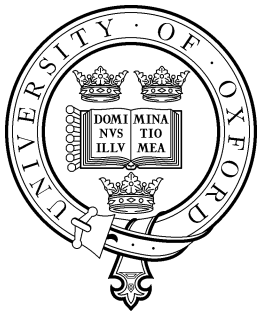
All-At-Once Solution of Time-Dependent PDE Problems



Eleanor McDonald
New College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2016

This thesis is dedicated to both
my grandfather, Jim McDonald,
and my father, Bob McDonald.

Acknowledgements

My sincerest thanks and gratitude must go to my supervisor, Andy Wathen, whose constant guidance and support have been invaluable to this thesis. Without exception, I never left a conversation with Andy without feeling more encouraged, motivated, and enthusiastic about my research and this has fuelled me throughout my time in Oxford. I have greatly appreciated his mentorship and advice and my research experience has been made infinitely more enjoyable because of it.

The support of the DPhil students who came before me has also been hugely helpful and my genuine thanks go, in particular, to Jen Pestana, John Pearson, and Tyrone Rees for their sound advice and for lending a friendly ear when required. Thanks must also go to the broader Numerical Analysis group at Oxford who have been so welcoming to me and have always been there for a chat, whether about maths or not. Patricio Farrell deserves special mention for being the best first officemate I could ask for.

Outside of maths, the New College MCR community has been a second home to me, from which I have gained many lasting friendships. In particular, thanks must go to Richard and Rupert for the countless coffees on the Rad Cam and teas in our kitchen, which kept me sane. My friends back in Australia have continued to be there for me, though miles away, and this support has been truly cherished.

To my family, in particular my parents, without your constant encouragement and belief in me, I would certainly not be where I am today. I am incredibly lucky to have been given so many opportunities in life and I will always be grateful for everything you have provided for me. Last but not least, thanks must go to John. I will forever be indebted to you for the sacrifices you made to allow me to come and live this dream. I am so glad I got to share this experience with you and I couldn't have done it without you. My time here has been made so much more special because you were here.

Abstract

In this thesis, we examine the solution to a range of time-dependent Partial Differential Equation (PDE) problems. Throughout, we focus on the development of preconditioners for the all-at-once system, which solves for all time-steps in a single coupled computation. The preconditioners developed are used with existing iterative methods and, due to their specific block structure, could be applied in parallel over time.

We first develop solvers for the heat equation and the transient convection-diffusion equation. For both of these forward problems, the all-at-once system is non-symmetric. Despite this, in certain cases, we are able to provide rigorous termination bounds for non-symmetric iterative methods, contrary to what is generally possible for non-symmetric systems.

The ideas developed for evolutionary PDEs are extended to develop preconditioners for time-dependent optimal control problems. By incorporating the methods designed for the forward problem, we are able to develop block diagonal Schur complement based preconditioners, which also could be implemented in parallel over time. We provide extensive eigenvalue analysis for each preconditioner and demonstrate their effectiveness through numerical computations for a variety of problems. We are able to describe solvers which are robust to various parameters, including the mesh size and number of time-steps.

Contents

CHAPTER 1

Introduction

The study of partial differential equations (PDEs) is prevalent across a diverse range of subjects. From engineering and finance to medicine and biology, PDEs are used to model some of nature's most interesting phenomena. Often, for these applications, it is not the state at equilibrium which is of most importance, but rather how the system changes through time. For this reason, time-dependent PDE problems are of great importance to the scientific community.

Such problems require further consideration and typically more computational work than a steady state problem. One such consideration is that in order to discretize a time-dependent problem, it is possible to use either an explicit or implicit time-stepping scheme. With an explicit scheme, restrictions are usually placed on the size of the time-steps that can be taken in order to maintain numerical stability. In contrast, implicit schemes are typically stable, often unconditionally, and therefore often are favoured for problems with a long time scale as larger time-steps can be taken. However, as the solution at each time step of an explicit scheme relies only on information from a previous step or steps, parallelization across the spatial domain can be readily achieved. This is not the case for implicit schemes.

Another aspect of evolutionary problems is that they are inherently sequential.

One must know the solution at a given step before it is possible to compute the solution at the next point in time. This means that any type of parallelization in time is typically difficult for both implicit or explicit schemes. This type of system is non-self-adjoint, regardless of the differential operator. This is due to the fact that, even for spatially symmetric operators such as diffusion, the system is naturally progressing forwards in time and is thus non-self-adjoint in the space-time domain.

