university of groningen

faculty of science
and engineering

mathematics and applied
mathematics

Approximating the solutions of Lyapunov equations for balanced truncation

Bachelor's Project Mathematics

March 2018

Student: S.E. Gonzalez

First supervisor: Dr.ir. B. Besselink

Second assessor: Dr.ir. F.W. Wubs

Abstract

Model reduction approximates a high-order system by one of lower-order and is often necessary for simulation and control of large-scale systems to be feasible. A popular method of model reduction is balanced truncation, since it preserves stability of the original model and has a computable error bound. However, balanced truncation involves solving Lyapunov equations, which are unfeasible to solve exactly for systems of orders larger than 10^3. In this paper, methods for approximating the solutions, which would make balanced truncation applicable to high-order models, are discussed and compared. The methods considered are the regular and extended Krylov subspace methods of Y. Saad and V. Simoncini, respectively. Theoretical discussions and numerical experiments are used to compare their convergence rates and suitability for balanced truncation. Numerical experiments demonstrate faster convergence for the extended Krylov subspace method than the regular Krylov subspace method. Futhermore, balanced truncation, when using the extended Krylov subspace method, shows increased accuracy in approximating the original model than when using the regular Krylov subspace method. The approximation accuracy shows to be particularly increased for low frequency inputs with the extended rather than the regular Krylov subspace method.

Contents

1 Introduction

Models of both physical phenomena and engineering systems often take the form of a system of differential equations. These systems, called mathematical models, can be used to analyse system behaviour through simulation, for example in predicting the weather. Moreover, such models can be used to design controllers that, when interconnected to the original (engineering) system, will induce desired behaviour. For example, a controller could be used to maintain an artificial satellite on a desired trajectory.

The complexity of a mathematical model is measured in terms of the number of differential equations or difference equations it includes and can be high for practical problems. This can result from the discretization of PDEs or from the inherent complexity of high-tech engineering problems such as the smart grid. Difficulties can arise when the model is too complex. High computational costs and memory limits may make simulating or controlling the system unfeasible. Therefore, it is often necessary to approximate the model with a system of lower order. This approximation is known as model reduction and various methods are described in [1].

One of the most popular techniques for model reduction is balanced truncation, which was first presented by Bruce Moore in [8]. This is because the reduced model closely approximates the input-output behaviour of the original system and particularly because it preserves stability and has a computable error bound.

Balanced truncation consists of two main steps. The first step is to order the states according to their contribution to the input-output behaviour of the system. For this, the controllability and observability Gramians of the system are found which makes it possible to determine which states are the easiest to control and observe. These are the states with the greatest contribution to the input-output behaviour. The second step is to create a new system consisting of only those states with the highest contribution. The number of states to include in the new, lower order system depends on the accuracy needed for the given application.

Carrying out the first step of balanced truncation requires solving two Lyapunov equations whose solutions are the controllability and observability Gramians of the system to be approximated. However, solving these equations is computationally expensive which limits the size of systems that balanced truncation can be used on.

For systems of order of up to 10^3, the corresponding Lyapunov equations can be solved exactly with the Bartels-Stewart algorithm. This is the standard algorithm for solving dense Lyapunov equations and was first presented in [3]. However, for systems of order larger than 10^3 it is not feasible to obtain the full solutions as they are too large to be stored explicitly in memory or would take too long to compute. Therefore, in order to use balanced truncation on large systems, we would have to approximate the solutions to the Lyapunov equations.

This motivates the questions, what kind of numerical strategies could be implemented to approximate the solutions of Lyapunov equations of order 10^4 and larger? Do methods already exist for doing this and, if so, what are they?

The work [13] gives an overview of the currently available methods for solving large Lyapunov equations. The article [11] also gives an overview of some existing methods. Although quite an amount of research has been carried out in the last 15-20 years to attempt to solve this problem, there still is not a universally agreed upon approach for solving high-order Lyapunov equations in the general case.

The main idea of the methods described is to project the Lyapunov equation onto a smaller space in order to obtain an equation which can be solved using standard methods. This results in a low rank approximation of the solution.

The success of this approach depends a great deal on the choice of the projection space. The use of so-called Krylov subspaces was presented by Y. Saad in [11]. A Krylov subspace of order r

is a subspace spanned by the images of an n-dimensional vector b under the first r powers, starting from 0, of an $n \times n$ matrix A. Efficient numerical methods exist for computing the basis of such a Krylov subspace, for example the Arnoldi method. Simoncini and co-authors further extended this idea in [12] by suggesting the use of an extended Krylov subspace, generated as a combination of Krylov subspaces in A and A^{-1}. This method was shown to give an equally accurate approximation of the solution using a significantly smaller dimensional projection space. Another proposed projection space is the so called rational Krylov subspace presented in [4].

The focus of this research paper will be to discuss and compare the effectiveness of using Krylov and extended Krylov subspaces as projection spaces for approximating the solutions of Lyapunov equations. In particular their effectiveness when applied to approximating the Gramians for balanced truncation. After a more detailed description of balanced truncation in Section 2, an outline of the projection methods, the regular and the extended Krylov subspace methods, will be given in Section 3. In Section 4, we apply the methods to an example and study the results. Section 5 concludes the paper.

2 Model Reduction

2.1 Problem Description

The aim of this section is to define the model reduction problem considered in this paper.

This paper focuses on systems which are linear and time-invariant of the form

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

where $u(t) \in \mathbb{R}^m$ represents the input and $y(t) \in \mathbb{R}^p$ represents the output. The state is given by $x(t) \in \mathbb{R}^n$ where n also gives the problem dimension. We consider cases where $m, p \ll n$. Moreover, we assume that the system is controllable and observable and that it is asymptotically stable, i.e., all eigenvalues of A have negative real part. The matrix A is taken to be negative definite. This system will also be represented by the notation $\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right) \in \mathbb{R}^{(n+m)\times(n+p)}$ throughout this thesis.

Problem statement: *The model reduction problem considered is to approximate a system of dimension n given as*

$$\Sigma = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right) \in \mathbb{R}^{(n+m)\times(n+p)}$$

with a system of dimension $k < n$

$$\hat{\Sigma} = \left(\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right) \in \mathbb{R}^{(k+m)\times(k+p)}$$

so that the following properties hold:

1. *The approximation error is small;*
2. *Asymptotic stability is preserved;*
3. *The procedure is computationally stable and efficient.*

2.2 Approximation by Projection

One approach for model reduction is to project the system onto a lower-dimensional subspace. This is equivalent to a truncation in a chosen basis.

Consider the change of basis $\bar{x} = Tx$ where $T \in \mathbb{R}^{n\times n}$ is nonsingular. The transformed system can be represented as:

$$\left(\begin{array}{c|c} TAT^{-1} & TB \\ \hline CT^{-1} & D \end{array}\right). \tag{2}$$

Since T is nonsingular, projecting the system onto the subspace spanned by the columns of T does not change the system, only the basis. Therefore, the transformed system (2) is equivalent to the original system Σ.

The approximated system $\hat{\Sigma}$ of dimension k then corresponds to keeping only the first k equations, i.e., truncating the last $n-k$ equations, of the transformed system. This is done by discarding the last $n-k$ rows of the transformation T and then transforming the original system by the resulting matrix. Using the MATLAB style column notation, if we set

$$T_1 = T_{(1:k,:)}, \quad T_2 = (T^{-1})_{(:,1:k)}$$

then the approximated system of dimension k is given as

$$\hat{\Sigma} = \left(\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array}\right) = \left(\begin{array}{c|c} T_1AT_2 & T_1B \\ \hline CT_2 & D \end{array}\right).$$

For $\hat{\Sigma}$ to be a "good" approximation, the influence of the neglected $n-k$ equations must be "small" in some desired sense. In the next section, an approach for obtaining $\hat{\Sigma}$ is discussed that satisfies these properties.

2.3 Balanced Truncation

One of the most popular techniques for model reduction is balanced truncation, which was first presented by Bruce Moore in [8]. This is because the reduced model closely approximates the input-output behaviour of the original system and particularly because it preserves stability and has a computable error bound.

