function

$$G(s) = C(sE - A)^{-1}B. \tag{2.12}$$

If E is invertible, (2.10-2.11) can be easily converted to standard state space form.

For generalized state space systems, the reduced system should have the form,

$$\begin{aligned} E_k^r \dot{x}_k^r(t) &= A_k^r x_k^r(t) + B_k^r u(t), & (2.13)\\ y_k^r(t) &= C_k^r x_k^r(t), & (2.14) \end{aligned}$$

with the transfer function

$$G_k^r(s) = C_k^r(sE_k^r - A_k^r)^{-1}B_k^r. \tag{2.15}$$

A projection method reduces (2.10-2.11) by choosing two k-dim projection spaces, $S_1, S_2 \subseteq \mathbb{R}^n$, so that the solution space is projected unto S_2, $x_k^r \in S_2$, and the residual of (2.10-2.11) is orthogonal to S_1. A realization of the reduced system satisfies the projection equations,

$$\begin{aligned} E_r^k &= V_k^T E U_k, \quad A_r^k = V_k^T A U_k, & (2.16)\\ B_r^k &= V_k^T B, \quad C_r^k = C U_k, & (2.17) \end{aligned}$$

where the columns of V_k and U_k form bases for S_1 and S_2, respectively,

$$col\,sp(V_k) = S_1, \quad V_k \in \mathbb{R}^{n\times k}, \quad col\,sp(U_k) = S_2, \quad U_k \in \mathbb{R}^{n\times k}. \tag{2.18}$$

If $S_1 = S_2$, the projection is orthogonal, otherwise it is oblique. The matrices V_k and U_k will be referred to as the left projection matrix and the right projection matrix, respectively. The following proposition shows that the choice of basis for S_1 and S_2 is not important.

Proposition 6. *If the columns of $\tilde{V}_k$ also form a basis for S_1, and the columns of $\tilde{U}_k$ also form a basis for S_2, then the reduced system obtained by projection with $\tilde{V}_k$ and $\tilde{U}_k$ according to (2.16-2.17), is equivalent to (has the same transfer function as) the reduced model obtained by projection with V_k and U_k.*

Proof. This follows from the existence of invertible $k \times k$ matrices, $R_{k\times k}$ and $W_{k\times k}$, such that

$$\begin{aligned} V_k &= \tilde{V}_k R_{k\times k}, & (2.19)\\ U_k &= \tilde{U}_k W_{k\times k}, & (2.20) \end{aligned}$$

so that

$$
\begin{aligned}
G_r(s) &= CU_k(sV_k^T EU_k - V_k^T AU_k)^{-1} V_k^T B && (2.21)\\
&= C\tilde{U}_k W_{k\times k}(sR_{k\times k}^T \tilde{V}_k^T E\tilde{U}_k W_{k\times k} - R_{k\times k}^T \tilde{V}_k^T A\tilde{U}_k W_{k\times k})^{-1} R_{k\times k}^T \tilde{V}_k^T B && (2.22)\\
&= C\tilde{U}_k(s\tilde{V}_k^T E\tilde{U}_k - \tilde{V}_k^T A\tilde{U}_k)^{-1}\tilde{V}_k^T B && (2.23)\\
&= \tilde{G}_r(s). && (2.24)
\end{aligned}
$$

□

Hence, the exact projection matrices are not important, only their column spans are.

Note if $V_k^T U_k \neq I_{k\times k}$, then the reduced system obtained according to (2.16-2.17) will not be in standard state space form even if the original system is in standard form. Thus, to preserve standard space form, the projection matrices V_k and U_k must be bi-orthogonal.

r 3

m i vi Kryl v S s s

This chapter describes the matching of transfer function moments, and how it is implemented as projection via Krylov subspaces.

3.1 Transfer function o ents

The category of moment matching methods includes all methods which seek to preserve, in the transfer function of the reduced system $G_r(s)$, some coefficients of a series expansion of the original transfer function $G(s)$. Generalized state-space form (2.10-2.11) will be used.

If $G(s)$ is expanded in powers of s^{-1}, i.e., around the point at infinity,

$$G(s) = \sum_{j=1}^{\infty} m_{-j} s^{-j}, \tag{3.1}$$

$$m_{-j} = C(E^{-1}A)^{j-1}E^{-1}B = g^{(j-1)}(t)|_{t=0}, \tag{3.2}$$

then the coefficients to be preserved are $m_{-j}, j = 1, \cdots, k$. The m_{-j}'s are called the Markov parameters, and they are the function value and derivatives of $g(t)$, the inverse Laplace transform of $G(s)$, evaluated at $t = 0$.

A reduced order model whose transfer function

$$G_r(s) = \sum_{j=1}^{\infty} m_{-j}^r s^{-j}, \tag{3.3}$$

$$m_{-j}^r = C_r(E_r^{-1}A_r)^{j-1}E_r^{-1}B_r = g_r^{(j-1)}(t)|_{t=0}, \tag{3.4}$$

preserves a number of the original Markov parameters,

$$m^r_{-j} = m_{-j}, \quad j = 1, \cdots, k, \tag{3.5}$$

is called a partial realization.

A partial realization generally results in good approximation to the original transfer function near $s = \infty$, but may not be accurate at low frequencies.

More often, $G(s)$ is expanded around one or more finite points in the complex plane. In this case, each series has the form,

$$\begin{aligned} G(s) &= \sum_{j_i=0}^{\infty} m_{j_i}(\sigma_i)(s-\sigma_i)^{j_i}, & (3.6)\\ m_{j_i}(\sigma_i) &= C((A-\sigma_i E)^{-1}E)^{j_i}(\sigma_i E - A)^{-1}B = \frac{G^{(j_i)}(s)|_{s=\sigma_i}}{j_i!}, & (3.7)\\ i &= 1, 2, \cdots, \bar{i}. & (3.8) \end{aligned}$$

The $m_{j_i}(\sigma_i)$'s are called the moments of the transfer function $G(s)$ at σ_i, which are the function value and derivatives of $G(s)$ evaluated at σ_i.

A reduced order model whose transfer function

$$\begin{aligned} G_r(s) &= \sum_{j_i=0}^{\infty} m^r_{j_i}(\sigma_i)(s-\sigma_i)^{j_i}, & (3.9)\\ m^r_{j_i}(\sigma_i) &= C_r((A_r-\sigma_i E_r)^{-1}E_r)^{j_i}(\sigma_i E_r - A_r)^{-1}B_r = \frac{G_r^{(j_i)}(s)|_{s=\sigma_i}}{j_i!}, & (3.10)\\ i &= 1, 2, \cdots, \bar{i}, & (3.11) \end{aligned}$$

preserves some moments of the original transfer function $G(s)$ at a number of points $\sigma_i, i = 1, \cdots, \bar{i}$, in the complex plane,

$$m^r_j(\sigma_i) = m_j(\sigma_i), \quad j = 1, \cdots, k_i, \quad , i = 1, \cdots \bar{i}, \tag{3.12}$$

is called a (multi-point) Padé approximant. The moment matching points $\sigma_i, i = 1, \cdots, \bar{i}$, can be real, imaginary, or complex.

Padé approximants result in good approximation to the original transfer function in neighborhoods around the points where moments are matched, but may not be accurate away from the expansion points.

3.2 I ple entation via Krylov su spaces

The usual implementation of moment matching uses projection via Krylov subspaces [8, 12, 15, 18, 40]. They are implicit moment matching methods, because the moments themselves are never explicitly computed. The choice of Krylov subspace determines where and to what order moments are matched. The assumption $B \in \mathbb{R}^n$ will be made throughout this section.

Definition 7. *The order m Krylov subspace $\mathcal{K}_m(A, B)$ of the $n \times n$ matrix A and the starting vector $B \in \mathbb{R}^n$ is the subspace,*

$$\mathcal{K}_m(A, B) = span\{B, AB, \cdots, A^{m-1}B\}. \tag{3.13}$$

Note $dim(\mathcal{K}_m(A, B)) \leq m$.

The following proposition connects projection via Krylov subspaces and the matching of Markov parameters.

Proposition 7. *(See [22]) If*

$$\mathcal{K}_{k^b}\left(E^{-1}A, E^{-1}B\right) = span\left\{E^{-1}B, E^{-1}AE^{-1}B, \cdots, E^{-1}A^{k-1}E^{-1}B\right\} \subseteq colsp\{U_k\}, \tag{3.14}$$

and

$$\mathcal{K}_{k^c}\left((E^{-1}A)^T, E^{-1}C^T\right) = span\left\{E^{-1}C^T, (E^{-1}A)^T E^{-1}C^T, \cdots, ((E^{-1}A)^T)^{k-1}E^{-1}C^T\right\} \subseteq colsp\{V_k\}, \tag{3.15}$$

then,

$$C(E^{-1}A)^{j-1}E^{-1}B = C_r(E_r^{-1}A_r)^{j-1}E_r^{-1}B_r, \tag{3.16}$$

for $j = 1, 2, \cdots, k^b + k^c$.

The following proposition connects projection via Krylov subspaces and the matching of moments at the points $\sigma_1, \cdots, \sigma_{\bar{i}} \neq \infty$.

Proposition 8. *(See [21]) If*

$$\bigcup_{i=1}^{\bar{i}} \mathcal{K}_{k_i^b}\left((A - \sigma_i E)^{-1}E, (A - \sigma_i E)^{-1}B\right) \subseteq colsp\{U_k\}, \tag{3.17}$$

and

$$\bigcup_{i=1}^{\bar{i}} \mathcal{K}_{k_i^c}\left((A - \sigma_i E)^{-T}E^T, (A - \sigma_i E)^{-T}C^T\right) \subseteq colsp\{V_k\}, \tag{3.18}$$

then,

$$-C\left\{(A-\sigma_i E)^{-1}E\right\}^{j_i-1}(A-\sigma_i E)^{-1}B \tag{3.19}$$

$$= -C_r\left\{(A_r-\sigma_i E_r)^{-1}E_r\right\}^{j_i-1}(A_r-\sigma_i E_r)^{-1}B_r, \tag{3.20}$$

$$\Longrightarrow \frac{d^{j_i-1}G(s)}{ds}|_{s=\sigma_i} = \frac{d^{j_i-1}G_r(s)}{ds}|_{s=\sigma_i}, \tag{3.21}$$

for $j_i = 1, 2, \cdots, k_i^b + k_i^c$ *and* $i = 1, 2, \cdots, \bar{i}$. *Note the inclusion rather than equality in (3.17-3.18).*

When certain processes are used to generate bases for the Krylov subspaces in (3.14-3.15) and (3.17-3.18), such as the Lanczos or the Arnoldi process, the reduced quantities $E_k^r, A_r^k, B_r^k, C_r^k$ in (2.16-2.17) may be obtained as part of the basis generation process, rather than projected explicitly via (2.16-2.17).

Regardless of how the Krylov subspaces are generated, the following two algorithms are examples of moment matching methods which use Krylov subspaces, and will be referred to in chapter 10 for numerical comparison. The systems they reduce are assumed to be in standard form, $E = I_{n\times n}$. They are not the most general of moment matching via Krylov subspaces methods. They assume bi-orthogonality of the two projection matrices, and make (3.17) and (3.18) equalities rather than inclusions. Algorithm 1 uses orthogonal projection, algorithm 2 uses oblique projection.

Algorithm 1 Moment matching via Krylov subspaces, orthogonal projection

0. Original system, $(I_{n\times n}, A, B, C)$.
1. Find $U_k = [u_1, \cdots, u_k]$ such that $U_k^T U_k = I_{k\times k}$ and

$$colsp\{U_k\} = \bigcup_{i=1}^{\bar{i}} \mathcal{K}_{k_i}\left((A-\sigma_i I)^{-1}, (A-\sigma_i I)^{-1}B\right), \tag{3.22}$$

$$k = k_1 + \cdots + k_m. \tag{3.23}$$

2. Obtain $E_k^r = I_{k\times k}, A_k^r, B_k^r, C_k^r$ such that (2.16-2.17) hold.

Moment matching methods require only matrix-vector products (3.14-3.15) or linear solves (3.17-3.18), hence they are very efficient. If the linear solves are done iteratively using a Krylov subspace method such as GMRES, all that is needed is the action of the system matrix A on a vector, which is advantageous when A is sparse, structured, or given only as a black box. However, there is no global error bound on the transfer function approximation error for moment matching methods. The error, $G(s) - G^r(s)$, will be small near points where moments are matched, but there is no guarantee that the error will be small elsewhere. These methods also may produce unstable reduced models even though the original system is stable. Further processing is needed to remove the unstable modes. [12, 22, 29]

Algorithm 2 Moment matching via Krylov subspaces, oblique projection

0. Original system, $(I_{n\times n}, A, B, C)$.
1. Find $U_k = [u_1, \cdots, u_k]$, $V_k = [v_1, \cdots, v_k]$ such that $V_k^T U_k = I_{k\times k}$ and

$$col\,sp\{U_k\} = \bigcup_{i=1}^{\bar{i}} \mathcal{K}_{k_i^b}\left((A-\sigma_i I)^{-1}, (A-\sigma_i I)^{-1}B\right), \tag{3.24}$$

$$col\,sp\{V_k\} = \bigcup_{i=1}^{\bar{i}} \mathcal{K}_{k_i^c}\left((A-\sigma_i I)^{-T}, (A-\sigma_i I)^{-T}C^T\right), \tag{3.25}$$

$$k = k_1^b + k_2^b + \cdots + k_m^b = k_1^c + k_2^c + \cdots + k_m^c. \tag{3.26}$$

2. Obtain $E_k^r = I_{k\times k}, A_k^r, B_k^r, C_k^r$ such that (2.16-2.17) hold.

A most important question associated with moment matching methods is how to pick moment matching points $\{\sigma_1, \cdots, \sigma_m\}$, and their orders $k_1, \cdots, k_m$, so that the global approximation error is small. This problem is not solved. Rather, it is tackled with heuristics [5, 6, 21], such as picking evenly or logarithmically spaced points on the imaginary or the real axis, as a function of the frequency range of interest.

In chapter 10, a criterion for picking good moment matching points, when the system is symmetric, will be given based on approximating the Truncated Balanced Realization method of model reduction.

r 4

r B l liz i

Truncated Balanced Realization (TBR) [11, 39, 45] produces a guaranteed stable reduced model, and has a frequency domain L^∞-error bound. There is no theoretical result concerning the optimality or near optimality of the TBR reduction in the L^∞ norm. However, TBR in general produces a reduced model with globally accurate frequency response approximation. This reduced model is usually superior to the models produced by moment matching methods.

The Square Root method of implementing TBR is proposed in [49, 53]. It has better numerical properties than the implementation in [19]. When referring to 'the TBR algorithm' in future chapters, the implementation in algorithm 3 is assumed.

Given a stable system in standard state space form (2.1-2.2), algorithm 3 produces the order k TBR reduction.

Algorithm 3 Square Root method to calculate the order k TBR reduction.

1. Find the Cholesky factors Z^B and Z^C of the solutions P and Q to (1.50-1.51),

$$P = Z^B (Z^B)^T, \quad Q = Z^C (Z^C)^T. \tag{4.1}$$

2. Calculate the singular value decomposition of $(Z^C)^T Z^B$,

$$U^L \Sigma (U^R)^T = (Z^C)^T Z^B, \tag{4.2}$$

where,

$$U^R = \begin{bmatrix} u_1^R & \cdots & u_n^R \end{bmatrix}, \quad U^L = \begin{bmatrix} u_1^L & \cdots & u_n^L \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix}. \tag{4.3}$$

3. If $\sigma_k > \sigma_{k+1}$, let

$$S^B = Z^B \left[u_1^R, \cdots, u_k^R \right] \begin{bmatrix} \frac{1}{\sqrt{\sigma_1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sqrt{\sigma_k}} \end{bmatrix}, \tag{4.4}$$

and

$$S^C = Z^C \left[u_1^L, \cdots, u_k^L \right] \begin{bmatrix} \frac{1}{\sqrt{\sigma_1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sqrt{\sigma_k}} \end{bmatrix}. \tag{4.5}$$

4. The order k Truncated Balanced Realization is given by

$$A_k^{tbr} = (S^C)^T A S^B, \quad B_k^{tbr} = (S^C)^T B, \quad C_k^{tbr} = C S^B. \tag{4.6}$$

The controllability and observability gramians of the order k reduced system $(A_k^{tbr}, B_k^{tbr}, C_k^{tbr})$ are diagonal and equal,

$$P_k^{tbr} = Q_k^{tbr} = \Sigma_1 = diag(\sigma_1, \sigma_2, ..., \sigma_k). \tag{4.7}$$

The resulting transfer function $G_k^{tbr}(s)$ has L^∞-error bound,

$$\|G(jw) - G_k^{tbr}(jw)\|_{L^\infty} := \sup_w \|G(jw) - G_k^{tbr}(jw)\|_2 \leq 2(\sigma_{k+1} + \sigma_{k+2} + ... + \sigma_n). \tag{4.8}$$

TBR is a projection method with left projection matrix S^C and right projection matrix S^B, such that $(S^C)^T S^B = I_{k\times k}$ and

$$col\,sp(S^B) \subseteq col\,sp(Z^B), \quad col\,sp(S^C) \subseteq col\,sp(Z^C). \tag{4.9}$$

A merit of the Square Root method is that it relies on the Cholesky factors Z^B and Z^C of the gramians P and Q, rather than the gramians themselves, which has advantages in terms of numerical stability.

The vast majority of the work involved in algorithm 3 comes from step 1 to obtain Z^B and Z^C, and step 2, the balancing singular value decomposition. Both steps 1 and 2 are $O(n^3)$ if done exactly, even if the system matrix A is sparse, which makes algorithm 3 impractical for problems with more than a few hundred components in the state vector. For this reason, TBR has long been considered too expensive to apply to large problems.

r 5

w k A r xi i

5.1 otivation

Even though Truncated Balanced Realization produces a guaranteed stable, globally accurate reduced model with a L^∞-error bound, it has been almost entirely abandoned in favor of Krylov subspace-based moment matching methods for large problems such as the modeling of complicated interconnect structures [3, 37, 40]. The solution of two Lyapunov equations and the balancing SVD in (4.2) both have complexity $O(n^3)$, which is prohibitive for problems with more than a few hundred components in the state vector.

It is clear that even if the $n \times n$ Cholesky factors of the gramians are available, the complexity of the Square Root method is still prohibitive for large n, due to the SVD of the $n \times n$ matrix $(Z^C)^T Z^B$ in step 2.

However, the work in step 2 and the subsequent step of calculating S_B and S_C will be dramatically reduced if Z^B and Z^C each have only a few columns, or equivalently, they have low rank.

This chapter answers the question of whether it is possible to approximate TBR if low rank approximations to Z^B and Z^C are available. The contention of this dissertation is that the answer is affirmative for symmetric systems, but not definitive for non-symmetric systems, although there is numerical evidence that good approximation to TBR is possible even in the non-symmetric case.

The main goal of this chapter is to present the approaches that can be taken in trying to approximate TBR, at a cost that is comparable to the popular moment matching methods. These approaches should only require matrix-vector products and linear solves. Two approaches are examined and compared. One is the Low Rank Square Root method [41, 46], the other is the Dominant Gramian Eigenspaces method [34].

The question of how to obtain low rank approximations to Z^B and Z^C will be answered in subsequent chapters.

5.2 Opti al low rank gra ian approxi ation

If $X \in \mathbb{R}^{n\times n}$, a symmetric, positive semi-definite matrix, has eigenvalue (singular value) decomposition,

$$X = [u_1, \cdots, u_J, u_{J+1} \cdots, u_n] \begin{bmatrix} \sigma_1 & & \cdots & \cdots & & 0 \\ & \ddots & & & & \\ \vdots & & \sigma_J & 0 & & \vdots \\ \vdots & & 0 & \sigma_{J+1} & & \vdots \\ & & & & \ddots & \\ 0 & & \cdots & \cdots & & \sigma_n \end{bmatrix} [u_1, \cdots, u_J, u_{J+1}, \cdots, u_n]^T, \tag{5.1}$$

$$\sigma_1 \geq \cdots \sigma_J \geq \sigma_{J+1} \geq \cdots \geq \sigma_n \geq 0, \tag{5.2}$$

and if $\sigma_J > \sigma_{J+1}$, then

$$X_J^{opt} := [u_1, \cdots, u_J] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J \end{bmatrix} [u_1, \cdots, u_J]^T, \tag{5.3}$$

is the unique optimal rank J approximation to X in the 2-norm [20].

Clearly $\|X - X_J^{opt}\|_2 = \sigma_{J+1}$, and σ_{J+1} is the smallest achievable 2-norm error when approximating X by a rank J matrix. If σ_{J+1} is not small, then X cannot be well approximated by a rank J matrix.

Definition 8. *$Z_J^{opt} \in \mathbb{R}^{n\times J}$ is an optimal rank J Cholesky factor of X if*

$$Z_J^{opt}(Z_J^{opt})^T = X_J^{opt}. \tag{5.4}$$

If Z_J^{opt} has 'thin' singular value decomposition,

$$Z_J^{opt} = [u_1^{cf}, \cdots, u_J^{cf}] \begin{bmatrix} \sigma_1^{cf} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J^{cf} \end{bmatrix} [v_1^{cf}, \cdots, v_J^{cf}]^T, \tag{5.5}$$

$$\sigma_1^{cf} \geq \cdots \geq \sigma_J^{cf} > 0, \quad u_i^{cf} \in \mathbb{R}^n, \quad v_i^{cf} \in \mathbb{R}^J, \tag{5.6}$$

$$\tag{5.7}$$

then

$$
\begin{aligned}
X_J^{opt} &= Z_J^{opt}(Z_J^{opt})^T \\
&= [u_1^{cf}, \cdots, u_J^{cf}] \begin{bmatrix} \sigma_1^{cf} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J^{cf} \end{bmatrix} [v_1^{cf}, \cdots, v_J^{cf}]^T [v_1^{cf}, \cdots, v_J^{cf}] \begin{bmatrix} \sigma_1^{cf} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J^{cf} \end{bmatrix} [u_1^{cf}, \cdots, u_J^{cf}]^T, \qquad (5.8) \\
&= [u_1^{cf}, \cdots, u_J^{cf}] \begin{bmatrix} (\sigma_1^{cf})^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & (\sigma_J^{cf})^2 \end{bmatrix} [u_1^{cf}, \cdots, u_J^{cf}]^T. \qquad (5.9)
\end{aligned}
$$

Thus, u_i^{cf} is an eigenvector of X_J^{opt} associated with the eigenvalue $(\sigma_i^{cf})^2$ if and only if it is a left singular vector of Z_J^{opt} associated with the eigenvalue σ_i^{cf}. Therefore, the eigenvectors of X_J^{opt} can be obtained by finding the left singular vectors of Z_J^{opt}, which is inexpensive to do since Z_J^{opt} has only J columns.

A matrix $Z_J \in \mathbb{R}^{n \times J}$ is called an approximately optimal rank J Cholesky factor of X, if $Z_J Z_J^T \approx X_J^{opt}$.

This chapter provides analysis on approximating TBR when approximately optimal rank J Cholesky factors of P and Q are available. Subsequent chapters will address how to obtain the approximately optimal low rank Cholesky factors.

5.3 Sy etric syste s

Approximating TBR for symmetric systems is addressed first.

A symmetric state-space system has the form,

$$
\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t), \quad A = A^T, && (5.10) \\
y(t) &= B^T x(t). && (5.11)
\end{aligned}
$$

The system matrix A is symmetric, and the output coefficient matrix is simply the transpose of the input coefficient matrix.

A certain class of circuit models from modified nodal analysis, which has the form,

$$
\begin{aligned}
E\dot{x}(t) &= Ax(t) + Bu(t), && (5.12) \\
y(t) &= B^T x(t), && (5.13)
\end{aligned}
$$

where E and A are symmetric, and E positive definite, can be symmetrized as follows [38].

A symmetric, positive definite square root of E, $E^{\frac{1}{2}}$, can be found. The new state vector

$\tilde{x}$ is defined as $\tilde{x} := E^{\frac{1}{2}}x$. Multiplying (5.12) by $E^{-\frac{1}{2}}$ results in

$$\begin{aligned} E^{-\frac{1}{2}}E^{\frac{1}{2}}E^{\frac{1}{2}}\dot{x}(t) &= E^{-\frac{1}{2}}AE^{-\frac{1}{2}}E^{\frac{1}{2}}x(t) + E^{-\frac{1}{2}}Bu(t), & (5.14)\\ y(t) &= B^TE^{-\frac{1}{2}}E^{\frac{1}{2}}x(t). & (5.15)\end{aligned}$$

Thus, (5.12-5.13) become

$$\begin{aligned} \dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \quad \tilde{A} = \tilde{A}^T, & (5.16)\\ y(t) &= \tilde{B}^T\tilde{x}(t), & (5.17)\\ \tilde{A} &= E^{-\frac{1}{2}}AE^{-\frac{1}{2}}, \quad \tilde{B} = E^{-\frac{1}{2}}B. & (5.18)\end{aligned}$$

TBR for symmetric systems is simpler than for non-symmetric systems. The controllability gramian is equal to the observability gramian for symmetric systems since equations (1.50) and (1.51) are the same when $A = A^T$ and $C = B^T$. Hence, there is no need for the balancing SVD in step 2 of algorithm 3.