In this thesis, we will examine only linear time-dependent PDEs. In order to solve these problems, the simplest approach would be to find the solution at each time-step sequentially. If there were n degrees of freedom in an appropriate spatial discretization and ℓ time-steps in the temporal discretization, this corresponds to solving ℓ linear systems of size $n \times n$. Another approach would be to solve for all time-steps simultaneously in one coupled computation. This is able to be done by constructing the so-called 'all-at-once' system, which is a single linear system of size $n\ell \times n\ell$ which describes the solution at all time-steps. The coefficient matrix is naturally block structured and block lower triangular. Importantly, it is not required to construct this matrix explicitly in order to solve the system using an iterative method. Rather, all that is required is the spatial discretizations and a 'recipe' for completing a matrix-vector multiplication with the all-at-once matrix.

Many solution methods exist for such large, sparse problems [99]. In particular, iterative methods excel when matrix-vector multiplications are relatively cheap and this is the case for large, sparse matrices. However, preconditioning is typically of utmost importance to ensure that convergence is rapid for such methods. Preconditioners are designed so that the preconditioned system will have more favourable properties for the iterative method than the original system, while the preconditioner itself must be relatively easy to apply. In particular, we seek methods which have a total number of iterations independent of the problem parameters so that the cost of each iteration scales linearly with the total number of degrees of freedom. This is generally referred to as an *optimal solver*. Often the most important parameter, for which iteration counts should be independent, is the number of spatial degrees of freedom n. However, for time-dependent problems, it may also be highly desirable to achieve iteration numbers independent of the number of time-steps ℓ. Thus,

both cases should be considered for the development of methods for time-dependent problems.

As previously stated, the all-at-once system will always be non-symmetric. Typically it is much harder to prove robust convergence bounds for iterative methods for non-symmetric problems. This is because, unlike for symmetric problems, the eigenvalues of the preconditioned system alone will not determine convergence of an iterative method. In fact as shown by Greenbaum, Ptak, and Strakoš in [44], for a given set of eigenvalues, any (monotone) convergence curve is possible for GMRES for termination at the n-th step. However, for non-symmetric systems, we are able to make some statements about the termination of iterative methods if the structure of the Jordan form is known or diagonalizability is assumed. This will be the case for the preconditioners developed in Chapters 3 and 4. However, for practical implementation, these preconditioners will only be applied approximately through the use of a multigrid process. While convergence theory based on Jordan structure can be developed for the exact preconditioner, this will not be the case for the approximation as it is well known that the Jordan structure of a matrix is unstable to perturbation. However, in numerically testing we see convergence almost equal to what was predicted for the exactly applied preconditioner. This idea that the Jordan form of a nearby matrix may somehow influence convergence is an interesting concept. A similar type of concept is investigated in [102] but this phenomenon, to the best of our knowledge, remains largely unstudied.

When we think of time-dependent problems in a sequential fashion we tend to think that any parallelization over time will be limited. This is not to say however that methods have not been developed for such problems; space-time multigrid methods [28, 53, 75], the parareal algorithm [65] and domain decomposition methods [36, 48] are examples of possible approaches. However, when we consider a time-dependent problem as simply a large, block matrix system, we are in the realm of problems with which the numerical analysis community is intimately familiar. It is with this concept in mind that we have developed preconditioners for such problems which also have the potential to be applied in parallel. These preconditioners can then be used with existing iterative methods.

While constructing an all-at-once system is not typically done for time-dependent PDEs, which we will refer to as the forward problem, it is often used in time-dependent optimal control problems [3, 9, 86, 87, 105, 108, 109, 113, 121]. The optimal control, or PDE-constrained optimization, problems considered here, aim to find a control which ensures that the state variable is as close to a known desired state as possible, under the influence of a PDE. There are other types of optimal control problem which do not have this desired state property but instead aim to minimise a particular aspect of the problem. A common example is a drag minimisation problem, which aims to minimise drag on an object while under the influence of fluid flow PDEs such as the Navier-Stokes equation (see for example [20, 35, 39]). These types of problems will not be considered in this thesis. For time-dependent problems, the optimization problems need to visit each time-step and it is, therefore, natural to construct an all-at-once system in this context. Furthermore, in the optimization context, the adjoint of the constraint PDE is required. As we discussed, time-dependent problems are always non-self-adjoint, however, if care is taken, the discretization of adjoint problems can correspond to the matrix transpose of the all-at-once system for the forward problem. The adjoint, in this case, can equally be thought of as the backwards problem or a final value problem. Thus, within a time-dependent optimal control problem, we require the solution to the all-at-once system for a time-dependent PDE and its transpose. It is, therefore, advantageous to study the solution of such problems, even though in the context of simply the forward problem this is typically not done.