Model reduction by balanced truncation consists of two main steps:

1. Transforming the system so that the states are ordered by decreasing contribution to the input-output behaviour.
2. Discarding the states with the smallest contribution to the input-output behaviour.

First, we make explicit what is meant by a small or large contribution to the input-output behaviour. For this the controllability and observability functions are needed, $\epsilon_c(x_0)$ and $\epsilon_o(x_0)$, respectively, which are defined as follows:

Definition 2.1 (Controllability function)**.** The controllability function $\epsilon_c(x_0)$ gives the minimum amount of input energy needed to reach the state x_0 from the origin in infinite time. It is given by

$$\epsilon_c(x_0) = \inf_{\substack{u(\cdot)\in\mathcal{L}_2((-\infty,0]) \\ x(-\infty)=0, x(0)=x_0}} \int_{-\infty}^{0} \|u(t)\|^2 dt, \tag{3}$$

where $\mathcal{L}_2((-\infty,0])$ denotes the space of square integrable functions, defined on the domain $(-\infty,0]$.

For controllable linear systems of the form (1), the controllability function can be written as the quadratic form

$$\epsilon_c(x_0) = {x_0}^T P^{-1} x_0,$$

(see [8]), where the matrix P is the so-called controllability Gramian.

Theorem 2.2. *The controllability Gramian is given by*

$$P = \int_0^\infty e^{At} B B^T e^{A^T t} dt$$

and is finite if the system (1) *is asymptotically stable. In this case, the matrix P is the unique solution of the following Lyapunov equation:*

$$AP + PA^T + BB^T = 0. \tag{4}$$

If, additionally, (A,B) is controllable, then P is positive definite.

The proof of Theorem 2.2 can be found in textbooks on Systems Theory, for example in [14].

Definition 2.3 (Observability function)**.** The observability function $\epsilon_o(x_0)$ gives the output energy of the system when released from initial state x_0 with zero input. It is given by

$$\epsilon_o(x_0) = \int_0^\infty \|y(t)\|^2 dt, \qquad x(0) = x_0, u(t) = 0, \forall t \in [0,\infty). \tag{5}$$

Similarly to the controllability function, for linear systems of the form (1), the observability function can be written as the quadratic form

$$\epsilon_0(x_0) = {x_0}^T Q x_0,$$

(see [8]), where the matrix Q is the observability Gramian.

Theorem 2.4. *The observability Gramian is given by*

$$Q = \int_0^\infty e^{A^T t} C^T C e^{At} dt$$

and is finite if the system (1) *is asymptotically stable. In this case, the matrix Q is the unique solution of the following Lyapunov equation:*

$$A^T Q + QA + C^T C = 0. \tag{6}$$

If, additionally, the system is observable, then Q is positive definite.

Theorem 2.4 is the dual result of Theorem 2.2, and its proof can also be found in [14].

The states of (1) which are the easiest to control are those that require the least input energy, $\epsilon_c(x)$, to reach and the states which are the easiest to observe are those that yield the most output energy, $\epsilon_o(x)$. The states which are the easiest to control and to observe are the most important with respect to the input $u(t)$ and the output $y(t)$, respectively, and thus contribute the most to the input-output behaviour of the system.

In order to determine which states give the greatest contribution to the input-output behaviour, it is necessary to find the controllability and observability Gramians by solving the Lyapunov equations (4) and (6), respectively. This is because evaluating the controllability and observability functions directly is not feasible since it involves evaluating infinite integrals so the quadratic forms (3) and (5) are evaluated instead. Under the standing assumptions that the system (1) is asymptotically stable and a minimal realization, both of these equations have unique (positive definite) solutions which makes balanced truncation computationally feasible. For smaller systems of orders up to $\mathcal{O}(10^3)$, well-established methods exist for solving the corresponding Lyapunov equations. One of the most popular methods, due to its efficiency, is the Bartels-Stewart method presented in [3]. However, these methods are computationally costly and for systems of order greater than $\mathcal{O}(10^3)$ solving the corresponding Lyapunov equations exactly is not feasible. Moreover, the solution cannot be stored explicitly in memory. If we are able to find a sufficiently accurate approximation which is both efficient to compute and possible to store in memory, then balanced truncation could also be applied to large systems. Strategies to find such an approximation are the topic of the next chapter.

It is important to note that it is not guaranteed that in the current state space realization those states which are the easiest to control are those which are also easiest to observe and vice-versa. The idea behind balanced truncation is to first transform the system (1) in such a way so that the states which are easiest to control are also the easiest to observe.

The key result in balanced truncation is the observation that there exists a state-space realization such that the Gramians are equal and also diagonal:

$$\bar{P} = \bar{Q} = \Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix}. \tag{7}$$

The diagonal elements are equal to the squared eigenvalues of the product of the Gramians, i.e.,

$$\sigma_i = \sqrt{\lambda_i(PQ)}, \qquad i = 1, ..., n, \tag{8}$$

and are known as the Hankel singular values. In Σ they appear along the diagonal in descending order, i.e. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

In this balanced realization, the controllability and observability functions are given by

$$\epsilon_c(\bar{x}_0) = \bar{x}_0^T \bar{P}^{-1} \bar{x}_0, \;\; \epsilon_o(\bar{x}_0) = \bar{x}_0^T \bar{Q} \bar{x}_0. \tag{9}$$

We can now see that the state $\bar{x}_0 = e_1 = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}^T$ requires the least input energy to reach, since $\epsilon_c(\bar{x}_0) = \sigma_1^{-1}$ and σ_1 is the largest Hankel singular value. Similarly, $\bar{x}_0 = e_1$ yields the most output energy since $\epsilon_o(\bar{x}_0) = \sigma_1$. Therefore, $\bar{x}_0$ is both the easiest to control and to observe and contributes the most to the input-output behaviour. Using the same reasoning for $\bar{x}_0 = e_n = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix}^T$, we see that it is the most difficult to control and to observe, and therefore contributes the least to the input-output behaviour.

The transformation T which gives this balanced realization with transformed matrices:

$$\bar{A} = TAT^{-1},\ \bar{B} = TB,\ \bar{C} = CT^{-1},\ \bar{D} = D \tag{10}$$

and the transformed Gramians

$$\bar{P} = TPT^T,\ \bar{Q} = T^{-T}QT^{-1} \tag{11}$$

is obtained by setting $\bar{P} = \bar{Q} = \Sigma$. Here the Cholesky decomposition $P = UU^T$ is used, and the eigenvalue decomposition of $U^TQU = K\Sigma^2K^T$. The desired transformation T and its inverse can be verified to be

$$T = \Sigma^{1/2}K^TU^{-1},\ T^{-1} = UK\Sigma^{-1/2}. \tag{12}$$

This completes the first step of balanced truncation. The second step is to discard the states with the least influence on the input-output behaviour. First we partition $\bar{x}$ into $\hat{x} = \begin{bmatrix} x_1, ..., x_k \end{bmatrix}^T$ and $\tilde{x} = \begin{bmatrix} x_{k+1}, ..., x_n \end{bmatrix}^T$. With this partitioning, the system is of the form:

$$\dot{\bar{x}} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{pmatrix} \hat{x} \\ \tilde{x} \end{pmatrix} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} u, \tag{13}$$

$$\bar{y} = \begin{bmatrix} \bar{C}_1 & \bar{C}_2 \end{bmatrix} \begin{pmatrix} \hat{x} \\ \tilde{x} \end{pmatrix} + \bar{D}u. \tag{14}$$

We discard the states with the least influence by setting $\tilde{x} = 0$. Letting $\bar{A}_{11} = A_k,\ \bar{B}_1 = B_k,\ \bar{C}_1 = C_k$ and $\bar{D} = D_k$, the resulting reduced-order model of dimension k is

$$\begin{cases} \dot{\hat{x}} = A_k\hat{x} + B_ku, \\ \hat{y} = C_k\hat{x} + D_ku. \end{cases} \tag{15}$$

Relating this to the theory from Section 2.2, if we let $T_1 = T_{(1:k,:)}$ and $T_2 = (T^{-1})_{(:,1:k)}$ then the matrices of the reduced-order model are given by

$$A_k = T_1AT_2,\ B_k = T_1B,\ C_k = CT_2,\ D_k = D.$$

Therefore, we can see that the complete process of balanced truncation is equivalent to projection onto a smaller subspace of dimension k.