TBR for symmetric systems simply solves,

$$AP + PA + BB^T = 0, \tag{5.19}$$

for the single system gramian $P(= Q)$, and finds P's k dominant eigenvectors,

$$\{u_1^{gram}, \cdots, u_k^{gram}\}, \tag{5.20}$$

where

$$\begin{aligned} P &= [u_1^{gram}, \cdots, u_n^{gram}]\Sigma^{gram}([u_1^{gram}, \cdots, u_n^{gram}])^T, & (5.21)\\ \Sigma^{gram} &= diag(\sigma_1, \cdots, \sigma_n),\ \sigma_1 \geq \cdots \geq \sigma_k > \sigma_{k+1} \geq \cdots \geq \sigma_n. & (5.22)\end{aligned}$$

The left and right projection matrices U_k and V_k are chosen to be equal, and

$$U_k = V_k = [u_1^{gram}, \cdots, u_k^{gram}] := U_k^{gram}. \tag{5.23}$$

The system in (5.10-5.11) is reduced according to

$$A_k^{tbr} = (U_k^{gram})^T AU_k^{gram}, \quad B_k^{tbr} = (U_k^{gram})^T B. \tag{5.24}$$

Because the symmetric system in (5.10-5.11) is already balanced, the k dominant left singular vectors of an approximately optimal low rank Cholesky factor can simply be used in place of U_k^{gram}, to obtain 'Approximate TBR', given as algorithm 4.

The k dominant left singular vectors of Z_J are easy to find because Z_J has only J columns. If Z_J is exactly an optimal rank J Cholesky factor of P, $Z_JZ_J^T = P_J^{opt}$, then algorithm 4

Algorithm 4 Approximate TBR for Symmetric Systems

1. Compute $Z_J \in \mathbb{R}^{n\times J}$, $Z_J Z_J^T \approx P_J^{opt}$.
2. Find U_k, $k \leq J$, the matrix of the k dominant left singular vectors of Z_J.
3. Reduction: $A_k^r = (U_k)^T A U_k, \quad B_k^r = (U_k)^T B$.
   (Using U_k to approximate U_k^{gram}).

produces exactly the order k TBR reduction.

5.4 Non-sy etric syste s

The controllability and observability gramians of a non-symmetric system will not, in general, be equal. This section examines how to reduce a system if only approximately optimal low rank Cholesky factors of P and Q are available.

5.4.1 Low ank Square oot method

An idea that was proposed in [41] and [46], is to simply replace the exact Cholesky factors Z^B and Z^C, (possibly of full or, at least, high rank), in algorithm 3 by low rank Cholesky factors, $Z_{J_B}^B \in \mathbb{R}^{n\times J_B}$ and $Z_{J_C}^C \in \mathbb{R}^{n\times J_C}$. This reduces step 2 of the Square Root method to the SVD of a small, $J_C \times J_B$, matrix, which is much less work than the SVD of the full $n \times n$ exact Cholesky factor product $(Z^B)^T Z^C$. This idea, the Low Rank Square Root method, is shown as algorithm 5.

Algorithm 5 Low rank square root method

1. Compute $Z_{J_B}^B \in \mathbb{R}^{n\times J_B}$, $Z_{J_B}^B (Z_{J_B}^B)^T \approx P_{J_B}^{opt}$,
2. Compute $Z_{J_C}^C \in \mathbb{R}^{n\times J_C}$, $Z_{J_C}^C (Z_{J_C}^C)^T \approx Q_{J_C}^{opt}$,
3. Compute reduced system (A_k^r, B_k^r, C_k^r), $k \leq J_B, J_C$, by algorithm 3 using approximate Cholesky factors $Z_{J_B}^B$ and $Z_{J_C}^C$.

Even if $Z_{J_B}^B$ and $Z_{J_C}^C$ are optimal rank J_B and J_C Cholesky factors of P and Q, respectively, algorithm 5 will not, in general, produce a good approximation to TBR unless $Z_{J_B}^B (Z_{J_B}^B)^T$ and $Z_{J_C}^C (Z_{J_C}^C)^T$ are fairly accurate approximations to the matrices P and Q themselves. If $J_B, J_C \ll n$, this cannot happen unless P and Q are themselves close to low rank.

For example, if

$$(Z^B_{J_B})^T(Z^C_{J_C}) = 0, \tag{5.25}$$

then algorithm 5 cannot proceed even though the order k TBR reduction via algorithm 3 may be perfectly well defined.

The near low rank assumption on the exact gramians P and Q needs to be met for algorithm 5 to be an efficient and accurate method. Numerical results for the Low Rank Square Root method will be given in section 5.5.

5.4.2 Dominant Gramian Eigenspaces method

When P and Q are not close to low rank, the Low Rank Square Root method often does not produce a good reduced model. In this case another approach is needed.

In the TBR reduction, gramians P and Q are balanced so that they have the same eigendecomposition, namely, along the coordinate axes, in the same order. Then it makes sense to project the original system onto that single dominant eigenspace of both gramians.

Balancing the gramians requires knowledge of the entire eigenspaces of both gramians. For the situation when only approximately optimal rank J_B and J_C Cholesky factors of P and Q are available, and the rest of the eigenspaces are unknown but significant, the following algorithm is proposed.

The Dominant Gramian Eigenspaces method is an orthogonal projection method, and its projection space is the column span of the union of a subset of the dominant left singular vectors $Z^B_{J_B}$, and a subset of the dominant left singular vectors of $Z^C_{J_C}$ [34].

5.4.3 A Special case

The following theorem gives a condition under which both algorithms 5 and 6 will produce exactly the order k TBR reduction.

Theorem 1. *If the span of the k most controllable modes is the same as the span of the k most observable modes, and $\sigma^B_k > \sigma^B_{k+1}$, $\sigma^C_k > \sigma^C_{k+1}$, where $\sigma^B_1, \cdots, \sigma^B_n$ are the singular values of P in non-increasing order, and $\sigma^C_1, \cdots, \sigma^C_n$ are the singular values of Q in non-increasing order, and if $Z^B_{J_B}$ and $Z^C_{J_C}$, $J_B, J_C, \geq k$, in algorithms 5 and 6 are optimal rank J_B and J_C Cholesky factors of P and Q, then both algorithms 5 and 6 will produce exactly the order k TBR reduction.*

Algorithm 6 Dominant Gramian Eigenspaces method

1. Compute $Z^B_{J_B}$, $Z^B_{J_B}(Z^B_{J_B})^T \approx P^{opt}_{J_B}$.

2. Compute $Z^C_{J_C}$, $Z^C_{J_C}(Z^C_{J_C})^T \approx Q^{opt}_{J_C}$.

3. Calculate SVD: $Z^B_{J_B} = U^B_{n\times J_B} D^B_{J_B\times J_B}(V^B_{J_B\times J_B})^T, \quad Z^C_{J_C} = U^C_{n\times J_C} D^C_{J_C\times J_C}(V^C_{J_C\times J_C})^T$.

4. Choose $k \leq J_B, J_C$, $2k$ being the desired reduction order, and let

$$U^{ctob}_m = qr\left(\left[U^B_{n\times J_B}(:,1:k), \quad U^C_{n\times J_C}(:,1:k)\right]\right). \tag{5.26}$$

Note $k \leq m = rank(U^{ctob}_m) \leq 2k$.

5. Reduce the system:

$$A^r_m = (U^{ctob}_m)^T A U^{ctob}_m, \quad B^r_m = (U^{ctob}_m)^T B, \quad C^r_m = C U^{ctob}_m. \tag{5.27}$$

Proof. Let P and Q have SVDs,

$$\begin{aligned}
P &= U^B(\Sigma^B)^2(U^B)^T, && (5.28)\\
U^B &= [u^B_1, \cdots, u^B_n], \quad \sigma^B_1 \geq \cdots \geq \sigma^B_k > \sigma^B_{k+1} \geq \cdots \geq \sigma^B_n \geq 0, && (5.29)\\
Q &= U^C(\Sigma^C)^2(U^C)^T, && (5.30)\\
U^C &= [u^C_1, \cdots, u^C_n], \quad \sigma^C_1 \geq \cdots \geq \sigma^C_k > \sigma^C_{k+1} \geq \cdots \geq \sigma^C_n \geq 0. && (5.31)
\end{aligned}$$

Since the span of the k most controllable modes is the same as the span of the k most observable modes,

$$span\{u^B_1, \cdots u^B_k\} = span\{u^C_1, \cdots u^C_k\}. \tag{5.32}$$

Without loss of generality, assume Z^B, Z^C, exact Cholesky factors, and $Z^B_{J_B}, Z^C_{J_C}$, optimal rank J_B and J_C Cholesky factors, have the following form,

$$Z^B = [u^B_1, \cdots, u^B_n]\begin{bmatrix}\sigma^B_1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \sigma^B_n\end{bmatrix}, \quad Z^B_J = [u^B_1, \cdots, u^B_J]\begin{bmatrix}\sigma^B_1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \sigma^B_J\end{bmatrix}, \tag{5.33}$$

$$Z^C = [u^C_1, \cdots, u^C_n]\begin{bmatrix}\sigma^C_1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \sigma^C_n\end{bmatrix}, \quad Z^C_J = [u^C_1, \cdots, u^C_J]\begin{bmatrix}\sigma^C_1 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \sigma^C_J\end{bmatrix}. \tag{5.34}$$

Otherwise, they differ from the above forms only by right multiplication by orthogonal matrices, which will cancel out when defining projection matrices in (4.4) and (4.5).

Because of (5.32),

$$
\begin{aligned}
Z_C^T Z_B &= \begin{bmatrix} \sigma_1^C & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^C \end{bmatrix} \begin{bmatrix} (u_1^C)^T \\ \vdots \\ (u_n^C)^T \end{bmatrix} [u_1^B, \cdots, u_n^B] \begin{bmatrix} \sigma_1^B & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^B \end{bmatrix}, && (5.35) \\
&= \begin{bmatrix} \Sigma_k^C & 0 \\ 0 & \Sigma_{n-k}^C \end{bmatrix} \begin{bmatrix} (U_k^C)^T U_k^B & 0 \\ 0 & (U_{n-k}^C)^T U_{n-k}^B \end{bmatrix} \begin{bmatrix} \Sigma_k^B & 0 \\ 0 & \Sigma_{n-k}^B \end{bmatrix}, && (5.36) \\
&:= \begin{bmatrix} W_k & 0 \\ 0 & W_{n-k} \end{bmatrix} && (5.37)
\end{aligned}
$$

is $(k, n-k)$ block diagonal. The matrices $(U_k^C)^T U_k^B \in \mathbb{R}^{k\times k}$, $(U_{n-k}^C)^T U_{n-k}^B \in \mathbb{R}^{n-k\times n-k}$ are both orthogonal. Let $W_k = U_k \Sigma_k V_k^T$, and $W_{n-k} = U_{n-k}\Sigma_{n-k}V_{n-k}^T$, be singular value decompositions, then

$$
Z_C^T Z_B = \begin{bmatrix} U_k & 0 \\ 0 & U_{n-k} \end{bmatrix} \begin{bmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{n-k} \end{bmatrix} \begin{bmatrix} V_k^T & 0 \\ 0 & V_{n-k}^T \end{bmatrix}, \tag{5.38}
$$

is a SVD of $Z_C^T Z_B$, and $\sigma_1 \geq \cdots \geq \sigma_k \geq \sigma_k^B \sigma_k^C > \sigma_{k+1}^B \sigma_{k+1}^C \geq \sigma_{k+1} \geq \cdots \geq \sigma_n$.

Therefore, $\left[u_1^R, \cdots, u_k^R\right]$ in (4.4) has zeros in the last $n-k$ rows,

$$
\left[u_1^R, \cdots, u_k^R\right] = \begin{bmatrix} V_k \\ 0 \end{bmatrix}, \tag{5.39}
$$

and the right projection space for TBR is,

$$
col\,sp((S^B)^{tbr}) = col\,sp\left(Z^B \begin{bmatrix} V_k \\ 0 \end{bmatrix}\right) = col\,sp(Z^B(:, 1:k)) = span\{u_1^B, \cdots, u_k^B\}. \tag{5.40}
$$

Similarly, $\left[u_1^L, \cdots, u_k^L\right]$ in (4.5) has zeros in the last $n-k$ rows,

$$
\left[u_1^L, \cdots, u_k^L\right] = \begin{bmatrix} U_k \\ 0 \end{bmatrix}, \tag{5.41}
$$

and the left projection space for TBR is,

$$
col\,sp((S^C)^{tbr}) = col\,sp\left(Z^C \begin{bmatrix} U_k \\ 0 \end{bmatrix}\right) = col\,sp(Z^C(:, 1:k)) = span\{u_1^C, \cdots, u_k^C\}. \tag{5.42}
$$

The same argument, replacing Z^B, Z^C by $Z^B_{J_B}$, $Z^C_{J_C}$, and n by $max(J_B, J_C)$, gives the right and left projection spaces for the Low Rank Square Root method as,

$$col\,sp((S^B)^{lrsqrt}) = col\,sp(Z^B_{J_B}(:, 1:k)) = span\{u^B_1, \cdots, u^B_k\}, \tag{5.43}$$

and

$$col\,sp((S^C)^{lrsqrt}) = col\,sp(Z^C_{J_C}(:, 1:k)) = span\{u^C_1, \cdots, u^C_k\}. \tag{5.44}$$

Thus, TBR and the Low Rank Square Root method have the same projection spaces.

From (5.26),

$$\begin{aligned} col\,sp(U^{ctob}_m) &= qr\left(\left[u^B_1, \cdots, u^B_k, u^C_1, \cdots, u^C_k\right]\right) && (5.45)\\ &= span\{u^B_1, \cdots, u^B_k\} = span\{u^C_1, \cdots, u^C_k\}, && (5.46) \end{aligned}$$

and $m = k$. Thus, TBR and the Dominant Gramian Eigenspace method have the same projection spaces.

Therefore, all three methods produce equivalent reduced systems. □

5.5 u erical results

This section gives numerical results for algorithms 5 and 6 for non-symmetric systems, when optimal low rank Cholesky factors are used.

Figures 5-2 and 5-3 show an example of a non-symmetric system which resulted from the discretization of the transmission line shown in figure 5-1. The non-symmetric system matrix A is 256×256, and the system is single-input single-output.

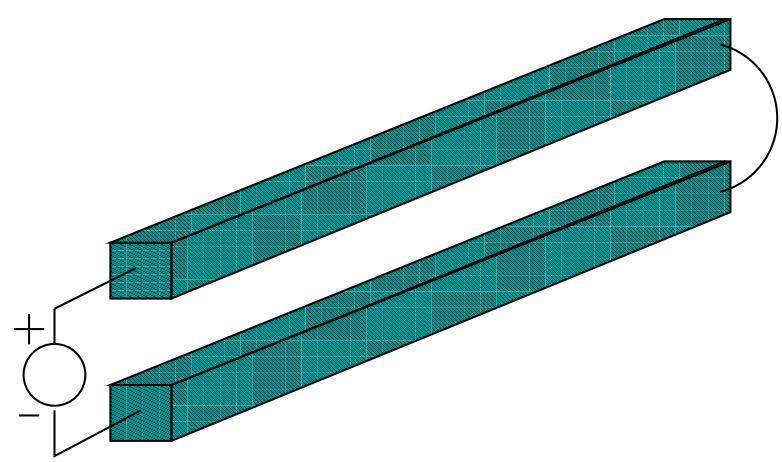


Figure 5-1: Transmission line.

Figure 5-2 shows the absolute value of the frequency responses, $|G(jw)|$, of the original

system and various reduced systems.

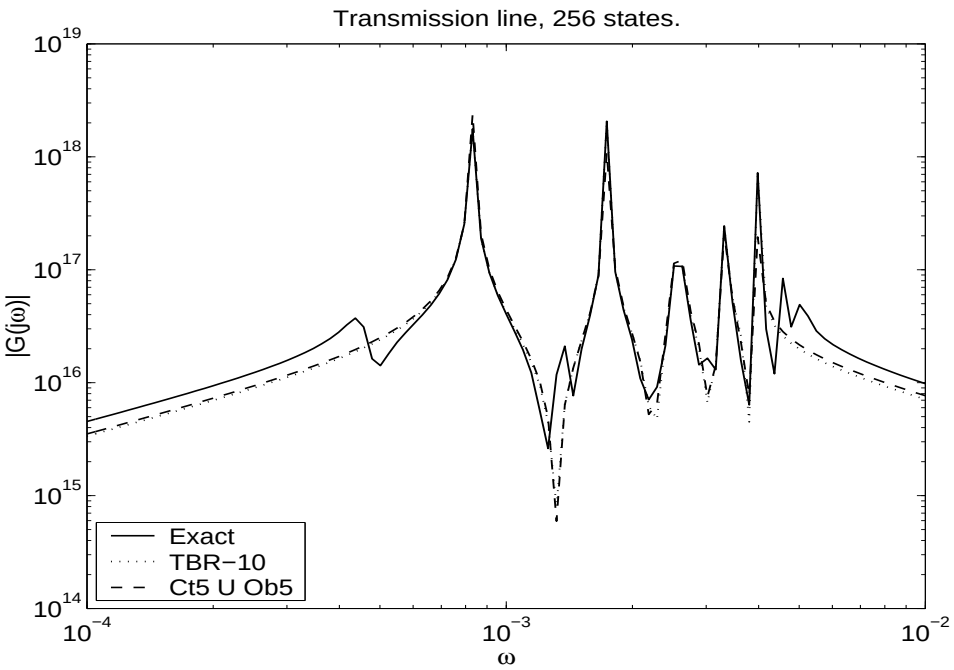
In figure 5-2(a), an order 10 reduced model obtained via the Dominant Gramian Eigenspaces method, 'Ct5 U Ob5', is compared to the order 10 reduced model from TBR, 'TBR-10'. The abbreviation 'Ct5 U Ob5' means that the column span of the union of the 5 most controllable modes and the 5 most observable modes is used as the projection space. In this case, the projection space has dimension 10. Optimal rank 5 Cholesky factors of P and Q are needed to produce the reduced model. It can be seen that the frequency response of reduced model from the Dominant Gramian Eigenspaces method is almost indistinguishable from the frequency response of the order 10 TBR reduced model.

In figure 5-2(b), order 10 and order 20 models obtained from the Low Rank Square Root method are shown as 'LR-sqrt-10', and 'LR-sqrt-20'. The order 10 model is obtained by balancing optimal rank 10 Cholesky factors of P and Q, the order 20 model by balancing optimal rank 20 Cholesky factors. The order 10 model from Low Rank Square Root is not a good approximation. Its system matrix also has many unstable eigenvalues. 'LR-sqrt-20' is a better approximation, with similar accuracy as 'Ct 5 U Ob5'. However, 'Ct 5 U Ob5' needs only two rank 5 Cholesky factors, whereas 'LR-sqrt-20' needs two rank 20 Cholesky factors.

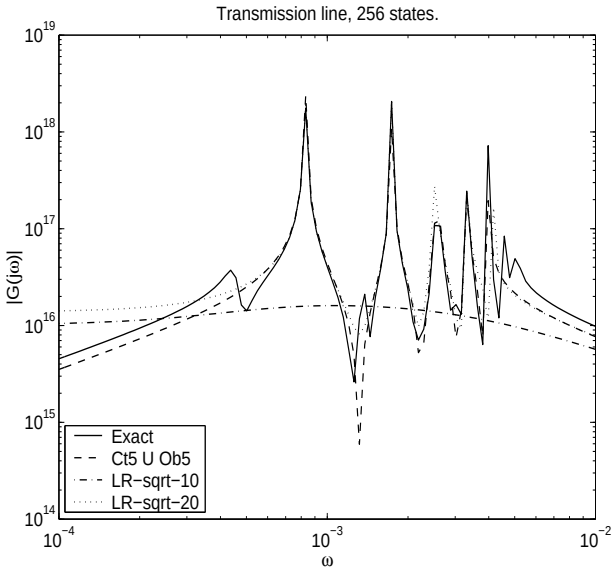
Figure 5-2(c) compares 'Ct 5 U Ob5' with projection by either the column span of the 10 most controllable modes, 'Ct-10', or by the column span of the 10 most observable modes, 'Ob-10'. Both 'Ct-10' and 'Ob-10' only need one rank 10 Cholesky factor. Neither 'Ct-10' nor 'Ob-10' comes close to capturing the frequency response behavior of the original system as well as using the union of 5 and 5.

Figures 5-3 shows that the dominant controllable and dominant observable modes are 'far' from each other. Figure 5-3(a) plots the projection of the observable modes onto the 10 most controllable modes, $\|(u_j^{ob})^T[u_1^{ct}, \cdots, u_{10}^{ct}]\|_2$. All are unit vectors. It can be seen that the 20 most observable modes have very little component in the span of the 10 most controllable modes, less than 0.01. Figure 5-3(b) shows a similar situation with the projection of the controllable modes onto the 10 most observable modes.

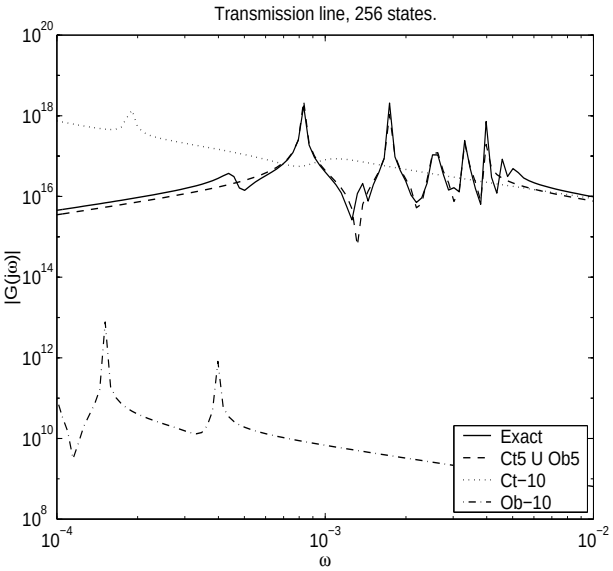
When the dominant controllable modes and the dominant observable modes are nearly orthogonal, and when the remaining eigenspace of either P or Q is not small, the Low Rank Square Root method does not produce good results. In that case, it is better to use the Dominant Gramian Eigenspaces method.



(a) CTOB close to TBR

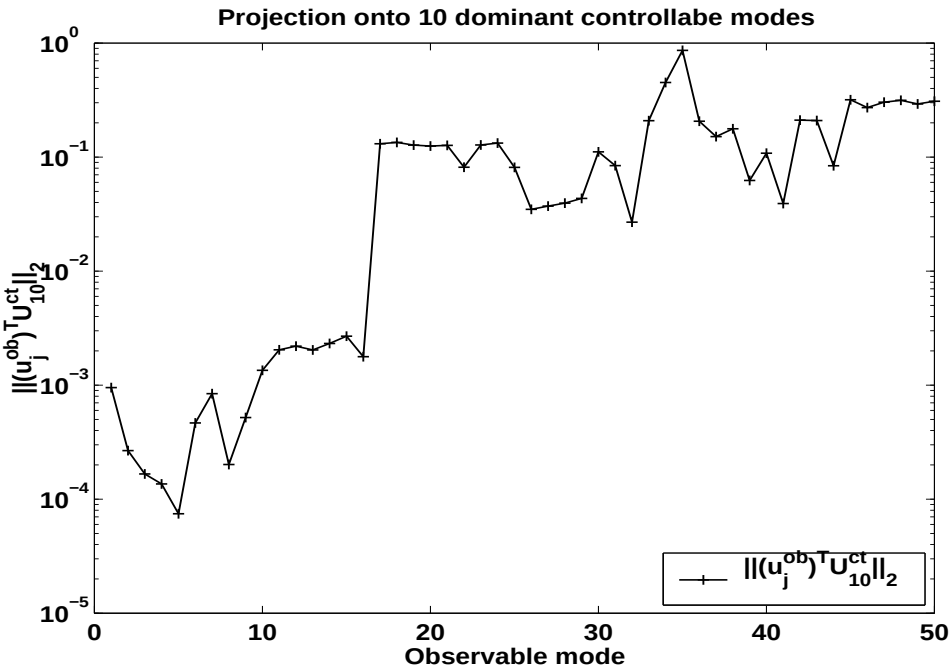


(b) CTOB vs LRSQRT

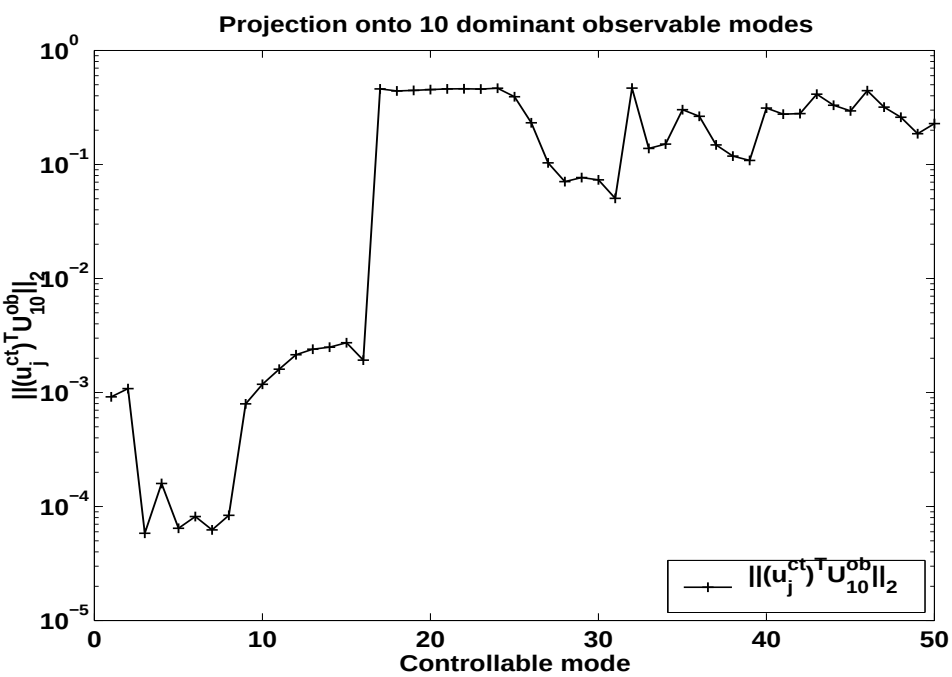


(c) CT and OB vs CTOB

Figure 5-2: Low Rank Square Root and Dominant Gramian Eigenspaces methods



(a) Projection of u_j^{ob} on U_{10}^{ct}



(b) Projection of u_j^{ct} on U_{10}^{ob}

Figure 5-3: Mutual projection of dominant controllable and dominant observable modes

r 6

l i i l

r l s s

This chapter contains a main theoretical result of this dissertation, given as theorem 2, which characterizes the different manifestations of the range of the solution to

$$AX + XA^T = -BB^T \tag{6.1}$$

as order n Krylov and rational Krylov subspaces with different starting vectors.