When it comes to the solution of optimal control problems, we find that the overall linear system is of saddle-point form. This is a well-studied area of linear algebra and many preconditioners have been developed for such problems [8]. Many of the preconditioners developed use the Schur complement of the system with a preconditioner. If we let the all-at-once system of the forward problem be given by the matrix $\mathcal{A}$, then the Schur complement effectively includes a term of the form $\mathcal{A}^T\mathcal{A}$. One might assume that if we have developed an effective preconditioner $\mathcal{P}$ for the matrix $\mathcal{A}$, then $\mathcal{P}^T\mathcal{P}$ might be an effective preconditioner for $\mathcal{A}^T\mathcal{A}$. In general, however, this is not the case. In fact, it was shown by Braess and Piesker in 1986 [13] that even for a small symmetric, positive definite example, $\mathcal{P}^T\mathcal{P}$ can be arbitrarily

bad as a preconditioner for $\mathcal{A}^T\mathcal{A}$ when $\mathcal{P}$ is effective for $\mathcal{A}$. Thus developing effective approximations to the Schur complement can often be more challenging than simply finding an effective approximation to the forward problem. However, this is typically the starting point. In this thesis, we will examine preconditioners for the forward problem and then use these ideas to develop analogous preconditioners for the time-dependent optimal control context.

This thesis is structured as follows. In Chapter 2 we will provide the necessary background theory required for the study of the time-dependent problems considered. This includes the formation of the all-at-once linear system through the use of the finite element method and temporal discretization using implicit schemes. We will also provide background to the iterative solvers used and details of the relevant convergence bounds, as well as an introduction to circulant preconditioners for Toeplitz systems which will be used to motivate preconditioners developed in Chapter 4. In Chapter 3, we will investigate the effectiveness of a simple block diagonal preconditioner for the forward problem through both supporting theory and numerical results. We continue to investigate the forward problem in Chapter 4 where we use the block Toeplitz structure of the all-at-once system to motivate block circulant preconditioners, which also have the ability to be applied in parallel over time.

In order to discuss time-dependent optimal control problems, we will provide an overview of such problems in Chapter 5. This will begin by looking at the steady-state Poisson control problem and then extending the ideas developed to the time-dependent heat control problem. We will incorporate the ideas developed for the forward problem in Chapters 3 and 4 to develop preconditioners for the heat control problem in Chapter 6. Through both eigenvalue analysis and numerical results, we will examine the effectiveness of each preconditioner for a range of parameter values. In Chapter 7, we extend these ideas to the time-dependent convection-diffusion control problem. Many additional considerations have to be made for the convection-diffusion control problem, due to the need for stabilization methods as well as the spatial non-self-adjointness of such problems. These considerations will be discussed and both theoretical convergence bounds and numerical results presented. Concluding remarks and suggested future work will be provided in Chapter 8.

CHAPTER 2

Background

In this chapter, we introduce some relevant background concepts crucial to the solution to time-dependent PDE problems. In Section 2.1, we detail the discretization of time-dependent problems, firstly through the use of the finite element method for spatial discretizations, and secondly, by using implicit time-stepping schemes for temporal discretization. We also introduce the concept of all-at-once solution for such problems. These concepts will be introduced for both the heat equation and the time-dependent convection-diffusion equation.

In Section 2.2 we discuss the solution to linear systems using iterative methods. The concept of preconditioning is introduced; we also introduce iterative methods, such as Chebyshev semi-iteration and multigrid, which can be used as preconditioning techniques. We then discuss iterative solvers for both non-symmetric and symmetric problems. The time-dependent PDE problems discussed in Chapters 3 and 4 will require non-symmetric solvers, while the optimal control problems examined in Chapters 6 and 7 will use symmetric solvers.

Lastly in Section 2.3, we introduce the theory of circulant preconditioners for Toeplitz matrices upon which the block circulant preconditioners discussed in Chapter 4 will rely.