The properties of the reduced-order system (15) are given in the following theorem.

Theorem 2.5. *The reduced-order state-space system* (15) *preserves stability of the original model. Furthermore, given the transfer functions $H(s) = C(sI - A)^{-1}B + D$ of the original model and $H_k(s) = C_k(sI_k - A_k)^{-1}B_k + D_k$ of the reduced-order model, an error bound is given by*

$$\|H(s) - H_k(s)\|_\infty \leq 2 \sum_{i=k+1}^{n} \sigma_i. \tag{16}$$

The σ_i's are the discarded Hankel singular values and $\|\cdot\|_\infty$ denotes the $\mathcal{H}_\infty$ norm defined as

$$\|H(s)\|_\infty = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(H(j\omega)), \tag{17}$$

with $\bar{\sigma}(\cdot)$ the largest singular value, where j is the imaginary unit.

The proof of stability preservation is given in [9] and the error bound is derived in [5] and [7].

3 Krylov projection methods

In the previous chapter, the problem of solving Lyapunov equations for finding the controllability and observability Gramians in balanced truncation was encountered. However, we saw that finding the exact Gramians is infeasible for systems of order larger than $\mathcal{O}(10^3)$. In particular, the explicit solutions for large systems are impossible to store in memory. A way to overcome this problem is to try to find low-rank approximations for the Gramians.

The general Lyapunov equation

$$AX + XA^T + bb^T = 0 \tag{18}$$

is considered in this section since both of the Lyapunov equations (4) and (6) are of this form. In this paper we restrict ourselves to cases with only one input and one output for simplicity. Then the matrices B and C^T in (4) and (6) are both single column vectors. Therefore, b in (18) is a column vector and is denoted in lower-case to emphasize this. The theory presented in this section can be extended to cases with multiple inputs and outputs.

Recall that we assumed from Section 2.1 that the system (1) is asymptotically stable and a minimal realization. Theorems 2.2 and 2.4 guarantee that (4) and (6) both have a unique, positive definite solution. Thus we consider only Lyapunov equations of the form (18) in which A is a stability matrix (i.e., all its eigenvalues have negative real part) and (A, b) is controllable so that the solution X of the general form is also guaranteed to be unique and positive definite. We also take the matrix A to be negative definite. We will see later that this is a sufficient condition to ensure the existence of an approximate solution to (19).

If we can find an approximation of the form $\tilde{X} = ZZ^T$ for X, then it is only necessary to store the factor Z. This is feasible for Z of sufficiently low rank. The rest of the section explores methods which give an approximation of this form.

3.1 Regular Krylov subspace method

The method described in this section, which we will refer to as the regular Krylov subspace method, was first presented in a work by Y. Saad, see [11]. The line of reasoning given by Saad to derive the regular Krylov subspace method is outlined in the following text.

Since we assumed A is a stability matrix, the unique solution X of the Lyapunov equation (18) can be written as the integral

$$X = \int_0^\infty e^{At} bb^T e^{A^T t} dt, \tag{19}$$

see [2].

Let $w(t) = e^{At}b$ and note that the solution is of the form

$$X = \int_0^\infty w(t)w(t)^T dt. \tag{20}$$

A way to approximate the solution X is to approximate the exponential function w. This can be done with a polynomial function $w_m(t) = q_m(At)b$ where $q_m(M) = c_0 I + c_1 M + \ldots + c_{m-1}M^{m-1}$ is a matrix polynomial of degree $m - 1$. It is clear that this approximation becomes arbitrarily accurate for m large enough and an appropriate choice of the polynomial q_m.

If we write out w_m explicitly, we have $w_m(t) = c_0 b + c_1 tAb + \ldots + c_{m-1}(tA)^{m-1}b$. Letting $V_m = [b, Ab, ..., A^{m-1}b]$ and $z_m(t) = [c_0, c_1 t, ..., c_{m-1}t^{m-1}]^T$, we can see that $w_m(t) = V_m z_m(t)$. For all t, $w_m(t)$ is an element of

$$\mathcal{K}_m(A, b) = \operatorname{span}\{b, Ab, ..., A^{m-1}b\}, \tag{21}$$

since $\operatorname{im} V_m = \mathcal{K}_m(A, b)$. The space $\mathcal{K}_m$ is known as the Krylov subspace. It is a subspace of all vectors in $\mathbb{R}^n$ that can be written as $x = p(A)b$ where p is a nonzero monic matrix polynomial of degree less than or equal to $m-1$. From the relationship $w_m(t) = V_m z_m(t)$, we have the approximate solution

$$X_m = V_m \Big(\int_0^\infty z_m(t) z_m(t)^T dt \Big) V_m^T. \tag{22}$$

Now we must consider how z_m should be chosen in order for $w_m(t) = V_m z_m(t)$ to accurately approximate $e^{At}b$. To answer this, we first generate an orthonormal basis, denoted by $[v_1, v_2, ..., v_m]$, for the Krylov subspace $\mathcal{K}_m(A,b)$ in order to be able to manipulate the columns of $\mathcal{K}_m(A,b)$ more easily. This can be done using the well-known Arnoldi algorithm (see Algorithm 1).

At each step of the Arnoldi algorithm, the previous Arnoldi vector v_j is multiplied by the matrix A. The resulting vector Av_j is then orthonormalized with respect to all previous v_i's by a standard Gram-Schmidt process.

Algorithm 1 Arnoldi(A, b, m)

```
Given $A$, $b$ and $m$
1. Initialize:
    Compute $v_1 := b/\|b\|_2$.
2. Iterate:
    for $j = 1, 2, ...m$ do
        1. Compute $h_{ij} = (Av_j)^T v_i$ for $i = 1, 2, ..., j$.
        2. Compute $w_j := Av_j - \sum_{i=1}^{j} h_{ij} v_i$.
        3. $h_{j+1,j} = \|w\|_2$.
        4. $v_{j+1} = w/h_{j+1,j}$.
    end for
return $V_m = [v_1, ...v_m]$ and $H_m = (h_{ij})$ for $i, j = 1, ..., m$
```

From now on we define $V_m = [v_1, v_2, ..., v_m]$ and we denote the $m \times m$ matrix consisting of the coefficients h_{ij} computed in Algorithm 1 by H_m. The latter is an upper Hessenberg matrix, i.e., all entries below the first sub-diagonal are zero. From the way H_m is constructed in the algorithm, we have the relation

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T, \tag{23}$$

where e_m is the m-th column of the identity matrix I_m. Due to the orthonormality of the vectors $v_1, ..., v_m, v_{m+1}$, the relationship $H_m = {V_m}^T A V_m$ follows. The matrix H_m represents the projection of the linear transformation A onto the Krylov subspace $\mathcal{K}_m(A,b)$.

To approximate $e^{At}b$ we want to minimize the norm $\|e^{At}b - V_m z_m(t)\|_2$ for all $t \geq 0$, where $\|\cdot\|_2$ is the standard Euclidean norm evaluated point-wise in time. It is easily seen that this is accomplished by $z_m(t) = V_m^T e^{At} b$. With v_1 as defined in Algorithm 1 and letting $\beta = \|b\|_2$, we have $b = \beta v_1 = \beta V_m e_1$, where e_1 is the first column of I_m. This gives us $z_m(t) = \beta V_m^T e^{At} V_m e_1$.