Proposition 9. *Let X be the solution to (6.1), then*

$$Range(X) = span\{B, AB, \cdots, A^{n-1}B\} = \mathcal{K}_n(A, B). \tag{6.2}$$

Proof. See [50]. □

The definition of a rational Krylov subspace is given below.

Definition 9. *An order m rational Krylov subspace $\mathcal{K}_m^{rat}(A, z_1, \mathbf{p_{m-1}})$, $A \in \mathbb{R}^{n\times n}, z_1 \in \mathbb{R}^n$, $\mathbf{p_{m-1}} = \{p_1, \cdots, p_{m-1}\}$, $p_i \in \mathbb{R}$, is the subspace,*

$$\mathcal{K}_m(A, z_1, \mathbf{p_{m-1}}) := span\left\{z_1, (A + p_1 I)^{-1} z_1, (A + p_2 I)^{-1}(A + p_1 I)^{-1} z_1, \cdots, \prod_{i=1}^{m-1}(A + p_i)^{-1} z_1\right\}. \tag{6.3}$$

Note that $dim(\mathcal{K}_m^{rat}(A, z_1, \mathbf{p_{m-1}}) \leq m$.

A main result of this dissertation is theorem 2, which shows the equivalence of an infinite number of order n Krylov and rational Krylov subspaces based on A and B.

Theorem 2. *Let A be invertible, $B \in \mathbb{R}^n$, and define the subspace $\mathcal{L}(A, B, \mathbf{p})$, $\mathbf{p} = \{\cdots, p_{-2}, p_{-1}, p_0, p_1, p_2 \cdots\}$, $p_i \in \mathbb{R}$, as*

$$\begin{aligned}
&\mathcal{L}(A, B, \mathbf{p}) \\
&:= span \quad \cdots, \quad \prod_{i=-j}^{-1} (A + p_i I)^{-1} B, \quad \cdots, \quad (A + p_{-2} I)^{-1} (A + p_{-1} I)^{-1} B, \\
&\qquad (A + p_{-1} I)^{-1} B, \quad B, \quad (A + p_0 I) B, \\
&\qquad (A + p_1 I)(A + p_0 I)^{-1} B, \quad \cdots, \quad \prod_{i=1}^{j} (A + p_i I) B, \quad \cdots \quad , \\
&= span\Big\{ \cdots, v_{-j}(\mathbf{p}), \cdots, v_{-2}(\mathbf{p}), \quad v_{-1}(\mathbf{p}), \quad v_0(\mathbf{p}), \quad v_1(\mathbf{p}), \quad v_2(\mathbf{p}), \quad \cdots, v_j(\mathbf{p}), \cdots \Big\},
\end{aligned} \tag{6.4, 6.5}$$

where

$$v_0(\mathbf{p}) = B, \quad v_j(\mathbf{p}) = \prod_{i=0}^{j-1} (A + p_i I) B, \ j > 0, \quad v_j(\mathbf{p}) := \prod_{i=j}^{-1} (A + p_i I)^{-1} B, \ j < 0, \tag{6.6}$$

and where all matrix inverses in (6.4) are well-defined. Then $\forall s$, $\forall \mathbf{p}$, $\forall \mathbf{r} = \{\cdots, r_{-1}, r_0, r_1, \cdots\}$, $\forall \mathbf{q} = \{\cdots, q_{-1}, q_0, q_1, \cdots\}$,

$$\begin{aligned}
\mathcal{L}(A, B, \mathbf{p}) &= span\{v_s(\mathbf{p}), \quad v_{s+1}(\mathbf{p}), \quad v_{s+2}(\mathbf{p}), \quad \cdots, v_{s+(n-1)}(\mathbf{p})\} && (6.7) \\
&= span\{B, AB, \quad \cdots, \quad A^{n-1} B\} && (6.8) \\
&= \mathcal{L}(A, v_s(\mathbf{r}), \mathbf{q}). && (6.9)
\end{aligned}$$

$\mathcal{L}(A, B) := \mathcal{L}(A, B, \mathbf{p})$ may be written without referring to the shifts.

The proof of theorem 2 needs the following lemmas. The dependence of the v_i's on $\mathbf{p}$ will be suppressed in the proofs unless needed.

Lemma 1. *If $m > n$, then $\mathcal{K}_m(A, B) = \mathcal{K}_n(A, B)$.*

Proof. First, it is shown that if $m > n$, then $A^{m-1} B \in \mathcal{K}_{m-1}(A, B)$. If $m > n$, there exist coefficients, $c_0, \cdots, c_{m-1}$, not all zero, such that

$$c_0 B + c_1 AB + \cdots + c_{m-2} A^{m-2} B + c_{m-1} A^{m-1} B = 0. \tag{6.10}$$

Choose $0 \le j \le m - 1$ such that $c_j \neq 0$, and $c_i = 0, \forall i > j$, then

$$c_0 A^{m-1-j} B + \cdots + c_j A^{m-1-j} A^j B = 0, \Longrightarrow c_j A^{m-1} B = -c_0 A^{m-1-j} B + \cdots + c_{j-1} A^{m-2} B. \tag{6.11}$$

Hence, $A^{m-1}B \in \mathcal{K}_{m-1}(A,B)$. Therefore, if $m > n$, $K_m(A,B) = K_{m-1}(A,B)$, and finally, $\mathcal{K}_m(A,B) = \mathcal{K}_{m-1}(A,B) = \cdots = \mathcal{K}_{n+1}(A,B) = \mathcal{K}_n(A,B)$. □

The order n Krylov subspace is also referred to simply as the Krylov subspace, $\mathcal{K}_n(A,B) := \mathcal{K}(A,B)$, without the subscript.

Lemma 2. *With the v_i's defined as (6.4),*

$$v_l \in span\{v_s, \quad v_{s+1}, \quad v_{s+2}, \quad \cdots, v_{s+(n-1)}\}, \tag{6.12}$$

whenever $l > s + (n-1)$.

Proof. From (6.4), it can be seen that,

$$v_i = (A + p_{i-1}I)v_{i-1}, \quad \forall i, \quad \Longrightarrow v_i \in span\{v_{i-1}, Av_{i-1}\}, \tag{6.13}$$

and therefore,

$$span\{v_s, \quad v_{s+1}, \quad v_{s+2}, \quad \cdots, v_l\} = span\{v_s, Av_s, \cdots, A^{l-s}v_s\} = \mathcal{K}_{l-s+1}(A, v_s). \tag{6.14}$$

From lemma 1,

$$\begin{aligned} span\{v_s, \quad v_{s+1}, \quad v_{s+2}, \quad \cdots, v_l\} &= \mathcal{K}_{l-s+1}(A, v_s) \\ &= \mathcal{K}_n(A, v_s) = span\{v_s, \quad v_{s+1}, \quad v_{s+2}, \quad \cdots, v_{s+(n-1)}\}. \end{aligned} \tag{6.15}$$

The result follows. □

Lemma 3. *With the v_i's defined as (6.4),*

$$v_l \in span\{v_s, v_{s+1}, v_{s+2}, \cdots, v_{s+(n-1)}\}, \tag{6.16}$$

whenever $l < s$.

Proof. First show that the lemma is true for $l = s-1$. Equivalently, because of (6.14), show that

$$(A + p_{s-1}I)^{-1}v_s \in span\{v_s, Av_s, \cdots, A^{n-1}v_s\}. \tag{6.17}$$

Shifts can be added in the right hand side of (6.17),

$$span\{v_s, Av_s, \cdots, A^{n-1}v_s\} = span\{v_s, (A + p_{s-1}I)v_s, \cdots, (A + p_{s-1}I)^{n-1}v_s\}, \tag{6.18}$$

without affecting its column span. Because $\{v_{s-1}, v_s, \cdots, v_{s+(n-1)}\}$ are $n+1$ vectors in $\mathbb{R}^n$, there exist coefficients, $c_0, \cdots, c_n$, not all zero, such that,

$$c_0 v_s + c_1(A + p_{s-1}I)v_s + \cdots + c_{n-1}(A + p_{s-1}I)^{n-1}v_s + c_n(A + p_{s-1}I)^{-1}v_s = 0, \tag{6.19}$$

If $c_n \neq 0$, (6.17) is proven.

Otherwise, choose $0 \leq j < n$ such that $c_j \neq 0$, and $c_i = 0, \forall i < j$. Then multiply (6.19) by $(A + p_{s-1}I)^{-(j+1)}$, to obtain

$$\begin{aligned} &c_j(A + p_{s-1}I)^{-1}v_s + c_{j+1}v_s + \cdots + c_{n-1}(A + p_{s-1}I)^{n-2-j}v_s = 0, \\ &\Longrightarrow c_j(A + p_{s-1}I)^{-1}v_s = -c_{j+1}v_s - \cdots - c_{n-1}(A + p_{s-1}I)^{n-2-j}v_s. \end{aligned} \tag{6.20}$$

Thus, (6.17) is proven, and (6.16) holds for $l = s-1$. If $l < s-1$,

$$\begin{aligned} v_l &\in span\{v_{l+1}, v_{l+2}, \cdots, v_{l+n}\} && (6.21)\\ &\subseteq span\{v_{l+2}, \cdots, v_{l+n+1}\} && (6.22)\\ &\vdots && (6.23)\\ &\subseteq span\{v_s, \cdots, v_{s+n-1}\}. && (6.24) \end{aligned}$$

Line (6.22) follows because each vector $v_{l+1}, \cdots, v_{l+n}$ is in $span\{v_{l+2}, \cdots, v_{l+n+1}\}$. □

Proof of theorem 2. Lemmas 2 and 3 show that

$$\mathcal{L}(A, B, \mathbf{p}) = span\{v_s(\mathbf{p}), \quad v_{s+1}(\mathbf{p}), \quad v_{s+2}(\mathbf{p}), \quad \cdots, v_{s+(n-1)}(\mathbf{p})\} \tag{6.25}$$

holds for all s and for all $\mathbf{p}$. (6.8) follows from

$$span\{v_0(\mathbf{p}), v_1(\mathbf{p}), \cdots, v_{n-1}(\mathbf{p})\} = span\{B, AB, \cdots, A^{n-1}B\}, \tag{6.26}$$

with the choice of $s = 0$, and $\mathbf{p} \equiv 0$. (6.9) follows from

$$\begin{aligned} \mathcal{L}(A, B, \mathbf{p}) &= \mathcal{L}(A, B, \mathbf{r}) && (6.27)\\ &= span\{v_s(\mathbf{r}), v_{s+1}(\mathbf{r}), \cdots, v_{s+(n-1)}(\mathbf{r})\} && (6.28)\\ &= span\{v_s(\mathbf{r}), Av_s(\mathbf{r}), \cdots, A^{n-1}v_s(\mathbf{r})\} && (6.29)\\ &= \mathcal{L}(A, v_s(\mathbf{r}), \mathbf{q}), \quad \forall \mathbf{p}, \quad \forall \mathbf{r}, \quad \forall \mathbf{q}. && (6.30) \end{aligned}$$

□

Corollary 1. *With the same notation as in theorem 2,*

$$\mathcal{L}(A, v_s(\mathbf{r}), \mathbf{q}) = range(X), \quad \forall s, \quad \forall \mathbf{r}, \quad \forall \mathbf{q}, \tag{6.31}$$

where X is the solution to (6.1).

Theorem 2 and corollary 1 can be taken to mean that to find the range of X, one can choose any starting vector v_s of the form,

$$v_s(\mathbf{r}) := B, \quad \text{or} \quad v_s(\mathbf{r}) := \prod_{i=1}^{j} (A + r_i I)B, \quad \text{or} \quad v_s(\mathbf{r}) := \prod_{i=1}^{j} (A + r_i I)^{-1}B, \tag{6.32}$$

for any $r_1, \cdots, r_j$, and let the remaining basis vectors $\{v_s, v_{s+1}, \cdots, v_{s+n-1}\}$ satisfy

$$v_i = (A + q_{i-s}I)v_{i-1}, \quad i = s+1, \cdots, s+n-1, \tag{6.33}$$

for any choice of $q_1, \cdots, q_{n-1}$.

To emphasize the choice of basis, the various manifestations of the space $\mathcal{L}(A, B, \mathbf{p})$ will be written as $\mathcal{L}_n(A, v_s(\mathbf{r}), \mathbf{q_{n-1}})$, if its basis representation satisfies (6.33). The vector of shifts $\mathbf{q_{n-1}} = \{q_1, \cdots, q_{n-1}\}$ now has only $n-1$ numbers.

Since only B and not any other v_s is given, if the starting vector is B, $v_s = B$, or powers of shifts of A multiplied by B, $v_s(\mathbf{r}) = \prod_{i=1}^{j}(A + r_i I)B$, then (6.33) is an efficient way to compute the basis $\{v_s, \cdots, v_{s+n-1}\}$. If $v_s(\mathbf{r}) = \prod_{i=1}^{j}(A + r_i I)^{-1}B$, and $j \geq n-1$, then it is more efficient to find the basis in reverse order, and choose $\mathbf{q_{n-1}} = \{r_1, \cdots, r_{n-1}\}$ so the shifts of A cancel out. The final vector is

$$v_{s+n-1} = \prod_{i=n}^{j} (A + r_i I)^{-1}B, \text{ if } j > n-1, \quad \text{or} \quad v_{s+n-1} = B, \text{ if } j = n-1, \tag{6.34}$$

and the rest of the basis is calculated according to

$$v_{i-1} = (A + r_{i-s}I)^{-1}v_i, \quad i = s+n-1, \cdots, s+1. \tag{6.35}$$

If $\{v_s, v_{s+1}, \cdots, v_{s+n-1}\}$ contains both vectors which are positive powers of shifts of A multiplied by B, and vectors which are inverse powers of shifts of A multiplied by B, the basis should be computed in two parts. One starts with B and finds a subset of the basis by multiplication by shifts of A, and then finds the remaining basis vectors by multiplication by inverses of shifts of A.

If $\mathcal{L}_n(A, v_s(\mathbf{r}), \mathbf{q_{n-1}})$'s basis contains only vectors which are positive powers of shifts of A multiplied by B, $\mathcal{L}_n(A, v_s(\mathbf{r}), \mathbf{q_{n-1}})$ will be denoted $\mathcal{K}_n^{sh}(A, v_s(\mathbf{r}), \mathbf{q_{n-1}})$, which is a Krylov subspace, but with $n-1$ shifts. If $\mathcal{L}_n(A, v_s(\mathbf{r}), \{r_1, \cdots, r_{n-1}\})$ contains only vectors which are inverse powers of shifts of A multiplied by B, then it is actually the rational Krylov subspace, $\mathcal{K}_n^{rat}(A, v_s(\mathbf{p}), \mathbf{q_{n-1}})$, where $\mathbf{q_{n-1}} = \{r_{n-1}, \cdots, r_1\}$, and $\mathbf{p} = \{r_n \cdots, r_j\}$

Quite simply, what $\mathcal{K}_n^{sh}(A, v_s(\mathbf{r}), \mathbf{q_{n-1}})$ means is that the basis $\{w_1, \cdots, w_n\}$ is obtained in the following way,

$$w_1 := v_s(\mathbf{r}) = \prod_{i=1}^{l} (A + r_i I)B, \quad l > 0, \quad \text{or} \quad w_1 := B, \quad l = 0, \tag{6.36}$$

$$w_i = (A - q_{i-1} I) w_{i-1}, \quad i = 2, \cdots, n. \tag{6.37}$$

Furthermore, $\mathcal{K}_n^{rat}(A, v_s(\mathbf{p}), \mathbf{q_{n-1}})$ means that the basis $\{w_1, \cdots, w_n\}$ is obtained thus,

$$w_1 := v_s(\mathbf{p}) = \prod_{i=1}^{l} (A + p_i I)^{-1} B, \quad l > 0, \quad \text{or} \quad w_1 := B, \quad l = 0, \tag{6.38}$$

$$w_i = (A - q_{i-1} I)^{-1} w_{i-1}, \quad i = 2, \cdots, n. \tag{6.39}$$

The following theorem gives a different characterization of $\mathcal{K}_J^{rat}(A, (A+p_1 I)^{-1}B, \{p_2, \cdots, p_J\})$ as the sum of m Krylov subspaces, where m is the number of distinct parameters in the list $\{p_1, \cdots, p_n\}$.

Theorem 3. *Let $\mathcal{K}_J^{rat}(A, (A + p_1 I)^{-1}B, \{p_2, \cdots, p_J\})$ be such that no $(A + p_i I)$ is singular, then*

$$\mathcal{K}_J^{rat}(A, (A + p_1 I)^{-1}B, \{p_2, \cdots, p_J\}), \tag{6.40}$$

$$= span \quad (A - p_1 I)^{-1}B, \cdots, \prod_{i=1}^{j} (A - p_i I)^{-1}B, \cdots, \prod_{i=1}^{J} (A - p_i I)^{-1}B \quad , \tag{6.41}$$

$$= \prod_{i=1}^{m} span\{(A - p_i I)^{-1}B, \cdots, (A - p_i I)^{-i_s}B\} \tag{6.42}$$

$$= \prod_{i=1}^{m} \mathcal{K}_{i_s}\left((A - p_i I), (A - p_i I)^{-1}B\right), \tag{6.43}$$

where $1_s + \cdots + m_s = n$, and each p_i appears in $\{p_1, \cdots, p_n\}$ a total of i_s times.

Proof. By partial fraction expansion. □

Theorem 3 will be used in chapter 10 to prove moment matching results.

r 7

Eq i s

This chapter describes several existing methods for finding or approximating the solution to the Lyapunov equation,

$$AX + XA^T = -BB^T, \quad \lambda_i(A) < 0, \forall i, \tag{7.1}$$

including the iterative Alternating Direction Implicit (ADI) method [2, 57] in some detail.

7.1 revious et o s

The Bartels-Stewart method [1], the Hammarling method [23], and the Alternating Direction Implicit (ADI) method [2, 57, 59] described in this chapter are appropriate for Lyapunov equations with a small, dense matrix A. They require matrix decompositions and have $O(n^3)$ complexity. Low rank approximations to the solution X were formulated in [25, 27].

7.1.1 Bartels-Stewart method

A well-known, exact method to solve Lyapunov equations is the Bartels-Stewart method [1]. It first transforms A to real Schur form, and then back solves for the solution of the transformed Lyapunov equation. The solution X is then obtained by a congruence transformation. Reducing a general, possibly sparse matrix to real Schur form requires $O(n^3)$ work, as does the congruence transformation to produce X. The flop count for the Bartels-Steward method calculated in [36] is $15n^3$.

7.1.2 Hammarling method

The Hammarling method [23] is another exact method which first transforms A to Schur form. It calculates the Cholesky factor of the solution X rather than X itself. It also has $O(n^3)$ complexity.

7.1.3 Low rank methods

In [25, 27], low rank approximations to the solution to (7.1) were proposed of the form

$$X \approx V_m X_m V_m^T, \tag{7.2}$$

where the columns of V_m form an orthonormal basis for the block Krylov subspace $\mathcal{K}_m(A, B)$,

$$col\,sp(V_m) = \mathcal{K}_m(A, B) = col\,sp[B, AB, A^2B, \cdots, A^{m-1}B]. \tag{7.3}$$

The columns of V_m are obtained via the block Arnoldi process with A and B. The matrix $X_m \in \mathbb{R}^{mp \times mp}$ is obtained by solving a smaller, order mp, matrix equation.

The residual of (7.1) is defined as

$$R_m(X_m) := A(V_m X_m V_m^T) + (V_m X_m V_m^T)A^T + BB^T. \tag{7.4}$$

The smaller Lyapunov equation that needs to be solved to satisfy a Galerkin condition on the residual $R_m(X_m)$ was found in both [25] and [27]. In [27] the smaller matrix equation that needs to be solved to satisfy a minimum residual condition on the residual was also given.

The block Arnoldi algorithm [60] is given here as algorithm 7.

Algorithm 7 Block Arnoldi algorithm

1. $B = Q_1 R_1$ (QR factorization), $p_1 :=$ number of columns of Q_1 .

FOR $i = 1 : m$

2a. $V_i = [Q_1, Q_2, \cdots, Q_i]$,

2b. Compute $\begin{bmatrix} A_{1i} \\ A_{2i} \\ \vdots \\ A_{ii} \end{bmatrix} = V_i^T A Q_i$.

2c. $Q_{i+1} A_{i+1,i} = AQ_i - \sum_{k=1}^{i} Q_k A_{ki}$ (QR factorization), $p_{i+1} :=$ number of columns of Q_{i+1}.

END

Let $A_{m\times m} \in \mathbb{R}^{mp\times mp}$, $V_m \in \mathbb{R}^{n\times mp}$, $B_m \in \mathbb{R}^{mp\times p}$ be the quantities obtained via algorithm 7 such that,

$$\begin{aligned} B &= V_m B_m, && (7.5)\\ AV_m &= V_m A_{m\times m} + V_{m+1} A_{m+1,m} E_m^T, && (7.6)\\ A_{m\times m} &= V_m^T A V_m. && (7.7) \end{aligned}$$

Here, $A_{m\times m}$ is a block upper-Hessenberg matrix, the columns of V_m form an orthonormal

basis for $\mathcal{K}_m(A, B)$, and E_m is the matrix formed by the last p columns of the $mp \times mp$ identity matrix. If $\lambda_i(A_{m\times m}) + \bar{\lambda}_j(A_{m\times m}) \neq 0$ for all i, j, ensuring an unique solution to (7.9) exists, then

$$V_m^T R_m(X_m) V_m = 0 \tag{7.8}$$

if and only if X_m satisfies

$$A_{m\times m} X_m + X_m A_{m\times m}^T + B_m B_m^T = 0. \tag{7.9}$$

Equation (7.9) is the order mp Lyapunov equation that needs to be solved to satisfy a Galerkin condition on $R_m(X_m)$ [25, 27].

On the other hand, the Frobenius norm of $R_m(X_m)$ is minimized if X_m satisfies

$$\begin{aligned} A_{m\times m}^T(A_{m\times m}X_m + X_m A_{m\times m}^T + B_m B_m^T) + (A_{m\times m}X_m + X_m A_{m\times m}^T + B_m B_m^T)A_{m\times m} \\ + E_m A_{m+1,m}^T A_{m+1,m} E_m^T X_m + X_m E_m A_{m+1,m}^T A_{m+1,m} E_m^T = 0. \end{aligned} \tag{7.10}$$

Equation (7.10) is the order mp linear matrix equation that needs to be solved to satisfy a minimal residual condition on $R_m(X_m)$ [27].