2.1 Solution of Time-Dependent Problems

Perhaps the most fundamental time-dependent PDE problem is the heat equation. As indicated by the name, one practical application of this equation is to model the temperature distribution u, in a spatial domain Ω, subject to an external heat source f. However, it can also describe many other situations where diffusive effects arise (see for example [80]).

For simplicity of derivation we will restrict ourselves to the problem with Dirichlet boundary conditions given by

$$\begin{aligned} u_t - \nabla^2 u &= f \qquad \text{in } \Omega \times (0,T], \quad \Omega \subset \mathbb{R}^2 \text{ or } \mathbb{R}^3, \\ u &= g \qquad \text{on } \partial\Omega \times (0,T], \\ u(x,0) &= u_0. \end{aligned} \tag{2.1}$$

In this formulation, u is the state variable while f describes the forcing on the system. The spatial domain specified can either be two- or three-dimensional and all theory provided in this thesis will apply to a general d-dimensional system, although we will only provide numerical results for 2D systems.

2.1.1 Discretization

Throughout this thesis, we will use the *finite element method* to spatially discretize the problems considered. We will provide a brief overview of this method here, however, for a more complete explanation we refer the reader to [14, 25, 92]. The finite element method requires the weak form of the problem, however, in order to specify this, we first describe several spaces from which our solution and test functions will be drawn.

We will begin by defining our test space. Let $\mathcal{H}^1(\Omega)$ be the Sobolev space of functions v such that $v \in L_2(\Omega)$ and the first weak derivative of v is also contained in $L_2(\Omega)$. We then define the test space to be,

$$\mathcal{H}^1_{E_0}(\Omega) := \left\{ v \in \mathcal{H}^1(\Omega) | v = 0 \text{ on } \partial\Omega \right\}. \tag{2.2}$$

Since we are investigating a time-dependent problem, our solution space must also

involve a time component. At each time t, we have that $u(t) \in \mathcal{H}^1_E$ where this set is defined as,

$$\mathcal{H}^1_E(\Omega) := \left\{ u(t) \in \mathcal{H}^1(\Omega) | u(t) = g \text{ on } \partial\Omega \right\}. \tag{2.3}$$

Thus $u : (0, T] \to \mathcal{H}^1_E(\Omega)$ and the solution will be an element of the Bochner space $L_2(0, T; H^1_E(\Omega))$. Now since $\nabla^2 u(t) \in \mathcal{H}^{-1}(\Omega)$ where $\mathcal{H}^{-1}(\Omega)$ is the dual space of $\mathcal{H}^1(\Omega)$, this implies that $f(t)$ and $u_t(t)$ must also both be elements of $\mathcal{H}^{-1}(\Omega)$.

In the usual fashion, we obtain a weak formulation by multiplying by a test function v and integrating over Ω. This results in a weak formulation of (2.1) given by:

Find $u \in L_2(0, T; H^1_E(\Omega))$ for $t \in (0, T]$ such that

$$\int_\Omega u_t(t) v + \int_\Omega \nabla u(t) \cdot \nabla v = \int_\Omega f(t) v, \quad \forall v \in \mathcal{H}^1_{E_0}(\Omega). \tag{2.4}$$

At this point, we will consider the temporal discretization. Time-stepping schemes can be generally classified as either *explicit* or *implicit schemes*. Explicit schemes use only the values at the previous time-steps to solve for the current time; implicit schemes use values from both previous and current time-steps. It is well known that explicit schemes require restrictions on the size of the time-step size τ to ensure stability, which will depend on the spatial grid size [55, 56]. Thus for small spatial grids, only small time-step sizes are able to be taken and consequently, a lot of time-steps will be needed to reach the final time.

Instead, we will use an implicit method which is stable and thus enables larger time-steps to be taken. Explicit schemes do have the advantage of easily incorporating parallelization in the spatial domain, while this is not possible for implicit schemes. For either scheme, each time-step depends directly on the solution at the previous time-step and, therefore, it appears that each time-step must be computed sequentially.

For this description, we will use the general class of time-stepping schemes known as θ-schemes. The scheme will be implicit when the parameter $\theta \in (0, 1]$ and is

typically unconditionally stable when θ is between $\frac{1}{2}$ and 1. The case when $\theta = \frac{1}{2}$ is equivalent to the trapezoidal rule or *Crank-Nicolson scheme* and when $\theta = 1$ it is equivalent to the *Backwards Euler scheme*. If we consider constant time-steps of size τ with $T = \tau\ell$ we can define u^k to be the solution at time $t = k\tau$.

