# Some Preconditioners in Optimization with Partial Differential Equations

**Dissertation**

zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)

Dem Fachbereich IV der Universität Trier
vorgelegt von

**Xuancan Ye**

Trier, 2013

Gutachter: Prof. Dr. Ekkehard Sachs
Dr. Andy Wathen

# Contents

# German Summary (Zusammenfassung)

Krylov Unterraum Methoden werden oft verwendet, um hochdimensionale lineare Gleichungen, wie sie bei Optimierungsproblemen mit partiellen Differentialgleichungen (PDE) entstehen, zu lösen. Um hierzu effiziente iterative Löser zu entwickeln, ist eine passende Präkonditionierung unerlässlich.

Die vorliegende Dissertation setzt sich aus zwei Teilen zusammen. Im ersten Teil werden zwei verschiedene Präkonditionierer für einen Konjugierten Gradienten (CG) Löser, welcher auf eine spezielle partielle Integro Differentialgleichung (PIDE) aus dem Finanzbereich angewendet wird, sowohl theoretisch als auch numerisch verglichen. Hierbei werden ebenfalls die Gitterunabhängigkeit und Konvergenzrate analysiert. Das dabei erlangte Wissen über die Präkonditionierung der PIDE wird im Anschluss daran auf ein verwandtes Optimierungsproblem angewendet. Der zweite Teil zielt darauf ab, mittels Einbindung von Modellen reduzierter Ordnung in sogenannte Deflated Krylov Unterraum Methoden eine neue Präkonditionierungstechnik zu entwickeln und diese auf die zugehörigen Optimierungsprobleme anzuwenden. Die dabei verwendeten Modelle reduzierter Ordnung stammen von nichtlinearen PDEs und werden mittels Proper Orthogonal Decomposition (POD) erzeugt. Dem schließen sich numerische Ergebnisse für eine Serie von Testproblemen an.

# English Summary

Krylov subspace methods are often used to solve large-scale linear equations arising from optimization problems involving partial differential equations (PDEs). Appropriate preconditioning is vital for designing efficient iterative solvers of this type.

This research consists of two parts. In the first part, we compare two different kinds of preconditioners for a conjugate gradient (CG) solver attacking one partial integro-differential equation (PIDE) in finance, both theoretically and numerically. An analysis on mesh independence and rate of convergence of the CG solver is included. The knowledge of preconditioning the PIDE is applied to a relevant optimization problem. The second part aims at developing a new preconditioning technique by embedding reduced order models of nonlinear PDEs, which are generated by proper orthogonal decomposition (POD), into deflated Krylov subspace algorithms in solving corresponding optimization problems. Numerical results are reported for a series of test problems.

# Acknowledgements

First and foremost, I would like to thank Prof. Ekkehard Sachs, my supervisor, for his constant support and generous assistance. Without his invaluable advice and patient guidance this thesis would not have been possible. I am sincerely grateful to Prof. Andy Wathen for making a special trip to Trier to serve as examiner.

I wish to express my gratitude to Prof. Bingsheng He for encouraging me to start my doctoral program in Germany. I would like to acknowledge the financial support from China Scholarship Council.

Moreover, I am obliged to many of my colleagues at the Department of Mathematics for giving me helping hands during my stay and providing the great atmosphere. In particular, I want to thank Dr. Timo Hylla, Dr. Christoph Käbe, Dr. Andre Lörx, Dr. Matthias Schu, Bastian Groß, Matthias Wagner, Marina Schneider in the research group of Prof. Sachs as well as my colleagues Dr. Christian Wagner, Martin Siebenborn, Roland Stoffel, Ulf Friedrich and Benjamin Rosenbaum.

I am deeply indebted to my beloved wife, Binbin, for faithfully staying beside me and keep me grounded.

Xuancan Ye
Trier, 2013

# Chapter 1.

# Introduction

A *PDE-constrained optimization problem* is an optimization problem with a partial differential equation (PDE) as its constraint. In some cases there are more restrictions on the optimization variable involved. This kind of problem could arise from various scientific and engineering applications such as optimal control, optimal design, parameter identification. Some references about examples of PDE-constrained optimization problems arising in different areas are listed in the introduction section of Benzi et al. (2011).

PDE-constrained optimization problems can be described mathematically by the general form

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & e(x) = 0 \\ & g(x) \leq 0, \end{aligned} \tag{1.1}$$

where the equality constraint $e(x) = 0$ involves a PDE and $g(x) \leq 0$ gives further restrictions on the *optimization variable* $x$ like its upper bound and lower bound. The optimization variable $x$ lies in an infinite dimensional space and has a particular feature that it can be partitioned into $x = (y, u) \in Y \times U$. Here the notation $y$ is commonly used for *state variable* and $u$ *control variable* in the context that $y$ can be determined by the *state equation* $e(y, u) = 0$ for any given $u$. $Y$ and $U$ are their domains respectively. Note that $u$ could be named differently according to the background of the model problem.

Solving PDE-constrained optimization problems is a challenging task. It has received much attention in the past few decades. Introductions to this field can be found in Biegler et al. (2003), Tröltzsch (2010), Hinze et al. (2009).

## 1.1. Preconditioning in PDE-Constrained Optimization

One way to classify the numerical methods for solving PDE-constrained optimization problems is based on how the PDE constraint is treated. It leads to so-called *black-box methods* and *all-at-once methods*, see e.g. Herzog (2010) and Herzog and Kunisch (2010).

In black-box methods, the PDE constraint in (1.1) is eliminated by a control-to-state map $y = \mathcal{S}(u)$ . Notice that $\mathcal{S} : U \longrightarrow Y$ is not necessary to be given explicitly. Instead, one can obtain $y$ by solving the PDE $e(y, u) = 0$ for any given $u \in U$, which makes the PDE constraint a 'black box' with $u$ input and $y$ output. Then the objective functional of the optimization problem (1.1) becomes

$$\hat{f}(u) := f\left(\mathcal{S}(u), u\right),$$

taking only $u$ as optimization variable. In the case that the inequality constraint $g(x) \leq 0$ is absent, such an approach changes a constrained optimization problem into an unconstrained one.

In contrast, an all-at-once method keeps the PDE as a side constraint and minimizes $f(y, u)$ with respect to both state and control variables, namely, $y$ and $u$. An all-at-once method typically forms the (augmented) Lagrangian of the optimization problem and derives the first order necessary conditions, i.e. Karush-Kuhn-Tucker (KKT) system (see e.g. Nocedal and Wright (2006)).

No matter which type of methods is adopted, solving PDE-constrained optimization problems numerically normally involves the solution of large-scale linear problems.

For instance, a black-box method may require repeated solution of the state equation for different given controls. Sometimes one also needs to solve the *adjoint equation*, which has similar form as the PDE constraint, for obtaining gradient information efficiently, see e.g. Tröltzsch (2010). The discretization of either state equation or adjoint equation gives rise to linear equations of large size.

In all-at-once methods, solving the KKT system leads to structured linear equations, more precisely, *saddle point problems*, which have the form

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}. \tag{1.2}$$

Here the coefficient matrix is usually symmetric and the blocks in it are of course problem dependent.

For solving the resulting linear problems, iterative solvers (see e.g. Saad (2003)) are most preferable, due to the large scale of the problems and sparsity of the coefficient matrices. Sparsity is actually important for an iterative method, because the major computational effort of iterative methods lies on matrix-vector multiplications and sparsity allows efficient calculation of matrix-vector products.

The efficiency of an iterative solver for solving a linear equation

$$\mathbf{Ax} = \mathbf{b}$$

is generally determined by the spectral properties of the coefficient matrix $\mathbf{A}$, such as its eigenvalue distribution and condition number. Unfortunately, the linear equations that occur to us in PDE-constrained optimization problems are commonly ill-conditioned. The condition number of a corresponding coefficient matrix can be very large so that directly applying an iterative solver is not realistic at all. To resolve this problem, combining a *preconditioner* $\mathbf{P}$ is vital for developing a practical iterative solver. In fact, using a good preconditioner could dramatically improve the rate of convergence of the iterative solver.

The preconditioner $\mathbf{P}$ works by transforming the linear equation $\mathbf{Ax} = \mathbf{b}$ into

$$\mathbf{P}^{-1}\mathbf{Ax} = \mathbf{P}^{-1}\mathbf{b},$$

which has the same solution as the original equation but the preconditioned matrix $\mathbf{P}^{-1}\mathbf{A}$ has more favorable spectral properties. This process is known as *preconditioning.*

How to choose a proper preconditioner depends on the specific linear problem we need to solve. In the context of PDE-constrained optimization problems, the numerical method we choose has influence on the preconditioning strategy in the first place.

As solving PDEs is one major part of black-box methods, one may use preconditioners which approximate the coefficient matrices, e.g. an incomplete LU preconditioner (see Golub and Van Loan (1996)), thus the preconditioned matrix is close to identity matrix.

Preconditioning saddle point problems arising from all-at-once methods often exploits the block structure of the coefficient matrices, consequently many block structured preconditioners are designed. Such preconditioners normally have very nice clustering effect on the eigenvalues of the saddle point matrices in theory, however, applying block preconditioners usually requires more considerations to achieve efficient implementation. In fact one may only be able to apply block preconditioners approximately in practice. This will be discussed in Section 4.3.2 and some relevant references are given there.

In this work, we deal with two equality constrained optimization problems and raise discussions on preconditioning issues for them. We use Krylov subspace methods as iterative solvers for different situations in the rest of this thesis. Notice that we confine ourselves to the discretize-then-optimize approach, which provides a discrete counterpart of the PDE-constrained optimization problem and applies an optimization method to the discrete problem. For discussions on discretize-then-optimize and optimize-then-discretize approaches, we refer to Hinze et al. (2009).

We remark that the inequality constraint in (1.1) can be handled by primal-dual active set (PDAS) strategy (see Bergounioux et al. (1999) for instance). If the PDAS strategy is combined in the framework of an all-at-once method, applying a block preconditioner for the resulting linear problem could be complicated. For such an application, we refer to Herzog and Sachs (2010).

## 1.2. Outline

This section is devoted to an overview of the contents of this thesis.

### Preliminaries: Krylov Subspace Methods and Preconditioning

In Chapter 2, knowledge about Krylov subspace methods are reviewed for later use.

We present some estimates to characterize the convergence behavior of the conjugate gradient (CG) method. This gives a clear view of how spectral properties affect the rate of convergence of the CG method. The discussion is extended to general Hilbert space.

We also collect some block preconditioners for saddle point problems. In particular some Bramble-Pasciak like preconditioners, which make the preconditioned matrices symmetric positive definite with respect to certain nonstandard inner products thus, the CG method is still applicable to certain indefinite linear systems, are also included.

Deflated (augmented) Krylov subspace methods are briefly introduced. They are Krylov subspace methods carried out with extra constraints on the residuals and searching directions throughout the iteration.

## Efficient Solver with Proper Preconditioners for a PIDE

Chapter 3 focuses on preconditioning issues in solving a partial-integral differential equation (PIDE) with preconditioned CG method. It continues the discussion on the efficient solver proposed in Sachs and Strauss (2008). The knowledge about preconditioning for such a problem is also useful to a relevant optimization problem.

We look for $y(t,x) \in C^{1,2}((0,T] \times \mathbb{R}) \cap C^0([0,T] \times \mathbb{R})$ satisfying

$$\begin{aligned} y_t - \frac{1}{2}\sigma^2 y_{xx} + (r+\lambda)y - \lambda \int_{-\infty}^{\infty} y(t,z) f(z-x)\,\mathrm{d}z = 0, \quad \text{on} \quad (0,T] \times \mathbb{R}, \\ y(0,x) = H(e^x), \quad \forall x \in \mathbb{R}, \end{aligned} \tag{1.3}$$

where $f(x)$ is a normal density function, $H(x) = \max\{0, x-K\}$ and $K$, $\sigma$, $\lambda$, $T$ are positive constants, $r \geq 0$.

In Sachs and Strauss (2008), the PIDE (1.3) is carefully discretized and solving (1.3) becomes solving a sequence of linear equations

$$T_n x^p = b^p. \tag{1.4}$$

Here $n$ denotes the number of spatial nodes and $p$ is the index of time steps. The coefficient matrix $T_n$ is dense. However, it has a special structure called *Toeplitz* (see Appendix A). Toeplitz structure is very beneficial because matrix-vector multiplication involving a Toeplitz matrix can be performed very efficiently using fast Fourier transform (FFT). Moreover, $T_n$ is proved to be symmetric positive definite, hence the CG method is applicable. Sachs and Strauss suggest to use Strang's preconditioner, which is a circulant matrix (see Section 3.1.3 and Appendix A), in the CG solver and mesh independent convergence is observed in their numerical experiments.

Obviously, Toeplitz structure of the coefficient matrix is the key feature for designing such an efficient solver, otherwise the dense matrix resulting from the integral term in (1.3) would be a pain for numerical solution. Other than the finite difference scheme used in Sachs and Strauss (2008), we prove that finite element method (FEM) can still maintain the Toeplitz structure under certain assumptions.

One major work in this thesis is the theoretical analysis for the approach proposed by Sachs and Strauss (2008). We provide an estimate of the condition number of $T_n$, which has the order $n^2/m$ with $m$ the total number of time steps. The estimate confirms the ill condition of the linear equation (1.4) theoretically. Eigenvalue distribution of $T_n$ is also characterized by the major part of $T_n$.

According to the numerical results in Sachs and Strauss (2008), Strang's preconditioner outperforms other circulant preconditioners in this case, hence we raise analysis on the preconditioned matrix only for Strang preconditioned system. We prove that for sufficiently large $m$, the Strang preconditioned matrix has all eigenvalues clustered in a small interval around 1 except for two outliers, which can be used to explain the mesh independent convergence of Sachs and Strauss's approach. The condition number of Strang preconditioned matrix is reduced to the order of $n/\sqrt{m}$, which is of course a significant improvement compared to $n^2/m$.

Noticing that the coefficient matrix is dominated by a tridiagonal matrix resulting from the elliptic operator in (1.3) and the dense part coming from the integral term is relatively small in scale, we propose to use the tridiagonal part as the preconditioner. The corresponding preconditioned matrix has all its eigenvalues clustered around 1 and the condition number is close to 1. Mesh independent superlinear convergence of the CG solver is also proved.

If the PIDE (1.3) is discretized by FEM, then the mesh independent superlinear convergence of the CG method combining a tridiagonal preconditioner can be obtained easily by following the analysis in Karátson (2005). It leads to an estimate only relying on the operators in the PIDE, thus the rate of convergence of the corresponding CG solver is independent of finite elements we choose for discretization.

Results of numerical experiments support our analytical statements. Eigenvalue distributions and condition numbers of Strang and tridiagonal preconditioned matrices are compared in figures. In addition, we illustrate a few intermediate estimates which are important for our main results. We carry out the CG solver with the two preconditioners for various discretization settings of the PIDE. Mesh dependent convergence is observed in both cases. By comparing the performance of the two preconditioners, we see that the CG solver using the tridiagonal preconditioner requires less iterations. The numerical effort for each CG iteration requires $O(n)$ operations, which is slightly cheaper than $O(n \log n)$ in Strang preconditioned CG solver. This is confirmed by the numerical results concerning CPU time.

According to the spectral analysis and the numerical results, we conclude that preconditioning the elliptic part is more beneficial than preconditioning the whole system in this case. This viewpoint is helpful when we deal with a relevant calibration problem, which is an optimization problem governed by a very similar PIDE, presented by (1.5).

$$
\begin{aligned}
\min_u \quad & J(u) := \frac{1}{2} \sum_{i \in \mathcal{I}} |D(u; x_i, T_i) - d(x_i, T_i)|^2 \\
\text{s.t.} \quad & D_T(x,T) - \frac{\sigma^2(x,T)}{2} D_{xx}(x,T) + \left( r(T) + \frac{\sigma^2(x,T)}{2} - \lambda\zeta \right) D_x(x,T) \\
& \quad + \lambda(1+\zeta) D(x,T) - \lambda \int_{-\infty}^{+\infty} D(x-y,T) e^y \, f(y) dy = 0 \\
& \text{where} \quad D(x,0) = \max\{S_0 - S_0 e^x, 0\} =: D_0(x).
\end{aligned}
\tag{1.5}
$$

Here $x \in (-\infty, \infty)$, $T \in (0, T_{max})$, $u$ is defined to be the parameter set $(\sigma, \lambda, f)$.

The PIDE (Dupire's equation, see Dupire (1994)) in (1.5) describes a mathematical model for European option pricing in finance. The calibration problem aims at correcting the model to fit a given data set with respect to the parameter set $u$ as much as possible.

A multi-level TRPOD algorithm is developed in Schu and Sachs (2010) for solving (1.5), which embeds reduced order models calculated by proper orthogonal decomposition (POD) in a trust region (TR) framework. POD extracts key information from known solution to the PIDE, such information can be used to project the discrete PIDE into a low dimensional subspace to build a reduced order model. Solving the reduced order model gives an approximate solution of the original PIDE very cheaply. The trust region strategy serves as an indicator of the effectiveness of the reduced order model, as well as a global strategy.

The multi-level TRPOD algorithm can be identified as a black-box method. Regardless of the realization of the algorithm, solving (1.5) following the path of Schu and Sachs (2010) requires repeated solution of the PIDE constraint, as being discussed in Section 1.1.

Different from the previous PIDE (1.3), discretization of the Dupire's equation in (1.5) leads to linear equations with nonsymmetric coefficient matrices. Hence the common CG method is not applicable anymore. One can use a generalized minimal residual (GMRES) method as the iterative solver, which is also a Krylov subspace method and its convergence behavior is very similar to the CG method.

A tridiagonal preconditioner resulting from the elliptic operator is used in (1.5), thus effective solution of the PIDE is achieved. An estimate is derived for characterizing the mesh independent convergence of the GMRES solver. There are also some numerical results to compare the numerical effort of the unpreconditioned method and preconditioned method, using the codes developed for Schu and Sachs (2010).

### A Preconditioning Strategy Using POD for Nonlinear Problems

POD is commonly used for accelerating the solving of time dependent PDEs or relevant optimization problems, generally by building reduced order models, see e.g. Kunisch and Volkwein (1999), Markovinović and Jansen (2006) and Schu and Sachs (2010). In this research, we are using POD in a very different way.

The aim of Chapter 4 is developing a POD-involved preconditioning strategy in solving nonlinear PDE-constrained optimization problems with an all-at-once method. We embed the POD basis (the vectors extracted by POD) directly in the main iteration loop rather than use it to build reduced order models first. The embedding is realized by using deflated Krylov subspace methods (see Section 2.3).

Our work is closely related to that in the recent paper by Simoncini (2012). As a result, We first review the approach in Simoncini (2012), then address the difference of our method.

Solving PDE-constrained optimization problems with all-at-once methods usually leads to saddle point problems like (1.2). In some cases, a part of the unknowns can be easily eliminated by simple calculation, which reduce the dimension of the linear equation to be solved. The resulting problem is still a saddle point problem, which can be solved by Krylov subspace methods combined with a block preconditioner. Since block preconditioners are normally applied approximately, the spectral properties of the preconditioned matrices are usually not as good as expected. In Olshanskii and Simoncini (2010), it is even proved that when a block preconditioner is used for a saddle point problem, there remains a cluster of eigenvalues around 0 in the preconditioned matrix under certain assumptions. This results in stagnation phases in the convergence history of a Krylov subspace method.

As a remedy, it is proposed in Simoncini (2012) to use approximate eigenvectors corresponding to the small eigenvalues (in magnitude) of the preconditioned matrix to overcome the stagnation. The idea is carried out by restricting the Krylov subspace solver in the subspace $\mathbf{A}$-orthogonal to $W$. Here $\mathbf{A}$ denotes the coefficient matrix of the saddle point problem and $W$ consists of the approximate eigenvectors. The restriction of the Krylov subspace solver is realized by a deflated Krylov subspace algorithm. The matrix consisting of the approximate eigenvectors is called *deflation matrix*.

The approach is tested on a simplified Monge-Kantorovich mass transfer problem which is a nonlinear PDE-constrained optimization problem and is first handled by an inexact Gauss-Newton approach, see Benzi et al. (2011). The required approximate eigenvectors are evaluated once by the Lanczos algorithm and used for all the Newton-type iterations. Notice that obtaining the approximate eigenvectors are not cheap. Nevertheless, it is still worthwhile since the deflation using these eigenvectors remove the stagnation phases thus significantly reduce CPU time in each Newton-type iteration.

What we are trying to do is use the POD basis to generate the vectors for deflation, which is of course not a common way to utilize this reduced-order-model technique.

The first problem would be whether we can obtain desired eigenvectors via POD. In order to show this, we design a series of numerical experiments for a very simple one-dimensional heat equation, which is

$$\begin{cases} u_t - u_{xx} = 0, & \forall\ (x,t) \in \Omega \times (0,T] \\ u(x,t) = 0, & x \text{ on } \partial\Omega \\ u(x,0) = u_0(x), & \end{cases} \tag{1.6}$$

where $\Omega = (0,1)$. Different initial conditions (i.e $u_0(x)$) are used in these numerical experiments.

We discretize (1.6) with FEM and implicit Euler scheme in space and time respectively, then solve the sequence of linear equations

$$(M + \tau K)u_j = Mu_{j-1}, \quad j = 1, 2, \dots, m, \tag{1.7}$$

to get numerical solutions (snapshots). Here $M$ and $K$ denote mass matrix and stiffness matrix respectively. $\tau = T/m$ is time difference. In the next step, we apply POD to the snapshots to generate a set of vectors (POD basis) and examine how well these vectors can approximate the eigenvectors of the coefficient matrix $M + \tau K$ which are related to the small eigenvalues. The results are in fact encouraging.

In practice, the POD basis is calculated by the singular value decomposition (SVD) or solving an equivalent eigenvalue problem (EVP), see B.2. Thus the numerical effort of POD for large problems is considerable. Taking care of this, we try to use different subsets of the snapshots for POD, which reduces the dimension of the corresponding EVP. It turns out that we only need some properly selected snapshots to obtain satisfying vectors. This raises the possibility of reducing the cost of POD itself.

We remark that an extra strategy for choosing appropriate subsets of snapshots would be necessary in a practical problem. For simplicity, we use all the snapshots in the numerical experiments.

If all the unknown time instances in (1.7) are listed in one long vector, i.e. let $\bar{u} = (u_1^T, u_2^T, \dots, u_m^T)^T$, then we are able to solve (1.7) by solving a large linear equation at once instead of solving the system of linear equations sequentially. In such a sense, it leads to so-called one-shot method, see e.g. Stoll (2011).

As being pointed out earlier, the POD basis extracted is used to construct deflation matrices for deflated Krylov subspace methods. We can easily construct a deflation matrix $W$ for the linear equations (1.7) with POD basis by letting $W$ having its columns to be the vectors obtained by POD. In a one-shot method, we embed the above defined $W$ in the diagonal of the deflation matrix, this is illustrated in Section 4.1.2. Numerical results confirm that such a matrix still catches the desired eigenvalues of the large coefficient matrix.

In order to verify the deflation matrix constructed for one-shot method in an optimization problem, we take a linear quadratic problem from Stoll and Wathen (2010) for test, which is

$$
\begin{aligned}
\min \quad & \frac{1}{2}\int_0^T \int_\Omega (\mathbf{y}(x,t) - \bar{\mathbf{y}}(x,t))^2 \,\mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega (\mathbf{u}(x,t))^2 \,\mathrm{d}x\mathrm{d}t \\
\text{s.t.} \quad & \mathbf{y}_t - \Delta \mathbf{y} = \mathbf{u} \qquad \text{in } \Omega, \\
& \mathbf{y} = 0 \qquad \text{on } \partial\Omega, \\
& \mathbf{y}(x,0) = \mathbf{y}_0(x).
\end{aligned} \tag{1.8}
$$

The problem (1.8) is solved by an all-at-once method via a discretize-then-optimize approach in Stoll and Wathen (2010), it comes to the KKT system having the form

$$
\begin{pmatrix} \tau A & 0 & B^T \\ 0 & \beta\tau A & -\tau C^T \\ B & -\tau C & 0 \end{pmatrix} \begin{pmatrix} y \\ u \\ p \end{pmatrix} = \begin{pmatrix} \tau A\bar{y} \\ 0 \\ d \end{pmatrix}. \tag{1.9}
$$

Note that the time instances are treated with a one-shot scheme.

Like in Simoncini (2012), we first remove the control variable from (1.9) to derive an equivalent linear system (1.10).

$$
\begin{cases}
\begin{pmatrix} \tau A & B^T \\ B & -\frac{\tau}{\beta} C A^{-1} C^T \end{pmatrix} \begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau A\bar{y} \\ d \end{pmatrix}, \\
u = \frac{1}{\beta} A^{-1} C^T p.
\end{cases} \tag{1.10}
$$

The state variable and adjoint variable left are in fact solution to the state equation and adjoint equation respectively, hence they can be well approximated in the POD subspace.

We construct a deflation matrix for the coefficient matrix in (1.10) and solve (1.10) by deflated MINRES algorithm using deflation matrices of different dimension. The results show that the deflation matrix we build from POD basis is indeed effective. We also change the regularization parameter $\beta$ of (1.8) in the experiments, which influences the effect of deflation according to the numerical results.

In a real problem, it is necessary to use a preconditioner, we test our deflation strategy in a deflated MINRES solver combining a block-diagonal preconditioner on the following nonlinear problem.

$$
\begin{aligned}
\min \quad & \frac{1}{2}\int_0^T \int_\Omega (\mathbf{y}(x,t) - \bar{\mathbf{y}}(x,t))^2 \,\mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_{\partial\Omega} (\mathbf{u}(x,t))^2 \,\mathrm{d}x\mathrm{d}t =: \mathbf{J}(\mathbf{y},\mathbf{u}) \\
\text{s.t.} \quad & \mathbf{y}_t - \Delta \mathbf{y} - \mathbf{f}(\mathbf{y}) = 0 \qquad \text{in } \Omega, \\
& \mathbf{y} = \mathbf{u} \qquad \text{on } \partial\Omega, \\
& \mathbf{y}(x,0) = \mathbf{y}_0(x).
\end{aligned} \tag{1.11}
$$

Here $\mathbf{f}(\mathbf{y})$ is a nonlinear functional and the control lies on the boundary.

The problem is solved by a Lagrangian-Newton method, which is Newton's method applied to the Lagrangian of (1.11), it is equivalent to sequential quadratic programming (SQP), see Herzog (2010). As a result, we need to solve a sequence of saddle point problems similar to (1.9). We downsize each of them and, as we do in the linear quadratic problem, the resulting linear equations are solved by a deflated MINRES algorithm using a block diagonal preconditioner. The results show that the deflation strategy works along with the preconditioner we choose.

This part of work is still in progress, we give some concluding remarks and outlooks in the end of Chapter 4.

All numerical results in this thesis are produced by MATLAB codes on a desktop PC with Intel® Core™2 Duo (3.00GHz) processor and 4GB RAM.

# Chapter 2.

# Krylov Subspace Methods and Preconditioning

Iterative methods are usually adopted for solving large dimensional linear equations when direct methods, e.g. Gaussian elimination, become prohibitively expensive or even not possible. Krylov subspace methods are no doubt among the most successful and popular iterative methods currently.

Consider a linear equation $Ax = b$ in $\mathbb{R}^n$, where $A$ is an $n \times n$ nonsingular matrix. For an arbitrary given initial guess $x_0$, a Krylov subspace of degree $k$ is defined by

$$\mathcal{K}(A, r_0; k) = \operatorname{span}\{r_0, Ar_0, A^2 r_0, \ldots, A^{k-1} r_0\},$$

with $r_0 = b - Ax_0$. A Krylov subspace method generates an approximation of the solution in the linear manifold $x_0 + \mathcal{K}(A, r_0; k)$ at the $k$-th iteration. In a practical algorithm, the sequence of Krylov subspaces $\mathcal{K}(A, r_0; k)$ ( $k = 1, 2, \ldots$) is generated by Arnoldi's procedure, which becomes the Lanczos algorithm when $A$ is symmetric. Details on how to derive a variety of Krylov subspace methods can be found in Saad (2003).

In each iteration of a Krylov subspace method, the major computational effort comes from the evaluation of matrix-vector product involving the coefficient matrix $A$, which is required for increasing the dimension of current Krylov subspace to update the search direction. Hence Krylov subspace methods are preferable when matrix-vector multiplication is cheap. This is naturally true when $A$ is sparse. Moreover, the matrix-vector multiplication can still be efficient when $A$ has a certain special structure, even if $A$ is a dense matrix. We will see such an application in Chapter 3.

Regarding symmetry and positive definiteness of $A$, different kinds of Krylov subspace methods are developed. Some commonly used ones are listed as follows:

- conjugate gradient method (CG), if $A$ is symmetric and positive definite;
- minimal residual method (MINRES), if $A$ is symmetric and indefinite;
- generalized minimal residual method (GMRES) and biconjugate gradient stabilized method (BiCGStab), if $A$ is nonsymmetric.

We refer to the review article Simoncini and Szyld (2007) for more methods of this class and some recent developments.

In exact arithmetic, a Krylov subspace method converges in no more than $n$ iterations for a linear equation of dimension $n$ if it does not break. In practice, Krylov subspace methods usually converge much faster to meet chosen stopping criteria. Notice that the $k$th iterate can be expressed by $x_0 + P_{k-1}(A)r_0$, where $P_{k-1}$ is a polynomial of degree not exceeding $k - 1$. The convergence behavior can be analyzed using such an expression, see e.g. Saad

(2003).

According to theoretical analysis, Krylov subspace methods have a characteristic in common, that is, the convergence behavior heavily rely on the spectrum of the coefficient matrix $A$. This gives rise to a large area of research, namely, preconditioning, which aims to transform the linear equation into an equivalent problem with more favorable spectral properties for accelerating convergence.

As a prototypical Krylov subspace method, the CG method is well studied and its convergence is well understood. In this chapter, we mainly focus on the convergence and preconditioning aspects of the CG method. Such knowledge can also be used to guide the preconditioning of other Krylov subspace methods.

In the rest of this chapter, we first review some useful estimates for characterizing convergence behavior of the CG method and discuss the spirit of preconditioning. Different perspectives of seeing preconditioning and extended discussion in general Hilbert space are also briefly covered. Then we introduce a few preconditioning techniques which are commonly used in the process of solving saddle point problems arising from PDE-constrained optimization. These techniques exploit the algebraic structure of corresponding linear systems. In the last section of this chapter, we introduce deflation strategy for Krylov subspace methods, which can also be treated as a preconditioning technique, as it is used for improving the spectrum of target problem.

## 2.1. Conjugate Gradient Method and Preconditioning

### 2.1.1. Standard Conjugate Gradient Method and Rate of Convergence

The CG method is a remarkable iterative approach for solving the linear equation

$$Ax = b, \tag{2.1}$$

where $A$ is an $n \times n$ symmetric positive definite matrix. We present the classical CG method developed in Hestenes and Stiefel (1952) by Algorithm 2.1. This algorithm is attractive since it requires cheap computational effort in each iteration and fairly low storage. It is strongly connected to the Lanczos algorithm, in the sense it can be derived from the Lanczos algorithm and vice verse, see e.g. Golub and Van Loan (1996).

For all $k > 0$, the search directions $p_k$ and residual vectors $r_k$ generated by Algorithm 2.1 satisfy the following conjugacy and orthogonality conditions respectively:

$$p_k^T A p_i = 0, \quad \text{for} \quad i = 0, 1, \ldots, k-1, \tag{2.2}$$

$$r_k^T r_i = 0, \quad \text{for} \quad i = 0, 1, \ldots, k-1. \tag{2.3}$$

In fact, Algorithm 2.1 applied to $Ax = b$ is equivalent to minimizing the quadratic function $\frac{1}{2}x^T Ax - b^T x$ along the sequence of search directions $p_k$ consecutively. Based on this fact one can derive so-called finite termination property of the CG method, i.e. the sequence $\{x_k\}$ generated by conjugate gradient method converges to the solution $x^*$ in at most $n$ steps. Such a discussion can be found in e.g. Nocedal and Wright (2006). This is not satisfying when $n$ is very large, which commonly appears if the linear equation is obtained from a problem

**Algorithm 2.1** The CG method

```
1: Given $x_0$;
2: Set $r_0 \leftarrow b - Ax_0$, $p_0 = r_0$, $k \leftarrow 0$;
3: while $r_k \neq 0$ do
4:     $\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}$;
5:     $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
6:     $r_{k+1} \leftarrow r_k - \alpha_k A p_k$;
7:     $\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$;
8:     $p_{k+1} \leftarrow r_{k+1} + \beta_{k+1} p_k$;
9:     $k \leftarrow k + 1$;
10: end while
```

involves PDE. Hence techniques for accelerating convergence is of great interest. Accordingly, we first need to know more information about the rate of convergence of the CG method.

Let $e_k = x_k - x^*$ and $\| \cdot \|_A$ be weighted norm, i.e $\|u\|_A = (u^T Au)^{\frac{1}{2}}$ for $\forall u \in \mathbb{R}^n$. A basic convergence estimate for the CG method is given by

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \min_{P_k \in \pi_k^1} \max_{\lambda \in \sigma(A)} |P_k(\lambda)|, \tag{2.4}$$

here $\pi_k^1$ denotes the set of monic polynomials (i.e. the polynomials with the coefficients of their highest order terms equal to 1) of degree $k$ and $\sigma(A)$ denotes the spectrum of $A$, see e.g. Axelsson (1994).

Many important estimates can be derived from (2.4) when appropriate polynomials $P_k(x)$ are chosen. The following theorem is one special case, which improves the finite termination statement when $A$ has certain spectrum, see e.g. Nocedal and Wright (2006).

**Theorem 2.1.1.** *If $A$ has only $r$ distinct positive eigenvalues, then the CG iteration will terminate at the solution in at most $r$ iterations.*

*Proof.* Let $\lambda_1$, $\lambda_2$, ..., $\lambda_r$ denote the $r$ distinct eigenvalues of $A$, then

$$P_r(\lambda) = \prod_{i=1}^{r} \left(1 - \frac{\lambda}{\lambda_i}\right) \in \pi_r^1 \quad \text{and} \quad P_r(\lambda_i) = 0,$$

for $1 \leq i \leq r$. According to (2.4), we have

$$\frac{\|e_r\|_A}{\|e_0\|_A} \leq \max_{1 \leq i \leq r} |P_r(\lambda_i)| = 0.$$

This directly leads to the conclusion. □

By Theorem 2.1.1, the number of iteration required by the CG method is considerably smaller when $r$ is much smaller than $n$. Therefore clustered eigenvalue distribution is a favorable spectral property for the CG method.

**Remark 2.1.2.** Theorem 2.1.1 is shown in exact arithmetic. Indeed, it is possible that a few more iterations are needed to assure the accuracy of the solution in practice because of numerical error.

For more general linear equations, we summarize two convergence estimates in Theorem 2.1.3. Both of them can be found in e.g. Nocedal and Wright (2006).

**Theorem 2.1.3.** *If $A$ has eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, we have that*

$$\|e_{k+1}\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|e_0\|_A^2. \tag{2.5}$$

*Let $\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \lambda_n / \lambda_1$ denote the Euclidean condition number of $A$, then*

$$\|e_k\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \|e_0\|_A. \tag{2.6}$$

The two estimates in Theorem 2.1.3 characterize the behavior of the CG method by the spectrum of $A$, more precisely, the eigenvalue distribution and condition number respectively. Notice that Theorem 2.1.1 can also be obtained by setting $k = r - 1$ in (2.5).

### 2.1.2. Preconditioned Conjugate Gradient Method

The analysis on the convergence behavior naturally motivates *preconditioning*, i.e. improving the spectral properties of the linear system, so as to accelerate the CG method. A common way to implement preconditioning is transforming the original linear equation into an equivalent one. Such a process can be carried out via variable substitution.

If $C$ is a nonsingular matrix, let $\hat{x} = Cx$, then $Ax = b$ is equivalent to

$$(C^{-T} A C^{-1}) \hat{x} = C^{-T} b. \tag{2.7}$$

Obviously the matrix $C^{-T} A C^{-1}$ is symmetric positive definite, hence the CG method is still applicable to the equivalent linear equation (2.7). Moreover, the convergence rate of the CG method applied to (2.7) depends on the spectrum of $C^{-T} A C^{-1}$, instead of that of $A$. We would expect the matrix $C^{-T} A C^{-1}$ has clustered eigenvalues, or much smaller condition number compared to $A$, thus we can achieve faster convergence of the CG method according to Theorem 2.1.1 and Theorem 2.1.3.

Specially, if all the eigenvalues of $C^{-T} A C^{-1}$ are clustered around 1, which indicates the condition number $\kappa(C^{-T} A C^{-1})$ is close to 1, then the convergence of the CG method can be very rapid according to the estimate (2.6) in Theorem 2.1.3. This happens in the case $C^T C$ approximates $A$. One common choice of such a matrix $C$ could be the triangular matrix computed by incomplete Cholesky factorization of $A$, see e.g. Golub and Van Loan (1996).

In the specific case that the coefficient matrix has its eigenvalues well clustered around 1, a useful estimate is given by Theorem 2.1.4, which is a variant of the estimate given in van der Vorst (1980) and can be found in e.g. Chan and Jin (2007) and Ng (2004).

**Theorem 2.1.4.** *If the eigenvalues* $\{\lambda_j\}_{j=1,2,\dots,n}$ *of* $A$ *are ordered such that*

$$0 < \eta < \lambda_1 \le \cdots \le 1-\epsilon \le \lambda_p \le \cdots \le \lambda_{n-q} \le 1+\epsilon \le \lambda_{n-q+1} \le \cdots \le \lambda_n,$$

*where* $0 < \epsilon < 1$*, then we have*

$$\frac{\|e_k\|_A}{\|e_0\|_A} \le 2\left(\frac{1+\epsilon}{\eta}\right)^p \epsilon^{k-p-q} \tag{2.8}$$

*for* $k \ge p+q$*.*

Instead of using $C$ explicitly, a practical algorithm exploits the *preconditioner* defined by $P = C^T C$, which is symmetric positive definite by construction. Further, the spectrum of $C^{-T}AC^{-1}$ is the same as that of $P^{-1}A$ since the two matrices are similar.

The preconditioned version of the CG method is given by Algorithm 2.2. This algorithm is beneficial for it does not require formulating equation (2.7) or explicit variable substitution. Moreover, it preserves $A$ for efficient matrix-vector multiplication when $A$ has favorable structure such like sparsity.