7.2 lternate irection I plicit Iteration

The Alternate Direction Implicit (ADI) method [2, 57–59] is another iterative Lyapunov equation solver, and is given as algorithm 8. It produces the approximation X_J^{adi} to the Lyapunov solution X according to the two step iteration in (7.12-7.13). The parameters $\{p_1, p_2, p_2, \cdots\}, Re\{p_j\} < 0$, are called the ADI parameters.

Algorithm 8 Alternate Direction Implicit algorithm

INPUT: A, B.

00. If $v \mapsto Av, v \in \mathbb{R}^n$, is not $O(n)$ work, tri-diagonalize A,

a. Find $\tilde{A}$ tri-diagonal, such that $\tilde{A} = TAT^{-1}$.

b. Set $\tilde{B} := TB$.

Otherwise, set $\tilde{A} := A, \tilde{B} := B$.

0. Choose ADI parameters, $\{p_1, \cdots, p_{J_{max}}\}$, $Re\{p_i\} < 0$, (real or complex conjugate pairs), according to section 7.2.2 and references, using spectral bounds on $\tilde{A}$.

1. Initial guess,

$$\tilde{X}_0 = 0_{n\times n}. \tag{7.11}$$

FOR $j = 1, 2, \cdots, J$

2. Do

$$(\tilde{A} + p_j I)\tilde{X}_{j-\frac{1}{2}} = -BB^T - \tilde{X}_{j-1}(\tilde{A}^T - p_j I), \tag{7.12}$$

$$(\tilde{A} + p_j I)\tilde{X}_j = -BB^T - \tilde{X}^T_{j-\frac{1}{2}}(\tilde{A}^T - p_j I). \tag{7.13}$$

END

3. If A was tri-diagonalized, recover solution,

$$X^{adi}_J := T^{-1}\tilde{X}_J T^{-T}. \tag{7.14}$$

Otherwise, $X^{adi}_J = \tilde{X}_J$.

OUTPUT: $X^{adi}_J \in \mathbb{R}^{n\times n}$, $X^{adi}_J \approx X$.

Remark 1. *In step 0, the spectral bounds required in section 7.2.2 are easy to find if $\tilde{A}$ is tri-diagonal.*

To keep the final ADI approximation $X^{adi}_{J_{final}}$ real, it is assumed that in the parameter list $\{p_1, p_2, \cdots, p_{J_{final}}\}$, each parameter is either real or comes as a part of a complex conjugate pair. Because $\tilde{A}$ is stable, since A is stable, and $Re\{p_j\} < 0$ for all j, $(\tilde{A}+p_j I)$ is non-singular and solutions to (7.12-7.13) exist for all j. The intermediate matrix $\tilde{X}_{j-\frac{1}{2}}$ in (7.12-7.13) may not be symmetric, but $\tilde{X}_{j-1}$ and $\tilde{X}_j$ are symmetric.

There are two matrix-matrix products and two matrix-matrix solves at each ADI step (7.12-7.13). The matrix $\tilde{X}_{j-1}$ and BB^T are symmetric and in general full. The first matrix-

matrix product in (7.12) is the multiplication of $(\tilde{A}-p_jI)$ by the full matrix $\tilde{X}_{j-1}$, the result of which is then transposed. The first matrix-matrix solve is $(\tilde{A}+p_jI)\tilde{X}_{j-\frac{1}{2}} = -BB^T - \tilde{X}_{j-1}(\tilde{A}^T - p_jI)$, with the full matrix $-BB^T - \tilde{X}_{j-1}(\tilde{A}^T - p_jI)$ as the right hand side. The solution $\tilde{X}_{j-\frac{1}{2}}$ is also a matrix. This matrix-matrix solve can be done by solving n linear systems with the matrix $(\tilde{A}+p_jI)$ and the columns of $-BB^T - \tilde{X}_{j-1}(\tilde{A}^T - p_jI)$ as n different right hand sides.

Thus, in each ADI step (7.12-7.13), $(\tilde{A}-p_jI)$ is multiplied by two full matrices, and $2n$ linear systems are solved with the matrix $(\tilde{A}+p_jI)$.

A general matrix A must be first reduced to a sparse form before proceeding with (7.12-7.13), to avoid full matrix-matrix products and full matrix-matrix solves, which would require $O(n^3)$ work per iteration [36, 58]. If $v \mapsto \tilde{A}v, v \in \mathbb{R}^n$, requires $O(n)$ work, then the two matrix-matrix products in (7.12-7.13) can be done in $O(n^2)$ work. The two matrix-matrix solves in (7.12-7.13) can also be done in $O(n^2)$ work, either under the assumption that $\tilde{A}$ is narrowly banded so that banded LU factorization can be used, or under the assumption that the solves are done iteratively using only multiplication by $\tilde{A}$, for example, via a Krylov subspace method, and that convergence is fast, which will result in an approximate solution of $\tilde{A}x = b, x, b \in \mathbb{R}^n$ in $O(n)$ work. In either case, solving with $2n$ right hand sides puts the total work for doing two matrix-matrix solves at $O(n^2)$.

Reducing a full matrix A to tri-diagonal form via a similarity transformation as a pre-processing step ensures that $v \mapsto \tilde{A}v, v \in \mathbb{R}^n$, has $O(n)$ complexity, and that solving $\tilde{A}x = b, x, b \in \mathbb{R}^n$, has $O(n)$ complexity. The final approximation X_J^{adi} is recovered via (7.14).

The flop count for ADI calculated in [36] is

$$\frac{19}{3}n^3 + 12Jn^2, \tag{7.15}$$

where J is the total number of ADI iterations. The $O(n^3)$ term comes from the tri-diagonalization of a general matrix A, and the transformation in (7.14) to obtain the final ADI approximation. If A is already sparse or structured so that the action of A on a vector is $O(n)$ work, there is no need to reduce A to tri-diagonal form. The ADI step (7.12-7.13) can be performed with the original matrix A. In either case, the $O(Jn^2)$ term in (7.15) comes from J iterations of (7.12-7.13) with the sparse matrix $\tilde{A}$. The ADI method is competitive with the Bartels-Stewart and Hammarling methods which are also $O(n^3)$ methods.

If the original matrix A is sparse, then ADI has an advantage over the exact methods, because it then does not need to reduce A to any special form, and its work requirement becomes $O(Jn^2)$. It is shown in later sections and chapters that frequently $J \ll n$, for a variety of reasons. On the other hand, the Bartels-Stewart and Hammarling methods still need to reduce a sparse A to Schur form, and so still require $O(n^3)$ work to obtain the solution.

7.2.1 ADI error bound

To obtain an error bound on the ADI approximation, it is convenient to consider (7.1) and (7.11-7.13) as order n^2 linear systems.

The Kronecker product of two matrices, $F \in \mathbb{R}^{m_1 \times n_1}$ and $G \in \mathbb{R}^{m_2 \times n_2}$, is defined as

$$E = F \otimes G := \begin{bmatrix} f_{11}G & f_{12}G & \cdots & f_{1n_1}G \\ f_{21}G & f_{22}G & \cdots & f_{2n_1}G \\ \vdots & \vdots & \ddots & \vdots \\ f_{m_11}G & f_{m_12}G & \cdots & f_{m_1n_1}G \end{bmatrix}, \tag{7.16}$$

$$E \in \mathbb{R}^{(m_1n_1)\times(m_2n_2)}. \tag{7.17}$$

The 'vec' operation on a matrix $X \in \mathbb{R}^{m\times n}$ is defined as

$$vec(X) = \begin{bmatrix} X(:,1) \\ \vdots \\ X(:,n) \end{bmatrix} \in \mathbb{R}^{(mn)\times 1}. \tag{7.18}$$

Clearly,

$$Y = GXF^T \Longleftrightarrow vec(Y) = (F \otimes G)vec(X). \tag{7.19}$$

It is possible to consider (7.1) as an order n^2 linear system,

$$AX + XA^T = -BB^T, \tag{7.20}$$

$$\Rightarrow (I_n \otimes A)vec(X) + (A \otimes I_n)vec(X) = vec(-BB^T), \tag{7.21}$$

$$\Rightarrow (I_n \otimes A + A \otimes I_n)vec(X) = vec(-BB^T), \tag{7.22}$$

and define $H \in \mathbb{R}^{n^2\times n^2}$, $V \in \mathbb{R}^{n^2\times n^2}$, $u \in \mathbb{R}^{n^2}$, $b \in \mathbb{R}^{n^2}$, as

$$H := (I_n \otimes A), \quad V := (A \otimes I_n), \tag{7.23}$$

$$u := vec(X), \quad b := vec(-BB^T). \tag{7.24}$$

Then (7.1) can be written as the order n^2 linear system,

$$(H + V)u = b. \tag{7.25}$$

Similarly, without loss of generality, assume A is not pre-processed, $A = \tilde{A}$, then (7.11-

7.13) become

$$
\begin{aligned}
u_0 &= 0_{n^2}, && (7.26)\\
(H + p_j I_{n^2}) u_{j-\frac{1}{2}} &= (p_j I_{n^2} - V) u_{j-1} + b, && (7.27)\\
(V + p_j I_{n^2}) u_j &= (p_j I_{n^2} - H) u_{j-\frac{1}{2}} + b, && (7.28)
\end{aligned}
$$

where (7.28) uses the fact that X_j is a symmetric matrix for all j. Hence,

$$
\begin{aligned}
u_j =& (V + p_j I_{n^2})^{-1} (p_j I_{n^2} - H) \left((H + p_j I_{n^2})^{-1} (p_j I_{n^2} - V) u_{j-1} + (H + p_j I_{n^2})^{-1} b \right) \\
&+ (V + p_j I_{n^2})^{-1} b.
\end{aligned} \tag{7.29}
$$

Define $e_j := u_j - u$, then (7.25) and (7.29) imply,

$$
e_j = R_j e_{j-1}, \tag{7.30}
$$

where

$$
R_j = (V + p_j I_{n^2})^{-1} (H - p_j I_{n^2}) (H + p_j I_{n^2})^{-1} (V - p_j I_{n^2}). \tag{7.31}
$$

Thus,

$$
e_J = \prod_{j=1}^{J} R_j \quad e_0. \tag{7.32}
$$

It can be seen that H and V commute,

$$
HV = (I_n \otimes A)(A \otimes I_n) = (A \otimes A) = (A \otimes I_n)(I_n \otimes A) = VH, \tag{7.33}
$$

thus,

$$
\prod_{j=1}^{J} R_j = \prod_{j=1}^{J} (V + p_j I_{n^2})^{-1} (V - p_j I_{n^2}) \prod_{j=1}^{J} (H - p_j I_{n^2}) (H + p_j I_{n^2})^{-1} \quad . \tag{7.34}
$$

A bound for the second part of (7.34) is

$$
\left\| \prod_{j=1}^{J} (H - p_j I_{n^2}) (H + p_j j I_{n^2})^{-1} \right\|_2 \leq \|G\|_2 \|G^{-1}\|_2 \max_{x \in spec(H)} \left| \prod_{j=1}^{J} \frac{(p_j - x)}{(p_j + x)} \right|, \tag{7.35}
$$

where G is a matrix of eigenvectors of H and $spec(H)$ the set of H's eigenvalues,

$$
H = GDG^{-1}, \quad D = diag(\lambda_1, \cdots, \lambda_{n^2}), \quad spec(H) = \{\lambda_i | i = 1, \cdots, n^2\}. \tag{7.36}
$$

Since $H = I_n \otimes A$, $G := I_n \otimes T$ is a matrix of H's eigenvectors, provided T is a matrix of A's eigenvectors. Also, $\|G\|_2 = \|I_n \otimes T\|_2 = \|T\|_2$, and $spec(H) = spec(I_n \otimes A) = spec(A)$. A similar argument can be made for the expression in (7.34) containing V.

The error expression in (7.32) can now be written in terms of the qualities from the original Lyapunov equation (7.1),

$$\begin{aligned} \|u_J - u\|_2 &\leq \|T\|_2^2 \|T^{-1}\|_2^2 k(\mathbf{p})^2 \|u_0 - u\|_2, \\ k(\mathbf{p}) &= \max_{x \in spec(A)} \left| \prod_{j=1}^{J} \frac{(p_j - x)}{(p_j + x)} \right|. \end{aligned} \tag{7.37}$$

If $u = vec(X)$, the 2-norm of u is the Frobenius norm of X. The final form for the ADI error bound is

$$\begin{aligned} \|X_J - X\|_F &\leq \|T\|_2^2 \|T^{-1}\|_2^2 k(\mathbf{p})^2 \|X_0 - X\|_F, \\ k(\mathbf{p}) &= \max_{x \in spec(A)} \left| \prod_{j=1}^{J} \frac{(p_j - x)}{(p_j + x)} \right|, \end{aligned} \tag{7.38}$$

where T is a matrix of A's eigenvectors, and $\mathbf{p} = \{p_1, p_2, \cdots, p_J\}$ are the ADI parameters.

7.2.2 ADI parameter selection

Optimal ADI parameters $\mathbf{p} = \{p_1, p_2, \cdots, p_J\}$ are the solution of the discrete rational min-max problem [58],

$$\min_{p_1, p_2, \cdots, p_J} \max_{\lambda \in spec(A)} \left| \prod_{j=1}^{J} \frac{(p_j - \lambda)}{(p_j + \lambda)} \right|, \tag{7.39}$$

and are a function of J.

However, since A's entire spectrum may not be easily available, the following continuous problem is usually posed instead,

$$\min_{p_1, p_2, \cdots, p_J} \max_{x \in \mathcal{R}} \left| \prod_{j=1}^{J} \frac{(p_j - x)}{(p_j + x)} \right|, \tag{7.40}$$

where

$$\lambda_1(A), \cdots, \lambda_n(A) \in \mathcal{R}. \tag{7.41}$$

The parameters $\{p_1, \cdots, p_J\}$ will be referred to as optimal if they solve (7.40). They do not need to be the solution to the discrete problem (7.39). The problem of finding optimal and

near-optimal parameters was investigated in several papers [10, 26, 51, 52, 55].

For example, if A's eigenvalues are strictly real and contained in the interval $[-b, -a]$,

$$-b \leq \lambda_1(A), \cdots, \lambda_n(A) \leq -a < 0, \tag{7.42}$$

then the ADI parameters are chosen to be the solution of

$$\min_{p_1, p_2, \cdots, p_J} \max_{x \in [-b, -a]} \left| \prod_{j=1}^{J} \frac{(p_j - x)}{(p_j + x)} \right|. \tag{7.43}$$

The solution to (7.43) is known [58] and is given below.

The solution to (7.40) is not known when $\mathcal{R}$ is an arbitrary region in the open left half plane. In [55, 58], 'approximately optimal' parameters were reported. [52] gave 'asymptotically optimal' parameters.

The following parameter selection procedure comes from [58].

Define the spectral bounds a, b, and α for the matrix A as,

$$a = \min_i (Re\{\lambda)i\}), \quad b = \max_i (Re\{\lambda)i\}), \quad \alpha = \tan^{-1} \max_i |\frac{Im\{\lambda_i\}}{Re\{\lambda)i\}}|, \tag{7.44}$$

where $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of $-A$. It is assumed that $-A$'s spectrum lies entirely inside the 'elliptic function domain' determined by a, b, α, as defined in [58]. If this assumption does not hold, one should try to apply a more general parameter selection algorithm.

Let

$$\cos^2 \beta = \frac{2}{1 + \frac{1}{2}(\frac{a}{b} + \frac{b}{a})}, \tag{7.45}$$

$$m = \frac{2 \cos^2 \alpha}{cos^2 \beta} - 1. \tag{7.46}$$

If $m < 1$, the parameters are complex, and are given in [10, 58]. If $m \geq 1$, the parameters are real, and define

$$k' = \frac{1}{m + \sqrt{m^2 - 1}}, \tag{7.47}$$

$$k = \sqrt{1 - k'^2}. \tag{7.48}$$

Note $k' = \frac{a}{b}$ if the eigenvalues of A are all real. Define elliptic integrals K and v as,

$$F[\psi, k] = \int_0^{\psi} \frac{dx}{\sqrt{1 - k^2 \sin^2 x}}, \tag{7.49}$$

$$K = K(k) = F[\frac{\pi}{2}, k], \tag{7.50}$$

$$v = F[\sin^{-1} \sqrt{\frac{a}{bk'}}, k']. \tag{7.51}$$

The number of ADI iterations required to achieve $k(\mathbf{p})^2 \leq \epsilon_1$ is

$$J = \lceil \frac{K}{2v\pi} \log \frac{4}{\epsilon_1} \rceil, \tag{7.52}$$

and the ADI parameters are given by

$$p_j = -\sqrt{\frac{ab}{k'}} dn[\frac{(2j-1)K}{2J}, k], \quad j = 1, 2, \cdots, J. \tag{7.53}$$

It was noted in [36] that for most problems ADI usually converges in a few iterations with these parameters.

r 8

l s - r D

A major contribution of this dissertation is the development of the Cholesky Factor ADI (CF–ADI) algorithm [33], which is presented in this chapter. CF–ADI is well-suited to solve the Lyapunov equation

$$AX + XA^T = -BB^T, \quad B \in \mathbb{R}^{n\times p}, \quad rank(B) = p \ll n, \tag{8.1}$$

whose right hand side has low rank. The matrix A is assumed to be stable. The right hand side $-BB^T$ has low rank compared to the size of A. For simplicity, it is assumed that B has full column rank. Otherwise, it is a simple matter to replace B by $\tilde{B}$, where $\tilde{B}$ has full column rank, and $\tilde{B}\tilde{B}^T = BB^T$.

Lyapunov equations of the form (8.1) occur frequently in the analysis of large, linear, time-invariant systems whose system matrix is stable, and where the number of inputs and the number of outputs are much smaller than the system size.

For the low rank right hand side problem (8.1), CF–ADI produces the same approximation as the ADI method described in chapter 7, but is much more efficient because it iterates on the Cholesky factor of the ADI approximation rather than the approximation itself.

8.1 erivation

This section derives the CF–ADI method from the ADI method. For simplicity, all quantities in algorithm 8 with tildes will be written in this chapter without the tildes.

The complexity of the ADI method is given in (7.15). Frequently, the system matrix A of a large, linear, time-invariant system is sparse, so that the action of A on a vector requires only $O(n)$ work. In this case the reduction of A to tri-diagonal form in step 0 of algorithm 8 is not necessary. Because X_{j-1} and $X_{j-\frac{1}{2}}$ in (7.12-7.13) are in general full, and the work of multiplying a sparse matrix by a full matrix, as well as doing a sparse matrix solve with a full matrix as the right hand side, require $O(n^2)$ work, the complexity of ADI is $O(Jn^2)$

if A is sparse, where J is the total number of ADI iterations. Unfortunately, $O(Jn^2)$ is still unacceptably high for many applications, where n can be on the order of $100,000$.

The fact that there are two matrix-matrix products and two matrix-matrix solves in (7.12-7.13) is of concern. The need for matrix-matrix operations rather than simply matrix-vector operations at each ADI step makes algorithm 8 extremely expensive. It is clear that a more efficient way to represent the full matrix X_j is needed.

The first step in developing CF–ADI is to combine (7.12) and (7.13) and obtain

$$\begin{aligned} X_j = &-2p_j(A+p_jI)^{-1}BB^T(A+p_jI)^{-T} \\ &+ (A+p_jI)^{-1}(A-p_jI)X_{j-1}(A-p_jI)^T(A+p_jI)^{-T}. \end{aligned} \tag{8.2}$$

From (8.2) and the fact that $X_0 = 0_{n\times n}$, it can be seen that X_j is symmetric for all $j \in \mathbb{Z}$, and the rank of X_j is at most the sum of the rank of X_{j-1} and the rank of B. Since iteration begins with the zero matrix initial guess, X_j will have rank at most jp, where p is the number of columns in B. Therefore, X_j can be represented as an outer product,

$$X_j = Z_j Z_j^T, \tag{8.3}$$

where Z_j has jp columns.

Definition 10. *A matrix Z is called a* **Cholesky factor** *of $X \in \mathbb{R}^{n\times n}$ if it satisfies,*

$$X = ZZ^T. \tag{8.4}$$

The matrix Z does not have to be a square matrix nor have lower triangular structure. Thus, in (8.3) $Z_j \in \mathbb{R}^{n\times jp}$ is a Cholesky factor of $X_j \in \mathbb{R}^{n\times n}$.

Replacing X_j by $Z_j{Z_j}^T$ in (7.11-7.13) results in

$$Z_0 = 0_{n\times p}, \tag{8.5}$$

$$\begin{aligned} Z_jZ_j^T = &-2p_j\big\{(A+p_jI)^{-1}B\big\}\big\{(A+p_jI)^{-1}B\big\}^T \\ &+ \big\{(A+p_jI)^{-1}(A-p_jI)Z_{j-1}\big\}\big\{(A+p_jI)^{-1}(A-p_jI)Z_{j-1}\big\}^T. \end{aligned} \tag{8.6}$$

The left hand side of (8.6) is an outer product, and the right hand side is the sum of two outer products. Thus, Z_j on the left hand side of (8.6) can be obtained simply by combining the two factors in the two outer products on the right,

$$Z_j = \big[\ \ \overline{-2p_j}\big\{(A+p_jI)^{-1}B\big\}, \big\{(A+p_jI)^{-1}(A-p_jI)Z_{j-1}\big\}\big]. \tag{8.7}$$

Thus, the ADI algorithm can be reformulated in terms of the Cholesky factor Z_j of X_j. There is no need to calculate or store X_j at each iteration, only Z_j is needed.

The preliminary form of Cholesky Factor ADI which iterates on the Cholesky factor Z_j

of X_j is

$$Z_1 = \quad \overline{-2p_1}(A+p_1I)^{-1}B, \quad Z_1 \in R^{n\times p} \tag{8.8}$$

$$Z_j = \left[\quad \overline{-2p_j}(A+p_jI)^{-1}B, (A+p_jI)^{-1}(A-p_jI)Z_{j-1}\right], \quad Z_j \in R^{n\times jp}. \tag{8.9}$$

In this formulation, at each iteration, the previous Cholesky factor $Z_{j-1} \in \mathbb{R}^{n\times(j-1)p}$ needs to be modified by multiplication on the left by $(A+p_jI)^{-1}(A-p_jI)$. Thus, the number of columns which need to be modified at each iteration increases by p.

The implementation in (8.8-8.9) was independently developed in [44].

Here, a further step is taken to keep the number of columns modified at each iteration constant.

8.2 ational rylov su space for ulation

The Jp columns of Z_J, the Cholesky factor of the Jth ADI approximation, can be written out explicitly,

$$Z_J = \Big[S_J \quad \overline{-2p_J}B, \quad S_J\,(T_JS_{J-1}) \quad \overline{-2p_{J-1}}B, \quad S_JT_JS_{J-1}\,(T_{J-1}S_{J-2}) \quad \overline{-2p_{J-2}}B, \\ \cdots, S_JT_J\cdots S_2\,(T_2S_1) \quad \overline{-2p_1}B\Big], \tag{8.10}$$

where

$$S_i = (A+p_iI)^{-1}, \quad T_i = (A-p_iI). \tag{8.11}$$

Note that the S_i's and the T_i's commute,

$$S_iS_j = S_jS_i, \quad T_iT_j = T_jT_i, \quad S_iT_j = T_jS_i, \quad \forall i,j. \tag{8.12}$$

The Cholesky factor Z_J then becomes

$$Z_J = [z_J, \quad P_{J-1}(z_J), \quad P_{J-2}(P_{J-1}z_J), \quad \cdots, P_1(P_2\cdots P_{J-1}z_J)], \tag{8.13}$$

where

$$z_J := \left(\quad \overline{-2p_J} \right) S_J B \quad = \quad \overline{-2p_J}(A + p_J I)^{-1} B, \tag{8.14}$$

$$P_l := \left(\frac{\sqrt{-2p_l}}{\sqrt{-2p_{l+1}}} \right) S_l T_{l+1} \quad = \quad \frac{\sqrt{-2p_l}}{\sqrt{-2p_{l+1}}} (A + p_l I)^{-1} (A - p_{l+1} I), \tag{8.15}$$

$$= \quad \left(\frac{\sqrt{-2p_l}}{\sqrt{-2p_{l+1}}} \right) [I - (p_{l+1} + p_l)(A + p_l I)^{-1}]. \tag{8.16}$$

It can be seen that if B only has one column, the columns of Z_J span the order J rational Krylov subspace $\mathcal{K}(A, z_J, \{p_{J-1}, \cdots, p_1\})$, with starting vector $z_J = \sqrt{-2p_J}(A + p_J I)^{-1} B$ and the shifts $\{p_{J-1}, \cdots, p_1\}$.

Since there is no significance to the order in which the ADI parameters appear, the index $1, \cdots, J$ in (8.13) can be reversed, to obtain

$$Z_J = [z_1, \quad P_1 z_1, \quad P_2 P_1 z_1, \quad \cdots, P_{J-1} P_{J-2} \cdots P_1 z_1], \tag{8.17}$$

where

$$z_1 \quad = \quad \left(\quad \overline{-2p_1} \right) (A + p_1 I)^{-1} B, \tag{8.18}$$

$$P_l \quad = \quad \left(\frac{\sqrt{-2p_{l+l}}}{\sqrt{-2p_l}} \right) \left[I - (p_{l+1} + p_l)(A + p_{l+1} I)^{-1} \right]. \tag{8.19}$$

The CF–ADI algorithm which comprises of (8.17-8.19) is given as algorithm 9.