Thus, using a θ-scheme to discretize in time and for simplicity assuming that the forcing f is constant in time, we can write the discretized form of (2.4) as

$$\int_\Omega \frac{u^{k+1} - u^k}{\tau} v + \theta \int_\Omega \nabla u^{k+1} \cdot \nabla v + (1-\theta) \int_\Omega \nabla u^k \cdot \nabla v = \int_\Omega f v, \quad \forall v \in \mathcal{H}^1_{E_0}(\Omega), \tag{2.5}$$

for $k = 0, 1, \ldots, \ell - 1$. It is now time to describe a finite dimensional approximation to this weak form. We assume that $V_0^h \subset \mathcal{H}^1_{E_0}(\Omega)$ is a finite n-dimensional vector space of test functions for which $\{\phi_1, \phi_2, \ldots, \phi_n\}$ is a convenient basis. As we have assumed Dirichlet boundary conditions, we need to extend the space in order to describe the boundary data. To do this we define functions $\phi_{n+1}, \ldots \phi_{n+n_\partial}$ and coefficients $U_{n+1}, \ldots, U_{n+n_\partial}$ such that $\sum_{j=n+1}^{n+n_\partial} U_j \phi_j$ interpolates the boundary data. Thus our solutions at each time-step will be from the space

$$V_E^h = \operatorname{span}\{\phi_1, \phi_2, \ldots, \phi_n\} + \sum_{j=n+1}^{n+n_\partial} U_j \phi_j, \tag{2.6}$$

and the finite element approximation at the time-step k is given by $u_h^k \in V_E^h$. This can be uniquely determined by the vector $\mathbf{u} = (U_1, \ldots, U_n)^T$ of unknown coefficients in the expansion

$$u_h^k = \sum_{j=1}^{n} U_j \phi_j + \sum_{j=n+1}^{n+n_\partial} U_j \phi_j. \tag{2.7}$$

This type of discretization for time-dependent systems is referred to as *Faedo-Galerkin method* and a more extended explanation can be found for example in [101, 92]. Using this approximation, we now have an discrete approximation to the weak formulation:

For $k = 0, \ldots, \ell - 1$ find $u_h \in V_E^h$ such that

$$\int_\Omega \frac{u_h^{k+1} - u_h^k}{\tau} v_h + \theta \int_\Omega \nabla u_h^{k+1} \cdot \nabla v_h + (1-\theta) \int_\Omega \nabla u_h^k \cdot \nabla v_h = \int_\Omega f v_h, \quad \forall v_h \in V_0^h. \tag{2.8}$$

Substituting in the approximation (2.7) and the test functions into (2.8), then for each time-step k, we find that this is equivalent to solving for U_j^{k+1} for $j = 1, \ldots, n$ where

$$\sum_{j=1}^{n+n_\partial} \frac{(U_j^{k+1} - U_j^k)}{\tau} \int_\Omega \phi_j \phi_i + \sum_{j=1}^{n+n_\partial} (\theta U_j^{k+1} + (1-\theta) U_j^k) \int_\Omega \nabla \phi_j \cdot \nabla \phi_i = \int_\Omega f \phi_i \tag{2.9}$$

for $i = 1, \ldots, n$. This is equivalent to solving the following linear system

$$(M + \theta \tau K)\mathbf{u}_{k+1} = (M - (1-\theta)\tau K)\mathbf{u}_k + \tau \mathbf{f} + \mathbf{d}, \tag{2.10}$$

for $k = 0, \ldots, \ell - 1$ where

$$K = [k_{ij}] \in \mathbb{R}^{n \times n}, \qquad k_{ij} = \int_\Omega \nabla \phi_j \cdot \nabla \phi_i \tag{2.11a}$$

$$M = [m_{ij}] \in \mathbb{R}^{n \times n}, \qquad m_{ij} = \int_\Omega \phi_j \phi_i \tag{2.11b}$$

$$\mathbf{f} = [f_i] \in \mathbb{R}^n, \qquad f_i = \int_\Omega f \phi_i \tag{2.11c}$$

$$\mathbf{u}_k = [u_i^k] \in \mathbb{R}^n, \qquad u_i^k = U_i^k \tag{2.11d}$$

$$\mathbf{d} = [b_i] \in \mathbb{R}^n, \qquad d_i = -\sum_{j=n+1}^{n+n\partial} U_j \nabla \phi_j \cdot \nabla \phi_i \tag{2.11e}$$

It still remains to make a choice of our basis functions ϕ_i. For the problems we will consider numerically, the domain Ω will be square although this is by no means a requirement of our methods. The natural spatial discretization on a square grid is to divide the square into a grid of smaller squares such that each one has a height equal to h. A plot of such a grid is given in Figure 2.1. We note that this type of discretization could be easily generalized to 3 dimensions by constructing a grid of cubes as well as to more general quadrilaterals (see [25, Section 1.3] for more detail).