**Algorithm 2.2** Preconditioned CG method

```
1: Given $x_0$, preconditioner $P$;
2: Set $r_0 \leftarrow b - Ax_0$;
3: Solve $Py_0 = r_0$ for $y_0$;
4: Set $p_0 = y_0$, $k \leftarrow 0$;
5: while $r_k \neq 0$ do
6:     $\alpha_k \leftarrow \frac{r_k^T y_k}{p_k^T A p_k}$;
7:     $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
8:     $r_{k+1} \leftarrow r_k - \alpha_k A p_k$;
9:     Solve $Py_{k+1} = r_{k+1}$;
10:    $\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$;
11:    $p_{k+1} \leftarrow y_{k+1} + \beta_{k+1} p_k$;
12:    $k \leftarrow k + 1$;
13: end while
```

The conjugacy property of search directions (2.2) still holds, while the orthogonality of the residuals (2.3) becomes

$$r_k^T P^{-1} r_i = 0, \quad \text{for} \quad i = 0, 1, \dots, k-1. \tag{2.9}$$

Obviously, the main effort of the standard CG method is to compute the matrix-vector product $Ap_k$, in addition, preconditioned CG method needs extra effort to solve a linear equation $Py_k = r_k$ in each iteration. To see this difference clearly, we can simply set $P = I$ in Algorithm 2.2 to recover the standard CG method.

Those linear equations which involve the preconditioner $P$ must be tractable, since they are constantly solved. In Algorithm 2.2, it is clear that applying preconditioner $P$ only requires solution of the relevant linear equations, hence it is not even necessary to have the explicit form of $P$. Instead, a linear solver dealing with $Py_k = r_k$ is sufficient for preconditioned CG method. In practice, approximate solvers are usually used to realize preconditioning for further reducing computational effort.

### 2.1.3. Conjugate Gradient Method in Hilbert Space

A more general form of the CG method can be derived in Hilbert space, which does not only provide an alternative way of treating preconditioning and analyzing the convergence rate of corresponding CG algorithm, but also gives rise to a variety of preconditioning techniques.

Solving linear equation $\mathbf{Ax} = \mathbf{b}$ in $\mathbb{R}^n$ with standard CG method (Algorithm 2.1) requires $\mathbf{A}$ to be symmetric and positive definite, for assuring the validity of the CG method. Now we consider it in a Hilbert space $H$, by which we mean $\mathbb{R}^n$ endowed with a given inner product $\langle\cdot,\cdot\rangle$. We first extend these two necessary properties of $\mathbf{A}$ for the Hilbert space $H$.

Symmetry of $\mathbf{A}$ is generalized to self-adjointness with respect to the given inner product $\langle\cdot,\cdot\rangle$, defined by

$$\langle \mathbf{Au}, \mathbf{v}\rangle = \langle \mathbf{u}, \mathbf{Av}\rangle, \quad \text{for } \forall \mathbf{u}, \mathbf{v} \in H. \tag{2.10}$$

At the same time, the positive definiteness is expressed by

$$\langle \mathbf{Au}, \mathbf{u}\rangle > 0, \quad \text{for } \mathbf{u} \neq 0. \tag{2.11}$$

If these two conditions are satisfied, the CG method implemented in $H$ can be obtained by simply replacing the dot product (scalar product) in Algorithm 2.1 with inner product $\langle\cdot,\cdot\rangle$. We omit the result here for it is trivial. If we define the inner product used in the general form of the CG method, i.e. the CG method in general Hilbert spaces, by $\langle \mathbf{u}, \mathbf{v}\rangle = \mathbf{u}^T\mathbf{v}$, then we can recover Algorithm 2.1.

The relaxation of symmetry to self-adjointness (2.10) allows the CG method to handle even nonsymmetric linear equations, when the right inner product in is chosen. As an example, we first rebuild Algorithm 2.2 in such a context.

Let $P$ be a preconditioner defined in the previous section, which is a symmetric positive definite matrix, the equation

$$P^{-1}Ax = P^{-1}b \tag{2.12}$$

has its coefficient matrix $P^{-1}A$ usually nonsymmetric. We define a weighted Hilbert space $H_P$ whose inner product is given by $\langle u, v\rangle_P = \langle Pu, v\rangle = u^T Pv$, then it is easy to see

$$\langle P^{-1}Au, v\rangle_P = \langle u, P^{-1}Av\rangle_P, \quad \text{for } \forall u, v \in H_P.$$

The positive definiteness directly follows that of matrix $A$. Thus (2.10) and (2.11) are satisfied in this case. Apply the CG method in terms of inner product $\langle\cdot,\cdot\rangle_P$ to (2.12), we immediately obtain Algorithm 2.3.

If we compare Algorithm 2.3 with Algorithm 2.2, noticing that $\tilde{r}_k = P^{-1}r_k$, then obviously Algorithm2.2 and Algorithm 2.3 are essentially the same. The orthogonality of preconditioned residuals $\tilde{r}_k$ here is given by

$$\langle \tilde{r}_k, \tilde{r}_i\rangle_P = 0, \quad \text{for } i = 0, 1, \ldots, k-1,$$

**Algorithm 2.3** Standard CG method with $P$-weighted inner product

```
1: Given $x_0$;
2: Set $\tilde{r}_0 \leftarrow P^{-1}(b - Ax_0)$;
3: Set $p_0 = \tilde{r}_0$, $k \leftarrow 0$;
4: while $\tilde{r}_k \neq 0$ do
5:     $\alpha_k \leftarrow \frac{\langle \tilde{r}_k, \tilde{r}_k \rangle_P}{\langle P^{-1}Ap_k, p_k \rangle_P}$;
6:     $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
7:     $\tilde{r}_{k+1} \leftarrow \tilde{r}_k - \alpha_k P^{-1}Ap_k$;
8:     $\beta_{k+1} \leftarrow \frac{\langle \tilde{r}_{k+1}, \tilde{r}_{k+1} \rangle_P}{\langle \tilde{r}_k, \tilde{r}_k \rangle_P}$;
9:     $p_{k+1} \leftarrow \tilde{r}_{k+1} + \beta_{k+1} p_k$;
10:    $k \leftarrow k + 1$;
11: end while
```

as generalization of the orthogonalization given by (2.3). By definition of inner product $\langle \cdot, \cdot \rangle_P$, it is straightforward that $\langle \tilde{r}_k, \tilde{r}_i \rangle_P = \tilde{r}_k^T P \tilde{r}_i = r_k^T P^{-1} r_i$. Hence we can derive (2.9) very conveniently using Algorithm 2.3.

The connection between Algorithm 2.3 and Algorithm 2.2 shows that preconditioned CG method is nothing special but an instance of the general form of the CG method in Hilbert space. In such a context, the choice of a preconditioner is equivalent to the choice of the inner product of the Hilbert space. This is discussed for more general case in Günnel et al. (2011).

## Convergence Analysis of the CG Method in Hilbert Space

Although Algorithm 2.3 is not practical for implementation, it brings more flexibility for developing analysis on the rate of convergence of the CG method since it does not require the coefficient matrix to be symmetric any more. Hence it is especially convenient for analyzing the preconditioned linear equation (2.12) whose coefficient matrix is normally nonsymmetric.

Now we confine ourselves to the linear equation (2.12), which is rewritten into $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A} = P^{-1}A$ and $\mathbf{b} = P^{-1}b$. We present some estimates regarding r-superlinear convergence of the CG method in weighted inner product $\langle \cdot, \cdot \rangle_P$ (Algorithm 2.3). These estimates can be directly used for characterizing preconditioned CG method in standard inner product (Algorithm 2.2) due to the equivalence between these two algorithms.

Based on the K-condition number of matrix $\mathbf{A}$, which is defined via

$$K(\mathbf{A}) = \left(\frac{1}{n}\operatorname{trace}(\mathbf{A})\right)^n / \det(\mathbf{A}) = \left(\frac{1}{n}\sum_{i=1}^{n} \lambda_i(\mathbf{A})\right)^n \left(\prod_{i=1}^{n} \lambda_i(\mathbf{A})\right)^{-1}, \tag{2.13}$$

an r-superlinear estimate on the convergence of the CG method applied to $\mathbf{Ax} = \mathbf{b}$ is proposed in Axelsson and Kaporin (2000), see Theorem 2.1.5 below.

**Theorem 2.1.5.** *Let $k < n$ be even and $k \geq 3 \ln K(\mathbf{A})$, then*

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \left( \frac{3 \ln K(\mathbf{A})}{k} \right)^{k/2}. \tag{2.14}$$

An estimate similar in form can be derived when the linear equation is defined in an infinite dimensional Hilbert space. For details, see Theorem 8 in Axelsson and Karátson (2002). Note that the corresponding CG method should also be given for general Hilbert space, see e.g. Patterson (1974).

Such knowledge is useful typically when the linear equation being solved iteratively comes from discretization of a PDE. In such a case, studying the CG method in infinite dimensional space can provide rate convergence estimates in limit sense.

In particular, if an infinite dimensional linear operator equation is discretized by finite element method (FEM), then the convergence estimate of the CG method in finite dimensional subspace can be determined by the original linear operator rather than the coefficient matrix in the discrete counterpart. To see this, we summarize the analysis of Karátson (2005) as below.

Let $\mathbf{A} = \mathbf{I} + \mathbf{B}$ with $\mathbf{I} \in \mathbb{R}^{n \times n}$ identity and

$$\mathcal{F}(\mathbf{B}) = \left( \sum_{i=1}^{n} \lambda_i(\mathbf{B})^2 \right)^{\frac{1}{2}}.$$

When $\mathbf{B}$ is symmetric, $\mathcal{F}(\mathbf{B}) = \|\mathbf{B}\|_F$, here $\|\mathbf{B}\|_F$ denotes the Frobenius norm of $\mathbf{B}$. If $\mathbf{B}$ has all its eigenvalues nonnegative, then the following inequality holds.

$$\ln K(\mathbf{I} + \mathbf{B}) \leq \frac{1}{2} \mathcal{F}(\mathbf{B})^2. \tag{2.15}$$

This inequality can be derived from the proof of Theorem 8 in Axelsson and Karátson (2002).

Using (2.15), the estimate (2.14) is transformed to (2.16) in Karátson (2005).

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \left( \frac{3 \mathcal{F}(\mathbf{B})^2}{2k} \right)^{k/2}. \tag{2.16}$$

For characterizing the rate of convergence of the CG method, (2.16) does not have much difference compared to (2.14), They are both r-superlinear estimates. However, if the linear equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ is obtained by applying FEM to a linear operator equation such like a PDE, then it can be used to derive mesh independent superlinear convergence under certain assumptions. We denote the linear operator equation by

$$\mathcal{L}u = f, \tag{2.17}$$

with some $f \in H$, here $H$ is a separable Hilbert space.

Suppose the following assumptions are satisfied:

(i) The operator $\mathcal{L}$ is decomposed as $\mathcal{L} = \mathcal{P} + \mathcal{Q}$ where $\mathcal{P}$ is self-adjoint operator in $H$,with dense domain $D$ and $\mathcal{Q}$ is a self-adjoint operator defined on the domain $H$;

(ii) There exists $p > 0$ such that $\langle \mathcal{P}u, u\rangle \geq p\|u\|^2$ $(u \in D)$;
(iii) $\langle \mathcal{Q}u, u\rangle \geq 0$ $(u \in H)$.
(iv) The operator $\mathcal{P}^{-1}\mathcal{Q}$, defined on the energy space $H_{\mathcal{P}}$, is a compact Hilbert-Schmidt operator, i.e.

$$\|\mathcal{P}^{-1}\mathcal{Q}\|_{HS}^2 := \left(\sum_{i=1}^{\infty} \lambda_i^2(\mathcal{P}^{-1}\mathcal{Q})\right) < \infty.$$

Here $\|\cdot\|_{HS}$ is known as Hilbert-Schmidt norm, which can be considered as generalization of Frobenius norm in general Hilbert space. Let $H_{\mathcal{P}}$ be the completion of $D$ under the weighted inner product $\langle\cdot,\cdot\rangle_{\mathcal{P}} = \langle \mathcal{P}\cdot,\cdot\rangle$ and the corresponding norm is denoted by $\|\cdot\|_{\mathcal{P}}$, then the preconditioned form of (2.17), which is $(\mathcal{I} + \mathcal{P}^{-1}\mathcal{Q})u = \mathcal{P}^{-1}f$, has weak formulation

$$\langle u, v\rangle_{\mathcal{P}} + \langle \mathcal{P}^{-1}\mathcal{Q}u, v\rangle_{\mathcal{P}} = \langle \mathcal{P}^{-1}f, v\rangle_{\mathcal{P}}, \quad \text{for } \forall v \in H_{\mathcal{P}}. \tag{2.18}$$

(2.18) is equivalent to

$$\langle u, v\rangle_{\mathcal{P}} + \langle \mathcal{Q}u, v\rangle = \langle f, v\rangle, \quad \text{for } \forall v \in H_{\mathcal{P}}, \tag{2.19}$$

and it has unique solution in $H_{\mathcal{P}}$ by the assumptions. For more details, see Karátson (2005).

Solving (2.17) numerically, we let $V = \text{span}\{\phi_1, \dots, \phi_n\} \subset H_{\mathcal{P}}$ and set matrices $P, Q \in \mathbb{R}^{n\times n}$ to be

$$P = \{\langle \phi_j, \phi_i\rangle_{\mathcal{P}}\}_{i,j=1}^n, \qquad Q = \{\langle \mathcal{Q}\phi_j, \phi_i\rangle\}_{i,j=1}^n,$$

Galerkin discretization of (2.19) leads to a linear equation in $\mathbb{R}^n$, which has form

$$(P + Q)x = b. \tag{2.20}$$

and the coefficient matrix $P + Q$ is symmetric positive definite, following the assumptions on $\mathcal{P}$ and $\mathcal{Q}$. Thus the CG method is applicable to (2.20) and $P$ is available as a preconditioner.

In such a case, it is proved in Karátson (2005) that

**Theorem 2.1.6.** *$\mathcal{F}(P^{-1}Q)^2 \leq \|\mathcal{P}^{-1}\mathcal{Q}\|_{HS}^2$ holds independently of $n$, i.e. the dimension of the finite dimensional subspace $V$.*

If we set $\mathbf{A} = I + P^{-1}Q$ ($\mathbf{B} = P^{-1}Q$ accordingly) in (2.16) and apply Algorithm 2.3 to the $P$-preconditioned equation

$$P^{-1}(P + Q)x = (I + P^{-1}Q)x = P^{-1}b,$$

then the residuals of the CG method satisfy

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \left(\frac{3\mathcal{F}(P^{-1}Q)^2}{2k}\right)^{k/2} \leq \left(\frac{3\|\mathcal{P}^{-1}\mathcal{Q}\|_{HS}^2}{2k}\right)^{k/2}. \tag{2.21}$$

Here the second inequality is obtained by using Theorem 2.1.6. Assumption (iv) assures the right-hand side is bounded, hence the estimate (2.21) guarantees r-superlinear convergence of the CG method regardless of the subspace $V$ chosen for Galerkin discretization.

Specially, when the linear operator equation (2.17) is treated via FEM and the subsequent linear system is solved by the above approach, the resulting rate of convergence is independent of mesh size within the FEM, which makes such a preconditioner $P$ optimal, in the sense of mesh independence.

An application of such analysis will be given in the next chapter, where an estimate very similar to (2.16) is used, given by Theorem 2.1.7.

**Theorem 2.1.7.** *If $\mathbf{A} = \mathbf{I} + \mathbf{B}$ has positive eigenvalues, the residuals of the CG method for $\mathbf{Ax} = \mathbf{b}$ satisfy*

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \left( \frac{2\mathcal{F}(\mathbf{B})}{\lambda_{min}(\mathbf{A})\sqrt{k}} \right)^k . \tag{2.22}$$

Notice that the estimate in Theorem 2.1.7 does not need all the eigenvalues of $\mathbf{B}$ to be nonnegative any more, thus relaxes the requirements for (2.16). Nevertheless, $\lambda_i(\mathbf{A}) \geq 0$ should hold for all indices.

The estimate (2.22) is derived based on the original work by Winther (1980), which asserts if $\mathbf{A} = \mathbf{I} + \mathbf{B}$ with $\lambda(\mathbf{B}) > -1$, then

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \left( \frac{2}{\lambda_{min}(\mathbf{A})} \right)^k \prod_{i=1}^{k} |\lambda_i(\mathbf{B})|. \tag{2.23}$$

Using the arithmetic-geometric means inequality in the form

$$\left( \prod_{i=1}^{k} |\lambda_i(\mathbf{B})| \right)^{\frac{1}{k}} = \left( \prod_{i=1}^{k} |\lambda_i(\mathbf{B})|^2 \right)^{\frac{1}{2k}} \leq \left( \frac{1}{k} \sum_{i=1}^{k} |\lambda_i(\mathbf{B})|^2 \right)^{\frac{1}{2}} ,$$

then (2.22) is straightforward, see Axelsson and Karátson (2009).

## Preconditioned CG Method in Hilbert Space

It is of course not necessary to force the inner product to be induced by the preconditioner when we are discussing preconditioning in a general Hilbert space $H$. Preconditioners can also be developed for certain nonstandard inner product.

If we define the inner product $\langle \cdot, \cdot \rangle_{\mathbf{H}}$ with a symmetric positive definite matrix $\mathbf{H}$ and solve $\mathbf{Ax} = \mathbf{b}$ with the CG method preconditioned by $\mathbf{P}$, then general preconditioned CG method can be given by Algorithm 2.4 when $\mathbf{P}^{-1}\mathbf{A}$ is self adjoint and positive definite with respect to $\langle \cdot, \cdot \rangle_{\mathbf{H}}$, like the requirements given by (2.10) and (2.11). Actually such requirements can be fulfilled when the matrix $\mathbf{HP}^{-1}\mathbf{A}$ is symmetric positive definite. For more details, see Stoll (2008).

Preconditioned CG method in non-standard inner product brings out some unusual preconditioning techniques, one surprising application is Bramble-Pasciak class preconditioners for symmetric saddle point problems, which will be reviewed in Section 2.2. It is even possible to combine different preconditioners to generate a new preconditioner for use. For such a discussion, see e.g. Stoll and Wathen (2008).

**Algorithm 2.4** Preconditioned CG method in non-standard inner product

```
1: Given $\mathbf{x}_0$;
2: Set $\mathbf{r}_0 \leftarrow \mathbf{P}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_0)$;
3: Set $\mathbf{p}_0 = \mathbf{r}_0$, $k \leftarrow 0$;
4: while $\mathbf{r}_k \neq 0$ do
5:     $\alpha_k \leftarrow \frac{\langle \mathbf{r}_k, \mathbf{r}_k \rangle_\mathbf{H}}{\langle \mathbf{P}^{-1}\mathbf{A}\mathbf{p}_k, \mathbf{p}_k \rangle_\mathbf{H}}$;
6:     $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \alpha_k \mathbf{p}_k$;
7:     $\mathbf{r}_{k+1} \leftarrow \mathbf{r}_k - \alpha_k \mathbf{P}^{-1}\mathbf{A}\mathbf{p}_k$;
8:     $\beta_{k+1} \leftarrow \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle_\mathbf{H}}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle_\mathbf{H}}$;
9:     $\mathbf{p}_{k+1} \leftarrow \mathbf{r}_{k+1} + \beta_{k+1}\mathbf{p}_k$;
10:    $k \leftarrow k + 1$;
11: end while
```

## 2.2. Preconditioning for Structured Systems

In the solution of PDE-constrained optimization problems, one special type of linear problem, namely, saddle point problem, has drawn a lot of attention, since it commonly arises from the first order necessity conditions (KKT system) or SQP steps. Krylov subspace methods are usually preferable for such saddle point problems, because they are often sparse and very large-scale. Unfortunately, PDEs normally bring in ill conditioning of the linear problems, thus preconditioning is vital for efficient solution of them.

Regarding the linear algebraic structure of saddle point problems, many block-wise preconditioners are designed, we briefly introduce some popular ones in this section. Some recent works, especially those on time dependent problems, are well summarized in Stoll (2011), see the references therein.

Here we deal with symmetric saddle point problems which have the following general form:

$$\mathcal{A}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}\begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{2.24}$$

with $A \in \mathbb{R}^{n\times n}$ symmetric, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times m}$ symmetric. The properties of these blocks are problem dependent.

The nonsingularity of saddle point matrix $\mathcal{A}$ is discussed in Benzi et al. (2005) and sufficient conditions are provided therein, which are summarized as follows.

**Theorem 2.2.1.** *(i) Assume $A$ is symmetric positive definite and $C$ is symmetric positive semidefinite. If* $\ker(C) \cap \ker(B^T) = \{0\}$*, then $\mathcal{A}$ is nonsingular. In particular, $\mathcal{A}$ is nonsingular if $B$ has full rank.*

*(ii) Assume $A$ is symmetric positive semidefinite and $C = 0$. If* $\ker(A) \cap \ker(B) = \{0\}$*, then $\mathcal{A}$ is nonsingular. The condition* $\ker(A) \cap \ker(B) = \{0\}$ *here actually implies $A$ is positive definite on the kernel of $B$.*

We refer to Benzi et al. (2005) for a comprehensive discussion on general saddle point problems and their numerical solution.

### 2.2.1. Ideal Block Preconditioners

The block preconditioners exhibited in this subsection involve the Schur complement of $A$ in the full matrix $\mathcal{A}$, denoted by $S := -(C + BA^{-1}B^T)$.

In addition to resolving other nonzero blocks in the preconditioners when they are applied, the computation of $S$ is costly since it requires the inverse of $A$. As a result, it is basically necessary to approximate these preconditioners for practical use. From this point of view, such block preconditioners are only *ideal* and many considerations should be taken in real applications.

Without regard to the implementation issues, we list some block preconditioners below.

When $A$ and $-S = C + BA^{-1}B^T$ are both nonsingular, an ideal block diagonal preconditioner is proposed in Rusten and Winther (1992), which is

$$\mathcal{P}_d = \begin{pmatrix} A & 0 \\ 0 & -S \end{pmatrix}.$$

In the special case $C = 0$, correspondingly $-S = BA^{-1}B^T$, it is not hard to see the preconditioned matrix

$$\mathcal{M} = \mathcal{P}_d^{-1}\mathcal{A} = \begin{pmatrix} I & A^{-1}B^T \\ -S^{-1}B & 0 \end{pmatrix}.$$

is nonsingular by assumption and satisfies

$$(\mathcal{M} - I)\left(\mathcal{M} - \frac{1}{2}(1+\sqrt{5})I\right)\left(\mathcal{M} - \frac{1}{2}(1-\sqrt{5})I\right) = 0.$$

It follows that $\mathcal{M}$ is diagonalizable and has only three distinct eigenvalues, namely 1, $\frac{1}{2}(1+\sqrt{5})$ and $\frac{1}{2}(1-\sqrt{5})$. In fact, symmetry of $\mathcal{A}$ is not necessary here. For the proof, see Murphy et al. (2000).

A similar diagonal block preconditioner can be given when $C$ and $-\bar{S} = A + BC^{-1}B^T$ are nonsingular, it has form

$$\bar{\mathcal{P}}_d = \begin{pmatrix} -\bar{S} & 0 \\ 0 & C \end{pmatrix}.$$

These preconditioners are usually symmetric and widely used. Specially, an analysis on the robustness of the above block diagonal preconditioners in parameter-dependent saddle point problems could be found in Zulehner (2010).

Other than the block diagonal preconditioners $\mathcal{P}_d$ and $\bar{\mathcal{P}}_d$, two block triangular preconditioners are derived from the following triangular factorization of the saddle point matrix, which reads

$$\mathcal{A} = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} = \begin{pmatrix} I & 0 \\ BA^{-1} & \pm I \end{pmatrix}\begin{pmatrix} A & 0 \\ 0 & S \end{pmatrix}\begin{pmatrix} I & A^{-1}B^T \\ 0 & \pm I \end{pmatrix}. \tag{2.25}$$

Here the nonsingularity of $A$ and its Schur complement $S$ is assumed. Computing the product of the first two blocks gives the left preconditioners (regarding positive sign and negative sign respectively)

$$\mathcal{P}_t = \begin{pmatrix} A & 0 \\ B & \pm S \end{pmatrix}.$$

The third matrix on the right-hand side of (2.25) indicates that the preconditioned matrix has spectrum

$$\lambda(\mathcal{P}_t^{-1}\mathcal{A}) = \begin{cases} 1, & \text{if } \; +S \; \text{ is used,} \\ \pm 1, & \text{if } \; -S \; \text{ is used.} \end{cases}$$

Note that applying $\mathcal{P}_t$ requires a nonsymmetric solver, for such a discussion, see e.g. Benzi et al. (2005) and Benzi and Wathen (2008).

**Remark 2.2.2.** In order to reduce the numerical cost of applying $\mathcal{P}_d$ and $\mathcal{P}_t$, usually the blocks $A$ and $S$ are approximated. The choice of the approximations are highly problem dependent. As applying preconditioners in iterative solvers is normally a major part of numerical cost, careful considerations concerning effective implementation should be made.

### 2.2.2. Bramble-Pasciak Type Preconditioning

As being pointed out in the previous section, preconditioned CG method in general Hilbert space does not require the preconditioner to be symmetric, instead, it only needs the preconditioned matrix to be self-adjoint and positive definite with respect to the inner product of the Hilbert space, see the discussion on Algorithm 2.4.

For solving the saddle point problem (2.24), a preconditioned CG method in non-standard inner product is developed in Bramble and Pasciak (1988), where a block triangular preconditioner is proposed coupled with a specific inner product.

Let $A_0$ be the symmetric approximation of $A$ such that $A_0$ and $A - A_0$ are both symmetric and positive definite. The triangular preconditioner is constructed as

$$\mathcal{P}_{BP} = \begin{pmatrix} A_0 & 0 \\ B & -I \end{pmatrix},$$

thus the inverse of $\mathcal{P}_{BP}$ is explicitly given by

$$\mathcal{P}_{BP}^{-1} = \begin{pmatrix} A_0^{-1} & 0 \\ BA_0^{-1} & -I \end{pmatrix}$$

and the preconditioned matrix is

$$\mathcal{M} = \mathcal{P}_{BP}^{-1}\mathcal{A} = \begin{pmatrix} A_0^{-1}A & A_0^{-1}B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{pmatrix}.$$

Obviously $\mathcal{M}$ is nonsymmetric. Bramble and Pasciak introduced the inner product

$$\langle u, v\rangle_{\mathcal{H}_{BP}} = u^T\mathcal{H}_{BP}v \quad \text{with} \quad \mathcal{H}_{BP} = \begin{pmatrix} A - A_0 & 0 \\ 0 & I \end{pmatrix}.$$

Surprisingly, $\mathcal{M}$ is self-adjoint in the $\mathcal{H}_{BP}$-inner product, this can be seen from

$$\mathcal{H}\mathcal{P}_{BP}^{-1}\mathcal{A} = \begin{pmatrix} AA_0^{-1}A - A & AA_0^{-1}B^T - B^T \\ BA_0^{-1}A - B & BA_0^{-1}B^T + C \end{pmatrix}.$$

Similarly, another triangular preconditioner

$$\widetilde{\mathcal{P}}_{BP} = \begin{pmatrix} A_0 & 0 \\ B & -S_0 \end{pmatrix}$$

is also available in the inner product

$$\langle u, v \rangle_{\widetilde{\mathcal{H}}_{BP}} = u^T \widetilde{\mathcal{H}}_{BP} v \quad \text{with} \quad \widetilde{\mathcal{H}}_{BP} = \begin{pmatrix} A - A_0 & 0 \\ 0 & S_0 \end{pmatrix},$$

proposed in Dollar et al. (2008). Here $S_0 = BA_0^{-1}B^T + C$.

In order to apply the CG method as Algorithm 2.4, the matrix $\mathcal{H}\mathcal{P}_{BP}^{-1}\mathcal{A}$ must be also positive definite. Hence the matrix $A_0$ should be properly chosen.

It could also necessary to scale $A_0$ to make sure $A - A_0$ is positive definite, which ensures the inner product regarding $\mathcal{H}_{BP}$ is well defined. A remedy is made to avoid the scaling in Stoll and Wathen (2007), the authors suggested to use the inner product

$$\langle u, v \rangle_{\bar{\mathcal{H}}} = u^T \bar{\mathcal{H}} v \quad \text{with} \quad \bar{\mathcal{H}} = \begin{pmatrix} A + A_0 & 0 \\ 0 & I \end{pmatrix},$$

and the coupled triangular preconditioner is given by

$$\mathcal{P}_1 = \begin{pmatrix} A_0 & 0 \\ -B & I \end{pmatrix} \qquad \text{or} \qquad \mathcal{P}_2 = \begin{pmatrix} A_0 & 0 \\ -B & S_0 \end{pmatrix}.$$

However, the authors also proved that such preconditioned matrices are always indefinite in the standard inner product.

Notice that although these couples of triangular preconditioner and non-standard inner product are developed for Bramble-Pasciak CG method, they can still be embedded in other symmetric Krylov subspace solvers such like MINRES, if the positive definiteness is violated. For such an application, see e.g. Rees, Stoll and Wathen (2010).

## 2.3. Deflated Krylov Subspace Methods

In order to accelerate the solution of linear systems with the same symmetric positive definite coefficient matrix but multi right-hand sides, augmented CG method was developed. The idea of augmented CG method is constraining CG iteration in standard Krylov subspace augmented with a few more vectors which possibly contain information of the solution, see e.g. Nicolaides (1987) and Erhel and Guyomarc'h (2000). Deflated CG method is essentially the same (see e.g. Meurant (2006)). The difference only lies in the subspace used as the constraint.

Recently, many works have been done to broaden the use of deflation (augmentation) technique to more Krylov subspace methods. In this section, we discuss deflated Krylov subspace methods for solving a nonsingular linear algebraic system

$$Ax = b, \tag{2.26}$$

constrained by a given subspace $\mathcal{W}$, where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. Notice that $\mathcal{W}$ should contain (approximate) eigenvectors of $A$. In a specific algorithm, the subspace $\mathcal{W}$ is represented by a matrix $W$ which is constructed by $l$ ($l < n$) linearly independent column vectors $\{w_i\}_{i=1}^{l}$ and has the form $W = [w_1, w_2, \ldots, w_l]$ with its columns orthonormal.

We mainly focus on the deflation technique which removes certain troublesome eigenvalues of the coefficient matrix to improve the spectrum. This is why we also treat deflation as a preconditioning technique here.

We illustrate deflated CG method first and move on to more general discussion afterwards.

### 2.3.1. Deflated Conjugate Gradient Method

As an instance of deflated Krylov subspace methods, we cite the deflated CG algorithm for real symmetric positive definite linear problems, which is derived and analyzed in Saad et al. (1999) (a preconditioned version of deflated CG algorithm can be found in the same paper):

**Algorithm 2.5** Deflated CG method

```
 1: Choose $l$ linearly independent vectors $w_1, w_2, \ldots, w_l$. Define $W = [w_1, w_2, \ldots, w_l]$.
 2: Choose an initial guess $x_0$ such that $W^T r_0 = 0$, where $r_0 = b - Ax_0$.
 3: Solve $W^T AW\mu_0 = W^T Ar_0$ for $\mu$ and set $p_0 = r_0 - W\mu_0$.
 4: for $k = 1, 2, \ldots,$ do
 5:     $\alpha_{k-1} = \dfrac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$
 6:     $x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$
 7:     $r_k = r_{k-1} - \alpha_{k-1} A p_{k-1}$
 8:     $\beta_{k-1} = \dfrac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$
 9:     Solve $W^T AW\mu_k = W^T Ar_k$ for $\mu$
10:     $p_k = \beta_{k-1} p_{k-1} + r_k - W\mu_k$
11: end for
```

Obviously, the major extra numerical cost comes from Step 9 in Algorithm 2.5, where a matrix-vector product is computed and a linear equation of dimension $k$ is solved. For minimizing such effort, $AW$ and $W^T AW$ could be computed once and explicitly stored at the beginning (see Olshanskii and Simoncini (2010)).

**Remark 2.3.1.** In fact, initial guess $x_0$ for deflated CG method only needs to satisfy $W^T r_0 = 0$, with $r_0 = b - Ax_0$. Hence $x_0$ can be generated from an arbitrary starting point $x_{-1} \in \mathbb{R}^{n\times n}$ by

$$x_0 = x_{-1} + W(W^T AW)^{-1} W^T r_{-1},$$

here $r_{-1} = b - Ax_{-1}$ (see e.g. Saad et al. (1999)). In the rest of this section, we confine our discussion to the case $x_{-1} = 0$, which gives $x_0 = W(W^T AW)^{-1} W^T b$.

As being pointed out in Saad et al. (1999), deflated CG method applied to (2.26) is equivalent to the standard CG method applied to

$$H^T AHx = H^T b, \tag{2.27}$$

with $H = I - W(W^T AW)^{-1}W^T A$ and $I$ identity matrix in $\mathbb{R}^{n\times n}$. Furthermore, the convergence behavior of Algorithm 2.5 is determined by the spectrum of auxiliary matrix $\bar{A} := H^T AH$.

Notice that $\bar{A}$ is a singular matrix. In fact, by simple calculation, we can obtain $\bar{A}W = 0$. This can not be a trouble for Algorithm 2.5, because the algorithm solves $Ax = b$ directly in $\mathcal{W}$ and the iteration is always restricted to the subspace $A$-orthogonal to $\mathcal{W}$.

When deflated CG is performed, an extra step

$$r_k := r_k - W(W^T W)^{-1}W^T r_k$$

after Step 7 is usually necessary for recovering orthogonality. This is also important for other deflated Krylov subspace methods.

## 2.3.2. Deflated Preconditioned Krylov Subspace Methods

In Gaul et al. (2011), a general framework for deflated and augmented Krylov subspace methods is proposed. The authors conclude that deflation (augmentation) technique can be embedded into any Krylov subspace method. We characterize deflation according to their framework and refer the readers to Gaul et al. (2011) for a more detailed discussion.

As discussed in the beginning of this chapter, a Krylov subspace method generates an approximate solution in $x_0 + \mathcal{K}(A, r_0; k)$ at $k$-th iteration, i.e.

$$x_k \in x_0 + \mathcal{K}(A, r_0; k). \tag{2.28}$$

Furthermore, the $k$-th residual $r_k := b - Ax_k$ satisfies

$$r_k \perp_B \mathcal{K}(A, r_0; k). \tag{2.29}$$

Here $B = I$ results in the CG method when $A$ is symmetric positive definite. If $B = A$, then (2.28) and (2.29) characterize MINRES and GMRES.

In general, a deflated Krylov subspace method

(i) directly solves the linear system (2.26) in $\mathcal{W}$, i.e. it solves the projected linear equation $W^T AW\bar{x} = W^T b$ and projects the solution $\bar{x}$ back to $\mathbb{R}^n$ by $W\bar{x}$, to generate an initial point $x_0$;

(ii) keeps the iteration in the subspace $A$-orthogonal to $W$.

In fact, (i) gives an initial point satisfying $W^T r_0 = 0$. (ii) can be achieved by searching

$$x_k \in x_0 + \mathcal{K}(A, r_0; k) + \mathcal{W} \tag{2.30}$$

and the residual $r_k$ satisfies

$$r_k \perp_B \mathcal{K}(A, r_0; k) \quad \text{and} \quad r_k \perp_B \mathcal{W}. \tag{2.31}$$

Derivation of deflated CG method, deflated MINRES and deflated GMRES in this framework can be found in Gaul et al. (2011).

Instead of using the original MINRES method developed in Saunders and Paige (1975), a preconditioned deflated MINRES algorithm is derived in Olshanskii and Simoncini (2010) based on a revised version of MINRES, see Algorithm 2.6. We cite it here for latter use in Chapter 4. The implementation requires careful considerations regarding efficiency and stability, similar to deflated CG method in Section 2.3.1. The extra numerical effort of Algorithm 2.6 is the same as that in Algorithm 2.5.

**Algorithm 2.6** Deflated preconditioned MINRES method

```
Given $A$, $b$, maxit, tol, $P$ and $W$ with orthonormal columns
$x_0 = W(W^TAW)^{-1}W^Tb$ starting approximation
$r = b - Ax_0$, $y = P^{-1}r$, $r_1 = r$, $k = 0$
$\beta_1 = \sqrt{r^Ty}$
$\beta = \beta_1$, $\beta_0 = 0$, $\bar{d} = 0$, $e = 0$, $\bar{\phi} = \beta_1$, $\chi_1 = \beta_1$, $\chi_2 = 0$
$c = -1$, $s = 0$, $w = 0$, $w_2 = 0$, $r_2 = r_1$
**while** $k <$ maxit **do**
  $k = k + 1$
  $v = y/\beta$
  $y = Av - AW(W^TAW)^{-1}W^TAv$
  if $k \geq 2$, $y = y - (\beta/\beta_0)r_1$
  $\alpha = v^Ty$
  $y = y - r_2\alpha/\beta$
  $r_1 = r_2$, $r_2 = y$
  $y = P^{-1}r_2$
  $\beta_0 = \beta$, $\beta = \sqrt{r_2^Ty}$
  $e_0 = e$, $\delta = c\bar{d} + s\alpha$, $\bar{g} = s\bar{d} - c\alpha$, $e = s\beta$, $\bar{d} = -c\beta$
  $\gamma = \max\{\|[\bar{g}, \beta]\|, e\}$, $c = \bar{g}/\gamma$, $s = \beta/\gamma$, $\phi = c\bar{\phi}$, $\bar{\phi} = s\bar{\phi}$
  $w_1 = w_2$, $w_2 = w$
  $w = (v - e_0w_1 - \delta w_2)/\gamma$
  $g = W(W^TAW)^{-1}W^TAw\phi$
  $x_k = x_{k-1} - g + \phi w$
  $\zeta = \chi_1/\gamma$, $\chi_1 = \chi_2 - \delta\zeta$, $\chi_2 = -e\zeta$
  Check preconditioned residual norm $\bar{\phi}$ for convergence
**end while**
```

# Chapter 3.

# Preconditioning for a Partial Integro-Differential Equation in Finance

## 3.1. Precondition a Partial Integro-Differential Equation

For pricing European call options, a number of extended mathematical models have been developed based on Black-Scholes model (Black and Scholes (1973)). Among them, a jump-diffusion model commonly leads to a partial differential equation with an additional non-local integral term, i.e. partial integro-differential equation (PIDE). It requires solution of linear systems with dense coefficient matrices when such PIDEs are tackled numerically. An efficient solver is of great interest.

In Sachs and Strauss (2008), the authors formulate a PIDE based on Merton's jump-diffusion model (Merton (1976)). In order to cure the numerical instability caused by the convection term in the underlying PIDE, variable transformation is suggested. As a result, the problem is turned into an equivalent one aiming to find a $y(t,x) \in C^{1,2}((0,T] \times \mathbb{R}) \cap C^0([0,T] \times \mathbb{R})$ satisfying

$$\begin{aligned} y_t - \frac{1}{2}\sigma^2 y_{xx} + (r+\lambda)y - \lambda \int_{-\infty}^{\infty} y(t,z) f(z-x)\,\mathrm{d}z = 0, \quad \text{on} \quad (0,T] \times \mathbb{R}, \\ y(0,x) = H(e^x), \quad \forall x \in \mathbb{R}, \end{aligned} \tag{3.1}$$

where $f(x) = \frac{1}{\sqrt{2\pi}\sigma_J} e^{-(x-\mu_J)^2/(2\sigma_J^2)}$ is normal density function ($\mu_J$ is set to be zero throughout this work), $H(x) = \max\{0, x-K\}$ is payoff function with strike price $K$, $\sigma$, $\lambda$, $T$ are positive constants and $r \geq 0$ (see Sachs and Strauss (2008) for more details). The discretization is carried out using backward difference formula of second order (BDF2) in time and central difference scheme for spatial derivatives. The resulting discrete linear equations share a positive definite coefficient matrix, which allows the usage of conjugate gradient (CG) method. Moreover, exploiting the Toeplitz structure of the coefficient matrix, efficient matrix-vector multiplication can be achieved utilizing fast Fourier transformation (FFT) in the CG iteration. Circulant type preconditioners are suggested for accelerating convergence.