It will be justified in section 8.5 that there is no need to tri-diagonalize A as a pre-processing step, even if A is a full matrix. As in the ADI method, it is assumed that each parameter in the parameter list $\{p_1, p_2, \cdots, p_J\}$ is either real or comes as a part of a complex conjugate pair, to ensure that the final approximation $X_J = Z_J Z_J^T$ is real. Again, because A is stable, and $Re\{p_j\} < 0$ for all j, $(A + p_j I)$ is non-singular for all j.

Algorithm 9 The Cholesky Factor ADI Algorithm.

INPUT: A, B.

0. Choose ADI parameters, $\{p_1, \cdots, p_{J_{max}}\}$, $Re\{p_i\} < 0$, (real or complex conjugate pairs).

Define: $P_i = \frac{\sqrt{-2p_{i+1}}}{\sqrt{-2p_i}} \quad [I - (p_{i+1} + p_i)(A + p_{i+1}I)^{-1}]$.

1a.

$$z_1 = \left(\quad \overline{-2p_1} \right) (A + p_1 I)^{-1} B, \tag{8.20}$$

1b.

$$Z_1^{cfadi} = \left[\; z_1 \; \right],$$

FOR $j = 2, 3, \cdots, J_{max}$

2a.

$$z_j = P_{j-1} z_{j-1}, \tag{8.21}$$

2b. If ($\|z_j\|_2 > tol_1$ or $\frac{\|z_j\|_2}{\|Z_{j-1}\|_2} > tol_2$) and ($j \leq J_{max}$)

$$Z_j^{cfadi} = \left[\; Z_{j-1}^{cfadi} \quad z_j \; \right]. \tag{8.22}$$

Otherwise, $J = j - 1$, stop.

END

OUTPUT: $Z_J^{cfadi} \in \mathbb{C}^{n \times Jp}$, $Z_J^{cfadi}(Z_J^{cfadi})^T \in \mathbb{R}^{n \times n}$, $X_J^{cfadi} := Z_J^{cfadi}(Z_J^{cfadi})^T \approx X$.

Theorem 4. *If X_J^{adi} is obtained by running J steps of algorithm 8, with the ADI parameters $\{p_1, p_2, \cdots, p_J\}$, and Z_J^{cfadi} is obtained by running J steps of algorithm 9, with the same parameters, in any order, then*

$$X_J^{adi} = Z_J^{cfadi}(Z_J^{cfadi})^T. \tag{8.23}$$

Proof. From the derivation of CF–ADI, it is clear that (8.23) is true when the order of the parameters is reversed. The fact that parameter order does not matter at all in either algorithm is shown by

$$\begin{aligned} X_j = (A + p_j I)^{-1}(A + p_{j-1}I)^{-1} \quad & (A - p_j I)(A - p_{j-1}I) X_{j-2} (A - p_j I)^T (A - p_{j-1}I)^T \\ - 2(p_j + p_{j-1})(ABB^T A^T + p_j p_{j-1} BB^T) \quad & (A + p_j I)^{-T}(A + p_{j-1}I)^{-T}. \end{aligned} \tag{8.24}$$

Clearly, this expression does not depend on the order of p_j and p_{j-1}. Any ordering of $\{p_1, \cdots, p_J\}$ can be obtained by exchanging neighboring parameters. □

As a matter of notation, define,

$$X_J^{cfadi} := Z_J^{cfadi}(Z_J^{cfadi})^T. \tag{8.25}$$

Both X_J^{cfadi} and Z_J^{cfadi} will be referred to as the Jth CF–ADI approximation, which one is meant will be made clear in context. The full matrix X_J^{cfadi} is usually not explicitly calculated. It will be used in subsequent sections for convergence analysis purposes only. The matrix X_J^{adi}, produced by the ADI algorithm, will be referred to as the Jth ADI approximation.

8.3 Stopping criterion

The stopping criterion $\|X_j^{cfadi} - X_{j-1}^{cfadi}\|_2 \leq tol^2$ can be implemented as $\|z_j\|_2 \leq tol$, since

$$\|Z_j Z_j^T - Z_{j-1} Z_{j-1}^T\|_2 = \|z_j z_j^T\|_2 = \|z_j\|_2^2. \tag{8.26}$$

Relative error can also be used, in which case the stopping criterion is $\frac{\|z_j\|_2}{\|Z_{j-1}\|_2} \leq tol$.

8.4 ara eter selection

The criterion for picking CF–ADI parameters, $\mathbf{p} = \{p_1, \cdots, p_{J_{max}}\}$, is exactly the same as for ADI parameters, which is given as (7.40). Section 7.2.2 gives a parameter selection procedure based on three spectral bounds of A,

$$a = \min_i(Re(\lambda_i)), \quad b = \max_i(Re(\lambda_i)), \quad \alpha = \tan^{-1} \max_i |\frac{Im(\lambda_i)}{Re(\lambda_i)}|, \tag{8.27}$$

where $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of $-A$. These three bounds for the matrix A may be estimated using the power and inverse power iterations, or Gershgorin's circles.

A numerical comparison of different choices of parameters in the model reduction context is given in section 10.2.

Power and inverse power iterations can be done at the cost of a few matrix-vector products and solves. The work to obtain the CF–ADI parameters, W^{param}, will be calculated in the next section.

8.5 CF– I algorit co plexity

The following definition is helpful when B has more than one column.

Definition 11. *A p-vector $v \in \mathbb{R}^{n \times p}$ is a matrix that has p columns.*

The final CF–ADI approximation Z_J^{cfadi} can be obtained from the starting p-vector z_1 and $J-1$ products of the form $P_i z_i$. The cost of applying P_i to a vector is essentially that of a linear matrix-vector solve. The starting p-vector z_1 is obtained after p matrix-vector solves with columns of $B \in \mathbb{R}^{n\times p}$ as the p right-hand sides (8.20). Each succeeding p-vector in Z_J^{cfadi} is obtained from the previous p-vector at the cost of p matrix-vector solves (8.21).

Thus, the work per iteration has been reduced from the two matrix-matrix products and two matrix-matrix solves in (7.11-7.12) of the original ADI method, to p matrix-vector solves in (8.21). Figure 8-1 illustrates this savings.

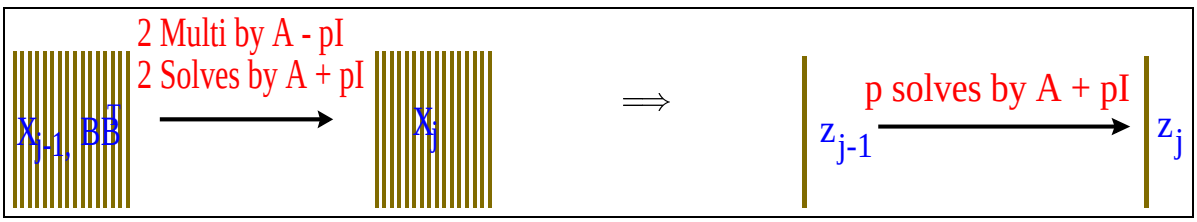


Figure 8-1: Savings from CF–ADI

As will be shown in later chapters, the Cholesky factor of the Lyapunov solution is precisely what is needed in model reduction. In general, if Z_J^{cfadi} is available, it is not necessary to calculate $X_J^{cfadi} = Z_J^{cfadi}(Z_J^{cfadi})^T$. Whereas if X_J^{adi} is available, it is often necessary to calculate its Cholesky factor in the subsequent model reduction procedure.

When comparing the complexities of the ADI algorithm and the CF–ADI algorithm, the work to generate X_J^{adi} after J steps of the ADI algorithm is compared with the work to generate Z_J^{cfadi} after J steps of the CF–ADI algorithm.

Table 8.1 summarizes the work of various matrix operations, depending on the sparsity pattern of A, which will be assumed and used to calculate the complexities of both algorithms.

	$v \mapsto Av$	$v \mapsto (A+p_iI)^{-1}v$	tri-diag(A)
Sparse	$O(n)$	$O(J_s n)$	$O(n^3)$
Full	$O(n^2)$	$O(J_s n^2)$	$O(n^3)$
Tri-diagonal	$O(n)$	$O(n)$	

Table 8.1: Work associated with matrix operations

The multiplication of a vector by a sparse matrix A requires $O(n)$ work. Iterative linear solve with the matrix $A+p_iI$ is assumed to be $O(J_s n)$ work, from the work of $O(J_s)$ matrix-vector products. This is true when an iterative Krylov-subspace such as GMRES is used to find $(A+p_iI)^{-1}v$. The number J_s indicates the speed of convergence of the iterative method. If A is full, and the same convergence speed is assumed, then calculating $(A+p_iI)^{-1}v$ is $O(J_s n^2)$. If A is tri-diagonal, calculating $(A+p_iI)^{-1}v$ is $O(n)$ work.

Unlike the ADI method, it is not necessary to tri-diagonalize a full matrix as a pre-processing step in CF–ADI if $J, J_s \ll n$. Since CF–ADI performs only p matrix-vector solves per iteration, and a matrix-vector solve requires $O(J_s n^2)$ work when A is full, J iterations of CF–ADI with a full matrix has $O(JpJ_s n^2)$ cost. If $p, J, J_s \ll n$, $O(JpJ_s n^2)$ cost is still better than the $O(n^3)$ cost of tri-diagonalization.

Exclusive of the work to obtain the ADI/CF–ADI parameters, J iterations of the ADI algorithm has $O(n^3 + 4Jn^2)$ cost when A is full and $O(J(2 + 2J_s)n^2)$ cost when A is sparse. In contrast, J iterations of CF–ADI has $O(JpJ_s n^2)$ cost when A is full and $O(JpJ_s n)$ cost when A is sparse.

Since the work to calculate the ADI/CF–ADI parameters after the spectral bounds in (8.27) have been obtained is negligible, the work to generate ADI/CF–ADI parameters, W^{param}, consists entirely of the work to calculate the spectral bounds.

Suppose J_p iterations of the power method and J_{ip} iterations of the inverse power method are run to generate the bounds. If A is sparse, $W^{param} = J_p n + J_{ip} J_s n$ for both ADI and CF–ADI. If A is full, $W^{param} = J_p n^2 + J_{ip} J_s n^2$ for CF–ADI. Since a full matrix A is first transformed to a tri-diagonal matrix in ADI, the spectral bounds can be obtained from the tri-diagonal matrix, and $W^{param} = J_p n + J_{ip} n$ for the ADI algorithm.

The complexity comparison between ADI and CF–ADI is shown in table 8.2. The first term is the work to generate the parameters and the second term is running J iterations in all entries except ADI/full A, where $O(n^3)$ is included for tri-diagonalization and the similarity transformation to obtain the final ADI solution.

	CF–ADI	ADI
Sparse (structured) A	$O((J_p + J_{ip}J_s)n) + O(pJJ_s n)$	$O((J_p + J_{ip}J_s)n) + O(J(2 + 2J_s)n^2)$
Full A	$O((J_p + J_{ip}J_s)n^2 + O(pJJ_s n^2)$	$O(n^3) + O(J_p n + J_{ip} n) + O(4Jn^2)$

Table 8.2: ADI and CF–ADI complexity comparison, $J, J_s, J_p, J_{ip} \ll n$.

Table 8.3 gives the complexities as a function of n, p, and J only.

	CF–ADI	ADI
Sparse (structured) A	$O(Jpn)$	$O(Jn^2)$
Full A	$O(Jpn^2)$	$O(n^3) + O(Jn^2)$

Table 8.3: ADI and CF–ADI complexity comparison, function of n, p, J.

Since p, the number of inputs, is by assumption much smaller than n, CF–ADI always results in an order of magnitude savings when A is sparse.

For many large system, $O(n)$ complexity is considered acceptable, and $O(n^2)$ is deemed too expensive. The work to run CF–ADI on a sparse matrix is $O(Jpn)$. Since $p \ll n$, $Jpn \ll n^2$ if and only if $J \ll n$. In other words, the total number of CF–ADI iterations

should be much smaller than the system size n, for CF–ADI to be practical on large problems. Thus, it is possible that algorithm 9 will terminate at a small J, before the error criterion is satisfied, to ensure that the work stays $O(n)$. Thus, $X_J^{cfadi} := Z_J^{cfadi}(Z_J^{cfadi})^T$, under the assumption that $J \ll n$, is necessarily a low rank approximation to the exact solution X.

8.6 eal – I for co plex para eters

The CF–ADI method in algorithm 9 will result in a complex Cholesky factor Z_J if there are complex ADI parameters, although $Z_J Z_J^T$ is guaranteed to be real if the CF–ADI parameters come in complex conjugate pairs.

A version of CF–ADI which only uses operations with real numbers is given as algorithm 10. It assumes that the CF–ADI parameters are either real or come in complex conjugate pairs, and that each complex conjugate pair is represented only once in the list $\{p_1, p_2, p_3, \cdots, p_J\}$. Thus, each complex number encountered in this list will result in $2p$ additional columns, p being the number of columns in B. The counter k in algorithm 10 indicates that the number of columns in Z_i is kp.

The matrices associated with a real parameter p_i

$$S_i := (A + p_i I)^{-1}, \quad T_i := (A - p_i I), \tag{8.28}$$

are a rational function and a polynomial of degree one in A. The matrices associated with the complex parameters p_i, $\bar{p}_i$,

$$\sigma_i = 2Re\{-p_i\}, \quad \tau_i = |p_i|^2, \tag{8.29}$$

$$Q_i \equiv (A^2 - \sigma_i A + \tau_i I)^{-1}, \quad R_i \equiv (A^2 + \sigma_i A + \tau_i I), \tag{8.30}$$

are a rational function and a polynomial of degree two in A.

8.7 u erical results

This section gives numerical results on the CF–ADI approximation to the solution of (8.1).

The example in figure 8-3 comes from inductance extraction of an on-chip planar square spiral inductor suspended over a copper plane [30], shown in figure 8-2. The original order 500 system has been symmetrized according to (5.16-5.18). The matrix A is a symmetric 500×500 matrix, and the input coefficient matrix $B \in \mathbb{R}^n$ has only one column.

Because A is symmetric, the eigenvalues of A are real and good CF–ADI parameters are easy to find. The procedure given in section 7.2.2 was followed. CF–ADI was run to convergence in this example, which took 20 iterations.

Figure 8-3 shows the relative error in the 2-norm of the CF–ADI approximation, $\frac{\|X - X_j^{cfadi}\|_2}{\|X\|_2}$,

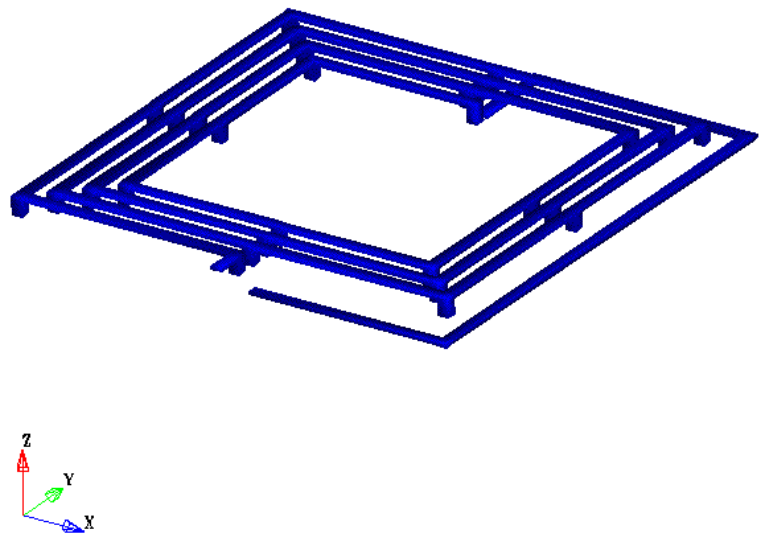


Figure 8-2: Spiral inductor, a symmetric system.

for $j = 1, \cdots, 20$. At $j = 20$, relative error has reached 10^{-8}, which is about the same size as the error of the optimal rank 11 approximation. The error estimate $\|z_{j+1}^{cfadi}\|_2^2$ approximates the actual error $\|X - X_j^{cfadi}\|$ closely for all j.

8.8 rylov vectors reuse

If the CF–ADI parameters $\{p_1, \cdots, p_J\}$ are distinct, then a re-organization of algorithm 9 can result in significant savings in computational cost, when an iterative Krylov subspace method such as GMRES is used to solve the shifted linear system in (8.21).

This re-organization involves converting the shifted linear system solve in (8.21) with the right hand side z_{j-1}, to one with the right hand side B. Then each solve in (8.21) can reuse the Krylov subspace built up during the previous solve.

8.8.1 Shifted linear systems with the same HS

This section describes how the CF–ADI approximation Z_J^{cfadi} can be produced after J linear systems solves with the right hand side B, if the CF–ADI parameters $\{p_1, \cdots, p_J\}$ are distinct.

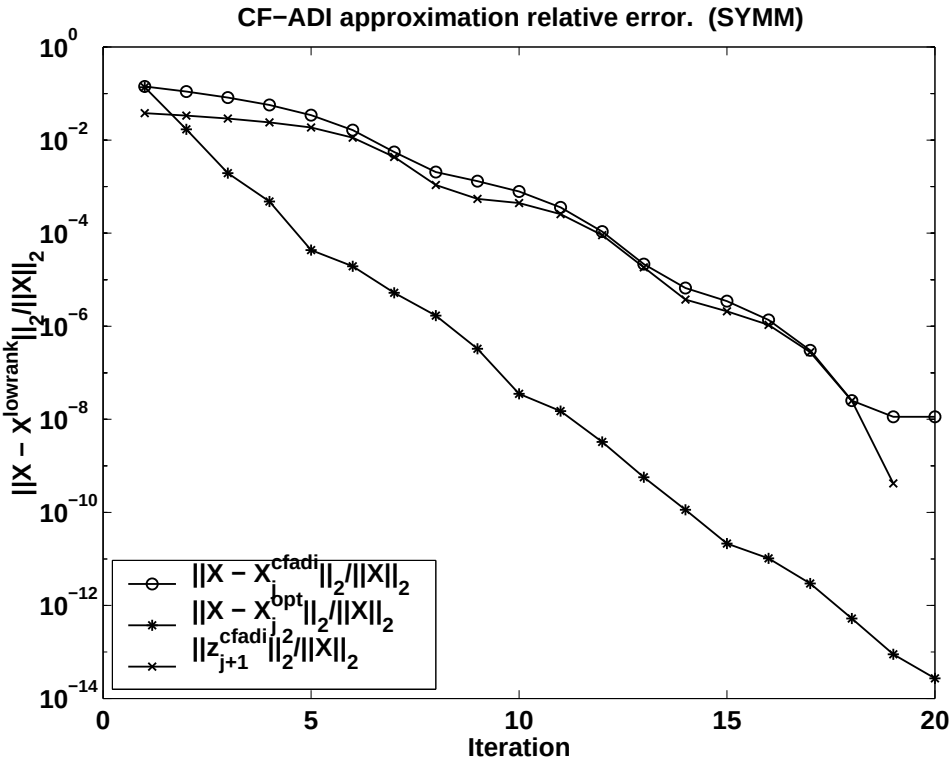


Figure 8-3: CF–ADI approximation.

The CF–ADI approximation Z_J^{cfadi} can be written out explicitly,

$$\begin{aligned} Z_J^{cfadi} = [\quad & \overline{-2p_1}(A+p_1I)^{-1}B, \\ & \overline{-2p_2}[I-(p_2+p_1)(A+p_2I)^{-1}](A+p_1I)^{-1}B, \\ & \vdots \\ & \overline{-2p_J}[I-(p_J+p_{J-1})(A+p_JI)^{-1}]\cdots \\ & \qquad \cdots[I-(p_2+p_1)(A+p_2I)^{-1}](A+p_1I)^{-1}B]. \end{aligned} \tag{8.31}$$

By expanding $\quad {}_{i=1}^{j}(A+p_iI)^{-1}$ into partial fractions,

$$\begin{aligned} {}_{i=1}^{j}(A+p_iI)^{-1} &= {}_{i=1}^{j}\Big(\quad {}_{k\neq i}\Big(\frac{1}{p_k-p_i}\Big)\Big)(A+p_iI)^{-1}, & (8.32) \\ p_1 &\neq p_2 \neq \cdots, \neq p_J, & (8.33) \end{aligned}$$

Z_J^{cfadi} becomes

$$Z_J^{cfadi} = V_J M_{J\times J} D_{J\times J}, \tag{8.34}$$

where

$$V_J = \Big[(A+p_1 I)^{-1}B, \quad (A+p_2 I)^{-1}B, \cdots, (A+p_J I)^{-1}B\Big], \tag{8.35}$$

$$M_{J\times J} = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1J} \\ 0 & m_{22} & \cdots & m_{2J} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & m_{JJ} \end{bmatrix}, \tag{8.36}$$

$$m_{11} = 1, \tag{8.37}$$

$$m_{ii} = -\sum_{j=1}^{i-1} m_{i-1,j}\left(\frac{p_{i-1}+p_i}{p_j - p_i}\right), \tag{8.38}$$

$$m_{j,i} = m_{j,i-1}\left(\frac{p_j + p_{i-1}}{p_j - p_i}\right), \quad j \neq i, \tag{8.39}$$

and

$$D_{J\times J} = \begin{bmatrix} \sqrt{-2p_1} & 0 & \cdots & 0 \\ 0 & \sqrt{-2p_2} & \cdots & 0 \\ \vdots & & \ddots & 0 \\ 0 & 0 & \cdots & \sqrt{-2p_J} \end{bmatrix}. \tag{8.40}$$

The matrices $M_{J\times J}$ and $D_{J\times J}$ are determined completely by the parameters $\{p_1, \cdots, p_J\}$ and cost very little to compute. Thus, the cost of calculating Z_J^{cfadi} via (8.34) comes almost entirely from the calculation of V_J.

It already follows from theorem 3 that $colsp(Z_J^{cfadi}) = colsp(V_J)$, but (8.34) makes the relationship between Z_J^{cfadi} and V_J precise.

8.8.2 Sharing of Krylov vectors

The columns of V_J (8.35) can be obtained either exactly, using J LU factorizations, or approximately, using J iterative linear system solves,

$$V_J = \Big[v_1, v_2, \cdots, v_J\Big], \tag{8.41}$$

$$(A+p_i I)v_i = B, \quad i = 1 \cdots J. \tag{8.42}$$

If an iterative Krylov subspace method such as GMRES is used, and if none of the solves in (8.42) is too difficult, the columns of V_J can be obtained in a much more efficient way than doing J separate solves. The solution of shifted systems is discussed in detail in [14].

For simplicity, assume B has only one column, $B \in \mathbb{R}^n$. GMRES solves the system $Ax = B$ by finding an approximate solution x_m in the m-dim Krylov subspace,

$$x_m \in \mathcal{K}_m(A, r_0) := span\left\{r_0, Ar_0, \cdots, A^{m-1}r_0\right\}. \tag{8.43}$$

It chooses $r_0 = B - Ax_0$. The difficulty of solving a system in (8.42), in other words, the dimension of the Krylov subspace required to find a satisfactory solution, depends on the shift p_i.

If zero is used as the initial guess for all system solves in (8.42), the Krylov subspace associated with each system is the same, namely, $\mathcal{K}_m(A, B)$, since shifts of A do not affect the Krylov subspace,

$$\begin{aligned} \mathcal{K}_m(A + p_i I, B) &:= span\left\{B, (A + p_i I)B, \cdots, (A + p_i I)^{m-1}B\right\}, & (8.44) \\ &= span\left\{B, AB, \cdots, A^{m-1}B\right\} := \mathcal{K}_m(A, B). & (8.45) \end{aligned}$$

Hence, one needs only one set of Krylov vectors for all solves in (8.42), which can be stored from solve to solve. When a more difficult shift is encountered, one simply adds to the list of stored Krylov vectors.

What is different for each solve in (8.42) is that the decomposition of a different Hessenberg matrix is needed. Let $\tilde{H}_m$ denote the Hessenberg matrix which comes from m steps of the Arnoldi process with the matrix A, for the system $Ax = B$, then $\tilde{H}_m + \begin{matrix} p_i I_{m\times m} \\ 0 \end{matrix}$ is the Hessenberg matrix associated with the shifted system $(A + p_i I)x_i = B$. But if none of the systems in (8.42) is too difficult, in other words, if they all can be solved in the Krylov subspace whose dimension is small compared to the size of A, then the cost of decomposing small Hessenberg matrices will be low compared to the cost of generating Krylov vectors. In that case the cost of solving J shifted systems is only marginally higher than the cost of solving the most difficult one.