The exponential e^{At} is unknown but since $H_m = {V_m}^T A V_m$ holds, $e^{H_m t}$ is a natural choice for approximating $V_m^T e^{At} V_m$. Therefore, for arbitrary t, we have

$$e^{At} b \approx \beta V_m e^{tH_m} e_1. \tag{24}$$

The following theorem from [6] gives an error bound for the approximation error with this choice of an approximation.

Theorem 3.1. *Let A be any square matrix and let $\rho = \|A\|_2$. Then the error of the approximation (24) for arbitrary $t \geq 0$ is such that*

$$\|e^{tA} b - \beta V_m e^{tH_m} e_1\|_2 \leq 2\beta \frac{(t\rho)^m e^{t\rho}}{m!}. \tag{25}$$

Results of experiments reported in [6] show that the approximation (24) can be very accurate even for moderate values of m. This was also confirmed in the numerical experiments which are presented in Section 4.

If we substitute the expression (24) in (19) we obtain the approximate solution

$$X_m = V_m\Big(\int_0^\infty e^{\tau H_m}(\beta e_1)(\beta e_1)^T e^{\tau H_m^T} d\tau\Big)V_m^T. \tag{26}$$

Defining $G_m = \int_0^\infty e^{\tau H_m}(\beta e_1)(\beta e_1)^T e^{\tau H_m^T} d\tau$, we see that the approximate solution is of the form

$$X_m = V_m G_m V_m^T.$$

Since we assumed that A is negative definite, it can be seen from the relationship $H_m = {V_m}^T A V_m$ that H_m is a stability matrix. Therefore, G_m is the unique, positive semi-definite solution of the $m \times m$ Lyapunov equation

$$H_m G_m + G_m H_m^T + \beta^2 e_1 e_1^T = 0. \tag{27}$$

The negative definitiveness of A is a sufficient condition for the existence of the solution G_m of (27).

The outcome of this section is that we have succeeded in reducing the dimension of the problem to m, which can be solved using standard methods for smaller-scale Lyapunov problems. Furthermore, since G_m is positive semi-definite, G_m has a Cholesky decomposition and can be written as $G_m = LL^T$ where L is a lower triangular matrix.

Let $Z_m = V_m L$. Then, the approximate solution of the Lyapunov equation (18) can be written as

$$X_m = (V_m L)(L^T V_m^T) = Z_m Z_m^T, \tag{28}$$

where only the slim factor Z_m needs to be stored in memory. This solves the problem of finding a computationally feasible method for finding an approximation of the solution X which is possible to store in memory for low-rank Z_m.

The regular Krylov subspace method can also be thought of as a projection of the Lyapunov equation (18) onto the Krylov subspace $\mathcal{K}_m(A, b)$. As in our first approach, we seek an approximate solution to the equation in our approximation subspace, $\mathcal{K}_m(A, b)$, of the form $X_m = V_m G_m V_m^T$. Note, the matrix V_m that we use is the same orthonormal matrix we generated using the Arnoldi algorithm. The projected problem is obtained by imposing the Galerkin condition that the residual $R = AX_m + X_m A^T + bb^T$ be orthogonal to the approximation space, i.e., that $V_m^T R V_m = 0$. Since V_m has orthonormal columns and we have $b = \beta v_1$, we obtain the following projected Lyapunov equation

$$\begin{aligned} 0 &= V_m^T A V_m G_m + G_m V_m^T A^T V_m + V_m^T bb^T V_m, \\ &= H_m G_m + G_m H_m^T + \beta^2 e_1 e_1^T. \end{aligned}$$

This equation is the same as (27) because of the relationship $H_m = V_m^T A V_m$. Therefore, this approach is equivalent to our first approach and yields the same approximation $X_m = Z_m Z_m^T$.

A summary of the regular Krylov subspace method is given below, in Algorithm 2. With each iteration, the algorithm increases the dimension m of the Krylov subspace that the Lyapunov equation (18) is projected onto. Each iteration produces an approximation, X_m, of the solution. It continues until the norm of the residual with X_m is smaller than a pre-chosen error tolerance. The output of the algorithm is the low-rank factor Z_m.

Note that in this version of the algorithm, the norm of the residual $\|AX_m + X_m A^T + bb^T\|_2$ is calculated at each iteration. This requires matrix computations involving matrices of dimension $n \times n$, where n is the dimension of the Lyapunov equation (18). This is computationally costly,

Algorithm 2 Regular Krylov subspace method $(A, b, errtol)$

```
1. Given $A$, $b$ and chosen error tolerance $errtol$:
2. Set $\beta = \|b\|_2$.
3. for $m = 1, 2, ...$ do
4.    $[V_m, H_m]$=arnoldi$(A, b, m)$.
5.    Set $c = \beta e_1$.
6.    Solve $H_m G_m + G_m H_m^T + cc^T = 0$.
7.    Use the Cholesky composition to find $G_m = L_m L_m^T$.
8.    Set $Z_m = V_m L_m$, the Cholesky factor of the approximation $X_m = Z_m Z_m^T$.
9.    if $\|AX_m + X_m A^T + bb^T\|_2 < errtol$ then
10.      return $Z_m$
11.   end if
12. end for
```

therefore for large n it is recommended to only evaluate the residual norm every five or ten iterations.

We should also point out that in practice it is not necessary to compute approximate solutions X_m for all m. This is done in Algorithm 2 in order to study the convergence behavior of the regular Krylov subspace method. However, for a given application one could start the iterative method starting from a moderate value of m, or only compute approximate solutions at increasing steps of m until one is satisfied with the approximation error.

3.2 Extended Krylov subspace method

Results of numerical experiments, which will be discussed in Section 4, show that the regular Krylov subspace method converges for moderate values of m. We explore in this section if the convergence rate could still be made faster.

Consider again the exponential function $w(t) = e^{At}b$. Since we assumed that A is symmetric, it is well known from linear algebra that it is diagonalizable. Therefore, we can rewrite $w(t)$ as a linear combination of the eigenvectors of A:

$$w(t) = \sum_{i=1}^{n} a_i e^{\lambda_i t} v_i, \tag{29}$$

where for $i = 1, .., n$, $a_i \in \mathbb{R}$ are constants and (λ_i, v_i) are the eigenpairs of A. Since we assumed that A is a stability matrix, all eigenvalues have negative real part. They are ordered here as $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| > 0$. Throughout the rest of this text we will use the notation $\lambda_{max} = \lambda_1$ and $\lambda_{min} = \lambda_n$, where the words maximum and minimum are considered in the absolute sense. The corresponding eigenvectors of A will be referred to as v_{max} and v_{min}, respectively.

As t goes to infinity, we can see from (29) that the most important component of $w(t)$ at time t will be in the direction of v_{min}. This is because $e^{\lambda_i t}$ converges to zero the slowest for λ_{min}. Therefore, the columns of

$$X = \int_0^\infty e^{At} bb^T e^{A^T t} dt \tag{30}$$

will also all have a large component in the direction of v_{min}.

The approximation we found for the solution X in Section 3.1 was of the form

$$X_m = V_m G_m V_m^T \tag{31}$$

where the columns of V_m form a basis for the Krylov subspace $\mathcal{K}_m(A, b) = \text{span}\{b, Ab, ..., A^{m-1}b\}$. From the well-known power method from Numerical Mathematics (a description of the method can be found in [10]), we know that $A^k b$ converges in the direction of v_{max} as $k \to \infty$. However, as we saw above, the columns of solution X have a significant component in the direction of v_{min}. This leads us to consider whether X would be more accurately approximated if this component were better represented in X_m.

In [12], V. Simoncini introduces the idea of extending the Krylov projection space to include information on both A and A^{-1}, that is to the subspace

$$\mathcal{K}_{2m}(A, A^{-m}b) = \text{span}\{b, A^{-1}b, Ab, ..., A^{-(m-1)}b, A^{m-1}b, A^{-m}b\} \tag{32}$$

which we will from now on refer to as the extended Krylov subspace. Since the eigenvalues of A^{-1} are equal to $\lambda_{max}^{-1}, ..., \lambda_{min}^{-1}$, $A^{-k}b$ converges in the direction of the eigenvector corresponding to λ_{min}^{-1}, i.e. to v_{min}. Let $\widehat{V}_m$ be an orthonormal matrix such that its columns span $\mathcal{K}_{2m}(A, A^{-m}b)$. If our approximation X_m is of the form

$$X_m = \widehat{V}_m Y_m \widehat{V}_m^T, \tag{33}$$

instead of the form given in (31), the component of the columns of X in the direction of v_{min} will be better represented and we expect that the approximation will converge faster. We will see in the results section that experiments indeed show faster convergence.