It is well known that the convergence behavior of CG iteration heavily relies on the spectral properties of the coefficient matrix, which can be significantly improved when a proper preconditioner is used. In this section, we follow the approach proposed by Sachs and Strauss (2008) and focus on preconditioning for PIDE (3.1). Other than circulant preconditioners, which aim to precondition the dense matrix taking advantage of its Toeplitz structure, we also suggest to use a tridiagonal preconditioner which only preconditions the PDE part in the discrete problem. These two types of preconditioners are compared theoretically and numerically. Rigorous spectral analysis for the unpreconditioned and preconditioned systems is developed. Mesh independent superlinear rate of convergence is proved for both preconditioned CG solvers. Notice that we confine our discussion on circulant preconditioners to Strang's preconditioner since it is easy to construct and outperforms other circulant preconditioners in this case according to the numerical results presented in Sachs and Strauss (2008).

If FEM is used for spatial discretization, Toeplitz structure of the resulting linear system can still be preserved under certain assumptions. Thus we are still able to apply the same CG solver and preconditioning techniques developed for the linear equations obtained by finite difference method (FDM).

In the context of FEM discretization, the tridiagonal preconditioner can be treated as discretized elliptic operator. A superlinear convergence estimate of CG solver using the tridiagonal preconditioner can be obtained by exploiting the relationship between the coefficient matrices in the discrete linear equations and the linear operators in corresponding differential equations. More precisely, the convergence result is independent of basis functions used for discretization and only related to the operators which show up in the original PIDE, thus mesh independent.

We acknowledge that, Pang et al. worked on PIDE (3.1) with $\mu_J \neq 0$ which results in a nonsymmetric Toeplitz system. They apply the CG method to normalized equations with Strang's preconditioner and a tridiagonal preconditioner respectively. They theoretically establish superlinear rate of convergence for both cases using a family of generating functions defined in Zhang (2010), which is quite different from our approach. Their results can be found in Pang et al. (2011) and Pang et al. (2012).

### 3.1.1. Numerical Treatment of the PIDE

Truncating the domain for $x$ from $\mathbb{R}$ into a sufficiently large subset $\Omega := (x_-, x_+)$, the following problem with approximate boundary conditions is solved numerically in Sachs and Strauss (2008) instead of (3.1):

$$\begin{gathered} y_t - \frac{1}{2}\sigma^2 y_{xx} + (r+\lambda)y - \lambda \int_{x_-}^{x_+} y(t,z) f(z-x)\,\mathrm{d}z - \lambda R(t,x,x_+) = 0, \quad \text{on} \quad (0,T]\times\Omega, \\ y(t,x_-) = 0, \quad y(t,x_+) = e^{x-\zeta t} - Ke^{-rt}, \\ y(0,x) = H(e^x), \quad \forall x \in \Omega. \end{gathered} \tag{3.2}$$

Here $R(t,x,x_+) := e^{-\zeta t} e^{x+\sigma_J^2/2} \Phi\left(\frac{x-x_+ + \sigma_J^2}{\sigma_J}\right) - Ke^{-rt}\Phi\left(\frac{x-x_+}{\sigma_J}\right)$ is the integral term for the region outside $\Omega$ with $\zeta$ constant and $\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-z^2/2}\,\mathrm{d}z$.

For finite difference discretization, the domain is partitioned as follows:

$$\begin{aligned} x_i &:= x_- + (i-1)h \quad \text{with} \quad i = 1,\dots,n+2, \quad h = (x_+ - x_-)/(n+1), \\ t_p &:= p\tau \qquad\qquad\quad \text{with} \quad p = 1,\dots,m, \qquad \tau = T/m. \end{aligned} \tag{3.3}$$

Note that the number of spatial nodes is different from that in Sachs and Strauss (2008). It is adjusted for simpler subscripts in related matrices.

Let $y^p(x) \in C^2(\Omega)$ approximate $y(t_p, x)$. For simplicity, we use the notation $y^p$ below. To obtain second order accuracy and stability in time, a backward difference formula of second order (BDF2) is used when $p \geq 2$, which is

$$y_t(t_p, x) \quad \approx \quad \begin{cases} (\frac{3}{2}y^p - 2y^{p-1} + \frac{1}{2}y^{p-2})/\tau, & \text{for } p \geq 2, \\ (y^p - y^{p-1})/\tau, & \text{for } p = 1. \end{cases}$$

Hence at the $p$th (without loss of generality, let $p \geq 2$) time step, the following PIDE is indeed discretized and solved on $\Omega$:

$$\begin{gathered} \big(\frac{3}{2} + (r+\lambda)\tau\big)y^p - \frac{1}{2}\sigma^2\tau y^p_{xx} - \lambda\tau\int_{x_-}^{x_+} y^p(t_p, z) f(z-x)\,\mathrm{d}z = b_p(x), \\ y^p(x_-) = 0, \quad y^p(x_+) = e^{x-\zeta t_p} - Ke^{-rt_p}, \end{gathered} \tag{3.4}$$

with $b_p(x) = 2y^{p-1} - \dfrac{1}{2}y^{p-2} + \lambda\tau R(t_p, x, x_+)$.

Using central difference scheme for the spatial derivative and composite trapezoidal rule for the integral term to discretize (3.4) and eliminating two trivial equations (the first and the last) of the consequent linear system, we come to a Toeplitz system

$$T_n x^p = b^p, \qquad p = 2, \ldots, m.$$

The diagonals of the $n \times n$ symmetric coefficient matrix $T_n$ (see (A.1)) are given by

$$\begin{aligned} a_0 &= \sigma^2\frac{\tau}{h^2} + (r+\lambda)\tau + \frac{3}{2} - \frac{\lambda\tau h}{\sqrt{2\pi}\sigma_J}, \\ a_1 = a_{-1} &= -\frac{\sigma^2}{2}\frac{\tau}{h^2} - \frac{\lambda\tau h}{\sqrt{2\pi}\sigma_J}e^{-h^2/(2\sigma_J^2)}, \\ a_i = a_{-i} &= -\frac{\lambda\tau h}{\sqrt{2\pi}\sigma_J}e^{-(ih)^2/(2\sigma_J^2)}, \quad 2 \leq i \leq n-1. \end{aligned} \tag{3.5}$$

Let $2\hat{x} = x_+ - x_-$, then in terms of $n$ and $m$ (3.5) is equivalent to

$$\begin{aligned} a_0 &= \frac{\sigma^2 T(n+1)^2}{4\hat{x}^2 m} + \frac{(r+\lambda)T}{m} + \frac{3}{2} - \frac{\alpha}{(n+1)m}, \\ a_1 = a_{-1} &= -\frac{\sigma^2 T(n+1)^2}{8\hat{x}^2 m} - \frac{\alpha}{(n+1)m}e^{-\beta/(n+1)^2}, \\ a_i = a_{-i} &= -\frac{\alpha}{(n+1)m}e^{-\beta i^2/(n+1)^2}, \quad 2 \leq i \leq n-1, \end{aligned} \tag{3.6}$$

here $\alpha = (x_+ - x_-)\lambda T/(\sqrt{2\pi}\sigma_J)$ and $\beta = (x_+ - x_-)^2/2\sigma_J^2$ are both positive constants according to the definitions of the variables.

It is proved in Sachs and Strauss (2008) that $T_n$ is a positive definite matrix when $n$ is sufficiently large, thus the CG method should be a good choice.

Now we concentrate on the semi-discretized PIDEs (3.4), whose discrete counterparts are actually solved by the CG method. Notice that these subproblems are not time dependent any more. We introduce notations in order to rewrite (3.4) into a concise linear operator form, then learn basic facts about the underlying linear operators in preparation for latter discussions involving Galerkin discretization.

Let $S = \big(\frac{3}{2} + (r+\lambda)\tau\big)I - \frac{1}{2}\sigma^2\tau\Delta$ be elliptic operator from $C^2(\Omega)$ into $C^0(\Omega)$, where $I$ is identity operator and $-\Delta$ is Laplacian defined in the same linear space as $S$, and $Q : L^2(\Omega) \to L^2(\Omega)$ is an integral operator defined by $(Qu)(x) = -\lambda\tau \int_\Omega f(z-x)u(z)\,\mathrm{d}z$, then (3.4) can be represented by

$$\begin{aligned} (S+Q)y^p &= b_p, \\ y^p(x_-) = 0, \quad y^p(x_+) &= e^{x-\zeta t_p} - Ke^{-rt_p}, \end{aligned} \tag{3.7}$$

In order to change the non-homogeneous boundary condition into a homogeneous one thus make $S$ be a self-adjoint operator on $C^2(\Omega)$, we transform (3.7) into an equivalent problem. Firstly, we choose a $y_0 \in C^2(\Omega)$ which satisfies

$$y_0(x) = \begin{cases} 0, & x_- \le x \le x_+ - h, \\ e^{x-\zeta t_p} - Ke^{-rt_p}, & x = x_+, \end{cases} \tag{3.8}$$

for a given mesh and assume $(S+Q)y_0 = \tilde{b}_p$. Moreover, assume that $\bar{y} \in C^2(\Omega)$ solves

$$\begin{aligned} (S+Q)y^p &= b_p - \tilde{b}_p =: \bar{b}_p, \\ y^p(x_-) = 0, \quad y^p(x_+) &= 0, \end{aligned} \tag{3.9}$$

then obviously $y_0 + \bar{y}$ is the solution of (3.7). Notice that on the interval $[x_-, x_+ - h]$, the solution of (3.9) is identical to that of (3.4), thus it is appropriate to solve (3.9) numerically instead of (3.4).

Define the scalar product on $L^2(\Omega)$ by $\langle u, v\rangle = \int_\Omega uv$ and denote the deduced Hilbert space by $H$, then it is natural to denote the corresponding norm by $||\cdot||_{L_2}$. It is well known that $S$ (with respect to the homogeneous Dirichlet boundary conditions in (3.9), the same below) is a self-adjoint operator on $C^2(\Omega) \subseteq H$.

The integral operator $Q$ is also self-adjoint due to its symmetric kernel, by which we mean $f(x,z) = f(z,x)$ in the definition of $Q$, according to the following proposition (see Section 2.2 and Section 5.1 in Gohberg et al. (2003)).

**Proposition 3.1.1.** *If* $\mathbf{k}(t,s)$ *is in* $L_2([a,b]\times[a,b])$ *and* $\overline{\mathbf{k}(t,s)} = \mathbf{k}(s,t)$ *a.e., then the integral operator* $K$ *defined by*

$$(Kf)(t) = \int_a^b \mathbf{k}(t,s)f(s)\mathrm{d}s$$

*is a compact self-adjoint operator on* $L_2([a,b])$*. Moreover,*

$$\|Kf\|_{L_2}^2 \le \int_a^b \left(\int_a^b |\mathbf{k}(t,s)f(s)|\mathrm{d}s\right)^2 \mathrm{d}t \le \|f\|_{L_2}^2 \int_a^b\int_a^b |\mathbf{k}(t,s)|^2\mathrm{d}s\mathrm{d}t. \tag{3.10}$$

For simplicity of notation, we define a constant

$$C_f := \int_\Omega \int_\Omega \big(f(z-x)\big)^2 \mathrm{d}z\,\mathrm{d}x = \iint_{\Omega\times\Omega} \big(f(z-x)\big)^2 \mathrm{d}z\,\mathrm{d}x, \tag{3.11}$$

obviously $C_f < \infty$ and the equality holds according to Fubini's theorem.

Using the properties of $S$ and $Q$, we can show

**Proposition 3.1.2.** *For sufficiently small $\tau$, $S+Q$ is coercive on $C^2(\Omega)$ , i.e. there exists a positive constant c, such that $\langle (S+Q)u, u\rangle \geq c\|u\|^2_{L_2}$ for all $u \in C^2(\Omega)$.*

*Proof.* It is easy to see

$$\langle Su, u\rangle = \big(\frac{3}{2} + (r+\lambda)\tau\big)\|u\|^2_{L_2} + \frac{1}{2}\sigma^2\tau\|u_x\|^2_{L_2}, \tag{3.12}$$

and

$$\begin{aligned} |\langle Qu, u\rangle| &\leq \|Qu\|_{L_2}\|u\|_{L_2} \\ &\leq \lambda\tau\sqrt{C_f}\|u\|^2_{L_2}. \end{aligned}$$

Here the Cauchy-Schwarz inequality is used in the first inequality and (3.10) is used in the second line. Thus for sufficiently small $\tau$ (small enough such that $\lambda\tau\sqrt{C_f} < \frac{1}{2}$), we have

$$\begin{aligned} \langle (S+Q)u, u\rangle &\geq \langle Su, u\rangle - |\langle Qu, u\rangle| \\ &\geq \frac{3}{2}\|u\|^2_{L_2} + \frac{1}{2}\sigma^2\tau\|u_x\|^2_{L_2} - \lambda\tau\sqrt{C_f}\|u\|^2_{L_2} \\ &\geq \|u\|^2_{L_2} + \frac{1}{2}\sigma^2\tau\|u_x\|^2_{L_2} \qquad (3.13)\\ &\geq \|u\|^2_{L_2}. \end{aligned}$$

□

Hence the CG method is applicable to the linear operator equation (3.9) in $H$. Such an approach could be seen as the infinite dimensional counterpart of the CG method applied to $T_n x^p = b^p$ and might provide convergence results in limit sense. Yet this is beyond our interest here.

Other than applying finite difference method to (3.9), we can also derive a finite dimensional linear equation by applying Galerkin discretization to the weak formulation of (3.9), i.e.

$$\langle Sy^p, v\rangle + \langle Qy^p, v\rangle = \langle \bar{b}_p, v\rangle, \quad \forall v \in H^1_0(\Omega). \tag{3.14}$$

For validity of FEM, the domain of $S$ is confined to $H^2(\Omega) \cap H^1_0(\Omega)$ henceforth.

Using the Friedrichs' inequality, we have $\|u\|_{L_2} \leq (x_+ - x_-)\|u_x\|_{L_2}$ with $u(x) \in H^1_0(\Omega)$, which leads to $\|u_x\|^2_{L_2} \geq \frac{1}{(x_+-x_-)^2+1}\|u\|^2_{H^1}$. Hence by inequality (3.13), for sufficiently small $\tau$, we get

$$\langle (S+Q)u, u\rangle \geq \|u\|^2_{L_2} + \frac{1}{2}\sigma^2\tau\|u_x\|^2_{L_2} \geq \frac{1}{2}\sigma^2\tau\|u_x\|^2_{L_2} \geq \frac{\sigma^2\tau}{2(x_+ - x_-)^2 + 2}\|u\|^2_{H^1},$$

which gives the coercivity of the bilinear form $\langle (S+Q)\cdot,\cdot\rangle$ on $H_0^1(\Omega)$.

The boundedness of the bilinear form $\langle (S+Q)\cdot,\cdot\rangle$ on $H_0^1(\Omega)$ can be shown similarly as the proof of Proposition 3.1.2, with help of the Cauchy-Schwarz inequality.

Thus we can conclude that for small enough $\tau$ the weak formulation (3.14) has an unique solution in $H_0^1(\Omega)$ by Lax-Milgram theorem.

Let $V = \operatorname{span}\{\phi_1,\dots,\phi_n\} \subset H_0^1(\Omega)$ be a given finite dimensional subspace,

$$S_n = \{\langle S\phi_j,\phi_i\rangle\}_{i,j=1}^n \quad \textit{and} \quad Q_n = \{\langle Q\phi_j,\phi_i\rangle\}_{i,j=1}^n \tag{3.15}$$

be the Gram matrices corresponding to $S$ and $Q$. We look for an approximation $y_V^p$ of $y^p$ in $V$. Let $y_V^p = \sum_{i=1}^n x_i^p \phi_i$, then $x^p = (x_1^p,\dots,x_n^p)^T \in \mathbb{R}^n$ is the solution of the equation

$$(S_n+Q_n)x^p = b^p, \tag{3.16}$$

where $b^p = \{\langle \bar{b}_p,\phi_i\rangle\}_{i=1}^n$ is also a column vector in $\mathbb{R}^n$.

The symmetry and positive definiteness of the matrix $S_n+Q_n$ are inherited from the self-adjointness and coercivity of the operator $S+Q$ respectively. Thus CG method can be applied to the finite dimensional linear equation (3.16).

Since the Toeplitz structure can bring great advantage to reducing computation effort, we hope the matrix $S_n+Q_n$ is Toeplitz like the coefficient matrix $T_n$ in the discrete equation derived by finite difference method. Fortunately, it is true if the basis functions $\{\phi_i\}_{i=1}^n$ are specially chosen. The following theorem establishes this fact.

**Theorem 3.1.3.** *Assume the basis functions $\{\phi_i\}_{i=1}^n$ have the same shape on an equidistant mesh, by which we mean:*

*(i) Let $\operatorname{supp}_i$ denote the support of $\phi_i$, then for $i=1,\dots,n$, $\operatorname{supp}_i = [x_- + (i-1)h, x_- + (i-1)h + L]$, where $L$ is the length of the interval and $h = (x_+ - x_- - L)/(n-1)$.*

*(ii) $\phi_i(x) = \phi_{i+1}(x+h)$.*

*Then the matrix $S_n+Q_n$ given by (3.15) is Toeplitz.*

*Proof.* It is trivial that the sum of two Toeplitz matrices is Toeplitz by definition. Thus we are going to show $S_n$ and $Q_n$ are Toeplitz respectively, i.e. $(S_n)_{ij} = (S_n)_{i+1,j+1}$ and $(Q_n)_{ij} = (Q_n)_{i+1,j+1}$ for any $i$, $j$ without exeeding bounds. Without loss of generality, we assume $j > i$.

Firstly, we have $(\phi_i)_x(x) = (\phi_{i+1})_x(x+h)$ from assumption. Utilizing the assumption and this equality, we have

$$\begin{aligned}
\langle S\phi_j,\phi_i\rangle &= \int_\Omega \left( (\frac{3}{2} + (r+\lambda)\tau)\phi_j - \frac{1}{2}\sigma^2\tau\Delta\phi_j \right)\phi_i \\
&= (\frac{3}{2} + (r+\lambda)\tau)\int_{\operatorname{supp}_i \cap \operatorname{supp}_j} \phi_j\phi_i + \frac{1}{2}\sigma^2\tau \int_{\operatorname{supp}_i \cap \operatorname{supp}_j} (\phi_j)_x(\phi_i)_x \\
&= (\frac{3}{2} + (r+\lambda)\tau)\int_{\operatorname{supp}_{i+1} \cap \operatorname{supp}_{j+1}} \phi_{j+1}\phi_{i+1} - \frac{1}{2}\sigma^2\tau \int_{\operatorname{supp}_{i+1} \cap \operatorname{supp}_{j+1}} \Delta\phi_{j+1}\phi_{i+1} \\
&= \langle S\phi_{j+1},\phi_{i+1}\rangle.
\end{aligned}$$

The last equality but one is obtained by substituting the integral variable $x$ by $x+h$. Thus

$S_n$ is Toeplitz.

Similarly,

$$\begin{aligned}\langle Q\phi_j, \phi_i\rangle &= \iint_{\Omega\times\Omega} f(z-x)\phi_j(z)\phi_i(x)\,\mathrm{d}z\mathrm{d}x \\ &= \int_{\mathrm{supp}_i}\int_{\mathrm{supp}_j} f(z-x)\phi_j(z)\phi_i(x)\,\mathrm{d}z\mathrm{d}x \\ &= \int_{\mathrm{supp}_i}\int_{\mathrm{supp}_j} f\big((z+h)-(x+h)\big)\phi_{j+1}(z+h)\phi_{i+1}(x+h)\,\mathrm{d}z\mathrm{d}x \\ &= \int_{\mathrm{supp}_{i+1}}\int_{\mathrm{supp}_{j+1}} f(z-x)\phi_{j+1}(z)\phi_{i+1}(x)\,\mathrm{d}z\mathrm{d}x \\ &= \langle Q\phi_{j+1}, \phi_{i+1}\rangle\end{aligned}$$

shows that $Q_n$ is Toeplitz. □

As a simple example of the above theorem, we can construct a Toeplitz system by utilizing a linear spline basis, which is defined by

$$\phi_i(x) = \begin{cases} (x-x_{i-1})/h, & x_{i-1} \le x \le x_i, \\ (x_{i+1}-x)/h, & x_i \le x \le x_{i+1}, \\ 0, & \textit{otherwise}, \end{cases}$$

where the partition of $\Omega$ is the same as (3.3). Moreover, it is not hard to see that the corresponding matrix $S_n$ is also tridiagonal, which means we are able to obtain the same structure of coefficient matrix via either FDM or FEM.

### 3.1.2. Analysis on Unpreconditioned Problem

To look into the spectrum of the Toeplitz system $T_n x^p = b^p$ , we would make use of the results related to so-called generating function (see e.g. Section 1.3.2 in Chan and Jin (2007)).

For a given $n$, by the definition in (A.2) with setting $a_k = 0$ $(k \ge n)$, we can obtain a finite series as the generating function of $T_n$ as follows

$$g^n(x) = -\frac{\alpha}{(n+1)m}\left(2\sum_{k=2}^{n-1} e^{-\beta k^2/(n+1)^2}\cos(kx)\right) + 2a_1\cos x + a_0. \tag{3.17}$$

Substitute (3.6) into (3.17), we get

$$\begin{aligned} g^n(x) &= \frac{3}{2} + \frac{(r+\lambda)T}{m} + \frac{\sigma^2 T(n+1)^2}{4\hat{x}^2 m}\big(1-\cos(x)\big) \\ &\quad -\frac{\alpha}{(n+1)m}\left(2\sum_{k=1}^{n-1} e^{-\beta k^2/(n+1)^2}\cos(kx) + 1\right). \end{aligned} \tag{3.18}$$

The bounds of $g^n(x)$ on $[-\pi, \pi]$ are established by

**Lemma 3.1.4.** *Let $g^n(x)$ be the function defined in (3.18) and $\lambda, r > 0$, then*

$$g^n(x) \geq \frac{3}{2} + \frac{rT}{m} + \frac{\lambda T}{m}\left(1 - erf\left(\frac{n-1}{n+1}\sqrt{\beta}\right)\right) - \frac{\alpha}{(n+1)m}, \tag{3.19}$$

$$g^n(x) \leq \frac{3}{2} + \frac{rT}{m} + \frac{\lambda T}{m}\left(1 + erf\left(\frac{n-1}{n+1}\sqrt{\beta}\right)\right) - \frac{\alpha}{(n+1)m} + \frac{\sigma^2 T(n+1)^2}{2\hat{x}^2 m}, \tag{3.20}$$

*where*

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\mathrm{d}t, \qquad x \geq 0.$$

*Proof.* Notice that

$$\begin{aligned}
&\left|\frac{1}{n+1}\sum_{k=1}^{n-1} e^{-\beta k^2/(n+1)^2}\cos(kx)\right| \\
\leq\ & \frac{1}{n+1}\sum_{k=1}^{n-1} e^{-\beta k^2/(n+1)^2}|\cos(kx)| \leq \frac{1}{n+1}\sum_{k=1}^{n-1} e^{-\beta k^2/(n+1)^2} \\
<\ & \int_0^{\frac{n-1}{n+1}} e^{-\beta x^2}\mathrm{d}x = \frac{1}{\sqrt{\beta}}\int_0^{\frac{n-1}{n+1}\sqrt{\beta}} e^{-y^2}\mathrm{d}y = \frac{\sqrt{\pi}}{2\sqrt{\beta}} erf\left(\frac{n-1}{n+1}\sqrt{\beta}\right)
\end{aligned} \tag{3.21}$$

and $\cos(x) \in [-1, 1]$ ($x \in [-\pi, \pi]$), the estimates (3.19) and (3.20) are straightforward. □

**Remark 3.1.5.** By (3.19) and Theorem A.1.2, we can conclude $T_n$ is positive definite when either $n$ or $m$ is sufficiently large.

Let

$$D_n = \begin{pmatrix} d_0 & d_1 & & \\ d_1 & \ddots & \ddots & \\ & \ddots & \ddots & d_1 \\ & & d_1 & d_0 \end{pmatrix}, \qquad E_n = \begin{pmatrix} e_0 & e_1 & \cdots & e_{n-1} \\ e_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & e_1 \\ e_{n-1} & \cdots & e_1 & e_0 \end{pmatrix}, \tag{3.22}$$

where $D_n$ is a tridiagonal matrix and the diagonals of $D_n$ and $E_n$ are given by

$$\begin{aligned}
d_0 &= \frac{3}{2} + \frac{(r+\lambda)T}{m} + \frac{\sigma^2 T(n+1)^2}{4\hat{x}^2 m}, \\
d_1 &= -\frac{\sigma^2 T(n+1)^2}{8\hat{x}^2 m}, \\
e_k &= -\frac{\alpha}{(n+1)m} e^{-\beta k^2/(n+1)^2}, \qquad 0 \leq k \leq n-1.
\end{aligned}$$

We can split $T_n$ by

$$T_n = D_n + E_n. \tag{3.23}$$

For convenience of further discussion, we set

$$w_1 = \frac{3}{2} + \frac{(r+\lambda)T}{m}, \qquad w_2 = \frac{\sigma^2 T(n+1)^2}{8\hat{x}^2 m},$$

thus

$$D_n = \begin{pmatrix} w_1+2w_2 & -w_2 & & \\ -w_2 & \ddots & \ddots & \\ & \ddots & \ddots & -w_2 \\ & & -w_2 & w_1+2w_2 \end{pmatrix} = w_2 \begin{pmatrix} 2+w_1/w_2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2+w_1/w_2 \end{pmatrix}.$$

Since it's trivial that all entries of $E_n$ tends to 0 very fast when $n$ grows, we consider to use $D_n$ as an approximation of $T_n$ and study the spectrum of $T_n$ by analyzing that of $D_n$. To start we need the following lemma.

**Lemma 3.1.6.** *The sequence* $\{\|E_n\|_2 : n \in \mathbb{N}\}$ *is bounded, more precisely,*

$$\|E_n\|_2 < \frac{C}{m}, \tag{3.24}$$

*where* $C = \alpha\left(\frac{1}{2} + \sqrt{\frac{\pi}{\beta}}\right)$ *is a positive constant.*

*Proof.* Notice that the maximum absolute row sum of $E_n$ is attained at the $[\frac{n}{2}]$th row, where $[x] := \max\{i \in \mathbb{Z} : i \leq x\}$ returns the largest integer which does not exceed $x$, thus

$$\begin{aligned} \|E_n\|_\infty &\leq \frac{\alpha}{(n+1)m} + \frac{2\alpha}{m}\sum_{k=1}^{[\frac{n}{2}]} \frac{1}{n+1} e^{-\beta k^2/(n+1)^2} \\ &< \frac{\alpha}{(n+1)m} + \frac{2\alpha}{m}\int_0^{\frac{[\frac{n}{2}]}{n+1}} e^{-\beta x^2}\mathrm{d}x \\ &= \frac{\alpha}{(n+1)m} + \frac{2\alpha}{m\sqrt{\beta}}\int_0^{\frac{[\frac{n}{2}]}{n+1}\sqrt{\beta}} e^{-y^2}\mathrm{d}y \\ &= \frac{\alpha}{m}\left(\frac{1}{n+1} + \sqrt{\frac{\pi}{\beta}} erf\Big(\frac{[\frac{n}{2}]}{n+1}\sqrt{\beta}\Big)\right). \end{aligned}$$

Since $erf(x) \leq 1$,

$$\|E_n\|_\infty \leq \frac{\alpha}{m}\left(\frac{1}{2} + \sqrt{\frac{\pi}{\beta}}\right). \tag{3.25}$$

As $E_n$ is symmetric, it follows

$$\|E_n\|_2 \leq \sqrt{\|E_n\|_1\|E_n\|_\infty} = \|E_n\|_\infty \leq \frac{\alpha}{m}\left(\frac{1}{2} + \sqrt{\frac{\pi}{\beta}}\right).$$

□

Lemma 3.1.6 indeed gives the upper bound and lower bound of eigenvalues of $E_n$. If we know eigenvalues of $D_n$ and treat $E_n$ as a perturbation matrix, then we will have the estimates on the eigenvalues of $T_n$, according to the following Weyl's theorem (see e.g. Horn and Johnson (1985)).

**Theorem 3.1.7** (Weyl's Theorem). *Let $A$ and $E$ be Hermitian matrices and the eigenvalues $\lambda_i(A+E)$, $\lambda_i(A)$ and $\lambda_i(E)$ be arranged in increasing order. Then for $i = 1, 2, \ldots, n$,*

$$\lambda_i(A) + \lambda_1(E) \leq \lambda_i(A+E) \leq \lambda_i(A) + \lambda_n(E).$$

Thus Lemma 3.1.6 indicates that when $m$ is large, $T_n$ and $D_n$ have almost identical spectra. In fact, we can have the explicit expression of the spectrum of $D_n$, with which we obtain the estimates of eigenvalues of $T_n$.

Using the result of Example 7.2.5 in Meyer (2000), the eigenvalues of $D_n$ are given by the following proposition.

**Proposition 3.1.8.** *The eigenvalues of $D_n$ are*

$$w_1 + 2w_2\left(1 + \cos\frac{i\pi}{n+1}\right), \quad i = 1, 2, \ldots, n.$$

Hence we obtain the following corollary by directly applying Weyl's theorem.

**Corollary 3.1.9.** *Let $\{\lambda_i(T_n) : i = 1, 2, \ldots, n\}$ be the eigenvalues of $T_n$, then for all $1 \leq i \leq n$, $\lambda_i(T_n)$ satisfies*

$$|\lambda_i(T_n) - \lambda_i(D_n)| < \frac{C}{m}.$$

The next theorem gives the order of condition number of $T_n$. The notation $f(x) = \Theta(g(x))$ we use means there exist two constants $c_1$ and $c_2$ with $0 < c_1 \leq c_2 < +\infty$ such that $c_1|g(x)| \leq |f(x)| \leq c_2|g(x)|$ when $f(x)$ and $g(x)$ tend to infinity. This notation is preferred to big O notation because it provides double-sided bounds which are appropriate for division operation later on.

**Theorem 3.1.10.** *For sufficiently large $n$ and $m$, we have*

*(i)* $\left|\dfrac{\kappa(T_n) - \kappa(D_n)}{\kappa(D_n)}\right| < \dfrac{2C}{m}$.

*(ii) If* $\dfrac{n^2}{m}$ *is large, i.e.* $\dfrac{n^2}{m} \geq \bar{C}$ *for a given* $\bar{C} > 0$, $\kappa(D_n) = \Theta(\dfrac{n^2}{m})$.

*Proof.* We present the proof briefly as follows:

(i) Let $m$ be sufficiently large such that $C/m < 1/2$, thus $\lambda_{min}(D_n) - C/m > 1$, then

$$\begin{aligned}
\frac{\kappa(T_n) - \kappa(D_n)}{\kappa(D_n)} &\leq \left(\frac{\lambda_{max}(D_n) + C/m}{\lambda_{min}(D_n) - C/m} - \frac{\lambda_{max}(D_n)}{\lambda_{min}(D_n)}\right)\frac{\lambda_{min}(D_n)}{\lambda_{max}(D_n)} \\
&= \frac{C}{m}\frac{\lambda_{max}(D_n) + \lambda_{min}(D_n)}{(\lambda_{min}(D_n) - C/m)\,\lambda_{max}(D_n)} \\
&< \frac{C}{m}\frac{2\lambda_{max}(D_n)}{\lambda_{max}(D_n)} = \frac{2C}{m}.
\end{aligned}$$

Similarly we can show $\dfrac{\kappa(T_n) - \kappa(D_n)}{\kappa(D_n)} > -\dfrac{2C}{m}$.

(ii) We have

$$\begin{aligned}\lambda_{max}(D_n) &= w_1 + 2w_2\left(1+\cos\frac{\pi}{n+1}\right) &&= w_1 + 4w_2\left(1-\sin^2\frac{\pi}{2(n+1)}\right),\\ \lambda_{min}(D_n) &= w_1 + 2w_2\left(1+\cos\frac{n\pi}{n+1}\right) &&= w_1 + 4w_2\sin^2\frac{\pi}{2(n+1)}.\end{aligned}$$

For sufficiently large $n$ and $m$, utilizing the Taylor series of $\sin(x)$ at 0, which is

$$\sin(x) = \sum_{k=0}^{\infty}\frac{(-1)^k}{(2k+1)!}x^{2k+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots,$$

statement (ii) is straightforward.

□

**Remark 3.1.11.** The first statement of Theorem 3.1.10 implies $\lim_{m\to\infty}\kappa(T_n)/\kappa(D_n) = 1$, which allows us to use $\kappa(D_n)$ as good approximation of $\kappa(T_n)$ for sufficiently large $m$. Notice that it holds independently of $n$. When $n^2/m < \bar{C}$, $\kappa(D_n)$ is bounded by a constant. For a given $m$, the condition number of $T_n$ grows rapidly as $n^2$, thus it will lead to ill-conditioned problem when $n$ is enormous. One may raise $m$ to control condition number, however it means the number of time steps is increased thus more linear equations should be solved, thus is also costly.

When the CG method is applied, preconditioning would be definitely necessary because of the nonclustered spectrum of $T_n$ and its large condition number in a high dimensional problem.

### 3.1.3. Circulant Type Preconditioning

A circulant preconditioned CG solver is developed in Sachs and Strauss (2008) to solve the linear equations $T_n x^p = b^p$ with $T_n$ defined by (3.5), where the preconditioner is a circulant matrix (see Definition A.2.1) constructed based on $T_n$. Such a solver is confirmed to be very efficient by the numerical results.

In this section, we are going to establish some spectral properties of the circulant preconditioned system.

#### Circulant Preconditioners

Gilbert Strang first constructs a circulant preconditioner in Strang (1986) by keeping the central diagonals of the Toeplitz matrix $T_n$ and wrapping them around to get circulant structure. We denote this preconditioner by $\widehat{T_n}$, its entries on diagonals are given by

$$s_k = \begin{cases} a_k, & 0 \le k \le \text{floor}(n/2),\\ a_{k-n}, & \text{floor}(n/2) < k < n,\\ s_{n+k}, & 0 < -k < n,\end{cases} \tag{3.26}$$

where $\text{floor}(x) = \max\{i \in \mathbb{Z} : i \leq x\}$. In the case $n = 2l + 1$ $(l \in \mathbb{N})$,

$$\widehat{T}_n = \begin{pmatrix} a_0 & a_{-1} & \cdots & a_{-l} & a_l & \cdots & a_1 \\ a_1 & a_0 & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & a_l \\ a_l & & \ddots & \ddots & \ddots & & a_{-l} \\ a_{-l} & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & a_0 & a_{-1} \\ a_{-1} & \cdots & a_{-l} & a_l & \cdots & a_1 & a_0 \end{pmatrix}.$$

If the generating function $g$ (see (A.3) for definition) of $T_\infty$ is real-valued, meaning that $T_\infty$ is Hermitian, let $T_n$ be the $n \times n$ leading principal submatrix of $T_\infty$, then we have the following result.

**Theorem 3.1.12** (Chan (1989)). *Let $g$ be a positive function in the Wiener class, that means its Fourier coefficients are absolutely summable. Then for large $n$ the circulant matrices $\widehat{T}_n$ and $\widehat{T}_n^{-1}$ are bounded in the $l_2$-norm. In fact, for large $n$, the spectrum $\sigma(\widehat{T}_n)$ of $\widehat{T}_n$ satisfies*

$$\sigma(\widehat{T}_n) \subseteq [g_{min}, g_{max}].$$

When the CG method is valid for the Toeplitz system $T_n x = b$, i.e. $T_n$ is symmetric positive definite, it is trivial that $\widehat{T}_n$ constructed by (3.26) is also symmetric. Furthermore, if $T_n$ has positive generating function, Theorem 3.1.12 guarantees that $\widehat{T}_n$ is positive definite (when $n$ is sufficiently large) hence well-defined as a preconditioner.

To study the behavior of the preconditioned system, we will need the following theorem.

**Theorem 3.1.13** (Chan (1989)). *Let $g$ be a positive function in Wiener class and $\{T_n\}$ be the sequence of Toeplitz matrices generated by $g$. Then for all $\epsilon > 0$, there exists $N > 0$ such that for all $n > N$, at most $2N$ eigenvalues of $\widehat{T}_n - T_n$ have absolute values exceeding $\epsilon$.*

Combining Theorem 3.1.12 and Theorem 3.1.13, and using the fact that

$$\widehat{T}_n^{-1} T_n - I_n = \widehat{T}_n^{-1}(T_n - \widehat{T}_n),$$

we have the following valuable result:

**Corollary 3.1.14** (Chan (1989)). *Let $g$ be a positive function in Wiener class, then for all $\epsilon > 0$, there exists $N > 0$ such that for all $n > N$, at most $2N$ eigenvalues of $\widehat{T}_n^{-1} T_n - I_n$ have absolute values exceeding $\epsilon$.*

The following theorem provides certain optimality of $\widehat{T}_n$.

**Theorem 3.1.15** (Chan (1989)). *Let $T_n$ be an Hermitian Toeplitz matrix. The circulant matrix $\widehat{T}_n$ whose entries are given by (3.26) minimizes $\|C_n - T_n\|_1 = \|C_n - T_n\|_\infty$ over all possible Hermitian circulant matrices $C_n$.*

Besides Strang's preconditioner, there are some other well-known circulant preconditioners, which also cluster most eigenvalues of the preconditioned matrices around 1. These preconditioners have different optimal properties.

In Chan (1988) a circulant preconditioner $c(A_n)$, known as *optimal circulant preconditioner*, is proposed by Tony Chan. This preconditioner can be defined for any square matrix $A_n$, in addition, $c(A_n)$ inherits self-adjointness and positive definiteness from $A_n$.