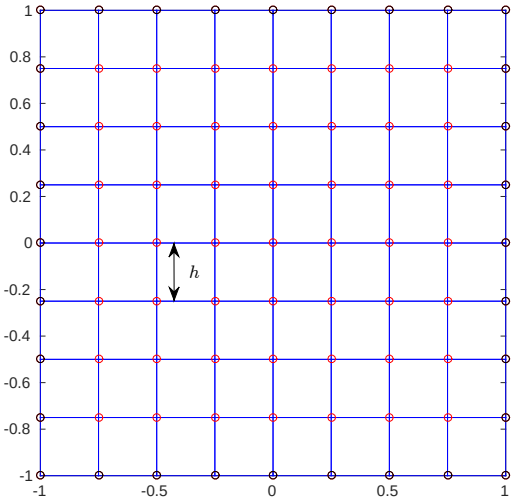


Figure 2.1: **Q1** finite element grid on the domain $\Omega = [-1,1] \times [-1,1]$.

We define our basis functions ϕ_i to be the bilinear functions such that at point $\mathbf{x}_j$ we have

$$\phi_i(\mathbf{x}_j) = \delta_{i,j},$$

where $\delta_{i,j}$ is the Kronecker delta function. These elements are referred to as **Q1** finite elements. As is also quite common, we also could divide our domain into a series of triangular (or tetrahedral) elements. If we defined linear elements in a similar way on such a grid, then we refer to these types of elements as **P1** finite elements.

As ϕ_i has been chosen such that it has small support, it is evident that many entries in both the mass matrix M and the stiffness matrix K will be zero. This sparsity will be particularly useful for our solution with iterative methods. It is also possible to state information relating to the spread of eigenvalues of each of the matrices.

Theorem 2.1 [25, Proposition 1.29 and Theorem 1.32]. *For a* **Q1** *or* **P1** *finite element approximation on a shape regular, quasi-uniform subdivision of* $\mathbb{R}^d$ *where* $d \in [2,3]$*, then for the mass matrix* M *and the stiffness matrix* K *as defined in* (2.11)

we have

$$c_1 h^d \leq \frac{\mathbf{v}^T M \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq c_2 h^d, \tag{2.12}$$

$$d_1 h^d \leq \frac{\mathbf{v}^T K \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq d_2 h^{d-2}, \tag{2.13}$$

for all $\mathbf{v} \in \mathbb{R}^d$*, where* h *is the longest edge of the mesh or grid and* $c_1, c_2, d_1,$ *and* d_2 *are constants independent of* h*.*

We can see from these bounds that all of the eigenvalues of M will behave like h^d and, therefore, M is spectrally equivalent to the identity matrix scaled by a factor of h^d. This is not the case, however, for the stiffness matrix, as we expect the minimal and maximal eigenvalues of K to differ by a factor of h^2. We will return to these bounds at several points during this thesis in order to bound the eigenvalues of matrices made up of mass and stiffness matrices.

2.1.2 All-At-Once Methods

We have seen that we can solve the heat equation by constructing a series of linear systems which we solve sequentially. We can think of the approach as the classical or traditional method for solving such a problem. Another alternative however, is to solve for all time-steps simultaneously by constructing the 'all-at-once' system. With constant time-steps we can easily do this be rewriting the series of equations described in (2.10) as

$$\mathcal{A}\mathbf{u} = \mathbf{b}, \tag{2.14}$$

where

$$\mathcal{A} = \begin{bmatrix} A & & & \\ B & A & & \\ & \ddots & \ddots & \\ & & B & A \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_\ell \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \tau\mathbf{f} + \mathbf{d} - B\mathbf{u}_0 \\ \tau\mathbf{f} + \mathbf{d} \\ \vdots \\ \tau\mathbf{f} + \mathbf{d} \end{bmatrix}, \tag{2.15}$$

with $A = M + \theta\tau K$, $B = -M + (1-\theta)\tau K$ and u_0 is the discretization of the initial conditions. We note that this system is now immense; rather than solving ℓ linear

systems of size $n \times n$, we are solving one system of size $n\ell \times n\ell$.