Note that for each iteration, two basis vectors are added to the approximation subspace, whereas in the regular Krylov subspace method only one vector is added per iteration. The two vectors added are first orthogonalized with respect to the existing basis, and then with respect to each other which can be done using the Gram-Schmidt method. An outline of the extended Krylov subspace method is given in Algorithm 3.

Algorithm 3 Extended Krylov subspace method $(A, b, errtol)$

```
Given $A$, $b$ and chosen error tolerance $errtol$, set $V_1 = gramSchmidt([b, A^{-1}b]), \mathcal{V}_0 = \emptyset$:
for $m = 1, 2, ...$ do
    $\mathcal{V}_m = [\mathcal{V}_{m-1}, V_m]$
    Set $\mathcal{T}_m = \mathcal{V}_m^T A \mathcal{V}_m$ and $E = \mathcal{V}_m^T b$
    Solve $\mathcal{T}_m Y + Y \mathcal{T}_m^T + EE^T = 0$ and set $Y_m = Y$
    If converged then $X_m = \mathcal{V}_m Y_m \mathcal{V}_m^T$ and stop
    Set $V_m^{(1)}$: first column of $V_m$; $V_m^{(2)}$: second column of $V_m$
    $V'_{m+1} = [AV_m^{(1)}, A^{-1}V_m^{(2)}]$
    $\hat{V}_{m+1} \leftarrow$ orthogonalize $V'_{m+1}$ w.r.t. $\mathcal{V}_m$
    $V_{m+1} = gramSchmidt(\hat{V}_{m+1})$
end for
```

3.3 Balanced truncation with approximated Gramians

Now that we are able to approximate the Gramians, we want to use these approximations in balanced truncation. To present how to do this we first consider again how balanced truncation is performed when the exact Gramians are available.

Recall from Section 2.3 that there exists a transformation under which the Gramians of the transformed system are equal and diagonal, i.e.,

$$\bar{P} = \bar{Q} = \Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n \end{bmatrix},$$

where the diagonal entries are the Hankel singular values. Balanced truncation consists of two main steps. The first step is to calculate the transformation T which gives the balanced realization. In order to do this, the Gramians, P and Q, are computed. Next the Cholesky decomposition $P = UU^T$ and the eigenvalue decomposition $U^TQU = K\Sigma^2K^T$ are found and the desired transformation matrix is calculated using

$$T = \Sigma^{1/2}K^TU^{-1}, \; T^{-1} = UK\Sigma^{-1/2}.$$

After transformation, the second step is to truncate the transformed system, giving the reduced-order model.

However, even when using the exact Gramians, the process outlined above can be numerically "unsafe". This is because the transformation matrix T is often highly ill-conditioned. This problem can be solved by using an alternative, but theoretically equivalent, algorithm which avoids formulas involving matrix inverses, for example the *square root algorithm*.

The *square root algorithm* uses the Cholesky decomposition of $Q = LL^T$, with L a lower triangular matrix, and the singular value decomposition of the product of the Cholesky factors U^T and L:

$$U^T L = W \Sigma V^T \tag{34}$$

which produces the orthogonal matrices W and V. Note that the Hankel singular values are the singular values of the product of U^T and L. In this algorithm the transformation

$$T = \Sigma^{-1/2} V^T L^T, \; T^{-1} = UW\Sigma^{-1/2},$$

is used. Using the relationship (34), it is easily verified that the transformed Gramians are given by $TPT^T = \Sigma = T^{-T} Q T^{-1}$. Therefore, under this transformation the system

$$\left(\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array}\right)$$

is also balanced.

We can partition the matrices obtained from the singular value decomposition of $U^T L$ in (34) as follows:

$$W = \begin{bmatrix} W_1 & W_2 \end{bmatrix}, \; \Sigma = \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix}, \; V = \begin{bmatrix} V_1 & V_2 \end{bmatrix},$$

where W_1, V_1 have k columns and $\Sigma_1 \in \mathbb{R}^{k\times k}$. Truncating the last $n-k$ columns of T as in Section 2.2 gives

$$T_1 = \Sigma_1^{-1/2} V_1^T L^T \in \mathbb{R}^{k\times n}, \; T_2 = UW_1\Sigma_1^{-1/2} \in \mathbb{R}^{n\times k}. \tag{35}$$

Applying this transformation balances and truncates the original system in a single step and yields the reduced-order system of dimension k

$$\hat{\Sigma} = \left(\begin{array}{c|c} T_1AT_2 & T_1B \\ \hline CT_2 & D \end{array}\right).$$

When the precise Cholesky factors of the Gramians are not available, we can apply the same method using low rank approximated Cholesky factors instead. For example, we can use the low rank factors obtained by the Krylov subspace methods presented in the previous sections. We then call this the *low rank square root method* (LRSRM). An outline of the algorithm is given below.

Algorithm 4 Low rank square root method (LRSRM) (A, B, C, D, k)

```
Compute low rank factors $U_m$ and $L_m$ by Algorithm 2 or 3 (or alternative low rank methods), such that $U_m U_m^T$ and $L_m L_m^T$ are approximations of the controllability and observability Gramians, respectively.
$[W, S, V]$=svd$(U_m^T L_m)$, $W_1 = W_{(:,1:k)}$, $S_1 = S_{(1:k,1:k)}$, $V_1 = V_{(:,1:k)}$.
$T_1 = S_1^{-1/2} V_1^T L_m^T$, $T_2 = U_m W_1 S_1^{-1/2}$.
$Ak = T_1 A T_2$, $B_k = T_1 B$, $C_k = C T_2$, $D_k = D$.
return $A_k, B_k, C_k, D_k$
```

When using the Krylov subspace methods to approximate the Cholesky factors in LRSRM, we have two parameters that can be varied. These are k, the order of the reduced-order model, and m, the dimension of the (extended) Krylov subspace. If we fix k, we expect that for the same value of m, the error of the reduced-order model will be smaller when the factors are approximated with the extended Krylov subspace method rather than with the regular Krylov subspace method. We expect this since numerical experiments show that the extended Krylov subspace method converges faster than the regular Krylov subspace method.

The Krylov subspace is constructed with a sequence of basis vectors $A^k b$ which converges in the direction of v_{max}. The extended Krylov subspace contains, in addition to this, a sequence $A^{-k}b$ which converges in the direction of v_{min}. Therefore, the extended Krylov subspace method approximates the components of the Gramians in the direction of v_{min}, the eigenvector corresponding to "slowest" eigenvalue, more accurately than the regular Krylov subpace method. Therefore, we are inclined to think that using LRSRM with the extended Krylov subspace method may give reduced-order models that approximate the behaviour for "slow", i.e., low frequency, inputs better than with the regular Krylov subspace method. Similarly, since the regular Krylov subspace contains a longer sequence $A^k b$ than the extended Krylov subspace with the same dimension, LRSRM with the regular Krylov subspace method may more accurately approximate the "fast", i.e., high frequency, behaviour of the model.

4 Numerical experiments

In this section the results of numerical experiments are reported in which the methods presented in Sections 3.1 and 3.2 are tested and compared. All reported experiments were preformed using Matlab 9.3, on a PC with a 3.10Ghz processor, and 8GBytes of RAM. The example used to illustrate the results of using the methods is described in the following section.