**Theorem 3.1.16** (Chan et al. (1991a))**.** *Let $A_n \in \mathbb{C}^{n\times n}$ and $c(A_n)$ be the minimizer of $\|C_n - A_n\|_F$ over all circulant $n \times n$ matrices $C_n$. Then $c(A_n)$ is uniquely determined by $A_n$. Moreover,*

$$c(A_n) = F_n^* \delta(F_n A_n F_n^*) F_n, \tag{3.27}$$

*where $F_n$ is the Fourier matrix defined in (A.4) and $\delta(\cdot)$ denotes the diagonal matrix with the same diagonal as the matrix in the argument.*

Let $\bar{T}_n$ be a Hermitian Toeplitz matrix, $a_{i-j} = (\bar{T}_n)_{ij}$, then the diagonals of $c(\bar{T}_n)$ are given by

$$c_k = \begin{cases} \big((n-k)a_k + k a_{k-n}\big)/n, & 0 \le k < n, \\ c_{n+k}, & -n < k < 0, \end{cases} \tag{3.28}$$

where $a_{-n}$ is taken to be 0 when $k = 0$. Thus optimal circulant preconditioner can be constructed in $O(n)$ operations for a Toeplitz matrix.

The following theorem gives the spectral property of $c(\bar{T}_n)$ preconditioned matrix, similar to Corollary 3.1.14. It can be found in Chan and Jin (2007).

**Theorem 3.1.17.** *Let $g$ be a positive function in Wiener class, then for all $\epsilon > 0$, there exist $M(\epsilon)$ and $N(\epsilon) > 0$ such that for all $n > N(\epsilon)$, at most $M(\epsilon)$ eigenvalues of $(c(A_n))^{-1}A_n - I_n$ have absolute values larger than $\epsilon$.*

Tismenetsky (1991) and Tyrtyshnikov (1992) independently propose a so-called *super-optimal preconditioner*, denoted by $t(A_n)$, which also has the property of inheriting self-adjointness and positive definiteness from $A_n$, just like optimal circulant preconditioner does.

**Theorem 3.1.18** (Chan et al. (1991a), Chan et al. (1991b))**.** *Let $A_n \in \mathbb{C}^{n\times n}$ be positive definite. Let $t(A_n)$ be the super-optimal circulant preconditioner for $A_n$, defined by*

$$\|I - t(A_n)^{-1}A_n\|_F = \min \|I - C_n^{-1}A_n\|_F,$$

*where the minimum is taken over all $n \times n$ nonsingular circulant matrices $C_n$. Then*

$$t(A_n) = \big(c(A_n A_n^*)\big) c(A_n^*)^{-1}, \tag{3.29}$$

*where $c(A)$ denotes optimal circulant preconditioner (T. Chan's preconditioner) constructed from $A$.*

Notice that $\widehat{C}_n = t(A_n)^{-1} = (c(A_n^*)c(A_n A_n^*))^{-1}$ can be used for preconditioning instead of $t(A_n)$ to avoid the evaluation of inverse.

Similar to optimal circulant preconditioner, we have the following result, see e.g. Chan and Jin (2007).

**Theorem 3.1.19.** *Let $g$ be a positive function in Wiener class, then for all $\epsilon > 0$, there exist $M(\epsilon)$ and $(\epsilon)N > 0$ such that for all $n > N(\epsilon)$, at most $M(\epsilon)$ eigenvalues of $(t(A_n))^{-1}A_n - I_n$ have absolute values larger than $\epsilon$.*

When $T_n$ is a Hermitian Toeplitz matrix, Chan (1989) proposes a preconditioner $R_n$ whose diagonals are given by

$$r_k = \begin{cases} a_{k-n} + a_k, & 0 \le k < n, \\ \bar{r}_{-k}, & 0 < -k < n, \end{cases}$$

where $a_{-n}$ is set to be 0 when $k = 0$. It is also said in Chan (1989) that $\lim_{n\to\infty} \|\widehat{T}_n - R_n\|_2 = 0$, thus we can expect similar performance of $\widehat{T}_n$ and $R_n$ for large $n$.

Huckle (1993) proposes a preconditioner $h(T_n)$ which is optimal in the sense of minimizing

$$\|I - C_n^{-1/2} T_n C_n^{-1/2}\|_F.$$

It is shown that this preconditioner and optimal circulant preconditioner are asymptotically equivalent with respect to the spectral norm. The eigenvalues of $h(T_n)^{-1}T_n$ are clustered around 1 for large $n$.

Besides circulant preconditioners, there are also skew circulant preconditioners which have similar properties, see e.g. Huckle (1992).

To sum up, the above mentioned circulant preconditioners all approximate the original Toeplitz coefficient matrix in some sense, and the preconditioned matrices have most eigenvalues clustered around 1.

The cost of applying a circulant preconditioner is fairly low. The preconditioning step in preconditioned CG algorithm, i.e. solving a linear equation with the circulant preconditioner as its coefficient matrix, takes only $O(n \log n)$ operations, more precisely, three FFTs and one vector multiply, see Section A.2.

Because Strang's preconditioner is simple to construct (almost free) and it outperforms other circulant preconditioners in solving the PIDE, the main numerical experiments are confined to Strang preconditioned systems in Sachs and Strauss (2008). For the same reason, the preconditioned matrices are only analyzed based on Strang's preconditioner in this section.

### Spectrum of Strang Preconditioned Matrix

Corollary 3.1.14 states that for sufficiently large $n$, Strang preconditioned systems have most of their eigenvalues clustered around 1, except a few so-called *outliers*. The number of these outliers for our case is the concern.

Notice that in Corollary 3.1.14 the result only holds for a given generating function $g$. In other words, changing the subscript $n$ of $T_n$ extracts different leading principal submatrices of $T_\infty$ determined by $g$ and Corollary 3.1.14 applies to all these matrices.

The situation is different in our case. When we change the mesh size for discretization, the entries of $T_n$ change with respect to $n$, thus the generating function also changes. We are not able to apply Corollary 3.1.14 for all discretization settings. Nevertheless, following the idea of Chan (1989), we can prove a result similar to Corollary 3.1.14 for $\widehat{T}_n^{-1}T_n$ with varying $n$.

The following theorem is useful for diagonally dominant matrices, we present it in the first place for later use.

**Theorem 3.1.20** (Varah (1975)). *Assume $A = (a_{ij})_{i,j=1,\dots,n}$, the following statements hold.*

*(i) If $A$ is diagonally dominant by rows and set $\xi_1 = \min_k \{|a_{kk}| - \sum_{j\neq k} |a_{kj}|\}$, we have $\|A^{-1}\|_\infty < 1/\xi_1$.*

*(ii) If $A$ is diagonally dominant by columns and set $\xi_2 = \min_k \{|a_{kk}| - \sum_{i\neq k} |a_{ik}|\}$, then $\|A^{-1}\|_1 < 1/\xi_2$.*

*(iii) If $A$ is diagonally dominant both by rows and by columns, $\|A^{-1}\|_2^{-1} < \sqrt{\xi_1\xi_2}$.*

**Lemma 3.1.21.** *If Toeplitz matrix $T_n$ is defined by (3.6), $\widehat{T}_n$ is the corresponding Strang's preconditioner, then for sufficiently large $m$, $\|T_n^{-1}\|_\infty < 1$ and $\|\widehat{T}_n^{-1}\|_\infty < 1$.*

*Proof.* It is not hard to see

$$\xi := \min_k \{|a_{kk}| - \sum_{j\neq k} |a_{kj}|\} > \frac{3}{2} + \frac{(r+\lambda)T}{m} - \|E_n\|_\infty > \frac{3}{2} + \frac{(r+\lambda)T}{m} - \frac{C}{m},$$

Thus for sufficiently large $m$, $\xi > 1$. By Theorem 3.1.20, this yields $\|T_n^{-1}\|_\infty < 1$. Similarly we obtain $\|\widehat{T}_n^{-1}\|_\infty < 1$. □

Now we come to the main result of this section.

**Theorem 3.1.22.** *Let $m$ be sufficiently large, $\forall\epsilon > 0$, there exists $N(\epsilon) > 0$ such that for all $n \geq N(\epsilon)$, at most $2N(\epsilon)$ eigenvalues of $\widehat{T}_n^{-1}T_n$ lie outside the interval $(1-\epsilon, 1+\epsilon)$. Here $N(\epsilon)$ is given by*

$$N(\epsilon) = \begin{cases} 1, & if \ \dfrac{\sqrt{\pi}\alpha}{2\sqrt{\beta}m} < \epsilon, \\ \operatorname{ceil}\left(\dfrac{n+1}{\sqrt{\beta}} erf^{-1}\left(1 - \dfrac{2\epsilon\sqrt{\beta}m}{\sqrt{\pi}\alpha}\right)\right), & otherwise, \end{cases} \tag{3.30}$$

*where* $\operatorname{ceil}(x) = \min\{i \in \mathbb{Z} : i \geq x\}$.

*Proof.* (i) First of all, we show $\|\widehat{T}_n^{-1}\|_2$ is bounded. Since $\widehat{T}_n$ is symmetric, $\|\widehat{T}_n^{-1}\|_1 = \|\widehat{T}_n^{-1}\|_\infty$. By Lemma 3.1.21, for sufficiently large $m$ we have the uniform bound

$$\|\widehat{T}_n^{-1}\|_2 \leq \sqrt{\|\widehat{T}_n^{-1}\|_1 \|\widehat{T}_n^{-1}\|_\infty} < 1. \tag{3.31}$$

(ii) We also need this statement for latter proof: $\forall\epsilon > 0$, there exists a $N(\epsilon) > 0$, for all $N \geq N(\epsilon)$

$$\sum_{k=N+1}^{n-N-1} |a_k| < \epsilon.$$

Actually, for all $N \geq 1$,

$$\begin{aligned}
\sum_{k=N+1}^{n-N-1} |a_k| &= \sum_{k=N+1}^{n-N-1} \left| -\frac{\alpha}{(n+1)m} e^{-\beta k^2/(n+1)^2} \right| \\
&= \frac{\alpha}{m} \sum_{k=N+1}^{n-N-1} \frac{1}{n+1} e^{-\beta k^2/(n+1)^2} \\
&< \frac{\alpha}{m} \int_{\frac{N}{n+1}}^{\frac{n-N-1}{n+1}} e^{-\beta x^2} \mathrm{d}x < \frac{\alpha}{\sqrt{\beta} m} \int_{\frac{N\sqrt{\beta}}{n+1}}^{\sqrt{\beta}} e^{-z^2} \mathrm{d}z \\
&= \frac{\sqrt{\pi}\alpha}{2\sqrt{\beta}m} \left( erf(\sqrt{\beta}) - erf\left(\frac{N\sqrt{\beta}}{n+1}\right)\right) \\
&< \frac{\sqrt{\pi}\alpha}{2\sqrt{\beta}m} \left( 1 - erf\left(\frac{N\sqrt{\beta}}{n+1}\right)\right).
\end{aligned}$$

If

$$\frac{\sqrt{\pi}\alpha}{2\sqrt{\beta}m} \left( 1 - erf\left(\frac{N\sqrt{\beta}}{n+1}\right)\right) < \epsilon, \tag{3.32}$$

holds, it follows the above statement. In fact, if $\dfrac{\sqrt{\pi}\alpha}{2\sqrt{\beta}m} < \epsilon$, then (3.32) always holds. In such a case we can set $N(\epsilon) = 1$. Otherwise, we can get (3.32) satisfied by setting $N(\epsilon) = \text{ceil}\left(\dfrac{n+1}{\sqrt{\beta}} erf^{-1}\left(1 - \dfrac{2\epsilon\sqrt{\beta}m}{\sqrt{\pi}\alpha}\right)\right)$. This gives (3.30).

(iii) (cf. Chan (1989)) Let $B_n = \widehat{T}_n - T_n$, $U_n$ be the matrix obtained from $B_n$ by replacing the $(n - N(\epsilon)) \times (n - N(\epsilon))$ leading principal submatrix of $B_n$ by zero matrix. Hence $\text{rank}(U_n) \leq 2N(\epsilon)$. Let $W_n = B_n - U_n$. It is not hard to find out that the maximum absolute column sum of $W_n$ is attained at the first column (or the $(n - N(\epsilon) - 1)$th column) by construction. If $N(\epsilon)$ is given by (3.30), according to (ii), we have (3.32), hence

$$\|W_n\|_1 \leq \sum_{k=N+1}^{n-N-1} |a_k| < \epsilon.$$

Since $W_n$ is symmetric by construction, $\|W_n\|_\infty = \|W_n\|_1 < \epsilon$. Thus

$$\|W_n\|_2 \leq \sqrt{\|W_n\|_1 \|W_n\|_\infty} < \epsilon.$$

Hence the spectrum of $W_n$ satisfies $\sigma(W_n) \subseteq (-\epsilon, \epsilon)$. Note that

$$\widehat{T}_n^{-1} T_n - I_n = \widehat{T}_n^{-1}(T_n - \widehat{T}_n) = \widehat{T}_n^{-1}(W_n - U_n) = \widehat{T}_n^{-1} W_n + \widehat{T}_n^{-1} U_n,$$

where $\|\widehat{T}_n^{-1} W_n\|_2 \leq \|\widehat{T}_n^{-1}\|_2 \|W_n\|_2 < \|W_n\|_2 < \epsilon$ (by (3.31)) and $\text{rank}(U_n) \leq 2N(\epsilon)$ leads to $\text{rank}(\widehat{T}_n^{-1} U_n) \leq \min\{n, \text{rank}(U_n)\} \leq 2N(\epsilon)$. Applying Weyl's theorem here, we conclude that at most $2N(\epsilon)$ eigenvalues of the $\widehat{T}_n^{-1} T_n$ lie outside the interval $(1-\epsilon, 1+\epsilon)$.

□

**Remark 3.1.23.** For a given matrix $T_n$, $N(\epsilon)$ given by (3.30) inspires us with the possibility of increasing $m$, i.e. refining time discretization, so as to limit the number of outlying eigenvalues to be small (in particular, equals 2). According to Theorem 2.1.4, the corresponding preconditioned CG method will take very few iterations in each time step.

Specially, when $m$ is sufficiently large such that $N(\epsilon) = 1$, the estimate in Theorem 2.1.4 is actually independent of $n$ thus mesh independent convergence of corresponding CG solver is obtianed.

On the other hand, if $m$ and small $\epsilon$ are fixed, by (3.30) we can tell that $N(\epsilon)$ will increase by order $O(n)$ as $n$ grows. Hence $2N(\epsilon) + 1$ can be taken as the upper bound of the CG method's iteration number.

### Condition Number of Strang Preconditioned System

In the numerical experiments, we observe that only a small part of Strang's preconditioner plays a major role. We believe this phenomenon is caused by the fact that the entries of $T_n$ except those of the three central diagonals vanish fast when the matrix size tends to infinity. Following this lead, we raise the following analysis on the condition number of Strang preconditioned system.

Similarly as (3.23) we can split Strang's preconditioner by the following statement, which is trivial by the construction of Strang's preconditioner.

**Proposition 3.1.24.** *Let $\widehat{D}_n$ and $\widehat{E}_n$ be Strang's preconditioner for $D_n$ and $E_n$ respectively, $\widehat{T}_n$ is Strang's preconditioner for $T_n$, then*

$$\widehat{T}_n = \widehat{D}_n + \widehat{E}_n.$$

Specially, to obtain the circulant structure, $\widehat{D}_n$ is obtained from $D_n$ by replacing $(D_n)_{n1}$ and $(D_n)_{1n}$ with $d_1$, i.e.

$$\widehat{D}_n = D_n + \widetilde{E}_n = \begin{pmatrix} d_0 & d_1 & & d_1 \\ d_1 & \ddots & \ddots & \\ & \ddots & \ddots & d_1 \\ d_1 & & d_1 & d_0 \end{pmatrix}, \qquad \widetilde{E}_n = \begin{pmatrix} 0 & \cdots & 0 & d_1 \\ \vdots & \ddots & & 0 \\ 0 & & \ddots & \vdots \\ d_1 & 0 & \cdots & 0 \end{pmatrix}.$$

Exploiting symmetry and tridiagonal Toeplitz structure, explicit inverse of $D_n$, given by its entries, can be obtained by modifying Theorem 2.8 in Meurant (1992). We present it by the following theorem.

**Theorem 3.1.25.** *Let $C_n = (c_{ij})_{i,j=1,\ldots,n}$ be the inverse of $D_n$ and*

$$\gamma_+ = \frac{w_1 + 2w_2 + \sqrt{w_1^2 + 4w_1w_2}}{2w_2}, \qquad \gamma_- = \frac{w_1 + 2w_2 - \sqrt{w_1^2 + 4w_1w_2}}{2w_2}, \tag{3.33}$$

*then*

$$c_{ij} = \begin{cases} \dfrac{(\gamma_+^i - \gamma_-^i)(\gamma_+^{n+1-j} - \gamma_-^{n+1-j})}{w_2(\gamma_+ - \gamma_-)(\gamma_+^{n+1} - \gamma_-^{n+1})}, & \text{if } j \geq i, \\ c_{ji}, & \text{otherwise.} \end{cases}$$

*Notice that $\gamma_- = 1/\gamma_+$. If we set $\psi = \log \gamma_+$, then $c_{ij}$ can also be written into*

$$c_{ij} = \begin{cases} \dfrac{\sinh(i\psi)\sinh((n+1-j)\psi)}{w_2 \sinh(\psi)\sinh((n+1)\psi)}, & \textit{if } j \geq i, \\ c_{ji}, & \textit{otherwise.} \end{cases}$$

*Specially,*

$$c_{11} = c_{nn} = \frac{\gamma_+^n - \gamma_-^n}{w_2(\gamma_+^{n+1} - \gamma_-^{n+1})} = \frac{\sinh(n\psi)}{w_2 \sinh((n+1)\psi)},$$
$$c_{1n} = c_{n1} = \frac{\gamma_+ - \gamma_-}{w_2(\gamma_+^{n+1} - \gamma_-^{n+1})} = \frac{\sinh(\psi)}{w_2 \sinh((n+1)\psi)}.$$

**Remark 3.1.26.** Recalling that

$$w_1 = \frac{3}{2} + \frac{(r+\lambda)T}{m}, \qquad w_2 = \frac{\sigma^2 T(n+1)^2}{8\hat{x}^2 m},$$

$\gamma_+$ and $\gamma_-$ defined by (3.33) are both dependent on $m$ and $n$. Obviously, $\gamma_+ > 1 > \gamma_- > 0$. Thus we have $c_{11} > c_{1n} > 0$.

The following theorem describes the spectrum of $\widehat{D}_n^{-1} D_n$.

**Theorem 3.1.27.** *Let $\lambda_1(\widehat{D}_n^{-1}D_n) \leq \lambda_2(\widehat{D}_n^{-1}D_n) \leq \cdots \leq \lambda_{n-1}(\widehat{D}_n^{-1}D_n) \leq \lambda_n(\widehat{D}_n^{-1}D_n)$ be the eigenvalues of $\widehat{D}_n^{-1}D_n$ and $C_n = (c_{ij})_{i,j=1,\dots,n}$ be the inverse of $D_n$, when $n^2/m$ is large, we have*

$$\lambda_i(\widehat{D}_n^{-1}D_n) = \begin{cases} \dfrac{1}{1 - w_2(c_{1n} - c_{11})} & , \ i = 1, \\ 1 & , \ i = 2, \cdots, n-1, \\ \dfrac{1}{1 - w_2(c_{1n} + c_{11})} & , \ i = n. \end{cases} \tag{3.34}$$

*For convenience, we denote $\lambda_1(\widehat{D}_n^{-1}D_n)$ and $\lambda_n(\widehat{D}_n^{-1}D_n)$ by $\lambda_n^-$ and $\lambda_n^+$ respectively. We have the following estimates for these two extreme eigenvalues of $\widehat{D}_n^{-1}D_n$:*

$$\lambda_n^- \in [\frac{1}{2}, 1], \quad \lambda_n^+ = \Theta(\frac{n}{\sqrt{m}}). \tag{3.35}$$

*Proof.* $D_n^{-1}\widehat{D}_n = D_n^{-1}(D_n + \widetilde{E}_n) = I_n + D_n^{-1}\widetilde{E}_n$. Since

$$D_n^{-1}\widetilde{E}_n = -w_2 \begin{pmatrix} c_{1n} & 0 & \cdots & 0 & c_{11} \\ c_{2n} & 0 & \cdots & 0 & c_{21} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{nn} & 0 & \cdots & 0 & c_{n1} \end{pmatrix},$$

the eigenvalues of matrix $-\dfrac{1}{w_2} D_n^{-1}\widetilde{E}$ are the $n$ zeros of equation

$$\lambda^{n-2}\big((\lambda - c_{1n})(\lambda - c_{n1}) - c_{11}c_{nn}\big) = \lambda^{n-2}\big((\lambda - c_{1n})^2 - c_{11}^2\big) = 0,$$

hence they are $c_{1n} - c_{11}$, $0$, $c_{1n} + c_{11}$ respectively. Thus the eigenvalues of $D_n^{-1}\widehat{D}_n$ are $1 - w_2(c_{1n} - c_{11})$, $1$ and $1 - w_2(c_{1n} + c_{11})$. By $D_n^{-1}\widehat{D}_n = (\widehat{D}_n^{-1}D_n)^{-1}$ , we obtain (3.34).

We can prove (3.35) by utilizing Theorem 3.1.25. Substituting $c_{1n}$ and $c_{11}$ by the explicit expressions, we get

$$\begin{aligned}
w_2(c_{1n} - c_{11}) &= \frac{\sinh(\psi)}{\sinh((n+1)\psi)} - \frac{\sinh(n\psi)}{\sinh((n+1)\psi)} \\
&= -\frac{2\sinh(\frac{n-1}{2}\psi)\cosh(\frac{n+1}{2}\psi)}{2\sinh(\frac{n+1}{2}\psi)\cosh(\frac{n+1}{2}\psi)} = -\frac{\sinh(\frac{n-1}{2}\psi)}{\sinh(\frac{n+1}{2}\psi)} \\
&= -\frac{\gamma_+^{\frac{n-1}{2}} - \gamma_+^{-\frac{n-1}{2}}}{\gamma_+^{\frac{n+1}{2}} - \gamma_+^{-\frac{n+1}{2}}} = -\frac{\gamma_+^n - \gamma_+}{\gamma_+^{n+1} - 1}.
\end{aligned}$$

Since $\gamma_+ > 1 > \gamma_- > 0$, we get $-1 \le w_2(c_{1n} - c_{11}) \le 0$. Thus $\lambda_n^- \in [\frac{1}{2}, 1]$.

Similarly,

$$\begin{aligned}
w_2(c_{1n} + c_{11}) &= \frac{\sinh(\psi)}{\sinh((n+1)\psi)} + \frac{\sinh(n\psi)}{\sinh((n+1)\psi)} \\
&= \frac{2\sinh(\frac{n+1}{2}\psi)\cosh(\frac{n-1}{2}\psi)}{2\sinh(\frac{n+1}{2}\psi)\cosh(\frac{n+1}{2}\psi)} = \frac{\cosh(\frac{n-1}{2}\psi)}{\cosh(\frac{n+1}{2}\psi)} \\
&= \frac{\gamma_+^{\frac{n-1}{2}} + \gamma_+^{-\frac{n-1}{2}}}{\gamma_+^{\frac{n+1}{2}} + \gamma_+^{-\frac{n+1}{2}}} = \frac{\gamma_+^n + \gamma_+}{\gamma_+^{n+1} + 1}.
\end{aligned}$$

Thus we have

$$\lambda_n^+ = \frac{\gamma_+^{n+1} + 1}{(\gamma_+^{n+1} + 1) - (\gamma_+^n + \gamma_+)} = \frac{\gamma_+^{n+1} + 1}{(\gamma_+^n - 1)(\gamma_+ - 1)} \tag{3.36}$$

For the purpose of estimating the order of $\lambda_n^+$, we first show that for sufficiently large $n$, $\dfrac{\gamma_+^{n+1} + 1}{\gamma_+^n - 1}$ is bounded by positive constants, which means this term does not influence the order of $\lambda_n^+$ and leaves us only the term $\gamma_+ - 1$ in (3.36) to estimate. In fact, we only need to prove that $\dfrac{\gamma_+^{n+1} + 1}{\gamma_+^n - 1}$ has a positive limit when $n$ tends to infinity.

Let

$$\omega = \frac{w_1}{w_2} = \frac{8\hat{x}^2\left(\frac{3}{2}m + (r+\lambda)T\right)}{\sigma^2 T(n+1)^2} = \frac{\nu(m)}{(n+1)^2},$$

where

$$\nu(m) = \frac{8\hat{x}^2\left(\frac{3}{2}m + (r+\lambda)T\right)}{\sigma^2 T},$$

then

$$\gamma_+ = 1 + \frac{1}{2}\omega + \frac{1}{2}\sqrt{\omega^2 + 4\omega}.$$

Since $\lim_{n\to\infty} \omega = 0$ and $\lim_{n\to\infty} (\gamma_+ - 1) = 0$, we have

$$\lim_{n\to\infty} \big(1 + (\gamma_+ - 1)\big)^{\frac{1}{\gamma_+ - 1}} = e.$$

Moreover,

$$\lim_{n\to\infty} (\gamma_+ - 1)n = \lim_{n\to\infty} \frac{1}{2}\left(\frac{\nu(m)n}{(n+1)^2} + \sqrt{\left(\frac{\nu(m)n}{(n+1)^2}\right)^2 + \frac{4\nu(m)n^2}{(n+1)^2}}\right) = \sqrt{\nu(m)}.$$

Hence

$$\lim_{n\to\infty} \gamma_+^n = \lim_{n\to\infty} \left(\big(1 + (\gamma_+ - 1)\big)^{\frac{1}{\gamma_+ - 1}}\right)^{(\gamma_+ - 1)n} = e^{\sqrt{\nu(m)}} > 1.$$

Obviously, we also have

$$\lim_{n\to\infty} \gamma_+^{n+1} = e^{\sqrt{\nu(m)}}.$$

As a result,

$$\lim_{n\to\infty} \frac{\gamma_+^{n+1} + 1}{\gamma_+^n - 1} = \frac{e^{\sqrt{\nu(m)}} + 1}{e^{\sqrt{\nu(m)}} - 1} = 1 + \frac{2}{e^{\sqrt{\nu(m)}} - 1}$$

It is not hard to see that

$$\frac{1}{\gamma_+ - 1} = \Theta(\frac{n}{\sqrt{m}}),$$

thus we conclude

$$\lambda_n^+ = \Theta(\frac{n}{\sqrt{m}}).$$

□

It directly follows

**Corollary 3.1.28.** *For large $n^2/m$,*

$$\kappa(\widehat{D}_n^{-\frac{1}{2}} D_n \widehat{D}_n^{-\frac{1}{2}}) = \frac{\lambda_n(\widehat{D}_n^{-1} D_n)}{\lambda_1(\widehat{D}_n^{-1} D_n)} = \Theta(\frac{n}{\sqrt{m}}).$$

We are going to show that $\kappa(\widehat{D}_n^{-\frac{1}{2}} D_n \widehat{D}_n^{-\frac{1}{2}})$ indeed approximates $\kappa(\widehat{T}_n^{-\frac{1}{2}} T_n \widehat{T}_n^{-\frac{1}{2}})$ when $m$ is sufficiently large. A similar result like the first statement in Theorem 3.1.10, which connects $\kappa(D_n)$ and $\kappa(T_n)$, can be proved. We first show some estimates heavily rely on $\|\cdot\|_\infty$ by the following lemmas.

**Lemma 3.1.29.** *The sequence $\{\|\widehat{E}_n\|_\infty : n \in \mathbb{N}\}$ is bounded, more precisely,*

$$\|\widehat{E}_n\|_\infty = \|E_n\|_\infty < \frac{C}{m}, \tag{3.37}$$

*where* $C = \alpha\left(\frac{1}{2}+\sqrt{\frac{\pi}{\beta}}\right)$ *is a positive constant.*

*Proof.* Notice that each absolute row sum of the circulant matrix $\widehat{E}_n$ is the same as the $[\frac{n}{2}]$th absolute row sum of $E_n$, which equals $\|E_n\|_\infty$, we have $\|\widehat{E}_n\|_\infty = \|E_n\|_\infty$. Thus the inequality (3.25) yields the result. □

The following lemma provides bounds for the spectrum of $D_n^{-1}\widehat{D}_n$.

**Lemma 3.1.30.**
*(i)* $\|D_n^{-1}\widehat{D}_n\|_\infty \leq 2$ *holds for all* $n$.
*(ii) If* $\frac{n^2}{m}$ *is large,* $\|\widehat{D}_n^{-1}D_n\|_\infty = \Theta(\frac{n}{\sqrt{m}})$ .

*Proof.* We already know that

$$D_n^{-1}\widehat{D}_n = I_n - w_2\begin{pmatrix} c_{1n} & 0 & \cdots & 0 & c_{11} \\ c_{2n} & 0 & \cdots & 0 & c_{21} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{nn} & 0 & \cdots & 0 & c_{n1} \end{pmatrix}.$$

To show (i), we have

$$\begin{aligned} \|D_n^{-1}\widehat{D}_n\|_\infty &\leq 1 + w_2 \max_i\{c_{in}+c_{i1}\} \\ &= 1+\max_i\left\{\frac{\sinh(i\psi)+\sinh\left((n+1-i)\psi\right)}{\sinh((n+1)\psi)}\right\} \\ &= 1+\max_i\left\{\frac{\cosh\left((i-\frac{n+1}{2})\psi\right)}{\cosh(\frac{n+1}{2}\psi)}\right\} \\ &= 1+\frac{\cosh(\frac{n-1}{2}\psi)}{\cosh(\frac{n+1}{2}\psi)} \leq 2. \end{aligned}$$

In the last inequality, we use the fact that $\psi = \log\gamma_+ > 0$ and $\cosh(x)$ is an monotonically increasing positive function when $x > 0$.

Notice that

$$\begin{aligned} \min_i\{1-w_2(c_{in}+c_{i1})\} &= \min_i\{1-w_2(c_{in}+c_{1i})\} \\ &= \min_i\left\{1-\frac{\sinh(i\psi)+\sinh\left((n+1-i)\psi\right)}{\sinh((n+1)\psi)}\right\} \\ &= 1-\max_i\left\{\frac{\cosh\left((i-\frac{n+1}{2})\psi\right)}{\cosh(\frac{n+1}{2}\psi)}\right\} \end{aligned}$$

$$\begin{aligned} &= 1 - \frac{\cosh(\frac{n-1}{2}\psi)}{\cosh(\frac{n+1}{2}\psi)} \\ &= 1 - w_2(c_{1n} + c_{11}), \end{aligned}$$

by Theorem 3.1.20 we have

$$\|\widehat{D}_n^{-1} D_n\|_\infty \leq \frac{1}{1 - w_2(c_{1n} + c_{11})} = \lambda_n^+.$$

It follows (ii).

□

**Lemma 3.1.31.** *For sufficiently large $m$ and large $n^2/m$,*

$$\lambda_n^- - c_1(m) \leq \lambda_{min}(\widehat{T}_n^{-1} T_n) \leq \lambda_n^- + c_2(m),$$

*where* $c_1(m) = \dfrac{3C(\lambda_n^-)^2}{m + 3C\lambda_n^-}$ *and* $c_2(m) = \dfrac{3C(\lambda_n^-)^2}{m - 3C\lambda_n^-}$.

*Proof.*

$$\begin{aligned} \|T_n^{-1}\widehat{T}_n - D_n^{-1}\widehat{D}_n\|_\infty &= \|T_n^{-1}\widehat{D}_n + T_n^{-1}\widehat{E}_n - D_n^{-1}\widehat{D}_n\|_\infty \\ &\leq \|T_n^{-1}(D_n - T_n)D_n^{-1}\widehat{D}_n\|_\infty + \|T_n^{-1}\widehat{E}_n\|_\infty \\ &\leq \|T_n^{-1}\|_\infty \|E_n\|_\infty \|D_n^{-1}\widehat{D}_n\|_\infty + \|T_n^{-1}\|_\infty \|\widehat{E}_n\|_\infty \\ &\leq \frac{3C}{m}. \end{aligned}$$

In the last inequality Lemma 3.1.21, Lemma 3.1.29, and Lemma 3.1.30 are used. Thus by Weyl's theorem, we get

$$\lambda_{max}(D_n^{-1}\widehat{D}_n) - \frac{3C}{m} \leq \lambda_{max}(T_n^{-1}\widehat{T}_n) \leq \lambda_{max}(D_n^{-1}\widehat{D}_n) + \frac{3C}{m},$$

using the notation $\lambda_n^-$, this is equivalent to

$$\frac{1}{\lambda_n^-} - \frac{3C}{m} \leq \lambda_{max}(T_n^{-1}\widehat{T}_n) \leq \frac{1}{\lambda_n^-} + \frac{3C}{m}.$$

We use this inequality to estimate the bounds for $\lambda_{min}(\widehat{T}_n^{-1}T_n)$ as follows:

$$\lambda_{min}(\widehat{T}_n^{-1}T_n) = \frac{1}{\lambda_{max}(T_n^{-1}\widehat{T}_n)} \leq \frac{1}{\frac{1}{\lambda_n^-} - \frac{3C}{m}} = \lambda_n^- + \frac{3C(\lambda_n^-)^2}{m - 3C\lambda_n^-}.$$

The lower bound can be computed in a similar way. □

Using the same approach, we can also find out the bounds for $\lambda_{max}(\widehat{T}_n^{-1}T_n)$.

**Lemma 3.1.32.** *For sufficiently large $m$ and large $n^2/m$,*

$$\lambda_n^+ - c_3(n,m) \le \lambda_{max}(\widehat{T}_n^{-1}T_n) \le \lambda_n^+ + c_3(n,m),$$

*where* $c_3(n,m) = \dfrac{C_1 n}{m^{\frac{3}{2}}} + \dfrac{C}{m}$ *with $C_1 > 0$ constant.*

*Proof.*

$$\begin{aligned}\|\widehat{T}_n^{-1}T_n - \widehat{D}_n^{-1}D_n\|_\infty &= \|\widehat{T}_n^{-1}D_n + \widehat{T}_n^{-1}E_n - \widehat{D}_n^{-1}D_n\|_\infty \\ &\le \|\widehat{T}_n^{-1}(\widehat{D}_n - \widehat{T}_n)\widehat{D}_n^{-1}D_n\|_\infty + \|\widehat{T}_n^{-1}E_n\|_\infty \\ &\le \|\widehat{T}_n^{-1}\|_\infty\|\widehat{E}_n\|_\infty\|\widehat{D}_n^{-1}D_n\|_\infty + \|T_n^{-1}\|_\infty\|\widehat{E}_n\|_\infty.\end{aligned}$$

Applying all the estimates and Weyl's Theorem, the statement is straightforward. □

**Theorem 3.1.33.** *For sufficiently large $m$ and large $n^2/m$,*

$$\left|\frac{\kappa(\widehat{T}_n^{-\frac{1}{2}}T_n\widehat{T}_n^{-\frac{1}{2}}) - \kappa(\widehat{D}_n^{-\frac{1}{2}}D_n\widehat{D}_n^{-\frac{1}{2}})}{\kappa(\widehat{D}_n^{-\frac{1}{2}}D_n\widehat{D}_n^{-\frac{1}{2}})}\right| \le \frac{C_2}{\sqrt{m}}$$

*with $C_2 > 0$ constant.*

*Proof.*

$$\begin{aligned} r &:= \frac{\kappa(\widehat{T}_n^{-\frac{1}{2}}T_n\widehat{T}_n^{-\frac{1}{2}}) - \kappa(\widehat{D}_n^{-\frac{1}{2}}D_n\widehat{D}_n^{-\frac{1}{2}})}{\kappa(\widehat{D}_n^{-\frac{1}{2}}D_n\widehat{D}_n^{-\frac{1}{2}})} \\ &= \left(\frac{\lambda_{max}(\widehat{T}_n^{-1}T_n)}{\lambda_{min}(\widehat{T}_n^{-1}T)} - \frac{\lambda_n^+}{\lambda_n^-}\right)\frac{\lambda_n^-}{\lambda_n^+} \\ &= \frac{\lambda_{max}(\widehat{T}_n^{-1}T_n)\lambda_n^- - \lambda_{min}(\widehat{T}_n^{-1}T_n)\lambda_n^+}{\lambda_{min}(\widehat{T}_n^{-1}T_n)\lambda_n^+}\end{aligned}$$

Using the estimates in Lemma 3.1.31 and Lemma 3.1.32, we get

$$r \le \frac{\big(\lambda_n^+ + c_3(n,m)\big)\lambda_n^- - \big(\lambda_n^- - c_1(m)\big)\lambda_n^+}{\big(\lambda_n^- - c_1(m)\big)\lambda_n^+} = \frac{c_3(n,m)\lambda_n^- + c_1(m)\lambda_n^+}{\big(\lambda_n^- - c_1(m)\big)\lambda_n^+},$$

and

$$r \ge \frac{\big(\lambda_n^+ - c_3(n,m)\big)\lambda_n^- - \big(\lambda_n^- + c_2(m)\big)\lambda_n^+}{\big(\lambda_n^- + c_2(m)\big)\lambda_n^+} = -\frac{c_3(n,m)\lambda_n^- + c_2(m)\lambda_n^+}{\big(\lambda_n^- + c_2(m)\big)\lambda_n^+}.$$

Compare the order of each term, the conclusion is straightforward. □

This theorem allows us to use $\kappa(\widehat{D}_n^{-\frac{1}{2}}D_n\widehat{D}_n^{-\frac{1}{2}})$ as good approximation of $\kappa(\widehat{T}_n^{-\frac{1}{2}}T_n\widehat{T}_n^{-\frac{1}{2}})$ when $m$ is sufficiently large. It directly leads to $\kappa(\widehat{T}_n^{-\frac{1}{2}}T_n\widehat{T}_n^{-\frac{1}{2}}) = \Theta(\frac{n}{\sqrt{m}})$.

According to the main results of this section, i.e. Theorem 3.1.27 and Theorem 3.1.33, the use of Strang's preconditioner improves the condition of the underlying Toeplitz system a lot, but the preconditioned system can still become ill-conditioned when the dimension tends to infinity. However, The condition number could be controlled by increasing time steps as in the unpreconditioned case. The condition number does not bring trouble to our numerical experiments, which benefits from the setting of parameters as we believe.

### 3.1.4. A Tridiagonal Preconditioner

Since $T_n \approx D_n$, i.e $D_n^{-1}T_n \approx I_n$, we can expect all the eigenvalues of the coefficient matrix can be clustered around 1 if $D_n$ is used as preconditioner. With this point in view, we raise the following analysis in this section.

#### Spectrum of Preconditioned Matrix

The following theorem establishes the expected spectrum of $D_n$ preconditioned matrix.