Figure 8-4 shows the speed-up in the calculation of the CF–ADI approximation Z_J^{cfadi} that comes from storing the Krylov vectors between solves. The matrix A is 500×500 and its eigenvalues are well-distributed for fast GMRES convergence. The flops required to generate Z_J^{cfadi}, as a function of J, are plotted. Doing J solves separately and not storing the Krylov vectors is denoted by $+$, generating V_J by storing Krylov vectors is represented as $\times$, the total cost of obtaining Z_J^{cfadi} from V_J, including the generation of the matrices $M_{J\times J}$ and $D_{J\times J}$ in (8.34), is shown as $\circ$.

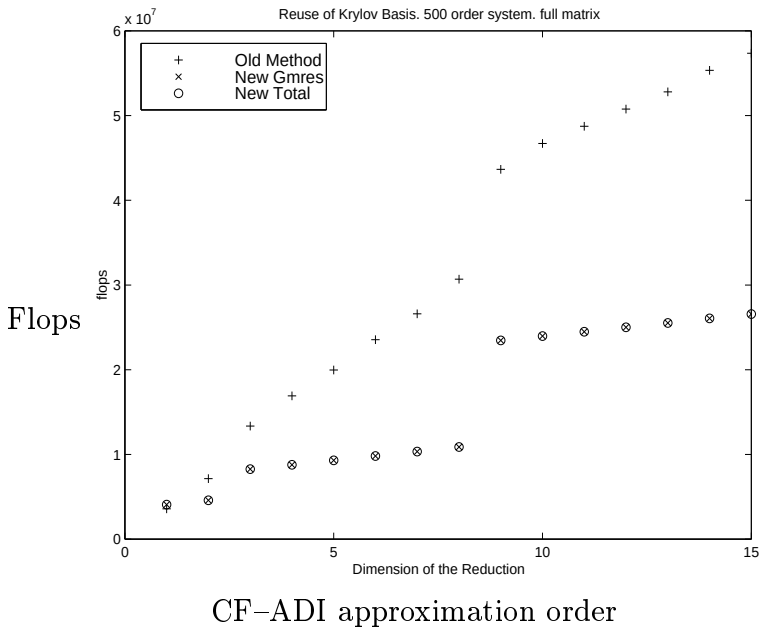


Figure 8-4: Cost of additional solves negligible.

Disregarding the jumps at $J = 3$ and $J = 9$ for the moment, it can be seen that when Krylov vectors are not stored, the cost of generating Z_J^{cfadi} grows linearly with J, whereas if the Krylov vectors are stored, the cost of generating Z_J^{cfadi} increases very little as J increases. The cost of generating Z_8 is only slightly higher than the cost of generating Z_3. The jump at $J = 3$ occurred because p_3 is a more difficult shift, so the solution of $(A+p_3I)v_i = B$ required more Krylov vectors than the previous two solves. But after the extra Krylov vectors were generated, more solves after $J = 3$ cost very little, until the next difficult shift at $J = 9$.

Algorithm 10 Real version of CF–ADI: $Z_J^{cfadi} \in \mathbb{R}^{n \times Jp}$.

- Define, $\sigma_i = 2Re\{-p_i\}, \quad \tau_i = |p_i|^2$.
- if p_1 is real,

$$\begin{aligned} v_1 &= S_1 B = (A + p_1 I)^{-1} B; \\ Z_1 &= \left[\quad \overline{-2p_1} v_1 \right]; \quad k = 1; \end{aligned}$$

- elseif p_1 is complex,

$$\begin{aligned} v_1 &= Q_1 B = \left(A^2 - \sigma_1 A + \tau_1 I\right)^{-1} B; \quad v_2 = A v_1; \\ Z_1 &= \left[\sqrt{2\sigma_1}\sqrt{\tau_1} v_1, \sqrt{2\sigma_1} v_2\right]; \quad k = 2; \end{aligned}$$

- for $i = 2, 3, \cdots, J$
  - if p_i is real,
    - if p_{i-1} is real,

$$\begin{aligned} v_{k+1} &= S_i T_{i-1} v_k = \left(I - (p_{i-1} + p_i)(A + p_i I)^{-1}\right) v_k; \\ Z_i &= \left[Z_{i-1}, \quad \overline{-2p_i} v_{k+1}\right]; \quad k = k + 1; \end{aligned}$$

    - elseif p_{i-1} is complex,

$$\begin{aligned} v_{k+1} &= S_i R_{i-1} v_{k-1} = \left(A + (\sigma_{i-1} - p_i) I + (\tau_{i-1} - p_i(\sigma_{i-1} - p_i))(A + p_i I)^{-1}\right) v_{k-1}; \\ &= v_k + \left((\sigma_{i-1} - p_i) I + (\tau_{i-1} - p_i(\sigma_{i-1} - p_i))(A + p_i I)^{-1}\right) v_{k-1}; \\ Z_i &= \left[Z_{i-1}, \quad \overline{-2p_i} v_{k+1}\right]; \quad k = k + 1; \end{aligned}$$

  - elseif p_i is complex,
    - if p_{i-1} is real,

$$\begin{aligned} v_{k+1} &= Q_i T_{i-1} v_k = \left(A^2 - \sigma_i A + \tau_i I\right)^{-1} (A - p_{i-1}) v_k; v_{k+2} = A v_{k+1}; \\ Z_i &= \left[Z_{i-1}, \sqrt{2\sigma_i}\sqrt{\tau_i} v_{k+1}, \sqrt{2\sigma_i} v_{k+2}\right]; \quad k = k + 2; \end{aligned}$$

    - elseif p_{i-1} is complex,

$$\begin{aligned} v_{k+1} &= Q_i R_{i-1} v_{k-1} = \left(I + ((\sigma_i + \sigma_{i-1}) A + (\tau_{i-1} - \tau_i) I)\left(A^2 - \sigma_i A + \tau_i I\right)^{-1}\right) v_{k-1}; \\ v_{k+2} &= A v_{k+1}; \\ Z_i &= \left[Z_{i-1}, \sqrt{2\sigma_i}\sqrt{\tau_i} v_{k+1}, \sqrt{2\sigma_i} v_{k+2}\right]; \quad k = k + 2; \end{aligned}$$

r 9

r i i

i i s

9.1 ow rank – I

The work required to run CF–ADI on a sparse matrix A is $O(Jpn)$, where p is the number of columns in B, and J is the number of CF–ADI iterations. For many large systems, $O(n)$ complexity is considered acceptable, and $O(n^2)$ is considered too expensive.

Since $p \ll n$, $Jpn \ll n^2$ if and only if $J \ll n$. Thus, the total number of CF–ADI iterations should be much smaller than the system size n, for CF–ADI to be practical on large problems. Therefore, it is possible that algorithm 9 will be terminated at a small J, before convergence, to ensure that the complexity of algorithm 9 stays $O(n)$. In that case, $X_J^{cfadi} := Z_J^{cfadi}(Z_J^{cfadi})^T$, $J \ll n$, is necessarily a low rank approximation to the exact solution X.

Since CF–ADI necessarily provides only a low rank approximation to the solution to (8.1), this section justifies the usefulness of a low rank approximation to the exact solution. The first part explains why a low rank matrix can often be a very good approximation to the exact solution to (8.1). The second part deals with the case when the exact solution cannot be well approximated by a low rank matrix, in which case CF–ADI provides a good approximation to an optimal low rank Cholesky factor, which is needed in the low rank model reduction methods proposed in chapter 5.

The effectiveness of the CF–ADI algorithm in each case is illustrated by numerical examples.

9.2 Exact solution close to low rank

In [36] it was noted that ADI converges in a few iterations for many problems if good parameters are chosen, which means that the exact solution in those cases is close to low

rank, since it can be well approximated by a low rank matrix.

The justification of why the solution to (8.1) is often close to low rank was given in [42] for symmetric A.

Proposition 10. *Let $A \in \mathbb{R}^{n\times n}$ be a stable, symmetric matrix with $\kappa = \kappa(A) = \frac{\lambda_n(A)}{\lambda_1(A)}$, $\lambda_n(A) \leq \lambda_1(A) < 0$, $B \in \mathbb{R}^{n\times p}$ a nonzero matrix, and $\lambda_i(X)$, $i = 1, \cdots, n$, the non-increasing ordered eigenvalues of X, then*

$$\frac{\lambda_{pk+1}(X)}{\lambda_1(X)} \leq \prod_{j=0}^{k-1} \frac{\kappa^{\frac{2j+1}{2k}} - 1}{\kappa^{\frac{2j+1}{2k}} + 1}^{2} \tag{9.1}$$

for $1 \leq pk < n$ [42].

□

A smaller κ value indicates faster decay. Figure 9-1, also taken from [42], illustrates the decay bound (9.1). The right hand side of (9.1) is smaller than 0.01 at $k = 20$, for all κ values ranging from 10 to 10^5.

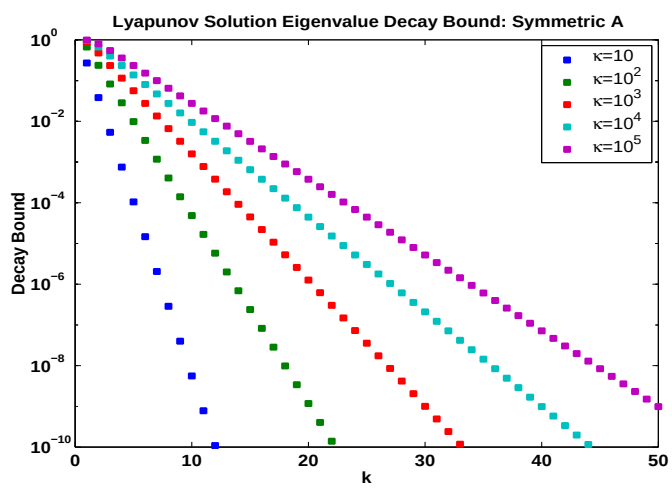


Figure 9-1: Eigenvalue decay bound, symmetric case

Thus, if κ is reasonably small, the exact solution to (8.1) when A is symmetric is a matrix which has very fast eigenvalue decay. Most of the solution's eigenvalues are negligible compared to the few largest ones. In other words, X, which is symmetric, has an eigenvalue (singular value) decomposition,

$$X = [u_1, \cdots, u_n] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix} [u_1, \cdots, u_n]^T, \tag{9.2}$$

where $\sigma_1 \geq \cdots \geq \sigma_J > \sigma_{J+1} \geq \cdots \geq \sigma_n \geq 0,\ \sigma_1 \gg \sigma_{J+1},\ J \ll n$. Therefore,

$$X = X_J^{large} + X_{n-J}^{small}, \tag{9.3}$$

$$X_J^{large} := [u_1, \cdots, u_J] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J \end{bmatrix} [u_1, \cdots, v_J]^T, \tag{9.4}$$

$$X_{n-J}^{small} := [u_{J+1}, \cdots, u_n] \begin{bmatrix} \sigma_{J+1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{n-J} \end{bmatrix} [u_{J+1}, \cdots, u_n]^T, \tag{9.5}$$

where $\|X_J^{large}\|_2 \gg \|X_{n-J}^{small}\|_2$. Hence, the exact solution X is close to low rank, in the following sense,

$$\|X - X_J^{large}\|_2 = \sigma_{J+1} \ll \sigma_1 = \|X\|_2, \quad rank(X_J^{large}) = J, \quad J \ll n. \tag{9.6}$$

There is no bound similar to (9.1) for a non-symmetric matrix A, but one also frequently encounters rapid eigenvalue decay when A is non-symmetric.

Figure 9.2 shows the eigenvalue decay of the solutions to (1.50) and (1.51) when A is non-symmetric. The matrix A comes from the discretized transmission line example shown in figure 5-1. It is a 256×256 matrix and B has one column. Figure 9.2 shows the rapid decay of the eigenvalues of $P \in \mathbb{R}^{256\times256}$, the solution to (1.50), and $Q \in \mathbb{R}^{256\times256}$, the solution to (1.51). The magnitude of each set of eigenvalues has decayed to 10^{-5} of the magnitude of the largest eigenvalue by $k = 20$. Thus, both $P \in \mathbb{R}^{256\times256}$ and of $Q \in \mathbb{R}^{256\times256}$ can be reasonably well approximated by rank 20 matrices.

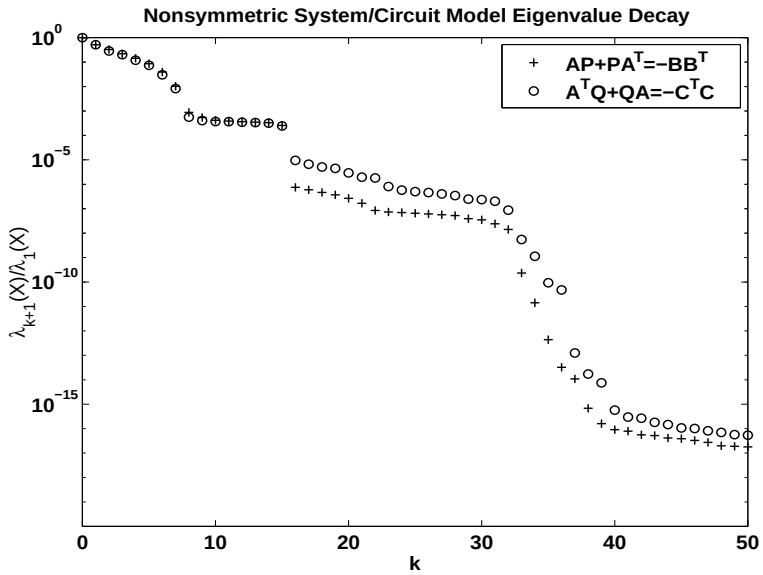


Figure 9-2: Discretized transmission line, 256 states.

9.3 o inant eigenspace of t e yapunov solution

In the case when the solution to (8.1) is not close to low rank as according to (9.6), and CF–ADI is still run only a small number of steps, a low rank approximation is produced. In this case, it is hoped that this low rank approximation will be close to optimal. To simplify the analysis, in this section assume B has only one column, thus $Z_J^{cf\,adi} \in \mathbb{R}^{n\times J}$. Also assume $Z_J^{cf\,adi}$ has full column rank.

If the exact solution X to (8.1) has an eigenvalue (singular value) decomposition,

$$X = [u_1, \cdots, u_n] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix} [u_1, \cdots, u_n]^T, \tag{9.7}$$

$$\sigma_1 \geq \cdots \geq \sigma_n \geq 0, \tag{9.8}$$

where the σ_i's do not necessarily decay rapidly, X can still be divided into two parts,

$$X = X_J^{large} + X_{n-J}^{small}, \tag{9.9}$$

$$X_J^{large} := [u_1, \cdots, u_J] \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J \end{bmatrix} [u_1, \cdots, u_J]^T, \tag{9.10}$$

$$X_{n-J}^{small} := [u_{J+1}, \cdots, u_n] \begin{bmatrix} \sigma_{J+1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{n-J} \end{bmatrix} [u_{J+1}, \cdots, u_n]^T. \tag{9.11}$$

With the assumption that $\sigma_J > \sigma_{J+1}$, X_J^{large} is the unique optimal rank J approximation to X in the 2-norm [20]. From now on, the optimal 2-norm rank J approximation (assumed unique) to X will be denoted by X_J^{opt}.

If σ_{J+1} is not small, then X cannot be well approximated by a rank J matrix. In that case, the most one can hope for regarding the rank J CF–ADI approximation, in the 2-norm, is that it is close to the optimal rank J approximation,

$$X_J^{cfadi} := Z_J^{cfadi}(Z_J^{cfadi})^T \approx X_J^{opt}. \tag{9.12}$$

The matrices X_J^{cfadi} and Z_J^{cfadi} both have rank J under the assumptions that B has only one column and Z_J^{cfadi} has full column rank.

In the model reduction context, it is often not important to capture the eigenvalues of X_J^{opt} exactly, rather, it is the eigenspace, $colsp([u_1, \cdots, u_J])$, associated with the large eigenvalues of X, which is significant.

Thus, it is hoped that

$$colsp([u_1^{cfadi}, \cdots, u_J^{cfadi}]) \approx colsp([u_1^{opt}, \cdots, u_J^{opt}]), \tag{9.13}$$

where

$$X_J^{opt} = [u_1^{opt}, \cdots, u_J^{opt}] \begin{bmatrix} \sigma_1^{opt} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J^{opt} \end{bmatrix} [u_1^{opt}, \cdots, u_J^{opt}]^T, \tag{9.14}$$

$$\sigma_1^{opt} \geq \cdots \geq \sigma_J^{opt} > 0, \tag{9.15}$$

and

$$X_J^{cfadi} = [u_1^{cfadi}, \cdots, u_J^{cfadi}] \begin{bmatrix} \sigma_1^{cfadi} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_J^{cfadi} \end{bmatrix} [u_1^{cfadi}, \cdots, u_J^{cfadi}]^T, \tag{9.16}$$

$$\sigma_1^{cfadi} \geq \cdots \geq \sigma_J^{cfadi} > 0, \tag{9.17}$$

are singular value decompositions, with the zero eigenvalues and their associated eigenvectors excluded.

The eigenvectors $\{u_1^{cfadi}, \cdots, u_J^{cfadi}\}$ can be obtained by finding the left singular vectors of Z_J^{cfadi}.

9.4 o inant eigenspace, rational rylov su spaces, – I

The CF–ADI approximation can be written in the following way,

$$X_J^{cfadi} = Z_J^{cf}(Z_J^{cf}) = U_J^{cf} Y_{J\times J} (U_J^{cf})^T, \tag{9.18}$$

where

$$(U_J^{cf})^T (U_J^{cf}) = I_{J\times J}, \quad colsp(U_J^{cf}) = colsp(Z_J^{cf}). \tag{9.19}$$

The columns of U_J^{cf} form a basis for the range of X_J^{cfadi}. Thus, the CF–ADI approximation has the same form as the low rank ideas proposed in [25, 27], and described in section 7.1.3. In all these low rank methods, X is approximated by a low rank matrix,

$$X \approx U_J Y_{J\times J} U_J^T, \tag{9.20}$$

where the columns of U_J form an orthonormal basis for the range of the low rank approximation. The important difference between the low rank methods in [25, 27] and CF–ADI is the choice of $colsp(U_J)$.

In [25, 27], the columns of U_J form an orthonormal basis for $\mathcal{K}_J(A, B)$,

$$colsp(U_J) = \mathcal{K}_J(A, B) = colsp[B, AB, A^2B, \cdots, A^{J-1}B]. \tag{9.21}$$

In CF-ADI, the choice is (9.19). The choice in (9.21) is intuitive because of proposition 9. Corollary 2, an immediate consequence of theorem 2 and lemma 3, shows that (9.19) is intuitive in the same way.

Corollary 2. *If $Z_n^{cfadi} = [z_1, \cdots, z_n]$ is the nth CF–ADI approximation, and $\{p_1, \cdots, p_n\}$ is any CF–ADI parameter set for which no $(A + p_i I)$ is singular, and $B \in \mathbb{R}^n$, then*

$$\begin{aligned} colsp(Z_n^{cfadi}) &= \mathcal{K}_n^{inv}(A, (A - p_1)^{-1}B, \{p_2, \cdots, p_n\}), & (9.22)\\ &= span\left(\left[B, AB, \cdots, A^{n-1}B\right]\right), & (9.23)\\ &= range(X), & (9.24) \end{aligned}$$

where X is the solution to (8.1).

The following two corollaries show what happens when a CF–ADI iterate is a linear combination of the previous iterates.

Corollary 3. *Let $Z_j^{cfadi} = [z_1, \cdots, z_j]$ be the jth CF–ADI approximation, and $\{p_1, \cdots, p_j\}$ be any CF–ADI parameter set for which no $(A + p_i I)$ is singular, and $B \in \mathbb{R}^n$. If z_{j+1} is a linear combination of $\{z_1, \cdots, z_j\}$, then z_l is a linear combination of $\{z_1, \cdots, z_j\}$ whenever $l \geq j + 1$.*

Proof. See lemma 3. □

Corollary 4. *If z_{j+1} at the $j + 1$th step of the CF–ADI iteration is a linear combination of the previous iterates, $z_1, \cdots, z_j$, and $B \in \mathbb{R}^n$, then*

$$Range(X) = span\{z_1, \cdots, z_j\}, \tag{9.25}$$

where X is the solution to (8.1).

Proof. Because

$$\begin{aligned} Range(X) &= span\{B, AB, \cdots, A^{n-1}B\}, & (9.26)\\ &= span\{z_1, z_2, \cdots, z_j, \cdots, z_n\}, & (9.27)\\ &= span\{z_1, z_2, \cdots, z_j\}. & (9.28) \end{aligned}$$

□

Corollary 4 says that if the $j + 1$st CF–ADI iterate, z_{j+1}, is a linear combination of the previous columns, then a basis for the range of the exact solution has been found.

If the goal is to find the range of the exact solution X, then iteration can stop when z_{j+1} is linear combination of the previous columns. If, however, the goal is to approximate the exact solution X by $Z_j^{cfadi}(Z_j^{cfadi})^T$, then iteration may have to continue, since even if $Z_j^{cfadi}(Z_j^{cfadi})^T$ has the same range as X, they may not be close as matrices.

The range of X can also be characterized in terms of its eigenvectors. Let the eigendecomposition of X be as in (9.7), and the eigenvalues ordered so that,

$$\sigma_1 \geq \cdots, \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0, \tag{9.29}$$

then $u_1, \cdots, u_r$, the eigenvectors of X associated with nonzero eigenvalues, span the range of X,

$$Range(X) = span\,\{u_1, \cdots, u_r\}\,. \tag{9.30}$$

Theorem 2 in chapter 6 shows that $Range(X) = \mathcal{L}(A, B, \mathbf{p})$, and $\mathcal{L}(A, B, \mathbf{p})$ can have an infinite number of characterizations, as Krylov subspaces, as rational Krylov subspaces, and as their sums.

A few examples of these characterizations appear below,

$$\begin{aligned}
span\,\{u_1, \cdots, u_r\} &= Range(X), && (9.31)\\
&= span\{B, AB, \cdots, A^{n-1}B\}, && (9.32)\\
&= span\{A^{-1}B, A^{-2}B, \cdots, A^{n}B\}, && (9.33)\\
&= span\{z_1^{cfadi}(\mathbf{p}), z_2^{cfadi}(\mathbf{p}), \cdots, z_n^{cfadi}(\mathbf{p})\},\ \text{any } \{p_1, \cdots, p_n\} && (9.34)\\
&= \bigcup_{i=1}^{m} \mathcal{K}_{i_s}\left((A - q_i I), (A - q_i I)^{-1}B\right), && (9.35)\\
& \quad 1_s + \cdots + m_s = n,\ \text{any } q_1, \cdots, q_m. && (9.36)
\end{aligned}$$

Therefore, the span of the eigenvectors of X associated with non-zero eigenvalues is the same as the span of the columns of the nth CF–ADI approximation Z_n^{cfadi}. Similarly, low rank methods which utilizes (9.32) [25, 27], and (9.33), instead of the CF–ADI choice of (9.34), also find the span of the eigenvectors of X associated with non-zero eigenvalues when run to full n steps.

Frequently it is not practical to run any of these Krylov subspace-based algorithms to n steps to find the full range of X. Instead, the measure of success for a low rank method is how well the partial basis it generates approximates X's dominant eigenvectors.

In other words, if $\{w_1, \cdots, w_J\}$ is the partial basis generated by J steps of a low rank method, how well does

$$span\{w_1, \cdots, w_J\} \approx span\,\{u_1, \cdots, u_J\}\,, \quad J \ll n, \tag{9.37}$$

where $u_1, \cdots, u_J$ are the dominant eigenvectors of X, in order of decreasing importance? To avoid ambiguity, the eigenvalue associated with u_J is assumed to be strictly larger than the eigenvalue associated with u_{J+1}.

Several possibilities for $\{w_1, \cdots, w_J\}$ are

$$span\{w_1, \cdots, w_J\} = span\{B, AB, \cdots, A^{J-1}B\}, \tag{9.38}$$

or

$$span\{w_1, \cdots, w_J\} = span\{z_1^{cfadi}(\mathbf{p}), z_2^{cfadi}(\mathbf{p}), \cdots, z_J^{cfadi}(\mathbf{p})\}, \text{ any } \{p_1, \cdots, p_J\}, \tag{9.39}$$

or some other set of J vectors from theorem 2. Due to practical considerations, the starting vector w_1 should not contain too many powers of shifts of A or inverse powers of shifts of A.

Clearly, the answer to the question of which choice of a partial basis approximates $span\{u_1, \cdots, u_J\}$ better depends on A, B, J, and the shift parameters. However, since there is more freedom in picking $\{w_1, \cdots, w_J\}$ according to (9.39), which amounts to picking the CF–ADI parameters $\{p_1, \cdots, p_J\}$, than according to (9.38), one expects to be able to approximate the span of the first J eigenvectors of X better with CF–ADI, if the CF–ADI parameters are well chosen.