4.1 Heat equation (continuous case)

The example used for the numerical experiments is the heat diffusion equation for the one-dimensional case given as

$$\begin{cases} \text{PDE} & \frac{\delta}{\delta t}T(x,t) = \alpha\frac{\delta^2}{\delta x^2}T(x,t) + u(x,t) & x \in (0,1);\ t > 0, \\ \text{BCs} & T(0,t) = 0 = T(1,t) & t > 0, \\ \text{IC} & T(x,0) = 0 & x \in (0,1). \end{cases}$$

In this equation $T(x,t)$ represents the temperature at point x on a thin rod at time t. The heat source is given by $u(x,t) = u(t)\delta_k(x)$. Note that the rod is only heated at a single point on the rod, at position $x = k$.

The solution is given by

$$T(x,t) = x(x-1) + \sum_{i=0}^{\infty} \frac{8}{(2i+1)^3\pi^3}\sin((2i+1)\pi x)e^{-(2i+1)^2\pi^2\alpha t} + \Big(\int_0^t u(s)ds\Big)\delta_k(x). \tag{36}$$

Evaluating the temperature exactly for arbitrary point x on the rod at time t would be infeasible since it would involve the computation of an infinite sum. However, if we discretize the spatial domain $[0,1]$ uniformly with $n+2$ points, i.e., if we cover the spatial domain with the mesh grid $[x_0, x_1, \dots, x_{n+1}]$, we can approximate the exact solution $T(x_i,t)$ at any point x_i, $1 \le i \le n$, of the grid. The grid points divide the spatial domain into $n+1$ equal segments of length $h = \frac{1}{n+1}$. To find an approximation $X_i(t)$ at arbitrary time $t > 0$ of the exact solution $T(x_i,t)$, we use the approximations

$$\frac{\delta}{\delta t}T(x_i,t) \approx \frac{\delta}{\delta t}X_i(t) \quad \text{and} \quad \frac{\delta^2}{\delta x^2}T(x_i,t) \approx \frac{X_{i+1}(t) - 2X_i(t) + X_{i+1}(t)}{h^2},$$

with the fact that $X_0 = X_{n+1} = 0$ to obtain a linear system of equations of the form

$$\dot{X}(t) = AX(t) + b, \quad X(0) = 0,$$

where

$$A = -\frac{\alpha}{h^2}\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{n\times n} \text{ and } b = \begin{bmatrix} u(x_1,t) \\ u(x_2,t) \\ \dots \\ u(x_n,t) \end{bmatrix} \in \mathbb{R}^n.$$

The solution $X(t) = [X_1(t), X_2(t), \dots, X_n(t)]$ gives the approximations of the temperature at each of the points x_i, $1 \le i \le n$, of the mesh grid at time t. The constant α is the thermal diffusivity of the rod.

Furthermore, if we want to measure the average temperature of the points in the mesh grid and apply the heat source on the leftmost point of the mesh grid on the rod, i.e., at point $x = x_1$, we obtain the system

$$\begin{cases} \dot{X}(t) = AX(t) + Bu(t), & X(0) = 0, \\ Y(t) = CX(t), \end{cases} \tag{37}$$

where

$$B = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \end{bmatrix}^T \in \mathbb{R}^n \text{ and } C = \frac{1}{n}\begin{bmatrix} 1 & 1 & \dots & 1 & 1 \end{bmatrix} \in \mathbb{R}^n.$$

We use this system (37) in the experiments and take $n = 200$. We use a system of relatively small order so that the approximated solutions of the corresponding Lyapunov equations can be compared to the exact solutions. If can be verified that the matrix A is negative definite, which is consistent with the assumptions we made since the beginning of this thesis.

4.2 Comparison of methods in approximating the Gramians

The controllability and observability Gramians of system (37) are the solutions P and Q, respectively, of the following Lyapunov equations:

$$\begin{aligned} (i) \quad & AP + PA^T + BB^T = 0, \\ (ii) \quad & A^TQ + QA + C^TC = 0, \end{aligned}$$

where the matrices A, B and C are defined as in Section 4.1. The goal of this section is to compare the convergence of the approximated Gramians when using the regular Krylov subspace method versus the extended Krylov subspace method.

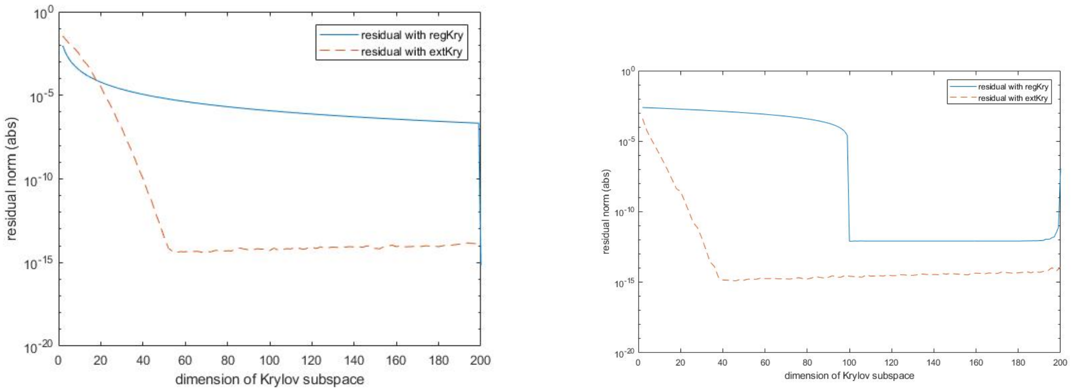
In the first experiment we approximate the controllability Gramian, P, using first the regular Krylov subspace method and then the extended Krylov subspace method. In both cases we measured the scaled Frobenius norm of the residual at each iteration, i.e., the quantity $\|AP_m + P_mA^T + BB^T\|_F$, where $\|Z\|_F = (tr[Z^TZ]/n)^{1/2}$, P_m is the approximation of P after m iterations and $n \times n$ is the dimension of matrix A. Note that since we are considering an example of "small" dimension, it is feasible to compute the residual at each step. However, as discussed in Section 3.1, this would be too computationally costly for a "medium" or "large" example. In these cases, only checking the residual at intervals of five or ten iterations would be recommended. We chose the error tolerance `tol`$= 10^{-6}$ and recorded the number of iterations needed before the norm of the residual was found to be less than `tol`. We also recorded the dimension of the Krylov projection subspace and the rank of the approximate solution at the point that the stopping criterion was met. We repeated the same steps to analyze the convergence of the observability Gramian. The results are shown in Table 1. For brevity, we use the short hand "regKrylov" to refer to the regular Krylov subspace method and "extKrylov" to refer to the extended Krylov subspace method in Table 1 and the figures shown throughout this chapter.

Gramian		reg. Krylov	ext. Krylov
controllability, P	no. iterations	108	14
	dim. Krylov space	108	28
	Solution rank	24	25
		reg. Krylov	ext. Krylov
observability, Q	no. iterations	100	6
	dim. Krylov space	100	12
	Solution rank	17	12

Table 1: Performance of the Krylov subspace methods

Recall that for each iteration of the extended Krylov subspace method, two new basis vectors are added to the subspace. On the other hand, only one new basis vector is added per iteration for the regular Krylov method. This is why we see in Table 1 that when using the extended Krylov subspace method, the dimension of the Krylov subspace is double the number of iterations when the stopping criterion is reached.

It is clear from the table that the subspace dimension required to obtain a residual norm below the error tolerance is significantly smaller when employing the extended rather than the regular Krylov subspace method. This is consistent with the predictions made in Chapter 3 based on the theory. The difference in speed of convergence is also seen clearly in Figure 1 where the residual norm is plotted against the dimension of the subspace for both methods.



(a) Convergence of controllability Gramian

(b) Convergence of observabiltiy Gramian

Figure 1: Residual norm versus dimension of approximation subspace

4.3 Comparison of methods in Balanced Truncation

Now that we have compared the effectiveness of the regular and extended Krylov subspace methods in approximating the Gramians, we want to analyze the effect of using the approximations in balanced truncation. In this section we use the *Low Rank Square Root Method*, LRSRM, described in Section 3.3 to approximate system (37) with reduced-order models.