**Theorem 3.1.34.** *If $D_n$ and $E_n$ are defined by (3.22), then*

$$\|D_n^{-1}T_n - I_n\|_2 < \frac{C_1}{3m + C_2},$$

*with $C_1 = \alpha\left(1 + 2\sqrt{\frac{\pi}{\beta}}\right)$ and $C_2 = 2(r+\lambda)T$ positive constants.*

*Proof.*

$$D_n^{-1}T_n - I_n = D_n^{-1}(D_n + E_n - D_n) = D_n^{-1}E_n,$$

thus

$$\|D_n^{-1}T_n - I_n\|_2 \le \|D_n^{-1}\|_2\|E_n\|_2 = \lambda_{max}(D_n^{-1})\|E_n\|_2 = \frac{1}{\lambda_{min}(D_n)}\|E_n\|_2.$$

Using (3.24) and substituting $\lambda_{min}(D_n)$ by $w_1 + 2w_2\left(1 + \cos\frac{n\pi}{n+1}\right)$, we obtain

$$\|D_n^{-1}T_n - I_n\|_2 < \frac{\alpha}{3m + 2(r+\lambda)T}\left(1 + 2\sqrt{\frac{\pi}{\beta}}\right).$$

□

If $m$ is sufficiently large, all the eigenvalues of $D_n$ preconditioned system are close to 1, thus the condition number is almost 1, independent of $n$. An alternative tridiagonal preconditioner could be the tridiagonal matrix only containing the three central diagonals of $T_n$.

To apply the preconditioner, a linear equation $D_n y = r$ with $y$ and $r$ being column vectors in $\mathbb{R}^n$ must be solved in each CG iteration, which can be done in $O(n)$ operations by forward elimination and backward substitution algorithm. This process can be accelerated by parallelizing techniques in numerical experiments.

As we notice in Theorem 3.1.10, $\kappa(D_n) = \Theta(n^2/m)$ has the same order as $\kappa(T_n)$, the ill condition of $T_n x^p = b^p$ is in fact transferred to $D_n y = r$ when $n$ is very large. As a result, numerical instability will likely occur. Dealing with this problem, we can compute $y = D_n^{-1}r$ by using the explicit inverse $D_n^{-1}$ (see Li et al. (2010) for example) to compute matrix-vector products. Theorem 3.1.25 can also be used to generate an algorithm for solving the linear equation $D_n y = r$ here. The overall cost grows to $O(n^2)$ then, which would compromise the efficiency of the tridiagonal preconditioner.

### Mesh Independent Superlinear Convergence

While $D_n$ is used as preconditioner for $T_n$, the preconditioned matrix is

$$D_n^{-1}T_n = I_n + D_n^{-1}E_n. \tag{3.38}$$

We first show that

$$\mathcal{F}(D_n^{-1}E_n)^2 := \sum_{i=1}^{n} \lambda_i(D_n^{-1}E_n)^2$$

has upper bound independent of $n$ in the following lemma:

**Theorem 3.1.35.** *$\mathcal{F}(D_n^{-1}E_n)^2 \le M$, where $M$ is a positive constant.*

*Proof.* We first prove that $\|E_n\|_F^2$ is bounded. Notice that by symmetry

$$\|E_n\|_F^2 = \sum_{i=1}^{n} \lambda_i(E_n)^2 = \sum_{i,j=1}^{n} e_{ij}^2 = (\lambda\tau)^2 \sum_{i,j=1}^{n} \left(\frac{h}{\sqrt{2\pi}\sigma_J} e^{-((i-j)h)^2/(2\sigma_J^2)}\right)^2,$$

in which $\sum_{i,j=1}^{n} \left(\frac{h}{\sqrt{2\pi}\sigma_J} e^{-((i-j)h)^2/(2\sigma_J^2)}\right)^2$ is a Riemann sum of $\int_{x_-}^{x_+-h}\int_{x_-}^{x_+-h} \big(f(x-z)\big)^2 \mathrm{d}x\mathrm{d}z$ with $f(x) = \frac{1}{\sqrt{2\pi}\sigma_J} e^{-x^2/(2\sigma_J^2)}$. Thus

$$\lim_{n\to\infty} \|E_n\|_F^2 \le \int_{x_-}^{x_+}\int_{x_-}^{x_+} \big(f(x-z)\big)^2 \mathrm{d}x\mathrm{d}z = C_f < \infty.$$

Note that $C_f$ is the same as in (3.11). Hence there exists a constant $\bar{C}_f$ such that for all $n$,

$$\|E_n\|_F^2 \le \bar{C}_f.$$

Furthermore, by Proposition 3.1.8, we can easily derive that $\|D_n^{-1}\|_2 < 2/3$. Utilizing the inequality

$$\|AB\|_F \le \min\{\|A\|_2\|B\|_F, \|A\|_F\|B\|_2\}, \qquad \forall A, B \in \mathbb{R}^{n\times n},$$

see Theorem 3.1.3 in Dennis and Schnabel (1996) for example, we obtain

$$\begin{aligned} \mathcal{F}(D_n^{-1}E_n)^2 &= \sum_{i=1}^{n} \lambda_i(D_n^{-\frac{1}{2}}E_nD_n^{-\frac{1}{2}})^2 = \|D_n^{-\frac{1}{2}}E_nD_n^{-\frac{1}{2}}\|_F^2 \\ &\le \|D_n^{-\frac{1}{2}}\|_2^2\|E_n\|_F^2\|D_n^{-\frac{1}{2}}\|_2^2 = \|D_n^{-1}\|_2^2\|E_n\|_F^2 \\ &\le (2/3)^2\bar{C}_f^2 =: M. \end{aligned}$$

Here $\|D_n^{-\frac{1}{2}}\|_2^2 = \lambda_{max}(D_n^{-\frac{1}{2}})^2 = \lambda_{max}(D_n^{-1}) = \|D_n^{-1}\|_2$ holds for $D_n$ symmetric. $\square$

Notice that $\lambda(D_n^{-1}T_n) \geq 1 - C_1/(3m + C_2) > 0$ for sufficiently large $m$ and the estimate is independent of $n$ according to Theorem 3.1.34. Thus combining the above theorem and Theorem 2.1.7, we come to a mesh independent superlinear convergence estimate for the CG method employing the tridiagonal preconditioner $D_n$, stated as follows.

**Corollary 3.1.36.** *For sufficiently large $m$, there exists a constant $C_m$, s.t.*

$$\frac{\|e_k\|_{T_n}}{\|e_0\|_{T_n}} \leq \left(\frac{C_m}{\sqrt{k}}\right)^k.$$

In fact, if we discretize (3.9) by FEM, we can always have a similar mesh independent superlinear convergence estimate, see the discussion below.

In Section 3.1.1, we denote the FEM discretized linear system of $(S + Q)y^p = \bar{b}_p$ by $(S_n + Q_n)x^p = b^p$, then under certain conditions, it is claimed that $\mathcal{F}(S_n^{-1}Q_n) \leq \|S^{-1}Q\|_{HS}$ by Theorem 1 in Karátson (2005), where $\|A\|_{HS}$ denotes the Hilbert-Schmidt norm of operator $A$, i.e. $\|A\|_{HS} := \left(\sum_{i=1}^{\infty} \lambda_i^2(A)\right)^{1/2}$ with $\lambda_i(A)(i \in \mathbb{N})$ the eigenvalues of $A$. We have already reviewed such an analysis in Section 2.1.3, now we summarize the result for our case by the following theorem.

**Theorem 3.1.37** (Karátson (2005)). *Let $H$ be a separable Hilbert space and consider a linear operator equation*

$$Ax = b$$

*with some $b \in H$, under the following assumptions:*

*(i) The operator $A$ is decomposed as $A = S + Q$ where $S$ is self-adjoint operator in $H$,with dense domain $D$ and $Q$ is a self-adjoint operator defined on the domain $H$;*

*(ii) There exists $c > 0$ such that $\langle Su, u\rangle \geq c\|u\|^2$ $(u \in D)$;*

*(iii) $\langle Qu, u\rangle \geq 0$ $(u \in H)$.*

*(iv) The operator $S^{-1}Q$, defined on the energy space $H_S$, is a compact Hilbert-Schmidt operator, i.e.*

$$\|S^{-1}Q\|_{HS}^2 := \left(\sum_{i=1}^{\infty} \lambda_i^2(S^{-1}Q)\right) < \infty.$$

*If all these assumptions are satisfied and $S_n$, $Q_n$ are the discrete matrices of $S$ and $Q$ via Galerkin discretization(see (3.15)), then*

$$\mathcal{F}(S_n^{-1}Q_n) \leq \|S^{-1}Q\|_{HS} \tag{3.39}$$

*holds independently of $n$.*

Similarly as the example in Karátson (2005), we set the Hilbert space $H$ to be $H_0^1(\Omega)$ and domain $D$ to be $H^2(\Omega) \cap H_0^1(\Omega)$, then assumption (i) is satisfied in our problem (3.9) since $H^2(\Omega) \cap H_0^1(\Omega)$ is dense in $H_0^1(\Omega)$ and the self-adjointness is obvious. Assumption (ii) is already shown by Proposition 3.1.2. In the proof of Theorem 1 in Karátson (2005), Assumption (iii) is nothing more than assuring the coercivity of the linear operator $A = S+Q$

together with Assumption (ii), it can be omitted if $A$ is coercive. Thus if assumption (iv) is fulfilled, we can apply Theorem 3.1.37 to our case.

We define the weighted inner product $\langle u, v\rangle_S = \left(\frac{3}{2} + (r+\lambda)\tau\right)\int_\Omega uv + \frac{1}{2}\sigma^2\tau\int_\Omega \nabla u\cdot\nabla v$ and the deduced norm is denoted by $\|\cdot\|_S$. Let $H_S$ be the completion of $H^2(\Omega)\cap H_0^1(\Omega)$. Notice that equation (3.12) implies that $\|\cdot\|_S$ is equivalent to $\|\cdot\|_{H^1}$, i.e. there exist two positive constants $M_1$ and $M_2$ such that $M_1\|x\|_S \le \|x\|_{H^1} \le M_2\|x\|_S$ for all $x\in H_S$. Hence $H_S = H_0^1(\Omega)$. This allows us to show that $S^{-1}Q$ is a Hilbert-Schmidt operator on $H_0^1(\Omega)$ rather than on $H_S$.

In fact, by Theorem 5.36 in Haroske and Triebel (2008), we have

**Proposition 3.1.38.** *$S^{-1}$ is a compact operator on $H_0^1(\Omega)$.*

The Hilbert-Schmidt norm of an integral operator is related to its kernel function by the following theorem (Theorem 1.2 in Section 5.1 of Gohberg et al. (2003)).

**Theorem 3.1.39.** *Suppose $\mathbf{k}(t,s)\in L_2([a,b]\times[a,b])$ and $\overline{\mathbf{k}(t,s)} = \mathbf{k}(s,t)$ a.e, the integral operator $K$ is defined by*

$$(Kf)(t) = \int_a^b \mathbf{k}(t,s)f(s)\mathrm{d}s,$$

*there exists a basic system of eigenvectors $\{\varphi_i\}$ and eigenvalues $\{\lambda_i\}$ of $K$ ($K\neq 0$) such that*

$$(Kf)(t) = \sum_{i=1}^{\infty}\lambda_i\left(\int_a^b f(s)\overline{\varphi_i(s)}\mathrm{d}s\right)\varphi_i(t).$$

*Furthermore,*

$$\int_a^b\int_a^b |\mathbf{k}(t,s)|^2\,\mathrm{d}t\mathrm{d}s = \sum_{i=1}^{\infty}\lambda_i^2.$$

Accordingly, for the integral operator $Q$ defined by $(Qu)(x) = -\lambda\tau\int_\Omega f(z-x)u(z)\,\mathrm{d}z$ in our case, we have

$$\|Q\|_{HS}^2 = \sum_{i=1}^{\infty}\lambda_i^2(Q) = \int_{x_-}^{x_+}\int_{x_-}^{x_+} |-\lambda\tau f(z-x)|^2\,\mathrm{d}z\mathrm{d}x = \lambda^2\tau^2 C_f < \infty,$$

thus $Q$ is a Hilbert-Schmidt operator.

Since $S^{-1}$ is a compact operator, there exists a constant such that $\|S^{-1}x\|_{H^1}\le C_S\|x\|_{H^1}$, $\forall x\in H_0^1(\Omega)$. Let $\{\phi_i\}_{i=1}^{\infty}$ be an orthonormal basis of $H_0^1(\Omega)$, then

$$\|S^{-1}Q\|_{HS}^2 = \sum_{i=1}^{\infty}\|S^{-1}Q\phi_i\|_{H^1}^2 \le C_S^2\sum_{i=1}^{\infty}\|Q\phi_i\|_{H^1}^2 = C_S^2\|Q\|_{HS}^2 = \lambda^2\tau^2 C_f C_S^2 < \infty, \quad (3.40)$$

where the first two equalities come from the definition of Hilbert-Schmidt norm given in Murphy (1990). It follows

**Proposition 3.1.40.** *$S^{-1}Q$ is a Hilbert-Schmidt operator on $H_0^1(\Omega)$.*

Since $\lambda_{max}(S_n^{-1}Q_n) \le \mathcal{F}(S_n^{-1}Q_n)$, combining (3.39) and (3.40), we obtain

$$\lambda_{max}(S_n^{-1}Q_n) \le \|S^{-1}Q\|_{HS} \le \lambda\tau\sqrt{C_f}C_S.$$

Hence for sufficiently large $m$, $\lambda_{max}(S_n^{-1}Q_n) < \frac{1}{2}$, thus $\lambda_{min}(I - S_n^{-1}Q_n) > \frac{1}{2}$.

By Theorem 2.1.7, we have the superlinear convergence estimate

$$\frac{\|e_k\|_{\bar{T}_n}}{\|e_0\|_{\bar{T}_n}} \le \left(\frac{4\|S^{-1}Q\|_{HS}}{\sqrt{k}}\right)^k, \tag{3.41}$$

where $\bar{T}_n$ denotes $S_n + Q_n$, for the CG method applied to $(S_n + Q_n)x^p = b^p$ using $S_n$ as the preconditioner .

### 3.1.5. Numerical Results

Throughout all the experiments in this section, the constant parameters being referred to in Section 3.1.1 are set to be: $T = 1$, $r = 0$, $\sigma = 0.2$, $\sigma_J = 0.5$, $\lambda = 0.1$, $K = 1$, $x_- := -\hat{x} + \zeta T$ and $x_+ := \hat{x} + \zeta T$ with $\hat{x} = 4$ and $\zeta := r - \frac{1}{2}\sigma^2 - \lambda\eta$. Here $\eta = e^{\sigma_J^2/2} - 1$. Details for the practical meaning of these parameters can be found in Sachs and Strauss (2008).

The behavior of CG solver and relevant spectral properties are illustrated as the number of space grid points $n$ and the number of time steps $m$ change.

### Spectral Property

As the eigenvalue distribution of the coefficient matrix plays a major role in the efficiency of CG iteration, we graph that of circulant preconditioned and tridiagonal preconditioned matrices. Note that the unpreconditioned Toeplitz matrix $T_n$ has distributed eigenvalues (c.f. Figure 1 in Sachs and Strauss (2008)).

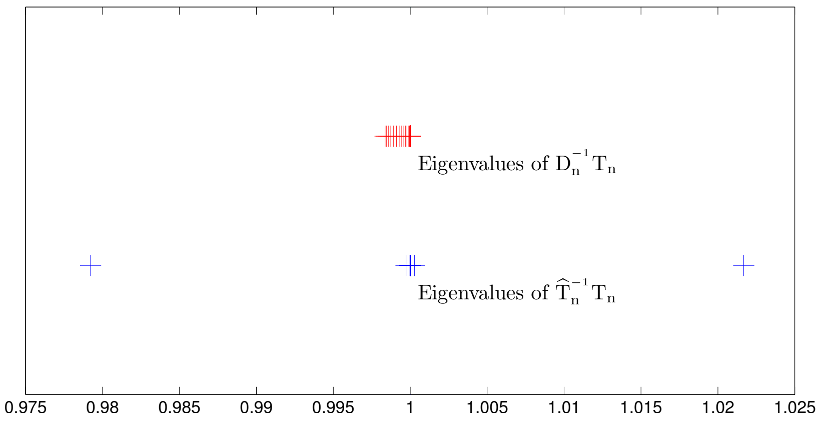


**Figure 3.1.: Spectra of Strang preconditioned matrix and tridiagonal preconditioned matrix with $n = 64$ and $m = 40$.**

The matrix $\widehat{T}_n^{-1}$ is computed using Theorem A.2.2 and $D_n^{-1}T_n$ is done by the Matlab inbuilt 'backslash'.

Figure 3.1 establishes the conclusion of Theorem 3.1.22, which states most eigenvalues of the Strang preconditioned matrix are clustered around 1, except two outliers. In fact, this depends on how large the 'neighborhood' of 1 is set. If $\epsilon$ is sufficiently small, it will turn to the second case of (3.30). When the tridiagonal part $D_n$ is used as the preconditioner, all the eigenvalues are clustered near 1 as Theorem 3.1.34 points out.

| $n$ | $m$ | $\rho$ |
|---|---|---|
| 512 | 5 | 0.012929709 |
| 512 | 10 | 0.006509528 |
| 512 | 20 | 0.003266120 |
| 512 | 40 | 0.001635938 |
| 512 | 80 | 0.000818698 |
| 512 | 160 | 0.000409534 |

**(a) $n$ fixed and $m$ variable.**

| $n$ | $m$ | $\rho$ |
|---|---|---|
| 128 | 40 | 0.001635736 |
| 256 | 40 | 0.001635886 |
| 512 | 40 | 0.001635938 |
| 1024 | 40 | 0.001635953 |
| 2048 | 40 | 0.001635957 |
| 4096 | 40 | 0.001635958 |

**(b) $n$ variable and $m$ fixed.**

**Table 3.1.: Bounds on the spectrum of $D_n^{-1}T_n$.**

If we set $\rho := \max_i\{|\lambda_i(D_n^{-1}T_n) - 1| : 1 \le i \le n\}$, which equals $\|D_n^{-1}T_n - I_n\|_2$, then by Theorem 3.1.34, $\rho < \frac{C_1}{3m+C_2}$ with $C_1$ and $C_2$ positive constants. This can be seen from Table 3.1a. When $m$ is fixed and $n$ changes, $\rho$ is only slightly influenced, see Table 3.1b.

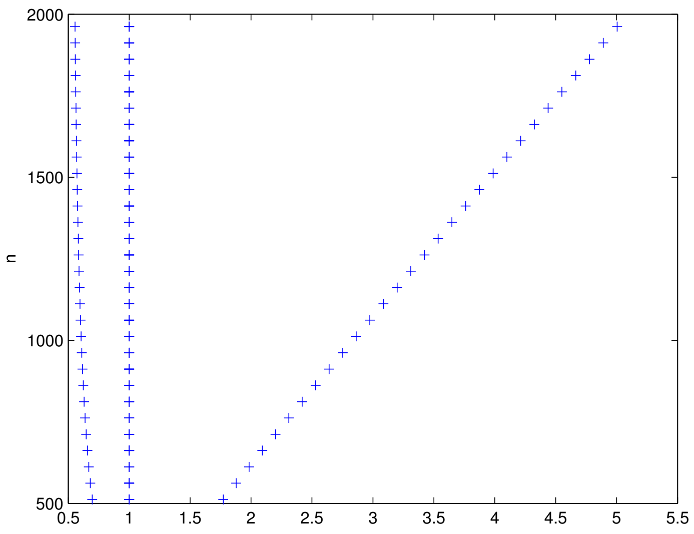
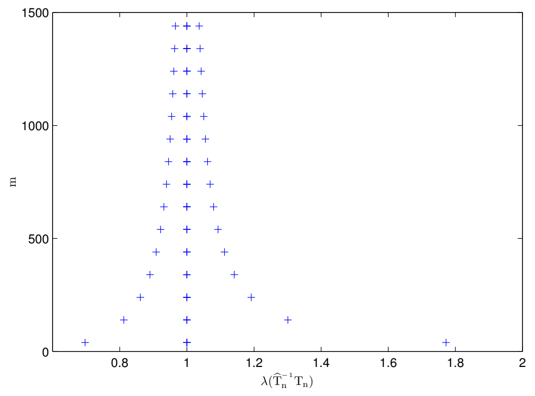


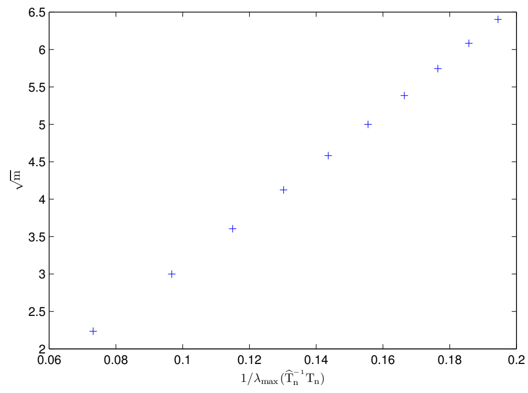
**Figure 3.2.: Spectrum of $\widehat{T}_n^{-1}T_n$ with $n$ variable and $m = 40$.**

Combining Theorem 3.1.27 , Lemma 3.1.31 and Lemma 3.1.32, we know that when a given $m$ is sufficiently large, the largest eigenvalue of Strang preconditioned matrix grows linearly as $n$ grows while the smallest eigenvalue enters a bounded interval. This fact is illustrated by Figure 3.2, in which the middle column gives the clustered eigenvalues of $\widehat{T}_n^{-1}T_n$ (almost equal to 1) for different $n$, the left column shows how the smallest eigenvalues change with respect to $n$ and the largest eigenvalues are in the right.

Figure 3.2 also implies the condition number of $\widehat{T}_n^{-1}T_n$ increases linearly under the same assumption. This is negative for the preconditioner, but in this case, we also notice that the condition number is indeed not large and the CG solver performs well (see Figure 3.4 and Table 3.2). As a matter of fact, in order to maintain second order accuracy, $n$ and $m$ are usually magnified simultaneously (see Table 3 in Sachs and Strauss (2008)), which means the condition number of the Strang preconditioned matrix can be well controlled in the application.



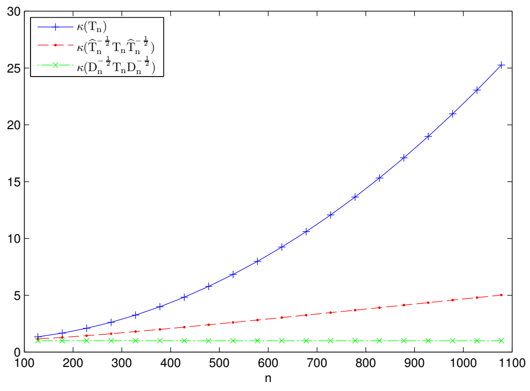
**(a) Spectrum of $\widehat{T}_n^{-1}T_n$ with $n = 512$ and $m$ variable.**



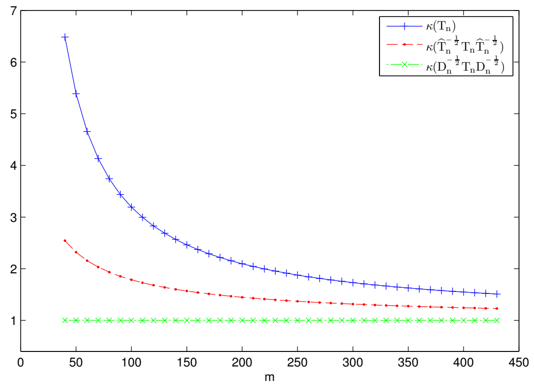
**(b) The relationship between $\frac{1}{\lambda_{max}(\widehat{T}_n^{-1}T_n)}$ and $\sqrt{m}$ with $n = 2048$.**

**Figure 3.3.: Eigenvalue distribution of $\widehat{T}_n^{-1}T_n$ with $n$ fixed and $m$ variable.**

On the other hand, if we fix $n$ and increase $m$, then the two extreme eigenvalues (outliers) shrink to 1 as shown in Figure 3.3a. More specifically, Figure 3.3b shows that the largest eigenvalue of Strang preconditioned matrix $\widehat{T}_n^{-1}T_n$ is inversely proportional to $\sqrt{m}$ when $n$ is fixed. Since it is always assumed that $n^2/m$ is large in the theoretical part, we fix $n$ to be relatively large (i.e. $n = 2048$) and try not to make $m$ large here.



**(a) $n$ variable and $m = 40$.**



**(b) $n = 512$ and $m$ variable.**

**Figure 3.4.: Euclidean condition numbers of unpreconditioned and conditioned systems.**

Condition number is another important character for evaluating the efficiency of preconditioning. We summarize those of unpreconditioned, tridiagonal preconditioned and Strang preconditioned matrices as follows:

(i) $\kappa(T_n) = \Theta(n^2/m)$ (see Theorem 3.1.10);

(ii) $\kappa(D_n^{-\frac{1}{2}} T_n D_n^{-\frac{1}{2}}) \approx 1$ (see Theorem 3.1.34);

(iii) $\kappa(\widehat{T}_n^{-\frac{1}{2}} T_n \widehat{T}_n^{-\frac{1}{2}}) = \Theta(n/\sqrt{m})$ (see Theorem 3.1.33 and Corollary 3.1.28).

Figure 3.4 demonstrates how the condition numbers of different systems change as $n$ and $m$ grow respectively. It is obvious that preconditioning with Strang's preconditioner improves the condition a lot and the tridiagonal preconditioner is almost perfect at this point, especially, the condition number of $D_n^{-\frac{1}{2}} T_n D_n^{-\frac{1}{2}}$ does not depend on $n$ any more.

To obtain the estimate of condition number of Strang preconditioned matrix $\widehat{T}_n^{-\frac{1}{2}} T_n \widehat{T}_n^{-\frac{1}{2}}$, the relevant proofs in Section 3.1.3 heavily rely on estimates in infinity norm. Indeed, numerical results show that these estimates are fairly sharp in this case.

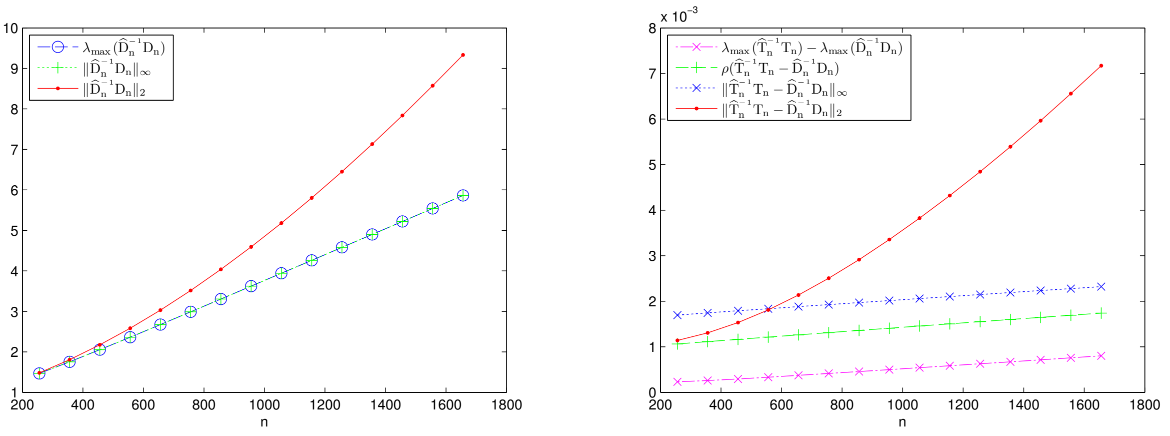


**Figure 3.5.: The left plot shows the largest eigenvalue of $\widehat{D}_n^{-1} D_n$ and two upper bounds of it with respect to infinity norm and 2-norm respectively. The right plot is about the difference between largest eigenvalues of $\widehat{T}_n^{-1} T_n$ and $\widehat{D}_n^{-1} D_n$, and three different upper bounds. In both plots, $m = 20$ and $n$ varies.**

As we can see in the left plot of Figure 3.5, $\|\widehat{D}_n^{-1} D_n\|_\infty$ is almost identical to $\lambda_{max}(\widehat{D}_n^{-1} D_n)$. $\|\widehat{D}_n^{-1} D_n\|_2$ is also plotted for comparison, which is obviously not a good estimate as the linear growth is violated. In the right plot, $\rho(\widehat{T}_n^{-1} T_n - \widehat{D}_n^{-1} D_n)$ denotes the spectral radius of $\widehat{T}_n^{-1} T_n - \widehat{D}_n^{-1} D_n$. For some $n$, the infinity norm estimate is not better than the 2-norm estimate. But once again, the 2-norm estimate does not keep the linearity of the growth and changes the order (with respect to $n$) of the quantity. Note that the gap in the right plot is indeed very small according to the scale of y-axis.

## Efficiency of CG Solver

In the CG solver, we use

$$\|r_k\|_\infty < \delta$$

as the stopping criterion instead of the one in weighted norm which is essential for all the error estimates in Section 2.1. Here $r_k$ denotes the residual after the $k$th CG iteration and $\delta$ is a positive constant describing the precision of the CG solver.

To solve the sequence of discrete problems with coefficient matrix defined by (3.6), CG solver using Strang's preconditioner needs $O(mn\log n)$ flops. If tridiagonal preconditioner is used, then the effort is $O(mn)$. It is not trivial that the overhead is linearly dependent on $m$, because the iteration number needed for each time step does not only rely on the spectrum of the coefficient matrix but also concerns the initial error $e_0$. In the numerical experiments, we always use the solution of previous time step as initial guess of the current linear equation, which assures that $\|e_0\|_2$ is of the same scale throughout all the time steps and results in stable iteration numbers.

| $n$ | $m$ | sec($D_n$) | #it($D_n$) | sec($\widehat{T}_n$) | #it($\widehat{T}_n$) |
|---|---|---|---|---|---|
| 2048 | 40 | 0.134 | 3 | 0.171 | 4 |
| 4096 | 80 | 0.478 | 3 | 0.687 | 4 |
| 8192 | 160 | 1.865 | 3 | 2.749 | 4 |
| 16384 | 320 | 5.605 | 2 | 11.041 | 4 |
| 32768 | 640 | 24.680 | 2 | 50.572 | 4 |
| 65536 | 1280 | 106.049 | 2 | 240.372 | 4 |

**Table 3.2.: CPU time and average iteration number of CG solver with tridiagonal and Strang's preconditioner respectively ($\delta = 10^{-8}$).**

Like in Sachs and Strauss (2008), we scale $m$ and $n$ by a factor of 2 at the same time in Table 3.2, the computational effort is reflected by CPU time (only that of CG iteration is counted). Further, the iteration numbers are smaller when the tridiagonal preconditioner $D_n$ is used, which can be even smaller when $m$ gets larger. Mesh independent convergence can also be observed for both cases.

| | | unprec. | | $D_n$ | | $\widehat{T}_n$ |
|---|---|---|---|---|---|---|
| $n$ | $m$ | $10^{-8}$ | $10^{-8}$ | $10^{-5}$ | $10^{-3}$ | tri. $\delta$ |
| 64 | 40 | 5 | 2 | 2 | 1 | 3 |
| 128 | 40 | 7 | 2 | 2 | 1 | 4 |
| 256 | 40 | 11 | 3 | 2 | 1 | 4 |
| 512 | 40 | 21 | 3 | 2 | 1 | 4 |
| 1024 | 40 | 43 | 3 | 2 | 1 | 4 |
| 2048 | 40 | 89 | 3 | 2 | 2 | 4 |

**Table 3.3.: Iteration numbers for different precision $\delta$ (m=40).**

In Table 3.3, the third column provides the iteration numbers of unpreconditioned CG method with $\delta = 10^{-8}$. The next three columns show the iteration numbers of CG solver with $D_n$ being the preconditioner and different values of $\delta$ are used which indeed influences the iteration numbers. The last column gives the iteration numbers of CG solver using Strang's preconditioner with respect to three different precisions (i.e. $\delta = 10^{-3}$, $\delta = 10^{-5}$ and $\delta = 10^{-8}$). Different from the case of tridiagonal preconditioned CG method, the iteration number does not vary when the required accuracy is changed. Roughly speaking, this is because $\widehat{T}_n^{-1}T_n$ has three clusters of eigenvalues while $D_n^{-1}T_n$ only has one.

We conclude that preconditioned CG method reduces the iteration number a lot and both ways of preconditioning achieve mesh independent convergence. The performance of tridiagonal preconditioner is slightly better than Strang's preconditioner for our application. Nevertheless, from theoretical point of view, it is much more reliable.

## 3.2. Precondition the PIDE in a Calibration Problem

In the previous section, the tridiagonal preconditioner coming from Laplacian outperforms circulant preconditioners, which aim at preconditioning the full linear equations derived from the PIDE. This fact helps us choose the preconditioner in solving the following calibration problem, which calibrates the mathematical model (constraint PIDE) with respect to parameter set $u$ by minimizing the difference between model data $D(u; x_i, a_i)$ (solution to the PIDE for given $u$) and known market data $d(x_i, a_i)$ $(i \in \mathcal{I})$ in terms of least squares.

$$\begin{aligned} \min_u \quad & J(u) := \frac{1}{2}\sum_{i\in\mathcal{I}} |D(u; x_i, a_i) - d(x_i, a_i)|^2 \\ \text{s.t.} \quad & D_T(x,T) - \frac{\sigma^2(x,T)}{2}D_{xx}(x,T) + \left(r(T) + \frac{\sigma^2(x,T)}{2} - \lambda\zeta\right) D_x(x,T) \\ & + \lambda(1+\zeta)D(x,T) - \lambda \int_{-\infty}^{+\infty} D(x-y,T)e^y \, f(y)dy = 0 \\ & \text{where} \quad D(x,0) = \max\{S_0 - S_0 e^x, 0\} =: D_0(x). \end{aligned} \tag{3.42}$$

Here $x \in (-\infty, \infty)$, $T \in (0, T_{max})$, $u$ is defined to be the parameter set $(\sigma, \lambda, f)$. For more detailed description of (3.42), we refer to Schu and Sachs (2010) and Schu (2012).

In Schu and Sachs (2010), a multilevel TRPOD algorithm, which employs POD in the framework of trust region method, is proposed for solving (3.42). We present it as follows:

**Algorithm 3.1** Multilevel TRPOD algorithm Schu and Sachs (2010)

```
1: Given $u_k$, solve the state and adjoint equation (PIDEs) on level $h_k$;
2: Use the solutions of Step 1 to build a POD model (reduced order model)
       $m_k^{POD}(u_k + s_k)$;
3: Compute an approximate minimizer $s_k$ for
       $\min_{||s_k|| \le \Delta_k} m_k^{POD}(u_k + s_k)$;
4: Compute $J(u_k + s_k)$ and $\rho_k = \frac{ared(s_k)}{pred(s_k)}$;
5: Update the trust region radius according to $\rho_k$,
   Refine, coarse or keep the discretization level $h_{k+1}$,
   $k \leftarrow k + 1$ and go to step 1 resp. step 3
```

In Algorithm 3.1, the major numerical effort lays on solving PIDEs on different levels in Step 1 and the POD subproblems in Step 3. Since the POD subproblems are of low dimension, we concern more about accelerating the process of solving the PIDEs. Note that the adjoint equations have similar structure as the constraint Dupire's equation.

The numerical treatment of the constraint PIDE is very similar to that in the previous section. We first need to truncate the infinite domain, $\sigma(x,T)$ and $g(x) := e^x f(x)$ are restricted to the truncated domain then. For time discretization, Crank-Nicolson scheme is employed while in space a linear spline basis is used for finite element discretization. More details can be found in Schu and Sachs (2010).

As a result, at each time step a linear equation having the form

$$(S_n + Q_n)x_n = b_n \tag{3.43}$$

is to be solved, where $S_n$ and $Q_n$ are from elliptic and integral parts respectively. Notice that $S_n$ is tridiagonal, not Toeplitz, nonsymmetric and $Q_n$ is dense, Toeplitz, nonsymmetric, see Schu and Sachs (2010). Since the coefficient matrix $S_n + Q_n$ is nonsymmetric, GMRES is used for solving (3.43). Fast Fourier transform (FFT) is adopted for efficient matrix-vector product evaluation because $Q_n$ is Toeplitz matrix.

Let $\tau$ denote time difference and $h$ be the mesh size in space, utilizing the explicit expressions of the entries of $S_n$ and $Q_n$ in Section 3.2.1 of Schu (2012), it can be easily shown that

**Lemma 3.2.1.** *(i) If $\tau$ is sufficiently small and $\sigma(x)$, $\sigma'(x)$, $\sigma''(x)$ are bounded, then $S_n$ is diagonally dominant by rows and columns.*

*(ii) $\|Q_n\|_1 \le \bar{c}\tau h, \|Q_n\|_\infty \le \bar{c}\tau h$ with $\bar{c}$ constant.*

Thus it is straightforward to have the following proposition according to Theorem 3.1.20.

**Proposition 3.2.2.** *$\|S_n^{-1}Q_n\|_\infty \le c_1\tau$, $\|S_n^{-1}Q_n\|_1 \le c_2\tau$. $c_1$ and $c_2$ are positive constants.*

This suggests if we use the tridiagonal matrix $S_n$ as preconditioner, the preconditioned matrix $I_n + S_n^{-1}Q_n$ has its eigenvalues clustered around 1 and its condition number is close to 1, when $\tau$ is sufficiently small.

With help of the following theorem, we can analyze the rate of convergence of GMRES in this case. A tighter estimate could be found in Beckermann et al. (2005).

**Theorem 3.2.3** (Elman (1982))**.** *If $A$ has a positive definite hermitian part $(A+A^T)/2$, the $k$-th residual $r_k$ of GMRES method for $Ax = b$ satisfies*

$$\frac{\|r_k\|_2}{\|r_0\|_2} \le \sin^k(\beta),$$

*where $\beta \in [0, \frac{\pi}{2}]$ is defined by* $\cos(\beta) := \dfrac{\lambda_{min}\big((A+A^T)/2\big)}{\|A\|_2}$.

Let $A = I_n + S_n^{-1}Q_n$ be the preconditioned matrix in our case, we have

$$\begin{aligned}\|A\|_2 &= \|I_n + S_n^{-1}Q_n\|_2 \\ &\leq 1 + \|S_n^{-1}Q_n\|_2 \\ &\leq 1 + \sqrt{\|S_n^{-1}Q_n\|_1 \|S_n^{-1}Q_n\|_\infty} \\ &\leq 1 + \tau\sqrt{c_1 c_2}\end{aligned}$$

and

$$\begin{aligned}\lambda_{min}\big((A + A^T)/2\big) &= \frac{1}{\lambda_{max}\big((A+A^T)/2\big)} = \frac{1}{1 + \frac{1}{2}\lambda_{max}\big(S_n^{-1}Q_n + (S_n^{-1}Q_n)^T\big)} \\ &\geq \frac{1}{1 + \frac{1}{2}\|S_n^{-1}Q_n + (S_n^{-1}Q_n)^T\|_1} \geq \frac{1}{1 + \frac{1}{2}\left(\|S_n^{-1}Q_n\|_1 + \|S_n^{-1}Q_n\|_\infty\right)} \\ &\geq 1 - \frac{1}{2}\left(\|S_n^{-1}Q_n\|_1 + \|S_n^{-1}Q_n\|_\infty\right) \\ &\geq 1 - \tau\frac{c_1 + c_2}{2}.\end{aligned}$$

Hence we can conclude that the residuals of GMRES iteration satisfy $\dfrac{\|r_k\|_2}{\|r_0\|_2} \leq \sin^k(\bar{\beta})$, where $\bar{\beta}$ satisfies $\cos(\bar{\beta}) \geq \dfrac{1 - \tau\frac{c_1+c_2}{2}}{1 + \tau\sqrt{c_1c_2}}$.