9.5 u erical results

This section provides numerical result on how well CF–ADI approximates the dominant eigenspace of X.

Figure 9-3 shows dominant eigenspace approximation, where the matrices A and B came from the spiral inductor problem considered in section 8.7. The matrix A is symmetric, 500×500, and B has one column. CF–ADI is run for 20 iterations. The relative error after 20 iterations is $\frac{\|X - X_j^{cfadi}\|_2}{\|X\|_2} = 10^{-8}$.

Figure 9-3(a) measures the closeness of the 20-dim dominant eigenspaces of X and X_{20}^{cfadi}. This measure is provided by the concept of principle angles between subspaces [20]. Let S^1 and S^2 be two subspaces, of dimension d_1 and d_2, respectively, and assume $d_1 \geq d_2$. Then the d_2 principle angles are defined as $\theta_1, \cdots, \theta_{d_2}$, such that

$$cos(\theta_j) = \max_{u^1 \in S^1, \|u^1\|=1} \max_{u^2 \in S^2, \|u^2\|=1} (u^1)^T u^2 = (u_j^1)^T u_j^2, \tag{9.40}$$

under the constraints that

$$(u^1)^T u_i^1 = 0, \quad (u^2)^T u_i^2 = 0, \quad i = 1 : j-1. \tag{9.41}$$

If the columns of U^1 are an orthonormal basis for S^1, and the columns of U^2 an orthonormal basis for S^2, and $(U^2)^T U^1$ has singular value decomposition,

$$(U^1)^T U^2 = U \Sigma V^T, \tag{9.42}$$

then

$$cos(\theta_j) = \Sigma(j,j), \quad u_j^1 = U^1 U(:,j), \quad u_j^2 = U^2 V(:,j). \tag{9.43}$$

Thus, these two bases, $\{u_1^1, \cdots, u_{d_2}^1\}$ and $\{u_1^2, \cdots, u_{d_2}^2\}$, are mutually orthogonal, $(u_i^1)^T u_j^2 = 0$, if $i \neq j$. And $(u_i^1)^T u_i^2 = cos(\theta_i)$ indicates the closeness of u_i^1 and u_i^2. If $S^1 = S^2$, then $cos(\theta_j) = 1$, $j = 1, \cdots, d_1 = d_2$. If $S^1 \perp S^2$, then $cos(\theta_j) = 0$, $j = 1, \cdots, d_2$. A basis for the intersection of S^1 and S^2 is given by those basis vectors whose principle angle is 0.

$$range(S^1) \cap range(S^2) = span\{u_1^1, \cdots, u_s^1\} = span\{u_1^2, \cdots, u_s^2\}, \tag{9.44}$$

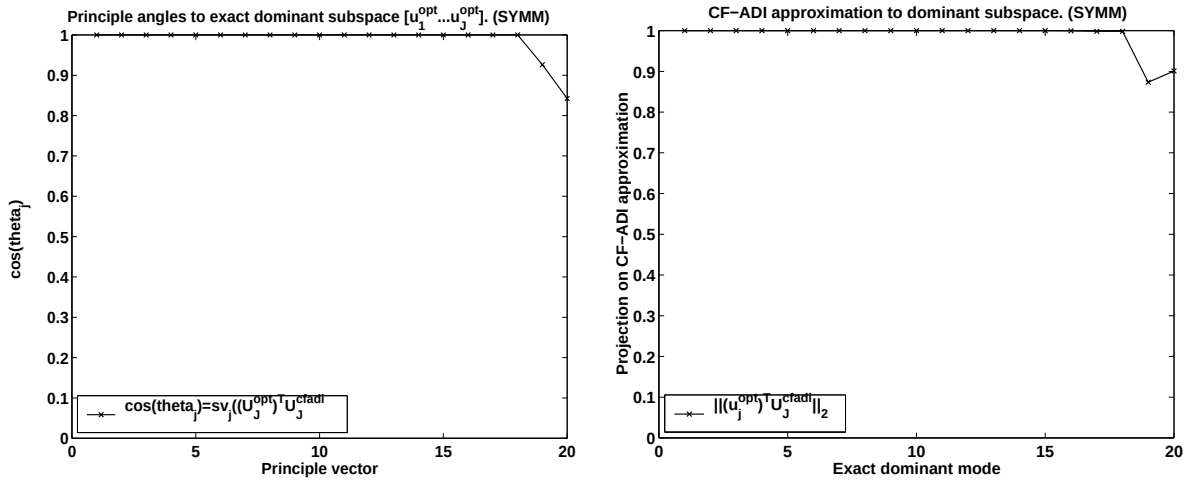
$$1 = cos(\theta_1) = \cdots = cos(\theta_s) > cos(\theta_{s+1}). \tag{9.45}$$

Thus, the closeness of two subspaces is measured by how many of their principle angles are close to 0.

In figure 9-3(a) the cosines of the principle angles between U_{20}^{cfadi} and U_{20}^{opt} are plotted. The cosines of 18 of the principle angles are 1, and the cosines of the last two are above 0.85, indicating close match of all dominant eigenvectors. This is not surprising since $\|X - X_{20}^{cfadi}\|/\|X\|$ is less than 10^{-8}.

Because the eigenvectors of X_{20}^{opt} associated with the larger eigenvalues are more important than the eigenvectors of X_{20}^{opt} associated with the smaller (non-zero) eigenvalues in view of later application to model reduction, as they indicate the more controllable or observable modes among the top 20, it is worthwhile to see how well each eigenvector of X_{20}^{opt} is individually matched by U_{20}^{cfadi}. This is measured by the norm of the projection of the exact dominant eigenvector, u_j^{opt}, onto U_{20}^{cfadi}. The direction u_j^{opt} is contained in the column span of U_{20}^{cfadi} if $\|(u_j^{opt})^T U_{20}^{cfadi}\|_2 = 1$. This is a different criterion than the one based on principle angles, as u_j^{opt} may not be one of the vectors in the orthogonal basis in (9.43).

As can be seen in figure 9-3(b), and not from figure 9-3(a), u_{20}^{opt} is better represented by the vectors in U_{20}^{cfadi} than is u_{19}^{opt}. Everything being equal, it is preferable for u_{19}^{opt} to be better represented than u_{20}^{opt}, because u_{19}^{opt} is more important in terms of controllability or observability.



(a) Principle angle

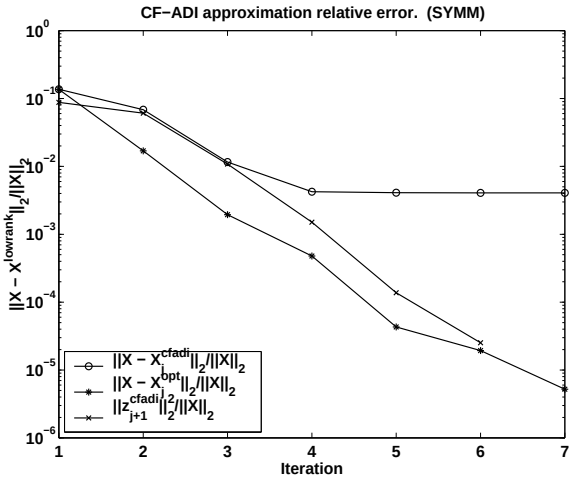
(b) Dominant subspace projection

Figure 9-3: Symmetric matrix, $n = 500$, 20 CF–ADI iterations, converged

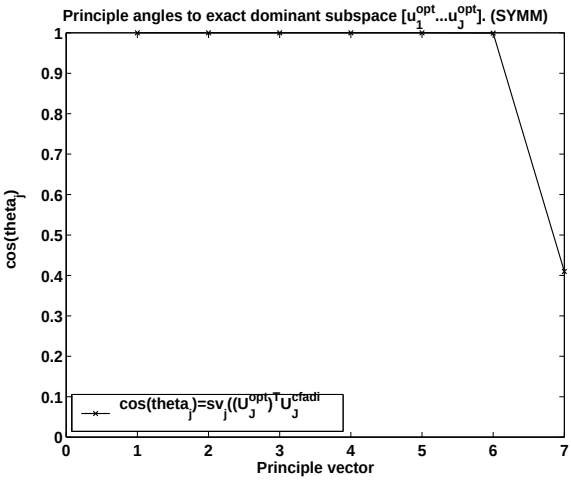
In contrast to figure 9-3, figures 9-4 and 9-5 demonstrate dominant subspace approximation when CF–ADI is not run to convergence.

Figure 9-4 is the same spiral inductor example as in figure 9-3, but CF–ADI is only run 7 steps. In figure 9-4(a), $\|z_7^{cfadi}\|$ is small but $\|X - X_j^{cfadi}\|_2$ has stagnated. The relative error $\frac{\|X - X_7^{cfadi}\|_2}{\|X\|_2}$ is between 10^{-2} and 10^{-3}, whereas the relative error of the optimal rank 7 approximation is 10^{-5}. However it can be seen from figure 9-4(b) that the intersection of the column span of U_7^{cfadi} and the column span of U_7^{opt} has dimension 6, since the cosines of 6 principles angles are 1. In figure 9-4(c), it can be seen that the top 5 dominant eigenvectors of X, the 5 most important modes, are contained entirely in the column span of U_7^{cfadi}. The norm of the projection of u_6^{opt} onto U_7^{cfadi} is around 0.9, while that of u_7^{opt} is around 0.5.

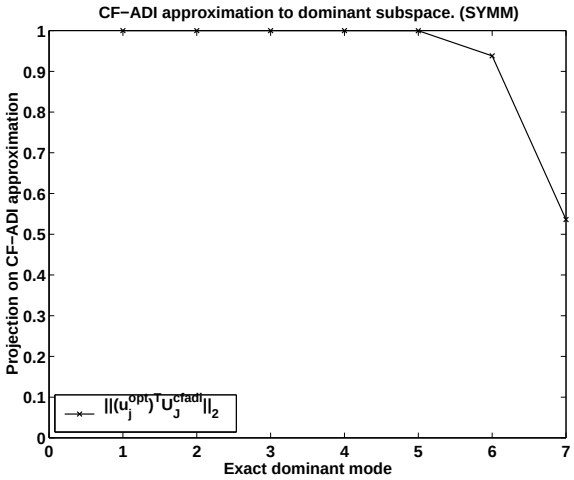
Thus, dominant eigenspace information about X can emerge, even when CF–ADI has not converged.



(a) CF–ADI convergence



(b) Principle angle



(c) Dominant subspace projection

Figure 9-4: Symmetric matrix, $n = 500$, 7 CF–ADI iterations, not converged

Figure 9-5 shows another example of running CF–ADI only a small number of steps, before convergence occurs. It comes from the transmission line example (figure 5-1). The system matrix A is 256×256, and the input matrix B has one column.

Figure 9-5 contains results for the solutions to the two Lyapunov equations (1.50-1.51). The solution to (1.50) is denoted by P, and the solution to (1.51) is denoted by Q.

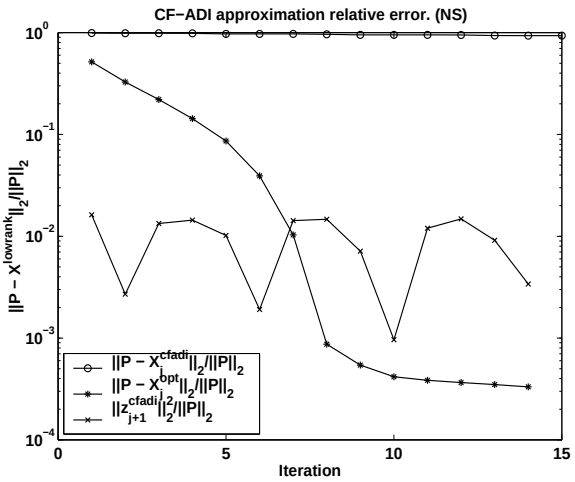
Compared to the Lyapunov solution associated with the spiral inductor example, whose system matrix is symmetric, the two Lyapunov solutions associated with the non-symmetric matrix A in this example have slower eigenvalue decay. In section 9.2 it is asserted that both P and Q are close to rank 20 matrices. Since the eigenvalues of a non-symmetric matrix can

be in an arbitrary region in the open left half plane, the problem of parameter selection is also more difficult for this example than for the symmetric example. The selection procedure in [58] was followed and the resulting parameters are complex.

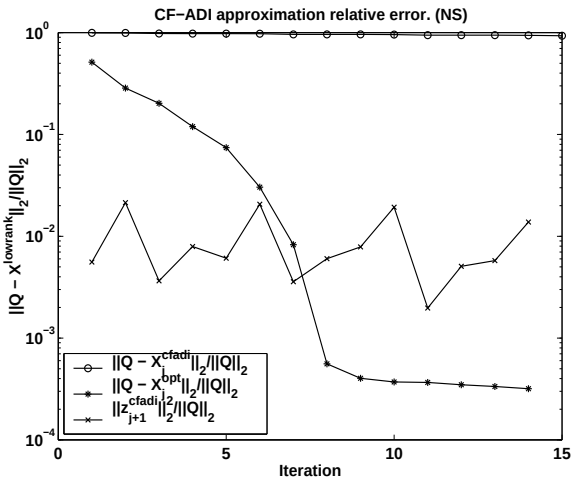
Figure 9-5(a) and 9-5(b) show that the CF–ADI error is not decreasing at all during 15 iterations. The relative error stagnates at 1. However, figure 9-5(c) shows that the intersection of the span of the 15 dominant eigenvectors of P and the span of the 15 dominant eigenvectors of the CF–ADI approximation has dimension 10 (almost 11). Similarly, the intersection of the span of the 15 dominant eigenvectors of Q and the span of the 15 dominant eigenvectors of the CF–ADI approximation has dimension 10.

Figure 9-5(d) provides an interesting picture. Recall that eigenvectors of P or Q associated with larger eigenvalues are more important than the eigenvectors associated with smaller eigenvalues. In figure 9-5(d), a lower index indicates a more important eigenvector. It can be seen that the 5 most important eigenvectors of P (Q) are represented almost completely in $span(U_{15}^{cfadi-P(Q)})$. What is interesting is that the 9th and 10th eigenvectors of Q are also completely represented, even though eigenvectors 7 and 8 are not. The eigenvectors of P display similar, if not as dramatic, behavior, whereby some middle eigenvectors are not as well captured as the eigenvectors to their left and right.

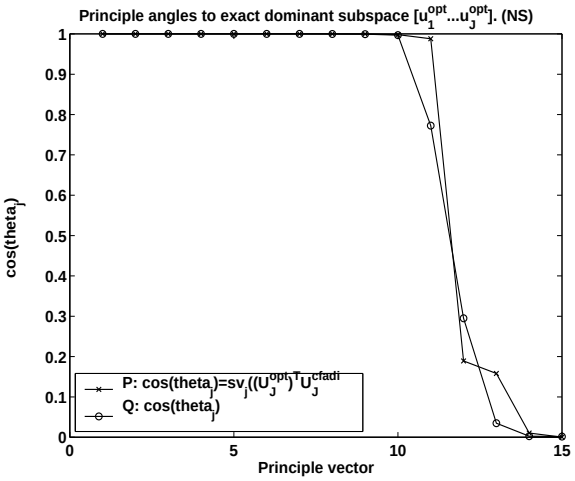
This example demonstrates that even if the CF–ADI error is large, some information about the dominant eigenspace can still emerge, although there may also be missing information.



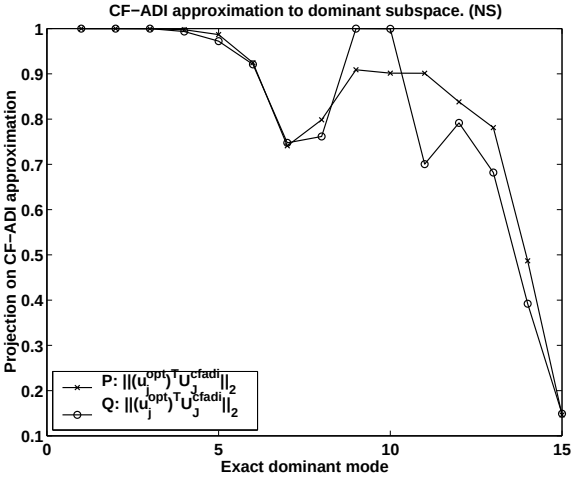
(a) CF–ADI convergence. $AP + PA^T = -BB^T$.



(b) CF–ADI convergence. $A^TQ + QA = -C^TC$.



(c) Principle angle



(d) Dominant subspace projection

Figure 9-5: Non-symmetric matrix, $n = 256$, 15 CF–ADI iterations, not converged

r 0

l i i –

Chapter 5 addressed the issue of how to utilize low rank approximations to the two system gramians in a model reduction method, with the goal of approximating the TBR reduced model. The solution is clear for symmetric systems, and two approaches, the Low Rank Square Root method and the Dominant Gramian Eigenspaces method, are developed for non-symmetric systems.

The question of how to obtain low rank approximations to the system gramians is answered with the development of the CF–ADI method. Other methods of generating low rank approximations, such as [27], can also be used in this context. In fact, an infinite variety of low order Krylov and rational Krylov bases can be used to generate low rank approximations. Any subset containing consecutive elements of the infinite spanning set $\{\cdots, v_{-j}, \cdots, v_0, \cdots, v_j, \cdots\}$ for the subspace $\mathcal{L}(A, B, \mathbf{p})$, whose characterization was given in theorem 2, suffices as a basis for the range of a low rank approximation.

Theoretical and numerical results in chapter 9 support the belief that CF–ADI can produce good approximately optimal low rank Cholesky factors.

This chapter uses the CF–ADI algorithm to generate the low rank Cholesky factors needed in the Dominant Gramian Eigenspaces method. Numerical results for symmetric and non-symmetric systems are given.

For symmetric systems, it is shown that, if the reduced model order equals the CF–ADI approximation order, Approximate TBR via CF–ADI (algorithm 11) results in a reduced model which is equivalent to the reduced model produced by a particular moment matching via rational Krylov subspaces method. Thus, from the point of view of moment matching, the problem of picking good moment matching points, so that the reduced model approximates the TBR reduction, can be approached by solving the CF–ADI parameter selection problem (7.43).

Numerical comparison of several CF–ADI parameter selection procedures are also made.

10.1 Sy etric syste s

A symmetric state-space system of the form (5.10-5.11) can be reduced according to algorithm 11.

Algorithm 11 Approximate TBR via CF–ADI for symmetric systems

INPUT: A, B.

1. Compute $Z_J \in \mathbb{R}^{n\times J}$, $Z_J Z_J^T \approx P_J^{opt}$, by CF-ADI, algorithm 9.
2. Obtain the order k reduced system $(A_k^r, B_k^r, (B_k^r)^T)$ according to algorithm 4.

10.1.1 Connection to moment matching

Here, for symmetric systems, a connection is established between Approximate TBR via CF–ADI and the moment matching method given in algorithm 1.

Theorem 5. *The reduced model obtained by algorithm 11 using $\{p_1, p_2, \cdots, p_J\}$ as the CF–ADI parameters, when the reduced model order k equals the CF–ADI approximation order J, is equivalent to the reduced model obtained by algorithm 1, which matches i_s moments at the point $-p_i$, where p_i appears i_s times in the parameter list $\{p_1, p_2, \cdots, p_J\}$.*

Proof. By theorem 3, algorithm 11 and algorithm 1 produce the same projection spaces, namely,

$$
\begin{aligned}
col(U_J^{algo-11}(\{p_1, \cdots, p_J\})) &= span\{z_1, \cdots, z_J\} && (10.1)\\
&= \{(A+p_1)^{-1}B, \cdots, \bigcup_{i=1}^{J} (A+p_iI)^{-1}\} && (10.2)\\
&= \bigcup_{i=1}^{m} \{(A+p_iI)^{-1}B, \cdots, (A+p_iI)^{-i_s}B\} && (10.3)\\
&= \bigcup_{i=1}^{m} \mathcal{K}_{i_s}\left((A-(-p_i)I), (A-(-p_i)I)^{-1}B\right), && (10.4)\\
&= col(U_J^{algo-1}(\{-p_1, \cdots, -p_J\})) && (10.5)
\end{aligned}
$$

where $1_s + \cdots + m_s = J$, and each p_i appears in $\{p_1, \cdots, p_J\}$ a total of i_s times. Hence the reduced models are equivalent by theorem 6. □

In trying to approximate TBR for symmetric systems via algorithm 11 with the parameters $\{p_1, p_2, \cdots, p_J\}$, and $k = J$, one obtains a reduced system which also matches moments of the original transfer function at $\{-p_1, -p_2, \cdots, -p_J\}$, with higher order moments denoted

by repeating the points. In this case, algorithm 11 can also be thought of as a moment matching method. The advantage of using algorithm 11 for symmetric systems instead of one of the moment matching algorithms described in chapter 3 is that the solution to the rational min-max problem on the real interval (7.43) is known, so optimal CF–ADI parameters can be found.

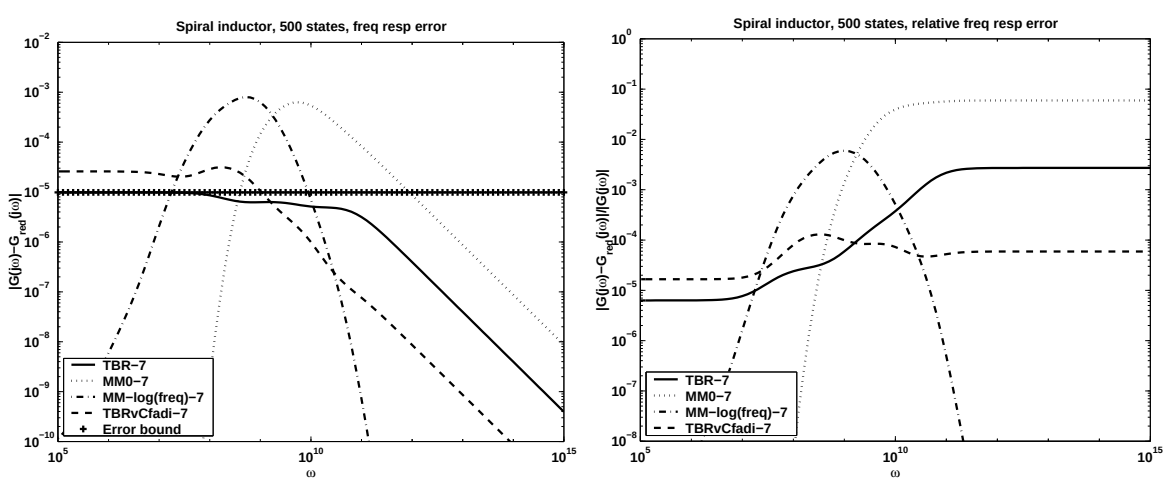
Thus, even if one starts from the view of matching transfer function moments, the question of which moment matching points to pick can be answered by solving the rational min-max problem (7.40), if the desire is to produce a reduced model which is close to the TBR reduction.

In addition, Z_B^J from CF–ADI contains more information than the projection matrix U_k obtained via moment matching. The columns of U_k are simply an orthonormal basis for the sum of several Krylov subspaces. The singular values of Z_B^J give an indication of approximately how controllable (and observable for symmetric systems) a mode is, and can be used in error estimation via (4.8).

Algorithm 11, as an approximation to the TBR method, is expected to produce a globally accurate reduced model.

10.1.2 Numerical results

Algorithm 11 was tested on the spiral inductor example (figure 8-2). The original system is single-input single-output, of order 500, and has been symmetrized according to (5.16-5.18).



(a) Frequency response error

(b) Relative error

Figure 10-1: Spiral inductor, order 7 reductions

Figure 10-1 compares four different order 7 reductions of the original system. One is TBR. The second is moment matching around $s = 0$, denoted 'MM0'. The third is moment

matching at 7 points distributed in the frequency range (as a real interval) $[10^5,\ 10^{15}]$ with log spacing, denoted 'MM-log(freq)'. The fourth is algorithm 11, denoted 'TBRvCfadi', where the reduced model order equals the CF–ADI approximation order, $J = k = 7$.

Figure 10-1(a) shows the magnitudes of the frequency response errors, $|G(j\omega) - G^{red}(j\omega)|$, of the four different approximations, as well as the TBR L^∞-error bound (4.8).

It can be seen that the TBR reduction has the smallest L^∞-error, $\sup_w |G(j\omega) - G^{red}(j\omega)|$, and it is below the TBR error bound. The L^∞-error of 'TBRvCfadi-7' is half an order of magnitude larger than TBR's. Both moment matching reductions' L^∞-errors are two orders of magnitude larger than TBR's.

Figure 10-1(b) shows the relative errors, $\frac{\|G(j\omega) - G_{red}(j\omega)\|}{\|G(j\omega)\|}$, of the same four order 7 reductions. Both the TBR and the Approximate TBR via CF–ADI reductions have comparatively flat relative errors, whereas the two moment matching reductions have regions with very small error and regions with much larger error.