The first step of LRSRM is to find the Cholesky factors of both the controllability and observability Gramians. In the first experiment we discuss in this section, we fixed k, the order of the reduced model, and ran LRSRM three times. In **Run 1**, we used the MATLAB function `lyapchol` to obtain the Cholesky factors. This function uses standard methods for solving Lyapunov equations. While this function cannot give the exact solutions, the model is small so we can take the Cholesky factors obtained with `lyapchol` as accurate to machine precision. In **Run 2** we approximated the factors using the regular Krylov subspace method and in **Run 3** we approximated the factors using the extended Krylov subspace method. Each run produced a reduced-order model approximation (of order k) of the original model. We will discuss and compare the accuracy of the three approximations. Any differences between the accuracy of the approximations are caused solely by the effect of using different methods to find the Cholesky factors, since all other steps of LRSRM were kept identical for each run.

To compare how accurately the reduced order models approximate the original model we compare the frequency response of the original system with those of the reduced systems in Figure 3. The figures display the function

$$\|\Delta H(jw)\| = \|H(jw) - H_k(jw)\| \tag{38}$$

for the frequency range $\omega \in [10^{-3}, 10^{5}]$ where H is the transfer model of the original system and H_k is the transfer model of the reduced-order model. Since heat diffusion is slow, we are most interested in the behaviour of the system when applying low frequency inputs, i.e., inputs with frequencies from the lower half of the chosen frequency range.

We show the frequency response of the original system in Figure 2 for reference. From the figure we see that for the chosen frequency range, the magnitude of the frequency response ranges between about 10^{-3} and 10^{-7}. It is logical that the magnitude of the frequency response is "small" because the thin rod is only heated at its leftmost end. Therefore, the temperature at the rightmost end is hardly effected by the heat source, and the average temperature of the rod is not greatly changed. Also, since heat diffusion is slow, we can see that the magnitude of the frequency response becomes smaller for higher frequency inputs. In order to have a "good" approximation of this model, the error should be at least four or five orders smaller on the entire range than the magnitude of the model's frequency response.

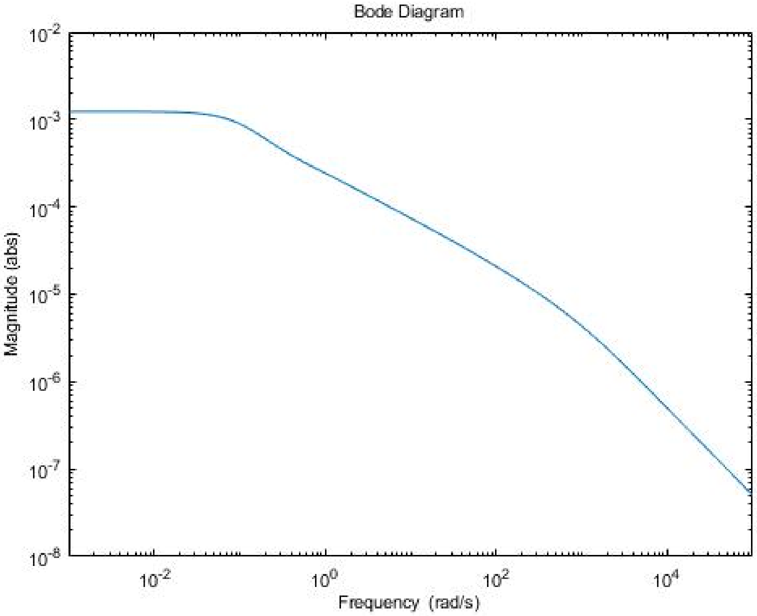


Figure 2: Frequency response of original system

For this experiment we chose to fix $k = 16$. We also chose to use an approximation subspace of dimension $m = 16$ when approximating the Cholesky factors using the Krylov subspace methods. The approximation errors from the three runs are shown in the Figure 3.

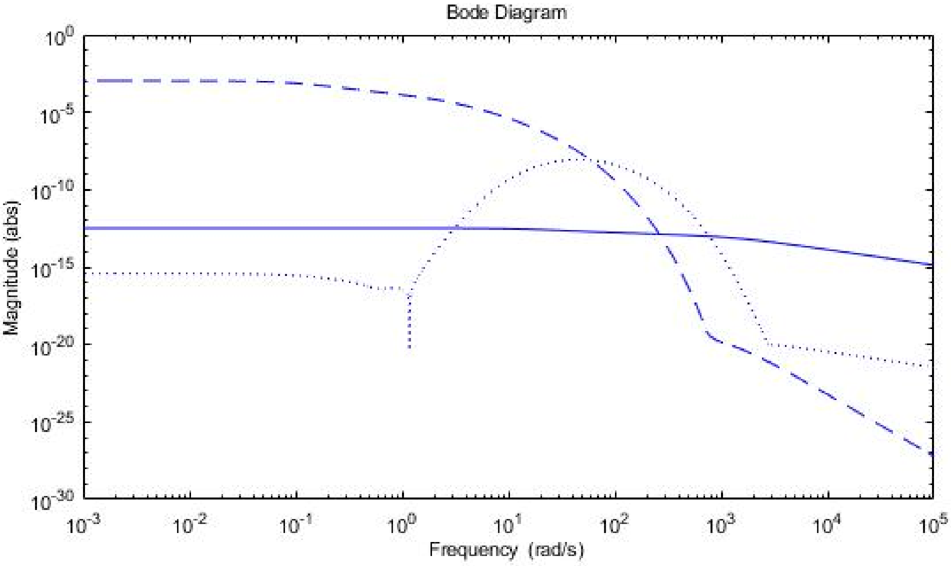


Figure 3: Error plots for Run 1 (solid line), Run 2 (dashed line) and Run 3 (dotted line).

First we see that the error of the approximation obtained in **Run 1**, shown by the solid line, is at least six orders smaller over the whole frequency range than the magnitude of the frequency response of the original model. Therefore, this gives us a good reference to compare the effectiveness of using the Krylov subspace methods against.

We see in Figure 3 that the error plots resulting from **Run 2** and **Run 3** are very different from each other. From the theory presented in Chapter 3 and the results shown in the previous section, we saw that the approximated Gramians converged more quickly to the true Gramians when using the extended rather than the regular Krylov subspace method. Therefore, we expect a more accurate reduced-order model from **Run 3** than from **Run 2**. We see from the approximation errors that our prediction is true for the frequencies from the lower half of our frequency range, particularly in the interval $[10^{-3}, 10^{1}]$. However, as the frequency of the input increases, the error of the model from **Run 2** decreases and after around $\omega = 10^2$ rad/s it is consistently smaller than the error from the model of **Run 3**. This could be explained by the prediction we made in Section 3.3, that LRSRM with the regular Krylov subspace method approximates the high frequency behaviour of the original model better than with the extended Krylov subspace method.

The error plot from **Run 3** in the interval $[10^0, 10^4]$ is especially notable since the approximation error suddenly begins to rise when entering the interval and peaks at around $\omega = 10^2$ rad/s before decreasing again. The approximation error at the peak is in the order of 10^8, which is only three orders smaller than the magnitude of the frequency response of the original system for the same frequency. Therefore, the approximation is unsatisfactory in the frequency range around $\omega = 10^2$ rad/s. It is logical to think that due to the structure of the extended Krylov subspace, the components of the Gramians in the directions of the eigenvectors corresponding to the intermediate eigenvalues are not as well approximated as the components in the directions of the eigenvectors of both of the extreme eigenvalues. This could explain why the reduced-order model obtained from **Run 3** does not approximate the behaviour of the model for moderate frequency inputs as accurately as for low or high frequency inputs. To improve the approximation for "moderate" frequency inputs, it may be necessary to increase the dimension m of the extended Krylov subspace.