Thus GMRES can reach at least linear convergence and the rate of convergence is mesh independent.

The numerical results are obtained using the same codes and the same setting in the numerical experiment of Schu and Sachs (2010). Table 3.4 compares the CPU time for solving calibration problem (3.42) using Algorithm 3.1, with or without preconditioner in Step 1. Numbers are given in second. The effect of using the preconditioner is significant.

| | Total | POD | PIDE |
|---|---|---|---|
| Unprec. | 8041 | 85 | 7956 |
| Prec. | 105 | 85 | 20 |

**Table 3.4.: CPU time (in second) comparison of multilevel TRPOD solver.**

| n | rep. | aver. #it unprec. | aver. #it prec. |
|---|---|---|---|
| 500 | 21 | 21 | 3 |
| 1000 | 3 | 35 | 3 |
| 2000 | 5 | 67 | 3 |
| 4000 | 4 | 141 | 3 |

**Table 3.5.: Average iteration number of GMRES for solving PIDEs with or without preconditioner.**

Table 3.5 gives the average iteration numbers of GMRES on different levels. The first column is the numbers of grid points of different levels of mesh, the second column records the number of PIDEs which are solved on a certain level. From the table we can see that the iteration numbers grow with respect to mesh size when GMRES solver works without preconditioner, however, the mesh independence is observed if the tridiagonal preconditioner is used.

# Chapter 4.

# Preconditioning with Proper Orthogonal Decomposition

Solving PDE-constrained optimization problems usually leads to linear systems of saddle point type, when an all-at-once method is adopted. Iterative solvers for such problems and corresponding preconditioning techniques are well discussed in Benzi et al. (2005). Developing efficient solvers for linear problems arising from PDE-constrained problems is an active field of research.

In this chapter, we aim at exploring the possibility of embedding a reduced-order-model (ROM) technique, namely proper orthogonal decomposition (POD), in preconditioned Krylov subspace methods and developing an efficient solver for optimization problem governed by a time dependent nonlinear PDE.

This work is inspired by the recent paper Simoncini (2012), in which the author introduces a preconditioned deflated MINRES solver for the linear systems resulting from a nonlinear PDE-constrained optimization problem, more precisely, a simplified Monge-Kantorovich mass transfer problem.

In Simoncini (2012), the optimization problem is first handled by an inexact Gauss-Newton approach, see Benzi et al. (2011). In each Newton-type iteration, part of the unknown variables are removed by simple calculation, which gives rise to an equivalent lower dimensional linear system. MINRES serves as the linear solver and a block diagonal preconditioner is employed. Since the preconditioner is applied approximately, some eigenvalues close to zero of the coefficient matrix are not well captured by the preconditioner. As a result, stagnation phases in the convergence history of MINRES are observed. In order to remove the stagnation, a few approximate eigenvectors related to the unpleasant eigenvalues of the preconditioned matrix are obtained by means of Lanczos process and a deflated MINRES algorithm using these eigenvectors is developed. The improvement of the convergence is established by numerical results.

Other than using the approximate eigenvectors of the preconditioned matrix for deflation, we propose to use POD basis to generate the deflation matrix. As is well known, POD is usually used for building reduced order models, because it can extract key information of the solution space out of given snapshots when a time dependent PDE is treated. This essential feature makes it possible to use POD to do more things.

Let us consider a simple nonsingular linear equation $Ax = b$ in $\mathbb{R}^n$. For simplicity, we assume that $A$ is symmetric positive definite. Let $\{\lambda_i\}_{i=1}^n$ denote the eigenvalues of $A$ satisfying $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$, the corresponding eigenvectors are denoted by $\{v_i\}_{i=1}^n$, then we can expand the right-hand side by $b = \sum_{i=1}^n c_i v_i$ for all $c_i \in \mathbb{R}$ constant and the solution has expression

$$x = A^{-1}b = \sum_{i=1}^{n} \frac{1}{\lambda_i} c_i v_i$$

When the linear equation is a discrete PDE, $A$ usually has eigenvalues close to 0, e.g. $A$ is discretized Laplacian operator in the previous Chapter. Despite the influence of $c_i$, the inverses of smallest eigenvalues of $A$ magnify the corresponding eigenvectors. Hence these eigenvectors are more likely dominating in the solution. If we have a set of such linear equations which share the same coefficient matrix but have different right-hand sides, then we have a good chance to extract $v_i$ of smallest indices from the solutions of all these linear equations via POD. This is exactly the progress of our first experiment in the next section.

We first test our idea of using POD basis to generate deflation matrix on a heat equation, then we move on to a linear quadratic problem and develop the deflation strategy for a down-sized KKT system. A POD deflated MINRES solver using a diagonal block preconditioner is proposed for a nonlinear PDE-constrained optimization problem at last. A number of numerical experiments are designed and carried out, affirmative results are obtained.

This chapter is organized as follows.

Section 4.1 consists of a series of numerical experiments on a simple 1D heat equation involving different initial conditions. We first extract POD basis from various subsets of snapshots obtained by three different ways, then we compare all these basis to the exact solution, in order to check how well POD can extract the desired information contained in the given snapshots. After that, we compare the eigenvalue distributions of POD-deflated matrix and the original coefficient matrix in the discrete system. The results show that POD basis can capture some smallest eigenvalues of the coefficient matrix. We also implement deflated CG algorithm which uses deflation matrix constructed by POD basis to solve the discrete system and observe obvious reduction of iteration numbers. In the end of this section, we propose a deflation matrix for solving the heat equation with one-shot method, which solves all time instances simultaneously instead of solving a sequence of linear equations.

In Section 4.2 we work on a linear quadratic problem taken from Stoll and Wathen (2010). We remove the control variable in the resulting KKT system to construct an equivalent linear system via simple calculation, as in Simoncini (2012). Then we propose deflation strategy for the resulting system by projecting state variable and adjoint variable into the POD subspace. A deflated MINRES solver is implemented for various experiment settings. The results confirm the effectiveness of our deflation strategy.

In Section 4.3, a boundary control problem with nonlinear constraint PDE is constructed for test. The problem is solved in an SQP framework thus a sequence of saddle point problems have to be solved. We also downsize each of them as we do in the linear quadratic problem, then propose a solver which contains a block diagonal preconditioner and combines the deflation strategy. Compared to same preconditioned MINRES solver, the embedding of deflation with POD improves the rate of convergence and reduces both iteration numbers and CPU time.

This research is still in progress, we give some concluding remarks for the work done and also outlook in the end of this chapter.

Notice that we refer to smallest eigenvalues in the sense of smallest magnitude in this chapter.

## 4.1. Observation: Catch Smallest Eigenvalues with POD

POD has been proved to be very helpful in solving many kinds of PDEs and PDE-constrained optimization problems. It is usually used to build reduced order models of very low dimension for PDEs. In consequence, solving such models provides good approximate solutions to the original problems at low cost.

We observe that the POD basis in fact includes eigen information of the underlying (discrete) PDE, which allows us to use POD as a pure numerical technique in an Krylov subspace method.

In this section, we first introduce how POD is realized numerically by example (a more general discussion is provided in Appendix B). Furthermore, the observation, that computed basis via POD can be used as approximate leftmost eigenspace (eigenvectors related to the smallest eigenvalues) of the linear equations arising from discretized PDE, is also illustrated for the same example.

### 4.1.1. Test Problem: Heat Equation with Various Initial Conditions

As a test problem, we take the following 1D heat equation:

$$\begin{cases} u_t - u_{xx} = 0, & \forall\ (x,t) \in \Omega \times (0,T] \\ u(x,t) = 0, & x \text{ on } \partial\Omega \\ u(x,0) = u_0(x), & \end{cases} \tag{4.1}$$

where $\Omega = (0,1)$.

By Fourier analysis, the solution $u(x,t)$ has the form

$$u(x,t) = \sum_{k=1}^{\infty} \alpha_k e^{-\lambda_k t} v_k(x), \tag{4.2}$$

with

$$\begin{aligned} \lambda_k &= (k\pi)^2, \\ v_k(x) &= \sqrt{2}\sin(\sqrt{\lambda_k}x), \qquad k = 1,2,3,\ldots \\ \alpha_k &= \langle u_0, v_k\rangle_{L^2(\Omega)}. \end{aligned}$$

For different initial conditions, the Fourier coefficients $\{\alpha_k\}_1^\infty$ are computed by evaluating $\langle u_0, v_k\rangle_{L^2(\Omega)}$ and presented as follows:

$$\alpha_k = \begin{cases} \frac{2\sqrt{2}}{k\pi}, & \text{if } k \text{ is odd} \\ 0, & \text{if } k \text{ is even} \end{cases}, \qquad \text{when} \quad u_0(x) = 1, x \in [0,1], \tag{4.3}$$

$$\alpha_k = \begin{cases} \frac{\sqrt{2}}{k\pi}, & \text{if } k \bmod 2 = 1 \\ -\frac{2\sqrt{2}}{k\pi}, & \text{if } k \bmod 4 = 2 \\ 0, & \text{if } k \bmod 4 = 0 \end{cases}, \qquad \text{when} \quad u_0(x) = \begin{cases} 0, x \in [0, \frac{1}{2}) \\ 1, x \in [\frac{1}{2}, 1] \end{cases}, \tag{4.4}$$

$$\alpha_k = \frac{2\sqrt{2}}{(k\pi\delta)^2}\sin(\frac{1}{2}k\pi)\,[1-\cos(k\pi\delta)] \quad \text{when} \quad u_0(x) = \begin{cases} \frac{1}{\delta}+\frac{1}{\delta^2}(x-\frac{1}{2}), & x\in[\frac{1}{2}-\delta,\frac{1}{2}] \\ \frac{1}{\delta}-\frac{1}{\delta^2}(x-\frac{1}{2}), & x\in[\frac{1}{2},\frac{1}{2}+\delta] \\ 0, & \text{otherwise} \end{cases}. \tag{4.5}$$

Here $u_0(x)$ is actually set to be 'constant', 'step', 'peak' function respectively. Note that some terms in the analytic solution (4.2) vanish when the relevant coefficients equals 0 in these three cases. This fact has impact on our numerical experiments.

## On the Solution

In preparation of latter discussion, we include two approximations of the analytic solution (4.2) in addition to a numerical solution.

For practical implementation, the *exact solution* $u^N$ of problem (4.1) is given by the partial sum

$$u^N = \sum_{k=1}^{N} \alpha_k e^{-\lambda_k t} v_k(x), \tag{4.6}$$

with a large $N \in \mathbb{N}$.

A *truncated solution* $u^{N_0}$ is given by the partial sum

$$u^{N_0} = \sum_{k=1}^{N_0} \alpha_k e^{-\lambda_k t} v_k(x), \tag{4.7}$$

where $N_0 \in \mathbb{N}$ and $N_0 \ll N$.

To obtain *FEM solution* $u^{FEM}$ of Problem (4.1), linear spline basis is used for spatial discretization and implicit Euler scheme is used for time discretization. The domain is partitioned as follows:

$$\begin{aligned} x_i &:= ih \quad \text{with} \quad i=1,\dots,n, \quad h=1/(n+1), \\ t_j &:= j\tau \quad \text{with} \quad j=1,\dots,m, \quad \tau=T/m. \end{aligned} \tag{4.8}$$

At each time step, a linear equation

$$(M+\tau K)u_j = Mu_{j-1} \tag{4.9}$$

should be solved, where

$$\text{mass matrix } M := \frac{h}{6}\begin{pmatrix} 4 & 1 & & \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{pmatrix}, \quad \text{stiffness matrix } K := \frac{1}{h}\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}.$$

The FEM solution $u^{FEM}$ is given by a $n \times m$ matrix whose $j$-th column is $u_j$. The other two solutions $u^N$ and $u^{N_0}$ are also evaluated on the same grid (4.8) and represented by $n \times m$

matrices in the following numerical experiments.

**Remark 4.1.1.** It is not hard to see that high frequency terms decay exponentially when $t$ moves away from 0 in the Fourier series (4.2), i.e. these terms are only visible around $t = 0$. Hence $u^N$ provides a solution to (4.1) with very high precision already and $u^{N_0}$ can be treated as very good approximation of the exact solution. This observation also plays an important role when we choose snapshots to extract the low frequency vectors, as we will see later.

### 4.1.2. Numerical Experiments

We carry on the numerical experiments step by step as follows. We first extract POD basis from different subsets of snapshots of the solution and compare the POD basis to the solution to see whether we can extract the major information of the given solution, more precisely, the nonzero terms in (4.2), by POD. Results are presented for different types of approximate solution and different initial conditions. Secondly, we illustrate the effect of removing the smallest eigenvalues from the coefficient matrix $M + \tau K$ using POD basis. We also try to use deflated CG method to compute the numerical solution and compare the efficiency to that of using standard CG method. In the end, some discussions are raised for one-shot method of solving (4.1) numerically, which compute all the time steps of the FEM solution at the same time.

#### POD and Practical Implementation

Extracting POD basis from the solution (snapshots) to a time dependent PDE (4.1) can be achieved by singular value decomposition (SVD), for detailed discussion, we refer to Volkwein (2011). In Section B.2, alternative implementations are presented, here we choose to solve an equivalent symmetric eigenvalue problem.

Assume we have a set of column vectors $\{u_{s_1}, u_{s_2}, \dots, u_{s_l}\} \subset \mathbb{R}^n$ $(l < n)$ and they form a matrix

$$X = (u_{s_1}, u_{s_2}, \dots, u_{s_l}),$$

we get a POD-like basis by:

(i) Solving the eigenvalue problem

$$X^T X V = V\Lambda, \text{ where } \Lambda = \begin{pmatrix} \lambda_1(X^T X) & & & \\ & \lambda_2(X^T X) & & \\ & & \ddots & \\ & & & \lambda_p(X^T X) \end{pmatrix} \tag{4.10}$$

with $\lambda_1(X^T X) \geq \lambda_2(X^T X) \geq \cdots \lambda_p(X^T X)$ the $p$ largest eigenvalues $(p < l)$ of the correlation matrix $X^T X$.

(ii) The basis is given by the columns of the normalized matrix

$$W = XV\Lambda^{-\frac{1}{2}}. \tag{4.11}$$

Note that the matrix $W$ is also referred to as POD basis later.

In the numerical implementation, if too small (maybe even negative) eigenvalues are detected, the corresponding eigenvectors are discarded. Moreover, an extra step of orthogonalization is applied to $W$ for correction.

Throughout this section, some necessary parameters are set as follows:

$$\begin{array}{lll} T = 0.06, & N = 10000, & N_0 = 100, \\ p = 4, & n = 999, & m = 900. \end{array}$$

See (4.6) and (4.7) for $N$ and $N_0$, (4.8) for $n$ and $m$. $p = \text{rank}(W)$. Note that $p$ is possible to be reduced by algorithm automatically in the case lower rank of $W$ is detected. We set $\delta = h$ in (4.5).

## Choice of Snapshots

The first question one may ask would be whether we can extract the required vectors from the solution. To answer this, we apply the above process to $u^N$, $u^{N_0}$ and $u^{FEM}$ on the same mesh, then compare the resulting matrix $W$ to the low frequency vectors given by the Fourier series (4.2).

The comparison is carried out by SVD. Let $V_{2p} = (v_1, v_2, \cdots, v_{2p})$ with $v_i$ being the finite dimensional representation of $v_i(x)$, $Z = (V_{2p}, W)$ is a matrix whose columns are all those of $V_{2p}$ and $W$. Here $2p$ $v$-vectors are chosen to cover all possible eigenvectors extracted from $W$ because some Fourier coefficients in (4.2) are 0 and the corresponding $v$-vectors are not in the solution thus will not be extracted by POD. We apply SVD to $Z$, then the $p$ smallest singular values indicate the strength of linear relationship of the column vectors of $Z$ (the closer to 0, the more linearly dependent). Notice that $V_{2p}$ has orthogonal column vectors and $W^TW = I_p$ by construction.

In the numerical experiments, $W$ should not contain small (with respect to norm) vectors, otherwise these vectors can result in singular values close to 0. The protection is made in the POD subroutine by adding a reorthogonalization step.

**Remark 4.1.2.** Using smallest singular values as a indicator of linear dependence of column vectors in $Z$ is in fact reasonable. To see this more clearly, we let $Z = (P, P) \in \mathbb{R}^{n\times 2m}$ with $P^TP = I_m$ $(m < n)$. Thus $Z^TZ = \begin{pmatrix} I_m & I_m \\ I_m & I_m \end{pmatrix}$. Obviously the kernel of $Z^TZ$ is comprised of all vectors in the form $\begin{pmatrix} v \\ -v \end{pmatrix}$ with $v \in \mathbb{R}^m$. This means the multiplicity of the eigenvalue 0 of $Z^TZ$ equals $m$. Since the singular values of $Z$ are the square roots of the eigenvalues of $Z^TZ$, we conclude that $Z$ has exactly $m$ singular values equal to 0.

Solving the eigenvalue problem (4.10) is not cheap in a high dimensional problem, hence it is more preferable that as few snapshots as possible are taken for POD. Different subsets of snapshots are chosen and the choices are compared in the following numerical tests. Note that according to Remark 4.1.1, the low frequency vectors could be more likely extracted out of the snapshots which are away from $t = 0$, while the high frequency terms are almost absent.

In Figure 4.1, the initial condition $u_0(x)$ is set to be constant function, c.f. (4.3). Each column of points are the $p$ (sometimes less than $p$) smallest singular values of matrix $Z$ in

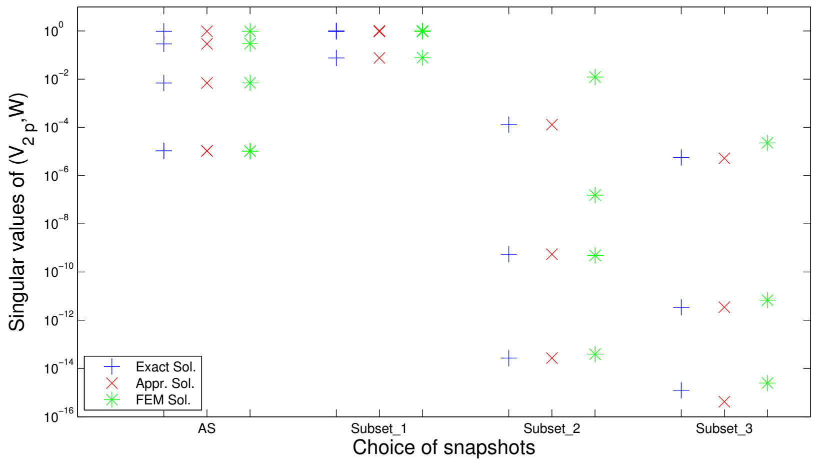


**Figure 4.1.: $p$ smallest singular values of $Z$ for different choices of snapshots and different types of solution. Initial condition is constant function.**

which the POD-like basis $W$ is computed from different subsets of snapshots of certain given solution. More specifically, the first three columns are the singular values of $Z$ when all snapshots are taken to generate $W$ and the snapshots come from exact, truncated, FEM solution respectively.

Here the marks of x-axis mean

| | |
|---|---|
| 'AS': | all snapshots , |
| 'Subset_1': | $p$ snapshots with $t \in [0, (p-1)\tau]$, |
| 'Subset_2': | $p$ snapshots with $t \in [0.02, 0.02 + (p-1)\tau]$, |
| 'Subset_3': | $p$ snapshots with $t \in [T - (p-1)\tau, T]$. |

That is to say, we use all snapshots and the subsets of snapshots taken from the beginning, the middle and the end of the time interval $[0, T]$ respectively. Notice that the last three choices only take $p$ vectors for POD, which leads to extremely low numerical cost of solving the corresponding eigenvalue problem.

We can conclude from Figure 4.1 that

(1) The three different types of solution provide very similar subspace $W$. This is natural as the solutions are almost equal.

(2) If proper snapshots are chosen for computing POD basis, the result could be better than applying POD to all snapshots, in the sense that the columns in $Z$ are more linearly dependent.

(3) The high frequency eigenvectors are more visible when $t$ is closer to 0, hence the snapshots farther away from 0 give better approximation of low frequency eigenvectors. Notice that when 'Subset_1' is taken, the singular values show highly linear independence of the column vectors in $Z$, as being pointed out at the beginning.

(4) When 'Subset_3' is taken, all the three types of solution only extracts 3 valid vectors. The reason could be when $t$ is away from 0, the solution is very small in infinite norm, thus suffers more from numerical error. The snapshot matrix may not even have full

rank.

Similar results can be seen from Figure 4.2 and Figure 4.3 where the initial condition is step function (4.4) and peak function (4.5) respectively.

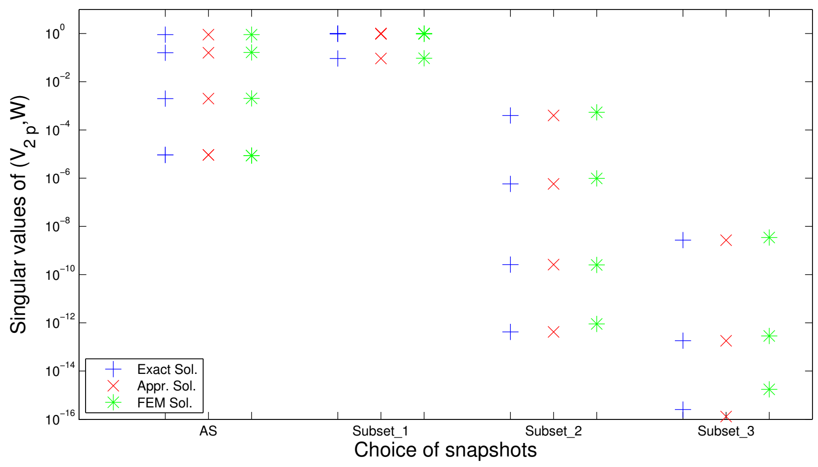


**Figure 4.2.:** $p$ **smallest singular values of** $Z$ **for different choices of snapshots and different types of solution. Initial condition is step function.**

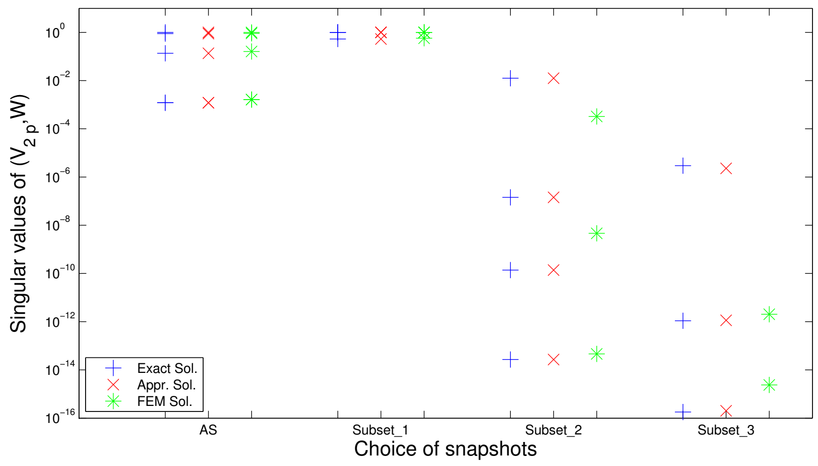


**Figure 4.3.:** $p$ **smallest singular values of** $Z$ **for different choices of snapshots and different types of solution. Initial condition is peak function.**

Note that the singular values related to FEM solution are sometimes different from exact or truncated solution. This depends on the initial condition and the choice of snapshots as well.

## Eigenvalue Deflation

We expect that the column vectors of $W$ well approximate the eigenvectors associated with the smallest eigenvalues of the coefficient matrix in the target linear equation so that we can use them for the deflation of these smallest eigenvalues to improve the condition.

As we can only have numerical solution in practice, the experiment is confined to FEM solution in this part.

The linear equation (4.9) can be solved by CG method or deflated-CG method, where the eigenvalue distribution of the coefficient matrix is critical for the efficiency of the corresponding algorithm. As a result, we compare the eigenvalues of the original coefficient matrix $A := M + \tau K$ to the auxiliary (deflated) matrix $B := H^T AH$ with $H = I - W(W^T AW)^{-1} W^T A$, (c.f. (2.27)). Eigenvalue distribution of matrix $B$ decides the rate of convergence for deflated CG method, as discussed in Section 2.3.1.

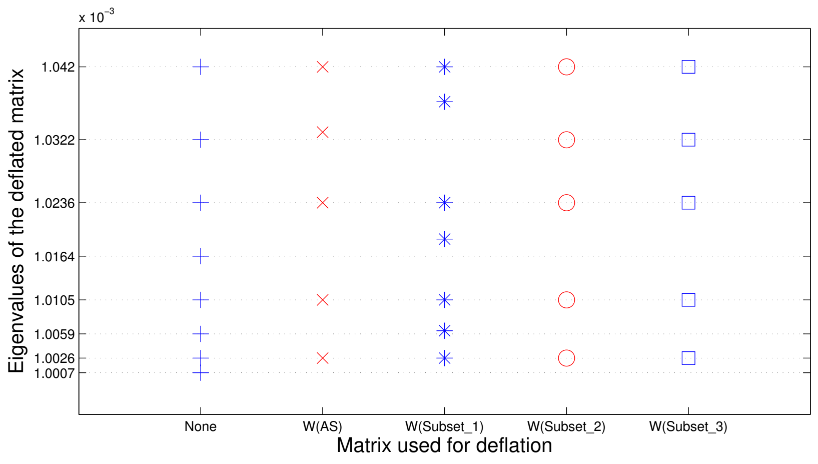


**Figure 4.4.: The smallest eigenvalues of $A$ and $B$ where $B$ is obtained by using different deflation matrix $W$. Initial condition is constant function.**

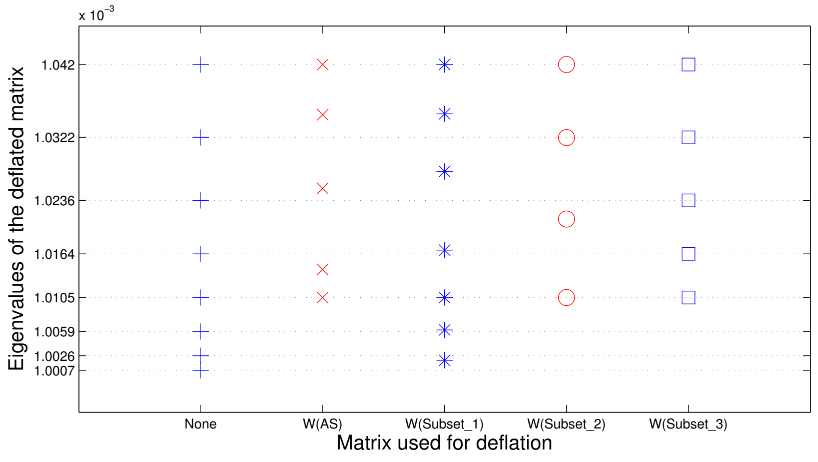


**Figure 4.5.: The smallest eigenvalues of $A$ and $B$ where $B$ is obtained by using different deflation matrix $W$. Initial condition is step function.**

Figure 4.4, Figure 4.5, Figure 4.6 show the smallest eigenvalues of $A$ and $B$, where $B$ is obtained by using different matrices $W$ extracted from different choices of snapshots, e.g. in Figure 4.4, the first column plots the smallest eigenvalues of $A$, the second plots those of $B$

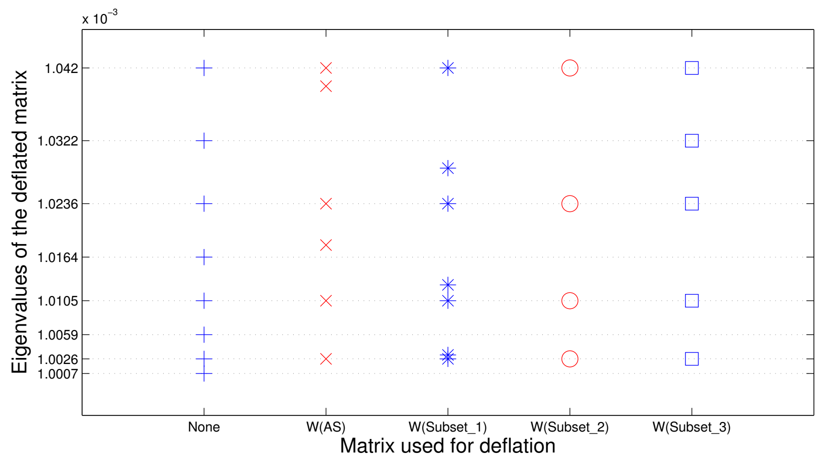


**Figure 4.6.: The smallest eigenvalues of $A$ and $B$ where $B$ is obtained by using different deflation matrix $W$. Initial condition is peak function.**

where the matrix $W$ used for deflation comes from all the snapshots.

In these three figures, the effect of deflation basically agrees with the corresponding analytic solution and the smallest singular values of $Z$. Shift of eigenvalues can be observed when $W$ provides less good approximation of eigenvectors. This is especially obvious when 'Subset_1' is taken to extract $W$.

One may notice that not all the small eigenvalues of $A$ are removed. For example, in the second column of Figure 4.4, only the odd-numbered small eigenvalues disappear. This is caused by the zero Fourier coefficients in (4.2), given by (4.3). More specifically, the terms related to such coefficients are actually not contained in the solution at all. Hence such information can not be extracted by POD, i.e. these eigenvectors are missing in $W$.

## Efficiency of CG Method

We now use (deflated) CG method to solve the heat equation step by step in time direction, then compare the average iteration numbers (total iteration numbers divided by $m$). Different deflation matrices are used. Specially, we also use $V_p$ (i.e. the first $p$ columns of $V$) for deflation. As being shown in the previous numerical experiment, the columns of $V_p$ span more or less the same subspace as those of $W$ do.

In the CG algorithm, the initial guess is given by the solution of previous linear equation, stopping criterion is based on relative reduction, i.e.

$$\frac{\|r_k\|_\infty}{\|r_0\|_\infty} \leq \epsilon,$$

the tolerance $\epsilon$ is set to be $1e-9$ here.

Note that when the initial condition is constant or step function, $W$(Subset_3) only contains 3 vectors, although we set $p = 4$ here. One invalid vector is discarded by the POD subroutine.

In Table 4.1, the first column presents the initial condition, the second column gives the

| Init. Cond. | None | W(AS) | W(Subset_1) | W(Subset_2) | W(Subset_3) | $V_p$ |
|---|---|---|---|---|---|---|
| constant | 29.6 | 11.9 | 24.4 | 8.5 | 17.3 | 19.0 |
| step | 32.9 | 21.5 | 29.9 | 16.8 | 20.9 | 20.9 |
| peak | 32.3 | 19.2 | 28.8 | 12.6 | 20.1 | 21.2 |

**Table 4.1.: Average iteration numbers of CG method with different deflation matrices for different initial conditions.**

average iteration numbers of standard CG method for solving linear equations (4.9). The next columns are the average iteration numbers for deflated CG method using different deflation matrix. The reduction in iteration number when deflated CG method is applied is reasonable since some small eigenvalues are deflated, c.f. Figure 4.4, Figure 4.5, Figure 4.6.

The corresponding CPU time is presented in Table 4.2. The figures are averages of 20 repeated computations.

| Init. Cond. | None | W(AS) | W(Subset_1) | W(Subset_2) | W(Subset_3) | $V_p$ |
|---|---|---|---|---|---|---|
| constant | 2.59 | 1.83 | 3.60 | 1.32 | 2.60 | 2.80 |
| step | 2.89 | 3.21 | 4.41 | 2.53 | 3.11 | 3.09 |
| peak | 2.83 | 2.86 | 4.26 | 1.93 | 4.61 | 3.20 |

**Table 4.2.: CPU time (in second) of CG method with different deflation matrices for different initial conditions.**

Combining the results in Table 4.1 and Table 4.2, we can conclude that our deflation strategy can possibly save much computational effort. However, in many cases, CPU time of deflated CG method is more than that of standard CG although iteration numbers are less. This is mainly because each iteration of deflated CG method involves one extra matrix-vector product and solution of a small linear equation, see Step 9 in Algorithm 2.5. Such additional cost takes significant proportion in 1D problem, however, its weight will be reduced in higher dimensional problems, when the coefficient matrix is more sparse. This will take place in the test problems of next sections.

## Deflation Matrix for One-Shot Method

Instead of solving time steps sequentially, one-shot method computes all the time steps at the same time. In the case of our test problem (4.1), it means we need to solve the following linear equation:

$$\begin{pmatrix} M+\tau K & & & \\ -M & M+\tau K & & \\ & \ddots & \ddots & \\ & & -M & M+\tau K \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} Mu_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} =: g. \tag{4.12}$$

Let $B$ be the coefficient matrix in (4.12), and $\bar{u} = (u_1^T, u_2^T, \dots, u_m^T)^T$, then (4.12) can be rewritten into

$$B\bar{u} = g.$$

Since $B$ is not symmetric, a nonsymmetric solver like GMRES can be applied to solve (4.12). The spectrum of $B$ is dominant for the rate of convergence.

However, our aim is not to solve (4.12), but to examine whether we can still attack the smallest eigenvalues of the big matrix $B$ using POD basis by constructing a proper deflation matrix.

We first split matrix $B$ and learn some characteristics of its eigenvalue distribution, which is surely related to the spectral properties of its nonzero blocks. Let

$$D = \begin{pmatrix} M+\tau K & & & \\ & M+\tau K & & \\ & & \ddots & \\ & & & M+\tau K \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 0 & & & \\ -M & 0 & & \\ & \ddots & \ddots & \\ & & -M & 0 \end{pmatrix},$$

then $B = D + R$.

It is not hard to see $\|R\|_2 = \|M\|_2 \leq h$. For any eigenvalue $\hat{\lambda}$ of $B$, there exists one eigenvalue $\lambda$ of $D$, such that

$$|\hat{\lambda} - \lambda| \leq \|R\|_2 \leq h, \tag{4.13}$$

see e.g. Corollary 6.3.4 in Horn and Johnson (1985). Hence $B$ has $n$ clusters of eigenvalues which have values close to the eigenvalues of $M + \tau K$.

Suppose we have a matrix $\widetilde{W}$ for deflation of the smallest eigenvalues of $M + \tau K$, we give a corresponding deflation matrix for $B$ by

$$W = \begin{pmatrix} \widetilde{W} & & & \\ & \widetilde{W} & & \\ & & \ddots & \\ & & & \widetilde{W} \end{pmatrix} = I_m \otimes \widetilde{W}, \tag{4.14}$$

where $\otimes$ denotes Kronecker product.

Using $W$ for deflation, we in fact project the equation (4.12) onto a lower dimensional subspace by projecting each $u_i$ $(i = 1, 2, \ldots, m)$ onto $\widetilde{\mathcal{W}}$ (subspace spanned by all the column vectors of $\widetilde{W}$).

Furthermore, all the column vectors of $W$ approximate the eigenvectors of $D$ based on the numerical results in the previous experiments, thus they can be treated as rough eigenvectors of $B$ regarding $R$ a perturbation matrix (c.f. (4.13)). Then we expect that the smallest $p \ \big(= \operatorname{rank}(\widetilde{W})\big)$ clusters of eigenvalues of $B$ can be deflated by $W$. Notice that matrix $B$ is not symmetric, thus it possibly has complex eigenvalues. We use absolute values for plotting instead.

The result in Figure 4.7 agrees with what we expect. It is very similar to Figure 4.4, except that the eigenvalues appear in clusters.

When the discretization gets finer by the same factor in time and space, the eigenvalues of $B$ are more distributed, so do the eigenvalues of the corresponding deflated matrices. Nevertheless, the deflation matrix still capture the right clusters of eigenvalues, according to Figure 4.8

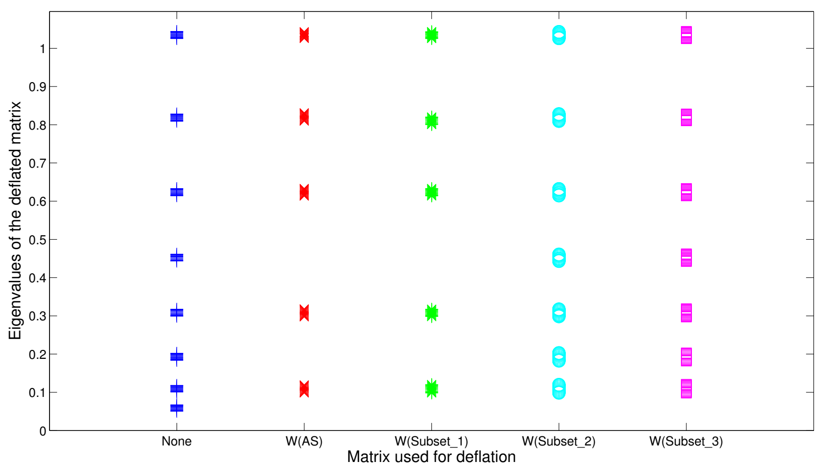


**Figure 4.7.: The smallest eigenvalues of $B$ and $B_{def}$, where $B_{def}$ is obtained by using different deflation matrix $\widetilde{W}$. Initial condition is constant function. $T = 1$, $n = 23$, $m = 24$. The eigenvalues here are given in absolute value.**

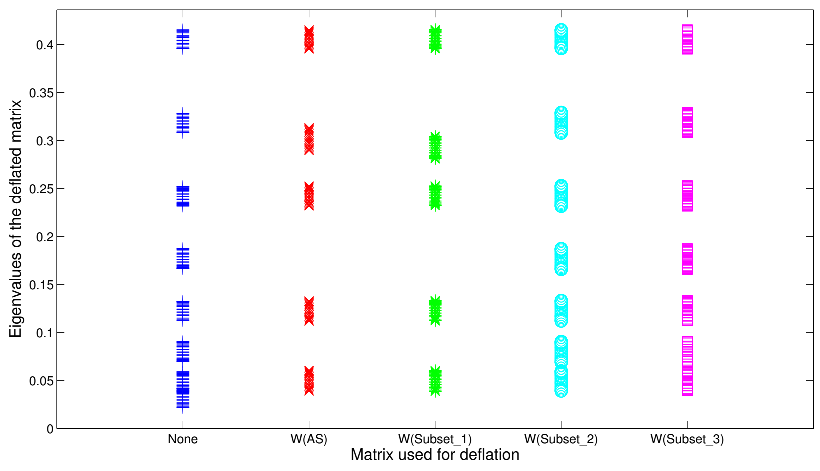


**Figure 4.8.: The smallest eigenvalues of $B$ and $B_{def}$ where $B_{def}$ is obtained by using different deflation matrix $\widetilde{W}$. Initial condition is constant function. $T = 1$, $n = 39$, $m = 40$. The eigenvalues here are given in absolute value.**

Preciser analysis on the smallest eigenvalues of $B$ could be tricky. The smallest eigenvalues seem to tend to 0 when the grid gets finer (according to Figure 4.7 and Figure 4.8), hence they are very sensitive to perturbation. This raises the difficulty of explaining why $W$ can approximate required eigenvectors of $B$.