The matrix $\mathcal{A}$ is lower block triangular and thus one method for solving such a system would be to perform block forward substitution. We note that using this method would correspond exactly to solving the series of equations (2.10) in the sequential manner described in the previous section.

Alternatively, we can treat the system as a whole and turn to iterative methods to solve this large, sparse system. This is perhaps an unusual approach for a time-dependent system, however, we will see in Chapter 5 that it is a common approach for time-dependent PDE-constrained optimization problems. For these problems, since the optimization problem needs to visit every time-step of the state, it is somehow natural to combine all of the time-steps in the one linear system. Indeed, an all-at-once system for a heat control problem with the same discretization methods as described here, will contain within it the exact matrix $\mathcal{A}$ described in (2.15).

Just as we have for the θ-method, we can also easily describe the all-at-once system for higher order time-stepping schemes such as the Backward Differentiation Formula (BDF) methods. For example, using the 2-step BDF method the equations in (2.10) are instead

$$(M + \frac{2}{3}\tau K)\mathbf{u}_{k+1} - \frac{4}{3}M\mathbf{u}_k + \frac{1}{3}M\mathbf{u}_{k-1} = \tau\mathbf{f} + \mathbf{d}, \tag{2.16}$$

for $k = 0, \ldots, \ell - 1$, noting that in this case we require two initial conditions $\mathbf{u}_0$ and $\mathbf{u}_{-1}$. The all at once system in this case is given by,

$$\mathcal{A}_{BDF2} = \begin{bmatrix} A_0 & & & & \\ A_1 & A_0 & & & \\ A_2 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & A_2 & A_1 & A_0 \end{bmatrix}, \tag{2.17}$$

with $A_0 = M + \frac{2}{3}\tau K$, $A_1 = -\frac{4}{3}M$ and $A_2 = \frac{1}{3}M$.

In addition to being block lower triangular, it is evident that these matrices are also block Toeplitz, i.e. a matrix with blocks repeated along the diagonals. This is

only the case as we have taken constant time-step sizes τ. It is certainly possible to construct the system with non-constant time-steps, however, we would lose block Toeplitz structure in this case. Furthermore, we would require the size of all time-steps to be known in advance; this is typically not the case with adaptive time-stepping. These issues are some of the main drawbacks of using the all-at-once method for time-dependent problems.

Another issue that arises with all-at-once methods is their extensive memory requirements. Although, as previously stated, it is not required to store the entire $n\ell \times n\ell$ coefficient matrix $\mathcal{A}$, the solution at all times-steps is stored. For large problems, in particular the optimal control problems discussed later in this thesis, this can be prohibitive. Several methods have been proposed for this issue including check pointing [45], multiple-shooting [52], and low-rank approaches [106]. We have not investigated any of these methods here and will assume throughout that sufficient storage is available for our methods. We see no reason, however, that methods such as the low-rank approach of Stoll and Breiten in [106] could not be used combined with our proposed approaches for the solution of optimal control problems discussed in Chapters 6 and 7.

2.1.3 Convection-Diffusion Equation

An extension of the heat equation, which models simple diffusive effects, is the time-dependent *convection-diffusion* problem. This can be stated as

$$\begin{aligned} u_t - \epsilon\nabla^2 u + \mathbf{w}\cdot\nabla u &= f \qquad && \text{in } \Omega \times (0,T], \quad \Omega \subset \mathbb{R}^2 \text{ or } \mathbb{R}^3, \\ u &= g \qquad && \text{on } \partial\Omega \times (0,T], \\ u(x,0) &= u_0, \end{aligned} \tag{2.18}$$

where $\epsilon > 0$ and $\mathbf{w}$ is the velocity with which the state, u, is being convected. As a simple example, consider u as a pollutant which is being transported by the velocity of the containing fluid, while simultaneously being subject to diffusive effects. Typically for practical application, the effect of diffusion is minimal in comparison to the convection and therefore we typically assume that $\epsilon \ll \|\mathbf{w}\|$ [25]. Additionally, we

assume that the velocity is divergence free so that $\nabla \cdot \mathbf{w} = 0$. For simplicity, we have assumed Dirichlet boundary conditions, however, other conditions could be applied and once again we assume that the forcing is constant in time.