It is surprising that the approximation error of the model obtained by **Run 3**, where the Cholesky factors of the Gramians are approximated, is smaller for the lower and upper halves of the the frequency range than for the model where `lyapchol` is used. We are not entirely certain why this occurs, however, a possible explanation could be that since the extended Krylov subspace method approximates particularly accurately the components of the Gramians in the directions of the eigenvectors corresponding to both the extreme eigenvalues, the behavior of the model for "low" and "high" frequency inputs is captured more accurately.

The extended Krylov subspace method gives a consistently small approximation error over the lower half of the frequency range, i.e. in the interval $[10^{-3}, 10^0]$ and considering we are more interested in the low frequency behaviour, it could be a suitable option for use in our application. However, if one is interested in approximating the behaviour of the model for inputs of frequency greater than $\omega = 10^0$rad/s, using the extended Krylov subspace method with $m = 16$ does not give a sufficiently accurate approximation. However, we will see from the results of the following experiment that increasing the dimension m of the extended Krylov subspace solves this problem.

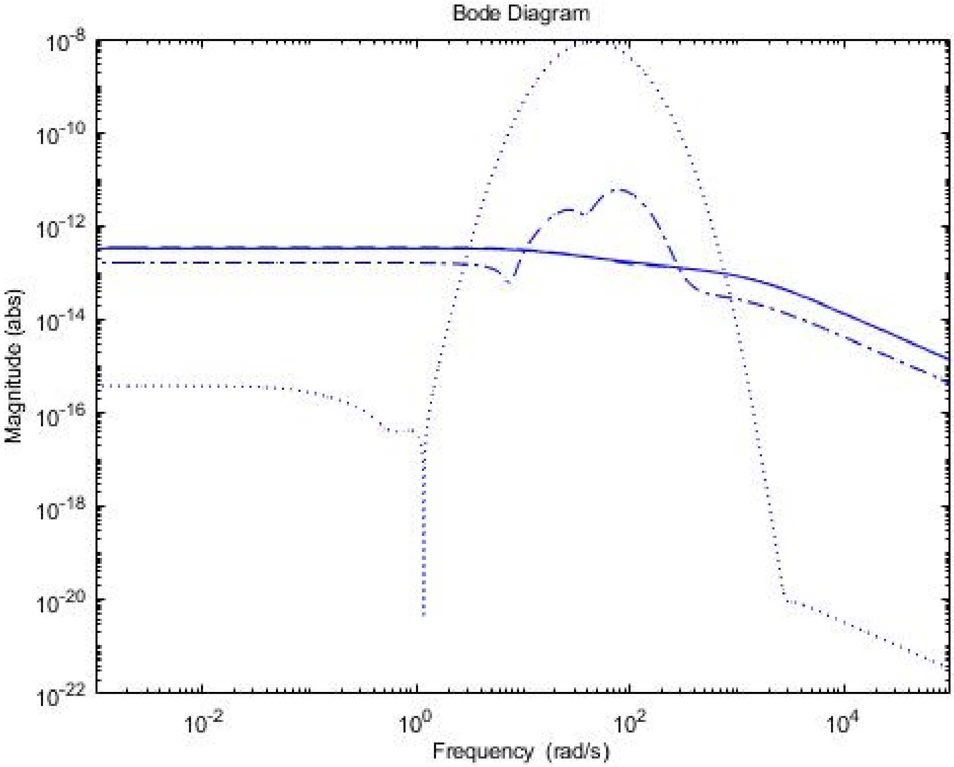


Figure 4: Error plots of the reduced-order model with $k = 16$ and $m = 16$ (dotted-line), $m = 24$ (dot-dashed line), $m = 32$ (dashed-line) and when using lyapchol (solid-line).

For this second experiment, we again fixed $k = 16$ but this time we approximated the Cholesky factors using the extended Krylov subspace method with increasing values of m. Then for each value of m, we used the corresponding approximations of the Cholesky factors in LRSRM. In Figure 4 we compare the effect of using different values of m on the accuracy of the reduced-order model. For comparison, we also plot the approximation error when using `lyapchol` to calculate the Cholesky factors in LRSRM.

We see in the figure, that increasing the value of m reduces the approximation error for the moderate frequency inputs. The approximation error over the entire range "evens out" and converges to the approximation error when using `lyapchol`. For dimensions $m = 32$ and higher, the error when using the extended Krylov subspace method becomes almost indistinguishable from the error when `lyapchol` is used.

This result is promising because it shows that increasing the dimension of the extended Krylov subspace sufficiently gives a satisfactory approximation of the original model when used in LRSRM. It also shows that using a good approximation for the Cholesky factors could give just as accurate a reduced-order model as using the exact factors. Therefore, balanced transformation (using the LRSRM algorithm) could still be applied to reduce models when the exact Gramians are not available.

5 Conclusion

We have presented and compared two methods for approximating the solutions to large Lyapunov equations. The two methods presented worked by projecting the large Lyapunov equation onto a smaller subspace, the Krylov subspace in the regular Krylov subspace method and the extended Krylov subspace in the extended Krylov subspace method. The latter showed faster convergence to the true solution than the former. The methods were used to approximate the Gramians (and their respective Cholesky factors) for model reduction by balanced truncation.

We also presented the *the Low Rank Square Root method* (LRSRM). This method is more numerically stable than balanced truncation and uses approximations of the Cholesky factors of the Gramians rather than the exact factors. LRSRM would be theoretically equivalent to balanced truncation if the exact Gramians were used. When the factors were approximated with the extended Krylov subspace method, LRSRM proved to give reduced-order models that were more accurate for low frequency inputs than when the factors were approximated with the regular Krylov subspace method. However, LRSRM with the extended Krylov subspace method gave reduced-order models that were less accurate for moderate frequency inputs than for high and low frequency inputs. For low dimensions of the extended Krylov subspace, the accuracy for moderate frequency inputs was unsatisfactory. However, the accuracy was shown to improve when the dimension of the extended Krylov subspace was increased. For dimensions high enough, which were still moderate in the numerical experiment shown, the approximation error was almost indistinguishable from the approximation error when the exact Gramians were used.

Overall, it seemed promising that balanced truncation could still be used to reduce large systems even when the exact Gramians are not available. In our example, the reduced-order model obtained using Gramians approximated with the extended Krylov subspace method, proved to be just as accurate as when using the exact Gramians for moderate dimensions of the extended Krylov subspace.

References

[1] Athanasios C. Antoulas. *Approximation of Large-Scale Dynamical Systems.* Society for Industrial and Applied Mathematics, January 2005.

[2] P. J. Antsaklis and A. N. Michel. *A Linear Systems Primer.* Birkhäuser Boston, 2007.

[3] Richard H. Bartels and George W. Stewart. Solution of the matrix equation AX+ XB= C [f4]. *Communications of the ACM*, 15(9):820–826, 1972.

[4] Vladimir Druskin and Valeria Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Systems & Control Letters*, 60(8):546–560, 2011.

[5] Dale F. Enns. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In *Decision and Control, 1984. The 23rd IEEE Conference on*, volume 23, pages 127–132. IEEE, 1984.

[6] Efstratios Gallopoulos and Yousef Saad. On the parallel solution of parabolic equations. In *Proceedings of the 3rd international conference on Supercomputing*, pages 17–28. ACM, 1989.

[7] Keith Glover. All optimal hankel-norm approximations of linear multivariable systems and their l,∞-error bounds. *International journal of control*, 39(6):1115–1193, 1984.

[8] Bruce Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.

[9] Lars Pernebo and Leonard Silverman. Model reduction via balanced state space representations. *IEEE Transactions on Automatic Control*, 27(2):382–387, 1982.

[10] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics.* Texts in applied mathematics. Springer, 2000.

[11] Youcef Saad. Numerical solution of large Lyapunov equations. 1989.

[12] Valeria Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM Journal on Scientific Computing*, 29(3):1268–1288, 2007.

[13] Valeria Simoncini. Computational methods for linear matrix equations. *SIAM Review*, 58(3):377–441, 2016.

[14] H. L. Trentelman, A. A. Stoorvogel, and M. Hautus. *Control Theory for Linear Systems.* Communications and Control Engineering. Springer London, 2012.