We show the following lemma, which gives a rough explanation by investigating a necessary condition.

**Lemma 4.1.3.** *Assume*

*(i) the mass matrix $M$ satisfies $\|M\|_2 \leq c_1 h^d$, with $c_1$ constant, $h$ denotes mesh size and $d$ is spatial dimension;*

*(ii) POD basis (matrix form) $\widetilde{W}$ approximates some eigenvectors of the stiffness matrix $K$ related to its smallest eigenvalues, i.e. $\|K\widetilde{W}\|_2 \leq c_2$, with $c_2$ small constant.*

*Let $W = I_m \otimes \widetilde{W}$, then we have $\|BW\|_2 \leq 2c_1h^d + \tau c_2$ and $\|B^TW\|_2 \leq 2c_1h^d + \tau c_2$, where $B$ is coefficient matrix in (4.12).*

*Proof.*

$$\begin{aligned}
\|BW\|_2 &= \left\| \begin{pmatrix} (M+\tau K)\widetilde{W} & & & \\ -M\widetilde{W} & (M+\tau K)\widetilde{W} & & \\ & \ddots & \ddots & \\ & & -M\widetilde{W} & (M+\tau K)\widetilde{W} \end{pmatrix} \right\|_2 \\
&\leq \left\| I_m \otimes \left( (M+\tau K)\widetilde{W} \right) \right\|_2 + \left\| \begin{pmatrix} 0 & & & \\ -M\widetilde{W} & 0 & & \\ & \ddots & \ddots & \\ & & -M\widetilde{W} & 0 \end{pmatrix} \right\|_2 \\
&= \|(M+\tau K)\widetilde{W}\|_2 + \|M\widetilde{W}\|_2 \\
&\leq \|M\widetilde{W}\|_2 + \tau\|K\widetilde{W}\|_2 + \|M\widetilde{W}\|_2 \\
&\leq 2\|M\widetilde{W}\|_2 + \tau c_2 \\
&\leq 2c_1h^d + \tau c_2
\end{aligned}$$

Similarly, $\|B^TW\|_2 \leq 2c_1h^d + \tau c_2$. □

Assumption (i) in Lemma 4.1.3 can be satisfied by choosing appropriate basis functions for spatial discretization. For some examples, see Section 1.6 in Elman et al. (2005). Assumption (ii) is reasonable based on the previous numerical tests.

**Remark 4.1.4.** When $\widetilde{W}$ well approximates some eigenvectors of $K$ related to nearly 0 eigenvalues, $K\widetilde{W}$ should be a almost null matrix, thus $\|K\widetilde{W}\|_2$ is very small. If $W$ approximates the left most eigenspace of $B$, then $\|BW\|_2$ should also be close to 0. The second term of the upper bound given by Lemma 4.1.3, i.e. $\tau c_2$, is indeed small by assumption (ii). Notice that $\tau c_2$ could be relatively larger if we embed some random matrices instead of $\widetilde{W}$ in $B$. In such a case, this term drives the upper bound away from 0.

## 4.2. Deflation with POD for a Linear Quadratic Problem

### 4.2.1. A Linear Quadratic Problem and its Optimality System

In the last section, we provided a deflation matrix for one-shot method in the case of solving a PDE. In order to illustrate the fact that a deflation matrix based on (4.14) can still capture desired eigenvalues when one shot scheme is adopted for solving PDE-constrained optimization problem, we first carry out some experiments on the following 3D linear quadratic problem:

$$
\begin{aligned}
\min \quad & \frac{1}{2}\int_0^T\int_\Omega (\mathbf{y}(x,t)-\bar{\mathbf{y}}(x,t))^2 \,\mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T\int_\Omega (\mathbf{u}(x,t))^2 \,\mathrm{d}x\mathrm{d}t \\
\text{s.t.} \quad & \mathbf{y}_t - \Delta \mathbf{y} = \mathbf{u} \qquad \text{in } \Omega, \\
& \mathbf{y} = 0 \qquad \text{on } \partial\Omega, \\
& \mathbf{y}(x,0) = \mathbf{y}_0(x).
\end{aligned} \tag{4.15}
$$

with $\Omega = (0,1)^3$, $T = 1$.

According to Stoll and Wathen (2010), the adjoint equation of the PDE in (4.15) is

$$
\begin{aligned}
-\mathbf{p}_t - \Delta \mathbf{p} &= \chi_\Omega(\mathbf{y} - \bar{\mathbf{y}}), \\
\mathbf{p} &= 0, \quad \text{on } \partial\Omega, \\
\mathbf{p}(x,T) &= 0,
\end{aligned} \tag{4.16}
$$

where $\chi_\Omega$ is an indicator function for the domain $\Omega$.

The discretization is done exactly like in Stoll and Wathen (2010). Using trapezoidal rule for approximating integral, linear spline basis for FEM on a uniform grid and implicit Euler for time discretization ($\tau := T/m$ denotes time difference), a discretize-then-optimize approach leads to such a KKT system

$$
\begin{pmatrix} \tau A & 0 & B^T \\ 0 & \beta\tau A & -\tau C^T \\ B & -\tau C & 0 \end{pmatrix}
\begin{pmatrix} y \\ u \\ p \end{pmatrix} = \begin{pmatrix} \tau A\bar{y} \\ 0 \\ d \end{pmatrix}, \tag{4.17}
$$

with $y = (y_1^T, \ldots, y_m^T)^T, u = (u_1^T, \ldots, u_m^T)^T \in \mathbb{R}^{mn^3}$, $A = \mathrm{blkdiag}(\frac{1}{2}M, M, \ldots, M, \frac{1}{2}M)$, $C = \mathrm{blkdiag}(M, \ldots, M) \in \mathbb{R}^{mn^3 \times mn^3}$, $d = ((My_0)^T, 0, \ldots, 0)^T + \tau Cf \in \mathbb{R}^{mn^3}$ and

$$
B = \begin{pmatrix} M+\tau K & & & \\ -M & M+\tau K & & \\ & \ddots & \ddots & \\ & & -M & M+\tau K \end{pmatrix}, \tag{4.18}
$$

Here $m$ and $n$ are number of time steps and number of grid points in each spatial dimension respectively.

We can either solve (4.17) directly or solve an equivalent linear system of lower dimension, which can be obtained by eliminating the control variable like in Simoncini (2012).

$$
\begin{cases}
\begin{pmatrix} \tau A & B^T \\ B & -\frac{\tau}{\beta} C A^{-1} C^T \end{pmatrix} \begin{pmatrix} y \\ p \end{pmatrix} = \begin{pmatrix} \tau A \bar{y} \\ d \end{pmatrix}, \\
u = \frac{1}{\beta} A^{-1} C^T p.
\end{cases} \tag{4.19}
$$

Notice that $CA^{-1}C^T = \mathrm{blkdiag}(2M, M, \ldots, M, 2M)$. Thus we only need to solve the linear equation in (4.19) (denoted by $\mathcal{A}\mathbf{x} = \mathbf{b}$) and evaluate $u$ explicitly to get the discrete solution of (4.15). The matrix $\mathcal{A}$ is symmetric but not necessarily positive definite, hence we choose to use MINRES.

### 4.2.2. Deflation Strategy

First of all, we take snapshots of both $y$ and $p$ (the discrete solution to the state equation and the adjoint equation respectively) and compute $\widetilde{W} \in \mathbb{R}^{n^3 \times l}$ via POD, such a $\widetilde{W}$ provides POD basis for the state equation and adjoint equation at the same time.

The matrix $\widehat{W}$ applied in the deflated MINRES algorithm can be constructed to be a block diagonal matrix as

$$\widehat{W} := \begin{pmatrix} W & \\ & W \end{pmatrix} \in \mathbb{R}^{2mn^3 \times 2ml}, \tag{4.20}$$

where

$$W = I_m \otimes \widetilde{W} = \begin{pmatrix} \widetilde{W} & & & \\ & \widetilde{W} & & \\ & & \ddots & \\ & & & \widetilde{W} \end{pmatrix} \in \mathbb{R}^{mn^3 \times ml} \tag{4.21}$$

is constructed similar to (4.14).

The deflation matrix $\widehat{W}$ actually projects each time instance of both state variable $y$ and adjoint variable $p$ into the POD subspace, this makes sense since $y$ and $p$ are solutions of PDEs for which POD can be applied.

If the assumptions of Lemma 4.1.3 are satisfied, we have

$$\begin{aligned}
&\left\| \begin{pmatrix} \tau A & B^T \\ B & -\frac{\tau}{\beta} C A^{-1} C^T \end{pmatrix} \begin{pmatrix} W & \\ & W \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \tau A W & B^T W \\ B W & -\frac{\tau}{\beta} C A^{-1} C^T W \end{pmatrix} \right\|_2 \\
\leq\ & \|\tau A W\|_2 + \|B^T W\|_2 + \|BW\|_2 + \left\| -\tfrac{\tau}{\beta} C A^{-1} C^T W \right\|_2 \\
\leq\ & 4c_1 h^d + 2\tau c_2 + \tau \left\| \operatorname{diag}\left( \tfrac{1}{2} M\widetilde{W}, M\widetilde{W}, \dots, M\widetilde{W}, \tfrac{1}{2} M\widetilde{W} \right) \right\|_2 \\
& + \tfrac{\tau}{\beta} \left\| \operatorname{diag}\left( 2M\widetilde{W}, M\widetilde{W}, \dots, M\widetilde{W}, 2M\widetilde{W} \right) \right\|_2 \\
\leq\ & 4c_1 h^d + 2\tau c_2 + \tau \|M\widetilde{W}\|_2 + \tfrac{2\tau}{\beta} \|M\widetilde{W}\|_2 \\
\leq\ & \left( 4 + \tau + \tfrac{2\tau}{\beta} \right) c_1 h^d + 2\tau c_2
\end{aligned} \tag{4.22}$$

Thus we expect the deflation matrix $\widehat{W}$ can still approximate expected eigenvectors of the downsized coefficient matrix thus capture the smallest eigenvalues of $\mathcal{A}$.

Using deflation strategy requires extra matrix-vector multiplication with $W$ in a deflated Krylov subspace method (see Section 2.3), but this can not be very expensive because of the sparsity of $W$. In fact, the number of nonzero entries of $W$ is no more than a dense matrix in $\mathbb{R}^{2mn^3 \times l}$.

According to (4.10), extracting $\widetilde{W}$ requires the solving of an eigenvalue problem in $\mathbb{R}^{2m}$ for $l$ eigenpairs. The size of the relative eigenvalue problem is much smaller compared to the linear system to be solved, thus it is no trouble in our tests. However, in a real application, $m$ could be large, the cost of solving eigenvalue problem should be taken into consideration. In such a case, using a smart subset of snapshots for POD could be necessary, as being illustrated in Section 4.1.2.

### 4.2.3. Numerical Experiments

In this part of numerical experiments, we set

$$\mathbf{y}_0(x) = 0, \quad \bar{\mathbf{y}}(x,t) = 1000tx_1(x_1-1)x_2(x_2-1)x_3(x_3-1),$$

thus the optimization problem (4.15) has solution

$$\mathbf{y}^*(x,t) = \bar{\mathbf{y}}(x,t),$$

$$\begin{aligned} \mathbf{u}^*(x,t) &= -2t\big(x_2(x_2-1)x_3(x_3-1) + x_1(x_1-1)x_3(x_3-1) + x_1(x_1-1)x_2(x_2-1)\big) \\ &\quad + x_1(x_1-1)x_2(x_2-1)x_3(x_3-1). \end{aligned}$$

This experiment is designed to verify our deflation strategy for the linear problem (4.19), rather than developing a solver for the linear quadratic problem (4.15).

Because we need the solution for computing POD basis, we first solve the linear system (4.19) with standard MINRES, then use the solution to build deflation matrix $\widehat{W}$ as described earlier and apply deflated MINRES to $\mathcal{A}\mathbf{x} = \mathbf{b}$. Notice that we do not use the known exact solution for POD but only for checking the numerical solution. The results obtained by such an approach are provided in the columns 'Defl. MINRES' in Table 4.3.

Deflated MINRES starts its iteration from the solution in the POD subspace (see Algorithm 2.6), in the sense that each time instance of $y$ and $p$ is in the POD subspace $\widetilde{W}$. More precisely, the initial point of Deflated MINRES is the solution of $\widehat{W}^T\mathcal{A}\widehat{W}\mathbf{x}_0 = \widehat{W}^T\mathbf{b}$, which can be considered to be the reduced order model of (4.19), hence a good initial guess in this case.

For comparison, we start standard MINRES from the same initial point as deflated MINRES, the results are presented under 'Init. MINRES' in Table 4.3.

Table 4.3 shows the results of the above solvers with respect to different values of regularization parameter $\beta$ and different dimension $l$ of POD subspace used for deflation. The discretized problem has 15 DOF in each spatial dimension and 16 time steps, hence the linear equation $\mathcal{A}\mathbf{x} = \mathbf{b}$ has dimension $108,000$. Notice that no preconditioner is used here.

Because of the good initial guess, the initial residuals are fairly small in number in all the cases. If the POD subspace used for deflation includes more vectors, it is natural that the initial guess is better, which results in even smaller initial residual of MINRES iteration. This fact is reflected by increasing the dimension of POD subspace $l$.

Another influencing factor in Table 4.3 is the regularization parameter $\beta$, which balances the weight of mass matrix and stiffness matrix in coefficient matrix $\mathcal{A}$ and leads to different efficiency of deflation. This effect could be highly problem dependent. The impact of $\beta$ in our case can be easily seen from the last column of Table 4.3, which denotes the percentage of iteration numbers reduced by embedding our deflation strategy.

| $\beta$ | $l$ | Init. Res. | Defl. MINRES | | Init. MINRES | | #it Red. |
|---|---|---|---|---|---|---|---|
| | | | #it | time | #it | time | |
| 1e-2 | - | 1.32e-5 | 526 | 30.30 | - | - | - |
| | 1 | 8.56e-7 | 133 | 9.02 | 344 | 19.91 | 61% |
| | 2 | 9.00e-9 | 57 | 4.17 | 132 | 7.66 | 57% |
| | 3 | 8.87e-11 | 33 | 2.54 | 39 | 2.39 | 15% |
| | 4 | 9.06e-13 | 9 | 0.87 | 11 | 0.72 | 18% |
| 1e-3 | - | 1.32e-5 | 779 | 44.51 | - | - | - |
| | 1 | 5.13e-6 | 224 | 14.66 | 487 | 27.94 | 54% |
| | 2 | 5.12e-8 | 100 | 7.11 | 276 | 16.08 | 64% |
| | 3 | 8.31e-10 | 68 | 5.26 | 104 | 6.08 | 35% |
| | 4 | 3.02e-11 | 17 | 1.45 | 26 | 1.59 | 35% |
| 1e-4 | - | 1.32e-5 | 1733 | 99.64 | - | - | - |
| | 1 | 3.10e-5 | 722 | 47.85 | 1157 | 66.19 | 38% |
| | 2 | 3.28e-6 | 467 | 32.53 | 742 | 42.95 | 37% |
| | 3 | 7.50e-9 | 205 | 15.23 | 372 | 21.29 | 45% |
| | 4 | 6.85e-10 | 110 | 8.69 | 209 | 12.06 | 47% |
| 1e-5 | - | 1.32e-5 | 1290 | 73.84 | - | - | - |
| | 1 | 6.65e-5 | 735 | 48.73 | 857 | 49.25 | 14% |
| | 2 | 1.09e-5 | 391 | 27.27 | 762 | 43.68 | 49% |
| | 3 | 3.61e-8 | 187 | 13.90 | 508 | 29.41 | 63% |
| | 4 | 5.38e-9 | 46 | 3.65 | 280 | 16.27 | 84% |
| 1e-6 | - | 1.32e-5 | 565 | 32.31 | - | - | - |
| | 1 | 8.29e-5 | 334 | 22.03 | 337 | 19.36 | 1% |
| | 2 | 7.89e-6 | 162 | 11.39 | 372 | 21.43 | 56% |
| | 3 | 3.10e-8 | 78 | 5.93 | 227 | 13.40 | 66% |
| | 4 | 1.50e-9 | 10 | 0.90 | 114 | 6.72 | 91% |

**Table 4.3.: Comparison of deflated MINRES and POD-initialized MINRES for distributed control problem (4.15). Time is given in second.**

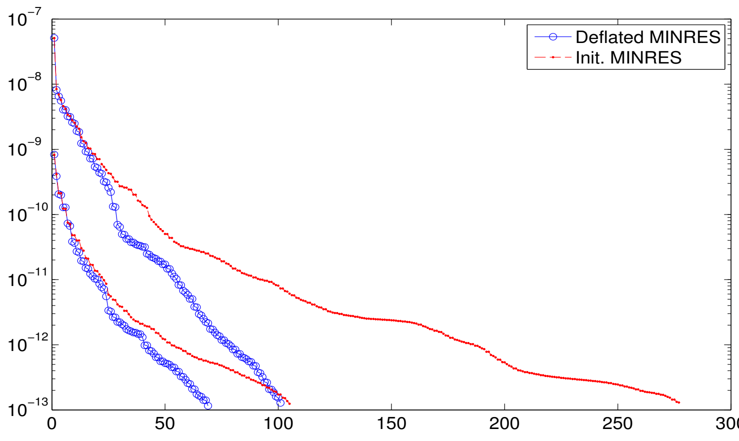


**Figure 4.9.: Iteration history of deflated MINRES and POD-initialized MINRES with different POD basis. $\beta = 1e-3$, $l = 2$ and $l = 3$.**

Figure 4.9 plots the iteration history of the case $\beta = 1e-3$, $l = 2$ and $l = 3$, which is intuitive that our deflation strategy indeed accelerates the convergence by improving the slope. The other cases perform in a similar way.

We conclude that deflation using POD subspace still works as expected when one shot scheme is used, in addition to what we verified in Section 4.1.2.

## 4.3. Preconditioned Deflated Algorithm for a Nonlinear Optimization Problem

For practical use, a preconditioner should be employed in a Krylov subspace method. We need to verify whether the deflation strategy developed in the last section works along with common preconditioning or not.

### 4.3.1. SQP Framework for Solving a Nonlinear Boundary Control Problem

We take the following 3D nonlinear Dirichlet boundary control problem for test:

$$\begin{aligned}
\min \quad & \frac{1}{2}\int_0^T\int_\Omega (\mathbf{y}(x,t)-\bar{\mathbf{y}}(x,t))^2\,\mathrm{d}x\mathrm{d}t + \frac{\beta}{2}\int_0^T\int_{\partial\Omega}(\mathbf{u}(x,t))^2\,\mathrm{d}x\mathrm{d}t =: \mathbf{J}(\mathbf{y},\mathbf{u}) \\
\text{s.t.} \quad & \mathbf{y}_t - \Delta\mathbf{y} - \mathbf{f}(\mathbf{y}) = 0 \qquad \text{in } \Omega, \\
& \mathbf{y} = \mathbf{u} \qquad \text{on } \partial\Omega, \\
& \mathbf{y}(x,0) = \mathbf{y}_0(x).
\end{aligned} \tag{4.23}$$

Here $\Omega = (0,1)^3$, $T = 1$, $\mathbf{f}(\mathbf{y})$ is a nonlinear function, $\bar{\mathbf{y}}$ is desired state.

The problem is handled with a discretize-then-optimize approach very similar to the linear quadratic problem in the previous section. As well, we use one-shot scheme to deal with the time instances.

We use $Q1$-element for the finite element discretization on a uniform grid in space. Implicit Euler scheme is used for time discretization. Like in Rees, Stoll and Wathen (2010), we first partition the mass matrix $M$ and stiffness matrix $K$ as follows:

$$M = \begin{pmatrix} M_{\mathrm{II}} & M_{\mathrm{BI}} \\ M_{\mathrm{IB}} & M_{\mathrm{BB}} \end{pmatrix}, \qquad K = \begin{pmatrix} K_{\mathrm{II}} & K_{\mathrm{BI}} \\ K_{\mathrm{IB}} & K_{\mathrm{BB}} \end{pmatrix}.$$

Here the subscript 'I' denotes the index set of interior nodes of $\Omega$ while 'B' denotes that of the boundary nodes. Thus for instance, $M_{\mathrm{BB}}$ denotes the boundary mass matrix. Obviously $M_{\mathrm{BI}} = M_{\mathrm{IB}}^T$ and $K_{\mathrm{BI}} = K_{\mathrm{IB}}^T$.

Using trapezoidal rule to approximate the integrals, the objective functional in (4.23) becomes

$$J(y,u) = \frac{\tau}{2}(y-\bar{y})^T A (y-\bar{y}) + \frac{\beta\tau}{2}u^T\widetilde{A}u, \tag{4.24}$$

where $y = (y_1^T, y_2^T, \ldots, y_m^T)^T$, $u = (u_1^T, u_2^T, \ldots, u_m^T)^T$ are column vectors consisting of solutions at all time steps, $A = \mathrm{blkdiag}(\frac{1}{2}M_{\mathrm{II}}, M_{\mathrm{II}}, \ldots, M_{\mathrm{II}}, \frac{1}{2}M_{\mathrm{II}})$, $\widetilde{A} = \mathrm{blkdiag}(\frac{1}{2}M_{\mathrm{BB}}$, $M_{\mathrm{BB}}$,

..., $M_{\mathrm{BB}}$, $\frac{1}{2}M_{\mathrm{BB}}$), $m$ and $\tau$ are the same as in the last section.

Let $C = \mathrm{blkdiag}(M_{\mathrm{II}}, \ldots, M_{\mathrm{II}})$, $\widetilde{C} = \mathrm{blkdiag}(M_{\mathrm{BI}}, \ldots, M_{\mathrm{BI}})$, $f_{\mathrm{I}}(y)$ and $f_{\mathrm{B}}(u)$ are pointwise finite dimensional approximations (vector valued) of $\mathbf{f}(\mathbf{y})$ evaluated at $y$ and $u$ respectively,

$$B = \begin{pmatrix} M_{\mathrm{II}} + \tau K_{\mathrm{II}} & & & \\ -M_{\mathrm{II}} & M_{\mathrm{II}} + \tau K_{\mathrm{II}} & & \\ & \ddots & \ddots & \\ & & -M_{\mathrm{II}} & M_{\mathrm{II}} + \tau K_{\mathrm{II}} \end{pmatrix},$$

$$\widetilde{B} = \begin{pmatrix} M_{\mathrm{BI}} + \tau K_{\mathrm{BI}} & & & \\ -M_{\mathrm{BI}} & M_{\mathrm{BI}} + \tau K_{\mathrm{BI}} & & \\ & \ddots & \ddots & \\ & & -M_{\mathrm{BI}} & M_{\mathrm{BI}} + \tau K_{\mathrm{BI}} \end{pmatrix},$$

then embedding boundary control into the PDE constraint (c.f. Rees, Stoll and Wathen (2010)) in (4.23) leads to the following discrete constraint:

$$e(y,u) := By - \tau C f_{\mathrm{I}}(y) + \widetilde{B}u - \tau \widetilde{C} f_{\mathrm{B}}(u) = 0. \tag{4.25}$$

Now we need to solve the nonlinear constrained optimization problem

$$\begin{aligned} \min \quad & J(y,u) \\ \text{s.t.} \quad & e(y,u) = 0. \end{aligned} \tag{4.26}$$

There are many ways to solve this kind of problem, for such a discussion see e.g. Herzog (2010). Here we adopt an SQP method. Equivalently, we apply Newton's method to the Lagrangian of (4.26), which is formulated as

$$L(y,u,p) = \frac{\tau}{2}(y-\bar{y})^T A (y-\bar{y}) + \frac{\beta\tau}{2} u^T \widetilde{A} u + p^T e(y,u), \tag{4.27}$$

where $p$ denotes the Lagrange multiplier. After simple calculation, the $k$-th Newton step reads

$$\begin{pmatrix} \tau A + P_k & 0 & Q_k^T \\ 0 & \beta\tau\widetilde{A} + \widetilde{P}_k & \widetilde{Q}_k^T \\ Q_k & \widetilde{Q}_k & 0 \end{pmatrix} \begin{pmatrix} \delta y_k \\ \delta u_k \\ \delta p_k \end{pmatrix} = - \begin{pmatrix} \tau A(y_k - \bar{y}) + Q_k^T p_k \\ \beta\tau\widetilde{A}u_k + \widetilde{Q}_k^T p_k \\ B(y_k) + \widetilde{B}(u_k) \end{pmatrix}, \tag{4.28}$$

with

$$\begin{aligned} P_k &= -\tau \mathrm{diag}\left(f_{\mathrm{I}}''(y_k) \circ (C^T p_k)\right), & Q_k &= B - \tau C \mathrm{diag}\left(f_{\mathrm{I}}'(y_k)\right), \\ \widetilde{P}_k &= -\tau \mathrm{diag}\left(f_{\mathrm{B}}''(u_k) \circ (\widetilde{C}^T p_k)\right), & \widetilde{Q}_k &= \widetilde{B} - \tau \widetilde{C} \mathrm{diag}\left(f_{\mathrm{B}}'(u_k)\right). \end{aligned}$$

Here "$\circ$" denotes (componentwise) Hadamard product, $f_{\mathrm{I}}'(y)$ and $f_{\mathrm{I}}''(y)$ are componentwise derivatives of $f_{\mathrm{I}}(y)$, the same to $f_{\mathrm{B}}'(u)$ and $f_{\mathrm{B}}''(u)$.

For a similar derivation of such an approach applied to a nonlinear PDE-constrained optimization problem, we refer to Benzi et al. (2011).

In terms of the new iterate, (4.28) can be rewritten into

$$\begin{pmatrix} \tau A + P_k & 0 & Q_k^T \\ 0 & \beta\tau\widetilde{A} + \widetilde{P}_k & \widetilde{Q}_k^T \\ Q_k & \widetilde{Q}_k & 0 \end{pmatrix} \begin{pmatrix} y_{k+1} \\ u_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} \tau A\bar{y} + P_k y_k \\ \widetilde{P}_k u_k \\ g(y_k, u_k) \end{pmatrix}, \tag{4.29}$$

with $g(y_k, u_k) = \tau C\big(f_{\rm I}(y_k) - f_{\rm I}'(y_k) \circ y_k\big) + \tau\widetilde{C}\big(f_{\rm B}(u_k) - f_{\rm B}'(u_k) \circ u_k\big)$.

Solving (4.29) is equivalent to solving the following system

$$\begin{cases} \begin{pmatrix} \tau A + P_k & Q_k^T \\ Q_k & -\widetilde{Q}_k\left(\beta\tau\widetilde{A} + \widetilde{P}_k\right)^{-1}\widetilde{Q}_k^T \end{pmatrix} \begin{pmatrix} y_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} \tau A\bar{y} + P_k y_k \\ h(y_k, u_k) \end{pmatrix}, \\ u_{k+1} = \left(\beta\tau\widetilde{A} + \widetilde{P}_k\right)^{-1}\left(\widetilde{P}_k u_k - \widetilde{Q}_k^T p_{k+1}\right). \end{cases} \tag{4.30}$$

with $h(y_k, u_k) = g(y_k, u_k) - \widetilde{Q}_k\left(\beta\tau\widetilde{A} + \widetilde{P}_k\right)^{-1}\widetilde{P}_k u_k$. The first equation in (4.30) will be referred to as $\mathcal{A}_k\mathbf{x}_k = \mathbf{b}_k$.

The idea of solving a relatively easier equivalent downsized system instead of the original system proposed in Simoncini (2012) is based on the assumption that part of the variables are easy to evaluate in the saddle point problem. In our case, transforming (4.29) into (4.30) requires $u_{k+1}$ is easy to evaluate, more precisely, the inversion of $\beta\tau\widetilde{A} + \widetilde{P}_k$ is cheap. To achieve this, we use diagonally lumped matrix $M$ throughout the numerical experiments in this section, which makes $\beta\tau\widetilde{A} + \widetilde{P}_k$ a diagonal matrix.

Eliminating the second equation of (4.29) and explicitly evaluating control variable $u$ leaves only $y$ and $p$ in the linear equation to be solved. In case the PDE constraint does not contain a nonlinear functional, $p$ could be interpreted as the solution to the adjoint equation, which is a PDE similar to the constraint, see (4.16). Hence $y$ and $p$ would be solutions to PDEs, the numerical solution of which could benefit from using POD basis. However, in the nonlinear problem (4.23), the coupling of $y$ and Lagrange multiplier $p$ is obscure because of nonlinearity. Despite this, we still exploit POD basis to gain numerical benefits.

### 4.3.2. Preconditioning

Let $n$ denote DOF in each dimension, then the coefficient matrix in (4.30)

$$\mathcal{A}_k = \begin{pmatrix} \tau A + P_k & Q_k^T \\ Q_k & -R_k \end{pmatrix} \in \mathbb{R}^{2mn^3 \times 2mn^3},$$

with $R_k = \widetilde{Q}_k\left(\beta\tau\widetilde{A} + \widetilde{P}_k\right)^{-1}\widetilde{Q}_k^T$.

Since $\mathcal{A}_k$ is symmetric but not necessarily positive definite, MINRES is preferable for solving the linear equation in (4.30). Now we focus on how to precondition this linear equation with an block structured preconditioner and using deflation strategy to accelerate the convergence of MINRES by further removing the smallest eigenvalues. For deflation we extract POD basis from the solution to a previous Newton step and expect it to not only provide a good initial guess for the current step but also capture the desired eigenvalues.

Aiming to use deflation strategy combined with a preconditioner, we employ the deflated MINRES algorithm derived in Olshanskii and Simoncini (2010), see Algorithm 2.6 in Section 2.3.2.

In general, selecting a proper preconditioner requires much consideration regarding the type of iterative solver being used as well as the structure and properties of the coefficient matrix, for such discussion we refer to Benzi et al. (2005).

Similarly as in Simoncini (2012), here we simply use the ideal block diagonal preconditioner suggested in Silvester and Wathen (1994):

$$\mathcal{P}_k^{bd} = \begin{pmatrix} \tau A + P_k & \\ & R_k + Q_k \left(\tau A + P_k\right)^{-1} Q_k^T \end{pmatrix}.$$

The clustering property of $\mathcal{P}_k^{bd}$ is analyzed for a discrete Stokes problem in Silvester and Wathen (1994), an extended analysis is raised in Zulehner (2010) for parameter dependent saddle point problems.

Applying a preconditioner is usually a lot numerical effort. In this application, applying $\mathcal{P}_k^{bd}$ means solving a linear equation with the preconditioner as coefficient matrix in each MINRES iteration. As we use lumped mass matrix, the (1,1)-block of $\mathcal{P}_k^{bd}$ is a diagonal matrix, thus the inversion of this block is trivial. Furthermore, $\beta\tau\widetilde{A} + \widetilde{P}_k$ is also a diagonal matrix, hence the Schur complement of $\tau A + P_k$ in $\mathcal{A}_k$, i.e. the (2,2)-block of $\mathcal{P}_k^{bd}$, can be evaluated explicitly in a very cheap way. The main effort of applying $\mathcal{P}_k^{bd}$ is to solve a linear equation involving its (2,2)-block, which can be performed using iterative solvers such as CG, MINRES or fast algorithm like algebraic multi-grid method (AMG).

Notice that in more general case, the block $\tau A + P_k$ in the preconditioner may not be easily invertible, Krylov subspace methods can be one choice. There are of course other solutions available, like Chebyshev semi-iteration (see Wathen and Rees (2009)). It is also common that the Schur complement of the first block is expensive to compute, thus a proper approximation is preferable in practice. A commonly used one is proposed in Rees, Dollar and Wathen (2010). Most recently, a regularization- robust approximation is proposed in Pearson and Wathen (2011) and its application in preconditioning can be found in Pearson et al. (2011).

Another fact should be noticed is, the linear equation involves the preconditioner could also be ill-conditioned, thus a second level preconditioning might be also necessary. For the experiment of our nonlinear test problem, this will be further discussed later.

### 4.3.3. Numerical Experiments

Our approach for solving the nonlinear problem (4.23) is summarized in Algorithm 4.1.

All the numerical experiments in this section are carried out in Matlab. For finite element discretization, we use the matlab interface of GetFem++ (Renard and Pommier (n.d.)). *Q1*-element is adopted.

For solving the sequence of linear systems (4.30), we apply Algorithm 4.1 and fix $\beta = 1e{-}4$, $l = 4$, $n = 15$, $m = 16$. Thus $\mathcal{A}_k \mathbf{x}_k = \mathbf{b}_k$ is of dimension 108,000. The desired state is set to be $\bar{\mathbf{y}} = 1000tx_1(x_1 - 1)x_2(x_2 - 1)x_3(x_3 - 1)$.

The deflated MINRES matlab routine is revised from the same code (a variant of MINRES

**Algorithm 4.1** Nonlinear solver using POD-based preconditioning

```
1: Solve $\mathcal{A}_1 x_1 = b_1$ by $\mathcal{P}_1^{bd}$-preconditioned MINRES and compute $u_1$ directly;
2: Compute POD basis for $y$ and $p$, build deflation matrix $W$;
3: for $k > 1$ do
4:     Solve $\mathcal{A}_k x_k = b_k$ by $\mathcal{P}_k^{bd}$-preconditioned deflated MINRES and compute $u_k$ directly;
5:     Update $W$ if necessary;
6:     $k \longleftarrow k + 1$;
7: end for
```

other than the original MINRES method proposed in Paige and Saunders (1975)) as in Olshanskii and Simoncini (2010) subject to the algorithm therein. Further implementation hints about deflated MINRES can also be found in the same paper.

The preconditioner $\mathcal{P}_k^{bd}$ is applied by solving the relevant linear equation to a certain low precision via MINRES. Notice that the ideal preconditioner is applied here, despite the inaccurate MINRES solver. The preconditioner itself still has large condition number, hence we use an additional diagonal preconditioner (the main diagonal of $\mathcal{P}_k^{bd}$) for the second level preconditioning. The above strategy of applying the preconditioner $\mathcal{P}_k^{bd}$ is chosen only for simplicity of implementation.

For comparison, we solve $\mathcal{A}_k \mathbf{x}_k = \mathbf{b}_k$ via $\mathcal{P}_k^{bd}$-preconditioned MINRES using the same preconditioning strategy for all $k$.

Since there is no globalization in Algorithm 4.1, we need start the outer iteration from an initial point which is close enough to the solution. That is $\mathbf{x}_k = 0$ in our case. The outer (SQP) iteration is stopped at a tolerance of $10^{-5}$ and the linear equations $\mathcal{A}_k \mathbf{x}_k = \mathbf{b}_k$ are solved to the precision $10^{-10}$.

Table 4.4 shows the performance of deflated MINRES and standard MINRES for solving all the linear subproblems when the nonlinear term is set to be $\mathbf{f}(\mathbf{y}) = \mathbf{y}^2$ and $h_k := \mathbf{x}_k - \mathbf{x}_{k-1}$.

| k | MINRES | | | Deflated-MINRES | | |
|---|---|---|---|---|---|---|
| | #it | time | $\|h_k\|_\infty$ | #it | time | $\|h_k\|_\infty$ |
| 1 | 52 | 102.45 | 6.93e+2 | - | - | - |
| 2 | 45 | 84.77 | 6.14e+1 | 32 | 51.72 | 6.14e+1 |
| 3 | 45 | 85.77 | 4.02e−1 | 30 | 37.75 | 4.02e−1 |
| 4 | 48 | 91.43 | 1.46e−5 | 28 | 37.00 | 1.46e−5 |
| 5 | 46 | 86.80 | 1.12e−7 | 29 | 39.27 | 1.43e−7 |

**Table 4.4.: Comparison of MINRES and deflated MINRES for solving nonlinear problem (4.23). $\mathbf{f}(\mathbf{y}) = \mathbf{y}^2$.**

Note that CPU time for deflated MINRES already includes the cost of POD, which is in fact ignorable here. This is because we only have 16 time steps, thus POD only needs the solving of a small eigenvalue problem in $\mathbb{R}^{32}$. When the number of time steps is high, in addition to choosing a smart subset of all the snapshots for POD, it could be necessary to reuse POD subspace for multiple steps in order to compute it less frequently.

Figure 4.10 presents iteration history of the second Newton step. The slope of deflated

MINRES is significantly steeper, which obviously benefits from the deflation with POD. The starting point is already very close the solution, thus the good initial guess is no more observable. Here we do not use the previous solution as starting point of next Newton step, otherwise there will be few MINRES iterations. Instead, we use the same starting point for all $k$, for better comparison.

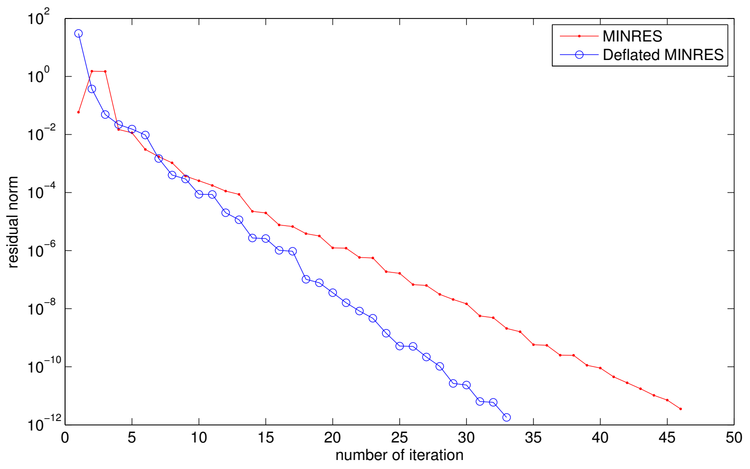


**Figure 4.10.: Iteration history of the second Newton step, when $\mathbf{f(y) = y^2}$**

| k | MINRES | | | Deflated-MINRES | | |
|---|---|---|---|---|---|---|
| | #it | time | $\|h_k\|_\infty$ | #it | time | $\|h_k\|_\infty$ |
| 1 | 54 | 109.76 | 6.18e+2 | - | - | - |
| 2 | 47 | 92.62 | 1.53e+1 | 28 | 77.00 | 1.53e+1 |
| 3 | 46 | 91.18 | 5.62e−2 | 28 | 74.23 | 5.62e−1 |
| 4 | 50 | 100.67 | 5.69e−5 | 28 | 75.24 | 5.69e−5 |
| 5 | 48 | 92.12 | 1.48e−7 | 28 | 74.74 | 2.49e−9 |

**Table 4.5.: Comparison of MINRES and deflated MINRES for solving nonlinear problem (4.23). $\mathbf{f(y) = \frac{100y}{1+y^2}}$.**

| k | MINRES | | | Deflated-MINRES | | |
|---|---|---|---|---|---|---|
| | #it | time | $\|h_k\|_\infty$ | #it | time | $\|h_k\|_\infty$ |
| 1 | 47 | 96.28 | 7.01e+2 | - | - | - |
| 2 | 59 | 121.68 | 2.87e+1 | 34 | 102.76 | 2.87e+1 |
| 3 | 53 | 109.48 | 3.58e−1 | 34 | 69.98 | 3.58e−1 |
| 4 | 53 | 106.67 | 5.91e−5 | 35 | 69.26 | 5.90e−5 |
| 5 | 53 | 107.13 | 4.63e−8 | 34 | 67.52 | 2.75e−7 |

**Table 4.6.: Comparison of MINRES and deflated MINRES for solving nonlinear problem (4.23). $\mathbf{f(y) = 5\sin y}$.**

Results for different nonlinear terms, i.e. $\mathbf{f(y)} = \frac{100\mathbf{y}}{1+\mathbf{y}^2}$ and $\mathbf{f(y)} = 5\sin\mathbf{y}$, are shown in

Table 4.5 and Table 4.6, which are very similar to Table 4.4. The efficiency of POD-based preconditioning varies with the nonlinear term.