Of course, the Approximate TBR via CF–ADI reduced model also matches moments at the negative of the CF–ADI parameters. Thus, one interpretation of the results shown in figure 10-1 is that the negative of the solution to the rational min-max problem on the real interval (7.43) is a better choice of moment matching points than log spaced points over the frequency range $[10^5,\ 10^{15}]$.

10.2 u erical co parison: – I para eters

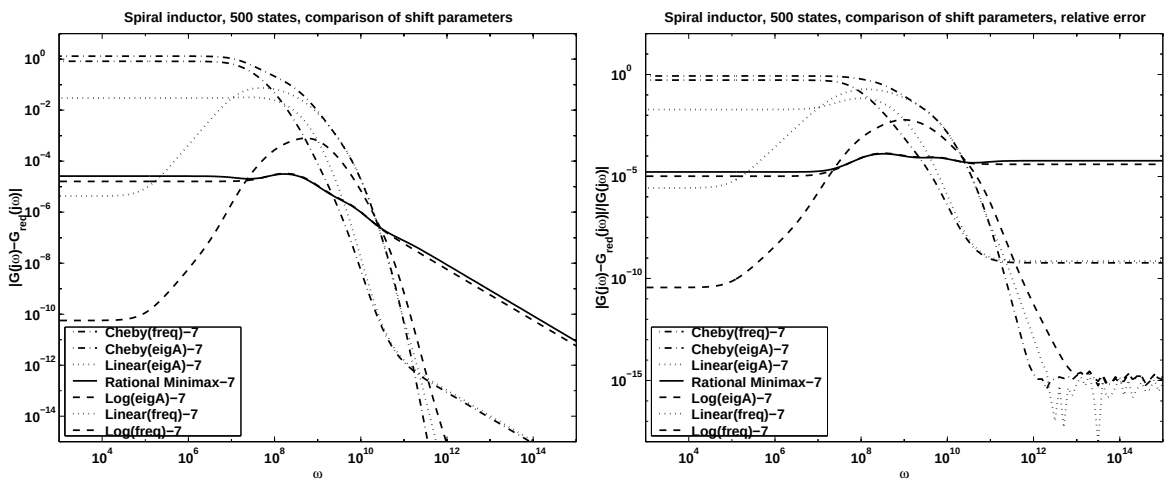
This section makes numerical comparison of several different selection procedures for the CF–ADI parameters, or equivalently, moment matching points, in terms of reduced model accuracy.

Figure 10-2 shows the frequency response errors, $|G(j\omega) - G_{red}(j\omega)|$, and relative errors, $|G(j\omega) - G_{red}(j\omega)|/|G(j\omega)|$, of seven parameter selection procedures for the spiral inductor example. One set of procedures chooses the parameters as a function of the frequency range of interest as a real interval, $[\omega_{min} = 10^5,\ \omega_{max} = 10^{15}]$. The other set chooses the parameters as a function of A's eigenvalue range, $[\lambda_{min} = -7.91 \times 10^{10},\ \lambda_{max} = -1.38 \times 10^7]$, which is also a real interval because A is symmetric. The spacings of the parameters are chosen to be linear, log, or Chebyshev on either $[\omega_{min},\ \omega_{max}]$, or $[\lambda_{min},\ \lambda_{max}]$. In addition, the solution of the real rational min-max problem (7.43) gives optimal parameters.

In figure 10-2(a), the legend is ordered so that the choices of parameters appear in the order their frequency response errors intersect the left vertical axis. The two Chebyshev choices have error 1 at low frequencies. Linear spacing on A's eigenvalue interval also attains highest error at low frequencies, around 3×10^{-2}. Linear and log spacing on the frequency interval have small errors at low frequencies, and attain maximum error in the middle frequency range, with L^∞-errors of 10^{-1} and 10^{-3}, respectively. The solution to the real min-max problem (7.43) and log spacing on A's eigenvalue interval have the smallest L^∞-errors of all

the choices, around 3×10^{-5}. In fact, these two choices picked parameter sets which are very close to each other.

Figure 10-2(b) shows the relative errors. Log spacing on A's eigenvalue interval and the solution to the real min-max problem (7.43) both have flat errors over the entire frequency range. It can be seen that knowing A's eigenvalue range helps one to pick good parameters. Without that knowledge, log spacing on the frequency range seems to work best.



(a) Frequency response error

(b) Relative error

Figure 10-2: Spiral inductor; shift parameters are important

10.3 on-sy etric syste s

In chapter 5 two low rank reduction methods were proposed, the Low Rank Square Root method, algorithm 5, and the Dominant Gramian Eigenspaces method, algorithm 6. CF–ADI can be used to produce low rank Cholesky factors for either method. If the CF–ADI error in algorithm 9 is small after only a small number of iterations on both (A, B) and (A^T, C^T), then algorithm 5 can be used. In that case, both gramians are close to low rank, and the CF–ADI approximations to them are fairly accurate. If the CF–ADI error is not small, then algorithm 6 should be used.

Because in chapter 5 it was shown that the Dominant Gramian Eigenspaces method, algorithm 6, generally produces a better reduced model than the Low Rank Square Root method, this section only shows results for using CF–ADI with algorithm 6.

10.3.1 Numerical results

This section uses the discretized transmission line example (figure 5-1) again.

Algorithm 12 Dominant Gramian Eigenspaces via CF–ADI

INPUT: A, B, C.

1. Compute $Z^B_{J_B}$, $Z^B_{J_B}(Z^B_{J_B})^T \approx P^{opt}_{J_B}$, by CF–ADI, algorithm 9, applied to (1.50).

2. Compute $Z^C_{J_C}$, $Z^C_{J_C}(Z^C_{J_C})^T \approx Q^{opt}_{J_C}$, by CF–ADI, algorithm 9, applied to (1.51).

3. Choose $k \leq J$, $2k$ being the desired reduction order.
   $U^{ctob}_m = qr(\begin{bmatrix} U^B_{n\times J}(:,1:k), & U^C_{n\times J}(:,1:k) \end{bmatrix})$
   note: $k \leq m = rank(U^{ctob}_m) \leq 2k$

4. Reduce the system:

$$A^r_m = (U^{ctob}_m)^T A U^{ctob}_m, \quad B^r_m = (U^{ctob}_m)^T B, \quad C^r_m = C U^{ctob}_m \tag{10.6}$$

Figure 10-3 shows numerical results obtained using algorithm 12. In figure 10-3(a), the frequency responses of three different reduced systems are shown. All three are order 10.

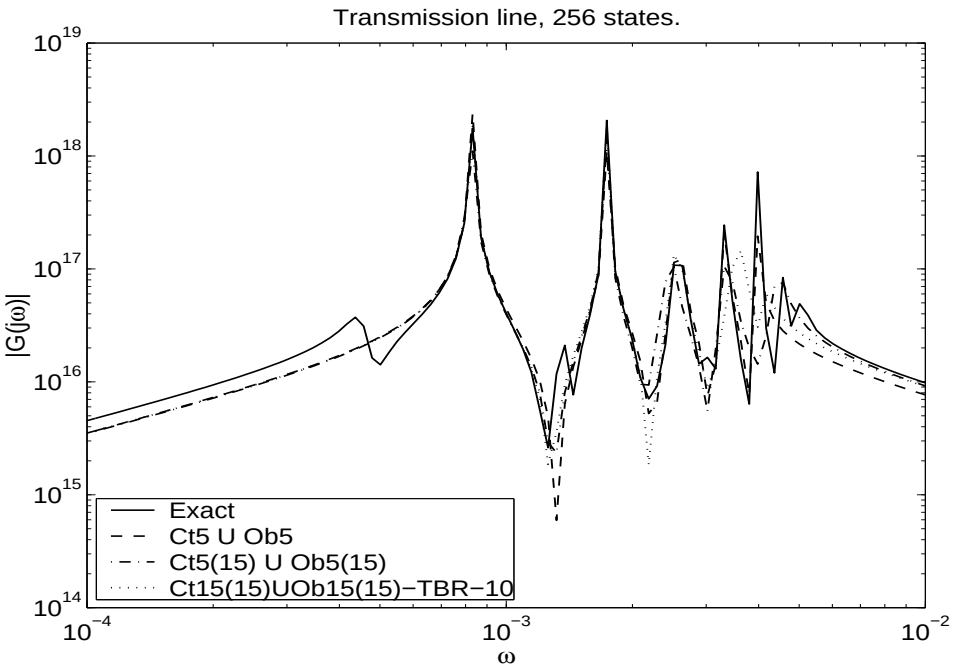
'Ct5 U Ob5' denotes using the 5 exact dominant controllable modes and the 5 exact dominant observable modes in steps 1 and 2 of algorithm 12. This reduction was shown to be indistinguishable from the order 10 TBR reduction in figure 5-2(a). 'Ct5(15) U Ob5(15)' denotes running 15 iterations of CF–ADI on (A, B) to obtain Z^B_{15}, and 15 iterations of CF–ADI on (A^T, C^T), to obtain Z^C_{15}, and letting $k = 5$ in step 3 of algorithm 12. 'Ct15(15) U Ob15(15)-TBR-10' denotes letting $k = 15$ instead, obtaining an order 30 reduced system, and then doing TBR on this reduced system to obtain the further reduced system of order 10.

The frequency responses of 'Ct5 U Ob5' and 'Ct5(15) U Ob5(15)' are close except at the last two peaks. 'Ct5 U Ob5' follows the next to last peak of the exact frequency response and then flattens out, whereas 'Ct5(15) U Ob5(15)' misses the next to last peak and finds the last one. 'Ct15(15) U Ob15(15)-TBR-10' has a peak in between the last two tall peaks. The fact that 'Ct15(15) U Ob15(15)-TBR-10' is not more accurate than 'Ct5(15) U Ob5(15)', rather, it is a bit worse, is surprising, since the projection spaces which produced the intermediate order 30 reduction contain the projection spaces which produced 'Ct5(15) U Ob5(15)'. The reason appears to be that the larger projection spaces have made the intermediate order 30 system unstable. Its system matrix has many eigenvalues with positive real parts.

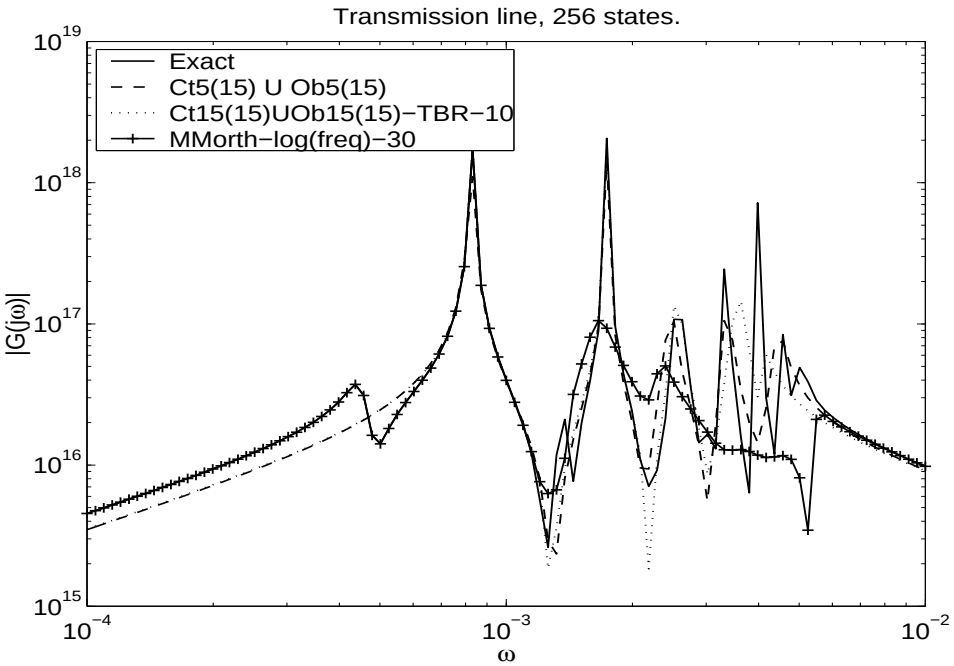
Figure 10-3(b) adds the frequency response of the reduced system obtained by the mo-

ment matching via orthogonal projection method given as algorithm 1. A total of 30 moment matching points were chosen in the frequency interval $[\omega_{min} = 10^{-4},\ \omega_{max} = 10^{-2}]$, with log spacing. 'MMorth-log(freq)-30' requires the same order of work as 'Ct5(15) U Ob5(15)'. It is an one-sided reduction as only a rational Krylov subspace with A and B is used. There is no contribution from the output coefficient matrix C.

It can be seen that 'MMorth-log(freq)-30' is extremely accurate at frequencies lower than 10^{-3}, but fails to capture any of the peaks beyond $\omega = 10^{-3}$. 'Ct5(15) U Ob5(15)' clearly captures the global frequency response behavior much better. It captures all but the next to last sharp peak. It averages the first tiny peak and small bumps between sharp peaks, which keeps the L^{∞}-error small without having to following every topographical feature exactly.



(a) Dominant Gramian Eigenspaces via CF–ADI



(b) Comparison with moment matching

Figure 10-3: Dominant Gramian Eigenspaces via CF–ADI

r

l si s r W r

In this dissertation, a low rank model reduction method, the Dominant Gramian Eigenspaces method, is proposed for the reduction of large, linear, time-invariant systems. This method utilizes low rank approximations to the exact system gramians.

Numerical comparison of the Dominant Gramian Eigenspaces method is made with another low rank model reduction method, the Low Rank Square Root method [41, 46]. It is shown that the Dominant Gramian Eigenspaces method often produces a better reduced model than the Low Rank Square Root method, when the low rank approximations to the system gramians have not converged to the exact gramians.

The system gramians are the solutions to two Lyapunov equations. In theorem 2 the range of the Lyapunov solution is characterized as order n Krylov and rational Krylov subspaces with different shifts and starting vectors. A connection is made between approximating the dominant eigenspace of the solution to the Lyapunov equation and the generation of various low order Krylov and rational Krylov subspaces.

The Cholesky Factor ADI algorithm is developed to generate a low rank approximation to the solution to the Lyapunov equation. Cholesky Factor ADI requires only matrix-vector products and linear solves, hence it enables one to take advantage of sparsity or structure in the system matrix.

The Cholesky Factor ADI algorithm is then used in conjunction with the Dominant Gramian Eigenspaces method in the model reduction of large, linear, time-invariant systems. It is demonstrated by numerical examples that this approach often produces a globally accurate reduced model, even when the low rank approximations to the system gramians have not converged to the exact gramians.

Finally, it is shown that, for symmetric systems, approximating Truncated Balanced Realization is achievable. Approximate TBR via CF–ADI for symmetric systems results in a reduction which also matches moments at the negative of the CF–ADI parameters, if the reduced model order is the same as the CF–ADI approximation order. It is shown that, from the point view of moment matching methods, the problem of picking points where moments

are to be matched, so that the reduced model is close to the TBR reduced model, can be approached by solving the rational min-max problem associated with CF–ADI parameter selection.

There is room for future research both in the area of low rank approximation to the Lyapunov solution and in low rank model reduction methods.

Further study is needed to characterize the eigenvalue behavior of the solution to the Lyapunov equation with a non-symmetric A matrix. It would be very useful to determine the conditions on A and B which will guarantee that the exact solution to the Lyapunov equation can be well approximated by a low rank matrix. [42] is the only work to the author's knowledge that addresses the issue of eigenvalue decay for the Lyapunov solution.

In the area of low rank model reduction methods, work needs to be done to find a method which genuinely approximates the TBR reduction for non-symmetric systems. Since the system gramians cannot be balanced without having the exact gramians, it is necessary to find a way to approximate the order k TBR projection matrices directly by low rank matrices, without referring to the gramians separately.

Finally, many of the results contained in this dissertation, on the solution of the Lyapunov equation and on low rank model reduction, can be extended to apply to the linear, time-varying model reduction problem [7, 54].

i li r

[1] R. H. Bartels and W. Stewart. Solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 1972.

[2] G. Birkhoff, R. S. Varga, and D. Young. Alternating direction implicit methods. In *Advances in Computers, Vol. 3*, pages 189–273. Academic Press, New York, 1962.

[3] J. Bracken, V. Raghavan, and R. Rohrer. Interconnect simulation with asymptotic waveform evaluation (AWE). *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(11):869–878, Nov. 1992.

[4] P. C. Chandrasekharan. *Robust control of linear dynamical systems.* Harcourt Brace, London ; San Diego, CA, 1996.

[5] E. Chiprout and M. Nakhla. Generalized moment-matching methods for transient analysis of interconnect networks. In *Proceedings of the 29th ACM/IEEE Design Automation Conference*, pages 201–206, 1992.

[6] E. Chiprout and M. Nakhla. Analysis of interconnect networks using complex frequency hopping (CFH). *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(2):186–200, Feb. 1995.

[7] P. Dewilde and A.-J. van der Veen. *Time-varying systems and computations.* Kluwer Academic Publishers, Boston, MA, 1998.

[8] I. Elfadel and D. Ling. A block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks. In *Proceedings of the International Conference on Computer-Aided Design*, pages 66 –71, 1997.

[9] I. Elfadel and D. Ling. Zeros and passivity of Arnoldi-reduced-order models for interconnect networks. In *Proceedings of the 34th Design Automation Conference*, pages 28–33, 1997.

[10] N. S. Ellner and E. L. Wachspress. Alternating direction implicit iteration for systems with complex spectra. *SIAM J. Numer. Anal.*, 28(3):859–870, 1991.

[11] D. F. Enns. Model reduction with balanced realizations: an error bound and frequency weighted generalizations. In *Proc. of 23rd Conf. on Decision and Control*, pages 127–132, Las Vegas, NV, Dec. 1984.

[12] P. Feldmann and R. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(5):639–649, May 1995.

[13] R. Freund and P. Feldmann. Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm. In *Proceedings of the IEEE Conference on Computer-Aided Design*, pages 280–287, 1996.

[14] R. W. Freund. Solution of shifted linear systems by quasi-minimal residual iterations. In *Numerical linear algebra (Kent, OH, 1992)*, pages 101–121. de Gruyter, Berlin, 1993.

[15] R. W. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. In *Applied and computational control, signals, and circuits, Vol. 1*, pages 435–498. Birkhäuser Boston, Boston, MA, 1999.

[16] K. Gallivan, E. Grimme, D. Sorensen, and P. Van Dooren. On some modifications of the Lanczos algorithm and the relation with Padé approximations. In *ICIAM 95 (Hamburg, 1995)*, pages 87–116. Akademie Verlag, Berlin, 1996.

[17] K. Gallivan, E. Grimme, and P. Van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Appl. Math. Lett.*, 7(5):75–80, 1994.

[18] K. Gallivan, E. Grimme, and P. Van Dooren. A rational Lanczos algorithm for model reduction. *Numer. Algorithms*, 12(1-2):33–63, 1996.

[19] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Internat. J. Control*, 39(6):1115–1193, 1984.

[20] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[21] E. Grimme. *Krylov projection methods for model reduction.* PhD thesis, University of Illinois at Urbana-Champaign, 1997.

[22] E. J. Grimme, D. C. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. *Numer. Algorithms*, 12(1-2):1–31, 1996.

[23] S. J. Hammarling. Numerical solution of the stable, nonnegative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2(3):303–323, 1982.

[24] R. A. Horn and C. R. Johnson. *Topics in matrix analysis.* Cambridge University Press, Cambridge, 1991.

[25] D. Y. Hu and L. Reichel. Krylov-subspace methods for the Sylvester equation. *Linear Algebra Appl.*, 172:283–313, 1992. Second NIU Conference on Linear Algebra, Numerical Linear Algebra and Applications (DeKalb, IL, 1991).

[26] M.-P. Istace and J.-P. Thiran. On the third and fourth Zolotarev problems in the complex plane. *SIAM J. Numer. Anal.*, 32(1):249–259, 1995.

[27] I. M. Jaimoukha and E. M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Numer. Anal.*, 31(1):227–251, 1994.

[28] I. M. Jaimoukha and E. M. Kasenally. Oblique projection methods for large scale model reduction. *SIAM J. Matrix Anal. Appl.*, 16(2):602–627, 1995.

[29] I. M. Jaimoukha and E. M. Kasenally. Implicitly restarted Krylov subspace methods for stable partial realizations. *SIAM J. Matrix Anal. Appl.*, 18(3):633–652, 1997.

[30] M. Kamon, F. Wang, and J. White. Recent improvements for fast inductance extraction and simulation [packaging]. In *Proceedings of the IEEE 7th Topical Meeting on Electrical Performance of Electronic Packaging*, pages 281–284, 1998.

[31] M. Kamon, F. Wang, and J. White. Generating nearly optimally compact models from krylov-subspace based reduced-order models. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(4):239–248, April 2000.

[32] K. J. Kerns, I. L. Wemple, and A. T. Yang. Stable and efficient reduction of substrate model networks using congruence transforms. In *IEEE/ACM International Conference on Computer Aided Design*, pages 207 – 214, San Jose, CA, November 1995.

[33] J.-R. Li, F. Wang, and J. White. An efficient Lyapunov equation-based approach for generating reduced-order models of interconnect. In *Proceedings of the 36th Design Automation Conference*, pages 1–6, 1999.

[34] J.-R. Li and J. White. Efficient model reduction of interconnect via approximate system gramians. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 380–383, 1999.

[35] Y. Liu and B. Anderson. Singular perturbation approximation of balanced systems. In *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 1355–1360, 1989.

[36] A. Lu and E. L. Wachspress. Solution of Lyapunov equations by alternating direction implicit iteration. *Comput. Math. Appl.*, 21(9):43–58, 1991.

[37] N. Marques, M. Kamon, J. White, and L. Silveira. A mixed nodal-mesh formulation for efficient extraction and passive reduced-order modeling of 3D interconnects. In *Proceedings of the 35th ACM/IEEE Design Automation Conference*, pages 297–302, San Francisco, CA, June 1998.

[38] L. Miguel Silveira, M. Kamon, I. Elfadel, and J. White. A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 288–294, 1996.

[39] B. C. Moore. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, 26(1):17–32, 1981.

[40] A. Odabasioglu, M. Celik, and L. Pileggi. PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(8):645–654, Aug. 1998.

[41] T. Penzl. Algorithms for model reduction of large dynamical systems. submitted for publication.

[42] T. Penzl. Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case. submitted for publication.

[43] T. Penzl. Numerical solution of generalized Lyapunov equations. *Adv. Comput. Math.*, 8(1-2):33–48, 1998.

[44] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418 (electronic), 1999/00.

[45] L. Pernebo and L. M. Silverman. Model reduction via balanced state space representations. *IEEE Trans. Automat. Control*, 27(2):382–387, 1982.

[46] P. Rabiei and M. Pedram. Model order reduction of large circuits using balanced truncation. In *Proceedings of the Design Automation Conference, Asia and South Pacific*, volume 1, pages 237–240, 1999.

[47] A. Ruhe. The rational Krylov algorithm for nonsymmetric eigenvalue problems. III. Complex shifts for real matrices. *BIT*, 34(1):165–176, 1994.

[48] A. Ruhe and D. Skoogh. Rational Krylov algorithms for eigenvalue computation and model reduction. In *Applied parallel computing (Umeå, 1998)*, pages 491–502. Springer, Berlin, 1998.

[49] M. G. Safonov and R. Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control*, 34(7):729–733, 1989.

[50] E. D. Sontag. *Mathematical Control Theory.* Springer-Verlag, 1998.

[51] G. Starke. Optimal alternating direction implicit parameters for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, 28(5):1431–1445, 1991.

[52] G. Starke. Fejér-Walsh points for rational functions and their use in the ADI iterative method. *J. Comput. Appl. Math.*, 46(1-2):129–141, 1993. Computational complex analysis.

[53] M. S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable nonminimal state-space system. *Internat. J. Control*, 46(4):1319–1330, 1987.

[54] E. I. Verriest and T. Kailath. On generalized balanced realizations. *IEEE Trans. Automat. Control*, 28(8):833–844, 1983.

[55] E. L. Wachspress. Optimum alternating-direction-implicit iteration parameters for a model problem. *J. Soc. Indust. Appl. Math.*, 10:339–350, 1962.

[56] E. L. Wachspress. Solution of the generalized ADI minimax problem. In *Information Processing 68 (Proc. IFIP Congress, Edinburgh, 1968), Vol. 1: Mathematics, Software*, pages 99–105. North-Holland, Amsterdam, 1969.

[57] E. L. Wachspress. ADI iterative solution of Lyapunov equations. In *Iterative methods in linear algebra (Brussels, 1991)*, pages 229–231. North-Holland, Amsterdam, 1992.

[58] E. L. Wachspress. The ADI model problem, 1995.

[59] O. B. Widlund. On the rate of convergence of an alternating direction implicit method in a noncommutative case. *Math. Comp.*, 20:500–515, 1966.

[60] J. H. Wilkinson. *The algebraic eigenvalue problem.* The Clarendon Press Oxford University Press, New York, 1988. Oxford Science Publications.