## 4.4. Conclusion and Outlook

The numerical results we obtain in this chapter are encouraging. We establish the fact that POD basis can be used as leftmost eigenvectors by the experiments on the heat equation in Section 4.1. The tests on the linear quadratic problem verify effectiveness of our deflation strategy for a downsized KKT system. We also get an alternative view about the deflated MINRES algorithm, i.e. the algorithm solves a reduced order model and improves the solution by MINRES iterations. The regularization parameter $\beta$ should be paid attention to because of its influence on the efficiency of deflation. By solving a nonlinear PDE-constrained optimization problem in Section 4.3, the deflated preconditioned MINRES solver outperforms same preconditioned MINRES without deflation, with many restrictions though.

As one may notice, the deflated preconditioned MINRES solver proposed in Section 4.3 does not include a global strategy, thus the experiments are carried out using initial points close to the solution. For practical use, it would be necessary to have globalization in Algorithm 4.1.

If we have large-scale problems, cheaper calculation of POD basis should also be taken under consideration. In order to further reduce the cost, reusing POD basis for several steps could be necessary, thus an strategy of updating POD basis from time to time is required.

We only test one block diagonal preconditioner in Section 4.3, more experiments involving different block structured preconditioners like those referred to in Section 2.2 and different ways of applying the preconditioners should be tested. Moreover, different iterative solver could also be tried and compared, e.g. the Bramble-Pasciak-like CG method discussed in Section 2.2.2.

The research in this chapter is still at the stage of numerical experiments, theoretical explanations and analysis are also of great interest.

Another issue is the downsizing in Section 4.2 and Section 4.3 heavily relies on the use of lumped mass matrix, this could be a restriction of using the deflated preconditioned MINRES solver. For overcoming this shortage, we propose the deflation matrix

$$Z = \begin{pmatrix} W & 0 \\ 0 & 0 \\ 0 & W \end{pmatrix}$$

for original linear system (4.17), where $W$ is defined as (4.21).

If the assumptions of Lemma 4.1.3 are satisfied, for (4.17) we have

$$\begin{aligned} & \left\| \begin{pmatrix} \tau A & 0 & B^T \\ 0 & \beta\tau A & -\tau C^T \\ B & -\tau C & 0 \end{pmatrix} \begin{pmatrix} W & 0 \\ 0 & 0 \\ 0 & W \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \tau AW & B^T W \\ 0 & -\tau C^T W \\ BW & 0 \end{pmatrix} \right\|_2 \\ \leq \; & \|\tau AW\|_2 + \|\tau C^T W\|_2 + \|B^T W\|_2 + \|BW\|_2 \\ \leq \; & \tau \left\| \operatorname{diag}\left( \frac{1}{2} M\widetilde{W}, M\widetilde{W}, \dots, M\widetilde{W}, \frac{1}{2} M\widetilde{W} \right) \right\|_2 + \tau \left\| \operatorname{diag}\left( M\widetilde{W}, \dots, M\widetilde{W} \right) \right\|_2 \end{aligned}$$

$$
\begin{aligned}
& +4c_1h^d + 2\tau c_2 \\
\leq \;& 2\tau\|M\widetilde{W}\|_2 + 4c_1h^d + 2\tau c_2 \\
\leq \;& (4+2\tau)c_1h^d + 2\tau c_2
\end{aligned}
$$

The upper bound is small, thus the deflation matrix $Z$ seems to be reasonable.

To examine this, we experiment on 2D version of the linear quadratic problem (4.15) like we do in Section 4.2, with

$$\mathbf{y}^*(x,t) = \bar{\mathbf{y}}(x,t) = tx_1(x_1-1)x_2(x_2-1),$$

and

$$\mathbf{u}^* = x_1(x_1-1)x_2(x_2-1) - 2t\big(x_1(x_1-1) + x_2(x_2-1)\big).$$

Here $\Omega = (0,1)\times(0,1)$, $T = 1$.

Furthermore, we set the parameters to be

$$\beta = \frac{1}{500}, \quad m = 30, \quad n = 20.$$

Thus the linear system to be solved has 24000 unknowns. The discretization and derivation of the KKT system are almost the same as in Section 4.2, hence we omit them here.

Deflated MINRES and POD-initialized MINRES are compared and the results are given by Table 4.7. We can obviously see that deflation improves the numbers of iteration a lot for different dimensional POD basis. However, CPU time is higher except the case $l = 1$. This is similar to what we see from the experiment on the efficiency of deflated CG method in Section 4.1. It could be changed if we shift to a 3D problem, where the extra matrix-vector product in deflated MINRES algorithm weights relatively less in the total computational effort.

| tol = 1e-8 | l | Init. Err. | Def-MINRES | | Init. MINRES | |
|---|---|---|---|---|---|---|
| | | | #it | CPU time | #it | CPU time |
| MINRES | - | 3.75e-3 | 3452 | 31.82 | - | - |
| MINRES($\widetilde{W}_1$) | 1 | 8.36e-7 | 1547 | 19.94 | 2493 | 22.67 |
| | 2 | 3.72e-7 | 1427 | 25.79 | 2409 | 22.12 |
| | 3 | 7.68e-8 | 1415 | 31.11 | 2453 | 21.96 |
| | 4 | 4.66e-8 | 1383 | 37.52 | 2449 | 21.90 |

**Table 4.7.: Comparison of deflated MINRES and POD-initialized MINRES for distributed control problem (4.15). Time is given in second.**

An interesting phenomenon in Table 4.7 is that the number of iterations does not decrease much when we use higher dimensional POD subspace for deflation. This is probably caused by the control variable $u$, more precisely, solving for $u$ requires certain amount of iterations, which can not be improved by deflation using $Z$.

If we use a matrix like $Z$ for deflation in Algorithm 4.1, then we will need a different preconditioner. The effect of such a deflation strategy combined with a proper preconditioner requires further study.

# Appendix A.

# Toeplitz Matrix

A Toeplitz matrix is a matrix with all its diagonals constant. The structure of Toeplitz matrices is very beneficial for storage and computation. Since we mainly focus on preconditioning for iterative methods, we introduce Toeplitz matrices with corresponding efficient matrix-vector multiplication here. For a comprehensive study of Toeplitz systems and related solvers, we refer to Chan and Jin (2007) and Ng (2004).

## A.1. Toeplitz Matrix and Generating Function

We first give the rigid definition of Toeplitz matrix as follows.

**Definition A.1.1.** *A matrix $T \in \mathbb{C}^{n\times n}$ is called* Toeplitz, *if $T$ is determined by the $2n-1$ scalars $a_{-(n-1)},\dots, a_{-1}, a_0, a_1, \dots, a_{n-1}$ with $T_{ij} = a_{i-j}$ for all $i$ and $j$, i.e. the Toeplitz matrix $T_n$ is of the following form:*

$$T_n = \begin{pmatrix} a_0 & a_{-1} & \cdots & a_{2-n} & a_{1-n} \\ a_1 & a_0 & \ddots & & a_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-2} & & \ddots & \ddots & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{pmatrix}. \tag{A.1}$$

Toeplitz matrices are usually dense, however, Toeplitz can be somewhat treated as a sparse structure, since storing a matrix $T_n$ defined by (A.1) only requires the storage of the $2n-1$ scalars $a_{-(n-1)},\dots, a_{-1}, a_0, a_1, \dots, a_{n-1}$.

In fact, an infinite Toeplitz matrix $T_\infty$ (by Definition A.1.1 with $n$ going to infinity) can be uniquely determined by a so-called *generating function* $g$, which is defined by the Fourier series

$$g(x) = \sum_{k=-\infty}^{\infty} a_k e^{-\mathrm{i}kx}, \quad x \in [-\pi, \pi]. \tag{A.2}$$

Precisely speaking, the entries of $T_\infty$ are given by

$$a_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} g(x)e^{\mathrm{i}kx}\,\mathrm{d}x, \qquad k = 0, \pm1, \pm2, \cdots. \tag{A.3}$$

Let $T_n$ be the matrix generated by $g$ in the sense that its entries are determined as (A.3) with $k = 0, \pm 1, \ldots, \pm(n-1)$, notice that when $g$ is a real-valued function,

$$a_{-k} = \frac{1}{2\pi}\int_{-\pi}^{\pi} g(x)e^{-\mathrm{i}kx}\,\mathrm{d}x = \bar{a}_k, \qquad \forall k \in \mathbb{Z},$$

that means for all $n \in \mathbb{N}$, $T_n$ must be Hermitian. Additionally, if $g$ is also even, i.e. $g(-x) = g(x)$, then $T_n$ is real and symmetric for all $n \in \mathbb{N}$.

The following theorem indicates the relationship between $g$ and the spectrum of $T_n$.

**Theorem A.1.2** (Grenander and Szegö (1984)). *Let $g$ be a real-valued function in $L^1[-\pi,\pi]$ defined by (A.2). Then the spectrum $\sigma(T_n)$ of $T_n$ satisfies*

$$\sigma(T_n) \subseteq [g_{min}, g_{max}], \quad \forall\ n \geq 1,$$

*where $g_{min}$ and $g_{max}$ are the essential infimum and the essential supremum of $g$ respectively. Moreover, if $g_{max} > g_{min}$, then*

$$g_{min} < \lambda_{min}(T_n) \leq \lambda_{max}(T_n) < g_{max}.$$

*In particular, if $g_{min} > 0$, then $T_n$ is positive definite for all $n \in \mathbb{N}$.*

## A.2. Circulant Matrix

As a special kind of Toeplitz matrices, circulant matrices have more remarkable properties, which are very important especially from a computational point of view.

**Definition A.2.1.** *$C \in \mathbb{C}^{n\times n}$ is called* circulant *if it is a Toeplitz matrix where each column is a circular shift of its preceding column or $a_{-i} = a_{n-i}$ in $T_n$ $(i = 1, ..., n-1)$, i.e. the circulant matrix $C_n$ is of the following form:*

$$C_n = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & \ddots & & c_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix}.$$

With respect to $n \times n$ Fourier matrix $F_n$ (unitary matrix), whose entries are given by

$$(F_n)_{ij} = \frac{1}{\sqrt{n}}\omega^{ij}, \qquad \omega = e^{-2\pi\mathrm{i}/n}, \quad i, j = 0, \ldots, n-1, \tag{A.4}$$

a very useful diagonalization of circulant matrices is given by the following theorem.

**Theorem A.2.2** (Davis (1979)). *Let $C_n \in \mathbb{R}^{n\times n}$ be circulant. Then it has the decomposition*

$$C_n = F_n^* \Lambda F_n, \tag{A.5}$$

*where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, $\lambda_j$ being the $j$th eigenvalue of $C_n$, $j = 1, \ldots, n$.*

As a result of Theorem A.2.2, more accurately, by comparing the first column of $F_nC_n$ and $\Lambda F_n$, we can get the relationship

$$(\lambda_1, \ldots, \lambda_n)^T = \sqrt{n}F_n(c_0, \ldots, c_{n-1})^T,$$

thus $\Lambda$ can be computed by one *Fast Fourier Transform* (FFT). Furthermore, the matrix-vector product $C_na = F_n^*\Lambda F_na$ can be computed by three FFTs and one vector multiply (see e.g. Golub and Van Loan (1996)). Since the computational complexity of $F_na$ via FFT is $O(n \log n)$, the evaluation of $C_na$ takes only $O(n \log n)$ operations instead of $O(n^2)$ for common dense matrices. Notice that $C_n^{-1}a = F_n^*\Lambda^{-1}F_na$, the numerical effort of evaluating $C_n^{-1}a$ is the same as computing $C_na$.

$\{\omega\}$-circulant is a generalized form of circulant structure, the definition of $\{\omega\}$-circulant is given by

**Definition A.2.3.** *Let $\omega = e^{i\theta_0}$ with $\theta_0 \in [-\pi, \pi]$. An $n \times n$ matrix $C_n^\omega$ is said to be an $\{\omega\}$-circulant matrix if it has the spectral decomposition*

$$C_n^\omega = \Omega_n^* F_n^* \Lambda_n F_n \Omega_n.$$

*Here $\Omega_n = \mathrm{diag}(1, \omega^{-1/n}, \ldots, \omega^{-(n-1)/n})$ and $\Lambda_n$ is a diagonal matrix containing the eigenvalues of $C_n^\omega$.*

In particular, $C_n^\omega$ is a circulant matrix when $\omega = 1$. If $\omega = -1$, $C_n^\omega$ is called skew-circulant matrix. Similar to matrix-vector product regarding a circulant matrix, FFT can also be used to accelerate the matrix-vector multiplication involving a skew-circulant matrix, the computational complexity is still $O(n \log n)$, see e.g. Ng (2004).

## A.3. Efficient Matrix-Vector Multiplication with Toeplitz Matrix

The computational advantage of circulant matrices is very attractive, fortunately, it can be employed to achieve efficient matrix-vector multiplication for general Toeplitz matrices. There are two ways to do so, the first method embeds a Toeplitz matrix in a large circulant matrix, the second is splitting a Toeplitz matrix into a circulant matrix and a skew circulant matrix. We present them as follows.

(i) For a given $n \times n$ Toeplitz matrix $T_n$ defined by (A.1) and a vector $x$ of dimension $n$, the product $T_nx$ can be computed within $O(n \log n)$ operations by first embedding $T_n$ into a $2n \times 2n$ circulant matrix

$$C_{2n} = \begin{pmatrix} T_n & B_n \\ B_n & T_n \end{pmatrix} \quad \text{with} \quad B_n = \begin{pmatrix} 0 & a_{n-1} & \cdots & a_2 & a_1 \\ a_1 & 0 & \ddots & & a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-2} & & \ddots & \ddots & a_{n-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & 0 \end{pmatrix}.$$

Obviously $C_{2n}$ is a circulant matrix, hence

$$C_{2n}\begin{pmatrix} x \\ 0 \end{pmatrix} = \begin{pmatrix} T_n x \\ * \end{pmatrix}$$

can be evaluated with three FFTs.

(ii) According to Pustyl'nikov (1980), any $n \times n$ Toeplitz matrix $T_n$ can be split into the sum of a circulant matrix $U_n$ and a skew-circulant matrix $V_n$, which are given by

$$U_n = \frac{1}{2}\begin{pmatrix} a_0 & a_{-1}+a_{n-1} & & & a_{1-n}+a_1 \\ a_1+a_{1-n} & a_0 & \ddots & & \\ & \ddots & \ddots & & \\ & & & \ddots & a_{-1}+a_{n-1} \\ a_{n-1}+a_{-1} & & & a_1+a_{1-n} & a_0 \end{pmatrix}$$

and

$$V_n = \frac{1}{2}\begin{pmatrix} a_0 & a_{-1}-a_{n-1} & & & a_{1-n}-a_1 \\ a_1-a_{1-n} & a_0 & \ddots & & \\ & \ddots & \ddots & & \\ & & & \ddots & a_{-1}-a_{n-1} \\ a_{n-1}-a_{-1} & & & a_1-a_{1-n} & a_0 \end{pmatrix}.$$

Thus we can compute $U_n x$ and $V_n x$ respectively and $T_n x$ is easily obtained by evaluating $U_n x + V_n x$.

The storage and computational effort of the above two methods are comparable. In Chapter 3, we use the second method though.

# Appendix B.

# Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) is a very useful method to generate low-dimensional approximate representation for large-scale dynamical systems and large data sets. It is mostly used as a reduced order modeling technique for time dependent partial differential equations since a low-dimensional POD basis is usually able to capture the characteristics of the system of interest. We first review some essential properties of POD following the track of Kragel (2005) and Kunisch and Volkwein (2002), then we introduce the numerical realization of POD. For more detailed discussion, we refer to Volkwein (2011).

## B.1. POD in Hilbert Space

Let $H$ be a separable Hilbert space endowed with inner product $\langle \cdot, \cdot \rangle_H : H \times H \to \mathbb{R}$. A corresponding norm is naturally given by $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle_H}$. For a given set of elements (*snapshots* in the context of dynamical systems) $y_1$, $y_2$, ..., $y_n \in H$, POD provides an orthonormal basis $\{\phi_j\}_{j=1}^k$ ($k$ is preassigned and $k \leq n$) solving

$$\begin{aligned} \min_{\{\psi_j\}_{j=1}^k} \quad & \sum_{i=1}^n \omega_i \|y_i - \sum_{j=1}^k \langle y_i, \psi_j \rangle_H \psi_j \|^2 \\ \text{s.t.} \quad & \langle \psi_i, \psi_j \rangle_H = \delta_{ij} \quad \text{for} \quad 1 \leq i, j \leq k. \end{aligned} \tag{B.1}$$

Here the solution $\{\phi_j\}_{j=1}^k$ is called a POD basis of rank $k$, $\{\omega_i\}_{i=1}^n$ are positive weights, $\delta_{ij}$ denotes the Kronecker delta, i.e.

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases}$$

In order to solve the constrained optimization problem (B.1), a bounded linear operator $\mathcal{Y}_n : \mathbb{R}^n \to H$ is introduced, which is defined by

$$\mathcal{Y}_n a = \sum_{i=1}^n \omega_i a_i y_i \quad \text{for} \quad a = (a_1, a_2, \ldots, a_n)^T \in \mathbb{R}^n.$$

If we further define a weighted inner product on $\mathbb{R}^n$ by

$$\langle a, b \rangle_{\mathbb{R}^n} = a^T W b \quad \text{for} \quad a, b \in \mathbb{R}^n$$

with $W = \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_n) \in \mathbb{R}^{n\times n}$, then $\mathcal{Y}_n^* : H \to \mathbb{R}^n$, the adjoint operator of $\mathcal{Y}_n$, is given by

$$\mathcal{Y}_n^* x = (\langle y_1, x\rangle_H, \langle y_2, x\rangle_H, \ldots, \langle y_n, x\rangle_H)^T \quad \text{for} \quad x \in H.$$

Obviously $\langle \mathcal{Y}_n a, x\rangle_H = \langle a, \mathcal{Y}_n^* x\rangle_{\mathbb{R}^n}$ for all $a \in \mathbb{R}^n, x \in H$ with such notations.

It follows that the *autocorrelation operator* $\mathcal{R}_n := \mathcal{Y}_n \mathcal{Y}_n^* \in \mathcal{L}(H)$ is given by

$$\mathcal{R}_n x = \sum_{i=1}^{n} \omega_i \langle y_i, x\rangle_H y_i, \quad \text{for} \quad x \in H.$$

Here $\mathcal{L}(H)$ denotes the Banach space of all bounded linear operators on $H$.

It is concluded in Volkwein (2001) that there exists an orthonormal basis $\{\phi_j\}_{j\in\mathbb{N}}$ of $H$ and a sequence $\{\lambda_j\}_{j\in\mathbb{N}}$ of nonnegative real numbers such that

$$\mathcal{R}_n \phi_j = \lambda_j \phi_j \quad \text{with} \quad \lambda_1 \geq \cdots \geq \lambda_p > 0 \quad \text{and} \quad \lambda_j = 0 \ (j > p). \tag{B.2}$$

Moreover, $\mathrm{span}\{y_1, \ldots, y_n\} = \mathrm{span}\{\phi_1, \ldots, \phi_p\}$. Using the expressions in (B.2), the solution to (B.1) and the relevant error formula is given by the following theorem.

**Theorem B.1.1.** *$\{\phi_j\}_{j=1}^k$ for $1 \leq k \leq p$ satisfying (B.2) solves Problem (B.1), the resulting truncation error is given by*

$$\sum_{i=1}^{n} \omega_i \| y_i - \sum_{j=1}^{k} \langle y_i, \phi_j\rangle_H \phi_j \|^2 = \sum_{j=m+1}^{p} \lambda_j. \tag{B.3}$$

The proof of Theorem B.1.1 can be found in Volkwein (2001).

## B.2. Calculate the POD Basis

For numerical realization, we confine ourselves to the special case $H = \mathbb{R}^m$. Thus the given snapshots $\{y_i\}_{i=1}^n$ are vectors in $\mathbb{R}^m$. If we let $Y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^{m\times n}$ denote the snapshot data matrix, then the linear operator $\mathcal{Y}_n$ has the matrix form $\bar{Y} := (\omega_1 y_1, \omega_2 y_2, \ldots, \omega_n y_n) = YW \in \mathbb{R}^{m\times n}$ and its adjoint operator $\mathcal{Y}_n^*$ is simply given by $\bar{Y}^T := WY^T \in \mathbb{R}^{n\times m}$. As a result, $\mathcal{R}_n = \mathcal{Y}_n \mathcal{Y}_n^*$ is represented by $R := \bar{Y}\bar{Y}^T \in \mathbb{R}^{m\times m}$. $R$ is symmetric and positive semidefinite by construction.

If the eigenvalues of $R$ are ordered as in (B.2) and the corresponding (normalized) eigenvectors are denoted by $\{u_j\}_{j=1}^m$, i.e.

$$R u_j = \lambda_j u_j \quad \text{with} \quad \lambda_1 \geq \cdots \geq \lambda_p > 0 \quad \text{and} \quad \lambda_j = 0 \ (j > p), \tag{B.4}$$

a POD basis of rank $k (1 \leq k \leq p)$ consists of the first $k$ eigenvectors $u_1, u_2, \ldots, u_k$ according to Theorem B.1.1. In practice, we use the matrix form of the POD basis, i.e. $U_k = (u_1, u_2, \ldots, u_k) \in \mathbb{R}^{m\times k}$. Note that the corresponding truncation error formula directly follows (B.3).

Directly solving the eigenvalue problem $Ru = \lambda u$ is usually not cheap, especially when $m$ is large, which is normally the case when the problem arises from discretization of a PDE.

We now introduce alternative ways based on singular value decomposition (SVD) which offer possibility of reducing such effort.

Assume $\bar{Y}$ has singular value decomposition

$$\bar{Y} = U\Sigma V^T = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V^T, \tag{B.5}$$

where $\Sigma \in \mathbb{R}^{m\times n}$, $D = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p) \in \mathbb{R}^{p\times p}$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p > 0$, $U = (u_1, u_2, \ldots, u_m) \in \mathbb{R}^{m\times m}$ and $V = (v_1, v_2, \ldots, v_n) \in \mathbb{R}^{n\times n}$ are orthogonal matrices. It is not hard to see

$$Ru_j = \bar{Y}\bar{Y}^T u_j = \sigma_j^2 u_j.$$

Thus $\lambda_j = \sigma_j^2 > 0$ and $U_k$ is indeed the left part (the first $k$ columns) of $U$.

Based on the decomposition (B.5), we also have

$$\bar{Y}v_j = \sigma_j u_j, \quad \bar{Y}^T u_j = \sigma_j v_j, \quad \bar{Y}^T\bar{Y}v_j = \sigma_j^2 v_j.$$

In order to get $U_k$, we can directly compute the singular value decomposition (B.5), or we solve the eigenvalue problem $\bar{Y}^T\bar{Y}v = \sigma_j^2 v$ first and compute $\{u_j\}_{j=1}^k$ by $u_j = \frac{1}{\sigma_j}\bar{Y}v_j$, in addition to solving (B.4).

We summarize the three ways of computing $U_k$ as follows:

(1) Do SVD to $\bar{Y} \in \mathbb{R}^{m\times n}$.

(2) Solve $\bar{Y}\bar{Y}^T u = \lambda_j u$ in $\mathbb{R}^{m\times m}$.

(3) Solve $\bar{Y}^T\bar{Y}v = \lambda_j v$ in $\mathbb{R}^{n\times n}$ for $V_k$ and $U_k = \bar{Y}V_k\Lambda_k^{-\frac{1}{2}}$.

Here $V_k = (v_1, v_2, \ldots, v_k) \in \mathbb{R}^{n\times k}$, $\Lambda_k = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_k) \in \mathbb{R}^{k\times k}$. If $m < n$, method (2) is preferable to method (3), and vice versa. For discussions on practical implementation, we refer to Fahl (2000) and the references therein.

We remark that POD and SVD have close connection in a more general sense, which is studied in Hilbert space in Volkwein (1999).

# Bibliography

Axelsson, O. (1994). *Iterative Solution Methods*, Cambridge University Press, Cambridge.

Axelsson, O. and Kaporin, I. (2000). On the sublinear and superlinear rate of convergence of conjugate gradient methods, *Numer. Algorithms* **25**: 1–22.

Axelsson, O. and Karátson, J. (2002). On the rate of convergence of the conjugate gradient method for linear operators in Hilbert space, *Numer. Funct. Anal. Optimization* **23**: 285–302.

Axelsson, O. and Karátson, J. (2009). Equivalent operator preconditioning for elliptic problems, *Numer. Algor.* **50**: 297–380.

Beckermann, B., Goreinov, S. and Tyrtyshnikov, E. (2005). Some remarks on the elman estimate for GMRES, *SIAM J. Matrix Anal. Appl.* **27**(3): 772–778.

Benzi, M., Golub, G. H. and Liesen, J. (2005). Numerical solution of saddle point problems, *Acta Numerica* **14**: 1–137.

Benzi, M., Haber, E. and Taralli, L. (2011). A preconditioning technique for a class of PDE-constrained optimization problems, *Adv. Comput. Math.* **35**(2-4): 149–173.

Benzi, M. and Wathen, A. (2008). Some preconditioning techniques for saddle point problems, *in* W. H. A. Schilders, H. A. Vorst and J. Rommes (eds), *Model Order Reduction: Theory, Research Aspects and Applications*, Vol. 13 of *Mathematics in Industry*, Springer Berlin Heidelberg, pp. 195–211.

Bergounioux, M., Ito, K. and Kunisch, K. (1999). Primal-dual strategy for constrained optimal control problems, *SIAM J. Control Optim.* **37**(4): 1176–1194.

Biegler, L., Ghattas, O., Heinkenschloss, M. and van Bloemen Waanders, B. (2003). *Large-Scale PDE-Constrained Optimization*, Vol. 30 of *Lecture Notes in Computational Science and Engineering*, Springer.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities, *Journal of Political Economy* **81**(3): 637 – 654.

Bramble, J. and Pasciak, J. (1988). A preconditioning technique for indeinite systems resulting from mixed approximations of elliptic problems, *Math. Comp.* **50**: 1 – 17.

Chan, R. (1989). Circulant preconditioners for Hermitian Toeplitz systems, *SIAM J. Matrix Anal. Appl.* **10**: 542–550.

Chan, R. and Jin, X. (2007). *An Introduction to Iterative Toeplitz Systems*, SIAM, Philadelphia.

Chan, R., Jin, X. and M., Y. (1991a). The circulant operator in the Banach algebra of matrices, *Linear Algebra Appl.* **149**: 41–53.

Chan, R., Jin, X. and M., Y. (1991b). The spectra of super-optimal circulant preconditioned Toeplitz systems, *SIAM J. Numer. Anal.* **28**: 871–879.

Chan, T. (1988). An optimal circulant preconditioner for Toeplitz systems, *SIAM J. Sci. Stat. Comput.* **9**: 766 – 771.

Davis, P. (1979). *Circulant Matrices*, John Wiley & Sons, New York.

Dennis, Jr., J. E. and Schnabel, R. B. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM.

Dupire, B. (1994). Pricing with a smile, *Risk* **7**: 18–20.

Elman, H. C. (1982). *Iterative Methods for Sparse Nonsymmetric Systems of Linear Equations*, PhD thesis, Yale University, Department of Computer Science.

Elman, H. C., Silvester, D. J. and Wathen, A. J. (2005). *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press.

Erhel, J. and Guyomarc'h, F. (2000). An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems, *SIAM J. Matrix Anal. Appl.* **21**(4): 1279–1299.

Fahl, M. (2000). *Trust-Region methods for flow control based on reduced order modelling*, PhD thesis, Universität Trier.

Gaul, A.and Gutknecht, M., Liesen, J. and Nabben, R. (2011). Deflated and augmented Krylov subspace methods: Basic facts and a breakdown-free deflated MINRES, Preprint 749. DFG Research Center Matheon, Mathematics for key technologies, Berlin, 2011.

Gohberg, I., Goldberg, S. and Kaashoek, M. A. (2003). *Basic Classes of Linear Operators*, Birkhäuser.

Golub, G. and Van Loan, C. (1996). *Matrix Computations*, 3rd edn, The Johns Hopkins University Press, Baltimore.

Grenander, U. and Szegö, G. (1984). *Toeplitz forms and their applications*, 2nd edn, Chelsea Publishing, New York.

Günnel, A., Herzog, R. and Sachs, E. (2011). A note on preconditioners and scalar products for Krylov methods in Hilbert space, *Technical report*, TU Chemnitz.

Haroske, D. and Triebel, H. (2008). *Distributions, Sobolev Spaces, Elliptic Equations*, European Mathematical Society.

Herzog, R. (2010). Algorithms and preconditioning in PDE-constrained optimization, Lecture Notes. TU Chemnitz.

Herzog, R. and Kunisch, K. (2010). Algorithms for PDE-constrained optimization, *GAMM-Mitteilungen* **33**: 163–176.

Herzog, R. and Sachs, E. (2010). Preconditioned conjugate gradient method for optimal control problems with control and state constraints, *SIAM Journal on Matrix Analysis and Applications* **31**(5): 2291–2317.

Hestenes, M. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards* **49**(6): 409 – 436.

Hinze, M., Pinnau, R., Ulbrich, M. and Ulbrich, S. (2009). *Optimization with PDE Constraints*, Vol. 23 of *Mathematical Modelling: Theory and Applications*, Springer, New York.

Horn, R. and Johnson, C. (1985). *Matrix Analysis*, Cambridge University Press.

Huckle, T. (1992). Circulant and skewcirculant matrices for solving Toeplitz matrix problems, *SIAM J. Matrix Anal. Appl.* **13**: 767 – 777.

Huckle, T. (1993). Some aspects of circulant preconditioners, *SIAM J. Sci. Stat. Comput.* **14**(1): 531–541.

Karátson, J. (2005). Mesh independent superlinear convergence estimates of the conjugate gradient method for some equivalent self-adjoint operators, *Appl. Math.* **50**(3): 277–290.

Kragel, B. (2005). *Streamline Diffusion POD Models in Optimization*, PhD thesis, Universität Trier.

Kunisch, K. and Volkwein, S. (1999). Control of the burgers equation by a reduced-order approach using proper orthogonal decomposition, *Journal of Optimization Theory and Applications* **102**: 345–371.

Kunisch, K. and Volkwein, S. (2002). Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM Journal on Numerical Analysis* **40**(2): 492–515.

Li, H. B., Huang, T. Z., Liu, X. P. and Li, H. (2010). On the inverses of general tridiagonal matrices, *Linear Algebra and its Applications* **433**(5): 965–983.

Markovinović, R. and Jansen, J. D. (2006). Accelerating iterative solution methods using reduced-order models as solution predictors, *International Journal for Numerical Methods in Engineering* **68**(5): 525–541.

Merton, R. (1976). Option pricing when underlying stock returns are discontinuous, *Journal of Financial Economics* **3**: 125–144.

Meurant, G. (1992). A review on the inverse of symmetric tridiagonal and block tridiagonal matrices, *SIAM J. Matrix Anal. Appl.* **13**(3): 707–728.

Meurant, G. (2006). *The Lanczos and Conjugate Gradient Algorithms*, Society for Industrial and Applied Mathematics.

Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia.

Murphy, G. (1990). *C*-Algebras and Operator Theory*, Academic Press.

Murphy, M., Golub, G. and Wathen, A. (2000). A note on preconditioning for indefinite linear systems, *SIAM Journal on Scientific Computing* **21**(6): 1969–1972.

Ng, M. K. (2004). *Iterative Methods for Toeplitz Systems*, Oxford University Press, New York.

Nicolaides, R. (1987). Deflation of conjugate gradients with applications to boundary value problems, *SIAM Journal on Numerical Analysis* **24**(2): 355–365.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*, 2nd edn, Springer Science+Business Media, New York.

Olshanskii, M. A. and Simoncini, V. (2010). Acquired clustering properties and solution of certain saddle point systems, *SIAM J. Matrix Anal. Appl.* **31**(5): 2754–2768.

Paige, C. and Saunders, M. (1975). Solution of sparse indefinite systems of linear equations, *SIAM Journal on Numerical Analysis* **12**(4): 617–629.

Pang, H., Zhang, Y. and Jin, X. (2012). Tri-diagonal preconditioner for pricing options, *Journal of Computational and Applied Mathematics* **236**(17): 4365 – 4374.

Pang, H., Zhang, Y., Vong, S. and Jin, X. (2011). Circulant preconditioners for pricing options, *Linear Algebra and its Applications* **434**(11): 2325 – 2342.

Patterson, W. E. (1974). *Iterative Methods for the Solution of a Linear Operator Equation in Hilbert Space - A Survey*, Vol. 394 of *Lecture Notes in Mathematics*, Springer-Verlag.

Pearson, J., Stoll, M. and Wathen, A. (2011). Regularization-robust preconditioners for time-dependent PDE constrained optimization problems, *submitted* .

Pearson, J. and Wathen, A. (2011). A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, *to appear in Numerical Linear Algebra with Applications* .

Pustyl'nikov, L. D. (1980). On the algebraic structure of spaces of Toeplitz and Hankel matrices, *Sov. Math. Dokl.* **21**(1): 141–144.

Rees, T., Dollar, S. and Wathen, A. (2010). Optimal solvers for PDE-constrained optimization, *SIAM J. Sci. Comput.* **32**: 271–298.

Rees, T., Stoll, M. and Wathen, A. (2010). All-at-once preconditioning in PDE-constrained optimization, *Kybernetika* **45**: 341–360.

Renard, Y. and Pommier, J. (n.d.). Getfem++, an open-source finite element library. `http://download.gna.org/getfem/html/homepage/index.html`.

Rusten, T. and Winther, R. (1992). A preconditioned iterative method for saddlepoint problems, *SIAM Journal on Matrix Analysis and Applications* **13**(3): 887–904.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*, 2nd edn, Society for Industrial and Applied Mathematics, Philadelphia, PA.

Saad, Y., Yeung, M., Erhel, J. and Guyomarc'h, F. (1999). A deflated version of the conjugate gradient algorithm, *SIAM J. Sci. Comput.* **21**(5): 1909–1926.

Sachs, E. and Strauss, A. (2008). Efficient solution of a partial integro-differential equation in finance, *Applied Numerical Mathematics* **58**: 1687–1703.

Schu, M. (2012). *Adaptive Trust-Region POD Methods and their Application in Finance*, PhD thesis, Universität Trier.

Schu, M. and Sachs, E. W. (2010). Reduced order models in PIDE constrained optimization, *Control and Cybernetics* **39**(3): 661–675.

Silvester, D. and Wathen, A. (1994). Fast iterative solution of stabilised Stokes systems part II: Using general block preconditioners, *SIAM J. Numer. Anal.* **31**(5): 1352–1367.

Simoncini, V. (2012). Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems, *Computational Optimization and Applications* **53**: 591–617.

Simoncini, V. and Szyld, D. (2007). Recent computational developments in Krylov subspace methods for linear systems, *Numerical Linear Algebra with Applications* **14**(1): 1–59.

Stoll, M. (2008). *Solving Linear Systems using the Adjoint*, PhD thesis, University of Oxford.

Stoll, M. (2011). One-shot solution of a time-dependent time-periodic PDE-constrained optimization problem, *Preprint MPIMD/11-04*, Max Planck Institute, Magdeburg.

Stoll, M. and Wathen, A. (2007). The Bramble-Pasciak preconditioner for saddle point problems, *Technical Report 07/13*, The Mathematical Institute, University of Oxford.

Stoll, M. and Wathen, A. (2008). Combination preconditioning and the Bramble-Pasciak$^+$ preconditioner, *SIAM Journal on Matrix Analysis and Applications* **30**(2): 582–608.

Stoll, M. and Wathen, A. (2010). All-at-once solution of time-dependent PDE-constrained optimization problems, *Technical Report 1017*, The Mathematical Institute, University of Oxford.

Strang, G. (1986). A proposal for Toeplitz matrix calculations, *Stud. Appl. Math.* **74**: 171–176.

Tismenetsky, M. (1991). A decomposition of Toeplitz matrices and optimal circulant preconditioner, *Linear Algebra Appl.* **156**: 105–121.

Tröltzsch, F. (2010). *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, Vol. 12 of *Graduate Studies in Mathematics*, American Mathematical Society.

Tyrtyshnikov, E. (1992). Optimal and superoptimal circulant preconditioners, *SIAM J. Matrix Anal. Appl.* **13**(2): 459–473.

van der Vorst, H. A. (1980). Preconditioning by incomplete decomposition, *Technical report*, Rijksuniversiteit te Utrecht.

Varah, J. M. (1975). A lower bound for the smallest singular value of a matrix, *Linear algebra and its applications* **11**: 3–5.

Volkwein, S. (1999). Proper orthogonal decomposition and singular value decomposition, *Technical Report SFB-Preprint No. 153*, Karl-Franzens-Universität Graz.

Volkwein, S. (2001). Optimal control of a phase-field model using proper orthogonal decomposition, *ZAMM - Journal of Applied Mathematics and Mechanics* **81**(2): 83–97.

Volkwein, S. (2011). Model reduction using proper orthogonal decomposition, Lecture Notes. Universität Konstanz.

Wathen, A. and Rees, T. (2009). Chebyshev semi-iteration in preconditioning for problems including the mass matrix, *Electronic Transactions on Numerical Analysis* **34**: 125–135.

Winther, R. (1980). Some superlinear convergence results for the conjugate gradient method, *SIAM J. Numer. Anal.* **17**(1): 14–17.

Zhang, Y. (2010). *Preconditioning Techniques for a Family of Toeplitz-Like Systems with Financial Applications*, PhD thesis, University of Macao.

Zulehner, W. (2010). Non-standard norms and robust estimates for saddle point prblems, *Technical Report 2010-07*, Institute of Computational Mathematics, Johannes Kepler University.