As with the heat equation, we require solutions $u(t)$ to be from the space $\mathcal{H}^1_E(\Omega)$. Thus, the weak formulation of this problem is the following:

Find $u \in L_2(0,T;\mathcal{H}^1_E(\Omega))$ such that for $t \in (0,T]$,

$$\int_\Omega u_t(t)v + \epsilon \int_\Omega \nabla u(t)\cdot\nabla v + \int_\Omega (\mathbf{w}\cdot\nabla u(t))v = \int_\Omega fv, \quad \forall v \in \mathcal{H}^1_{E_0}(\Omega). \tag{2.19}$$

To discretize this problem, we will again use Galerkin finite elements and for simplicity, we will restrict ourselves to Backward Euler time-stepping ($\theta = 1$), although this is by no means required. Using the approximation for u_h from (2.7) we obtain the following linear system

$$M\mathbf{u}_{k+1} - M\mathbf{u}_k + \tau(\epsilon K + N)\mathbf{u}_{k+1} = \tau\mathbf{f} + \mathbf{d}, \tag{2.20}$$

where M and K are the standard finite element mass and stiffness matrices as defined in (2.11). The terms $\mathbf{f}$ and $\mathbf{d}$ describe the forcing f and the boundary conditions as also defined in (2.11). We also have the effects of the convective term incorporated in the matrix $N \in \mathbb{R}^{n\times n}$ defined as

$$N = [\nu_{ij}] \in \mathbb{R}^{n\times n}, \quad \nu_{ij} = \int_\Omega (\mathbf{w}\cdot\nabla\phi_j)\phi_i. \tag{2.21}$$

One property of the convection-diffusion equation is that boundary or internal layers may be present in the solution. If the grid on which the solution is discretized is not sufficiently fine, it will not be able to accurately represent this layer and oscillations can appear in the numerical solution, typically for small ϵ. Furthermore, in some cases these oscillations can be propagated by the velocity field resulting in oscillations in all areas of the domain and not just limited to the region around the layer [71].

In order to account for this, various stabilization methods have been proposed which add an additional term $\widetilde{T}$ to the discretization to mitigate potential oscillations.

This results in solving

$$M\mathbf{u}_{k+1} - M\mathbf{u}_k + \tau\widetilde{K}\mathbf{u}_{k+1} = \tau\bar{\mathbf{f}}_k + \mathbf{d}, \tag{2.22}$$

where $\widetilde{K} = \epsilon K + N + \widetilde{T}$ and $\bar{\mathbf{f}}$ is often also redefined with additional terms.

One popular stabilization approach is the *Streamline Upwind Petrov-Galerkin* (SUPG) method which was introduced by Hughes and Brooks [54]. This method is widely discussed in the literature, for example, for the forward problem in [25, 34] and for the control problem in [19, 50, 94]. For this method we define

$$\widetilde{T} = [\tau_{ij}^{\delta}] \in \mathbb{R}^{n\times n}, \quad \tau_{ij}^{\delta} = \delta \int_{\Omega} (\mathbf{w}\cdot\nabla\phi_i)(\mathbf{w}\cdot\nabla\phi_j) - \epsilon\delta \sum_k \int_{\Delta_k} (\nabla^2\phi_i)(\mathbf{w}\cdot\nabla\phi_j) \tag{2.23}$$

$$\bar{\mathbf{f}} = [f_i] \in \mathbb{R}^n, \qquad f_i = \int_{\Omega} f\phi_i + \sigma \int_{\Omega} f\mathbf{w}\cdot\nabla\phi_i. \tag{2.24}$$

The parameter δ is a stabilization parameter and is typically defined for each element Δ_k individually. As suggested in [25] we define the parameter as

$$\delta = \begin{cases} 0 & \text{if } \mathrm{Pe} \le 1, \\ \frac{h}{2\|w\|_2}\left(1 - \frac{1}{Pe}\right) & \text{if } \mathrm{Pe} > 1, \end{cases} \tag{2.25}$$

where Pe is the *element Peclet number* defined as

$$\mathrm{Pe} = \frac{h\|\mathbf{w}\|}{2\epsilon}. \tag{2.26}$$

As pointed out in [94], the SUPG method is not adjoint consistent and for optimal control problems, this is typically desired. We will discuss this further in Chapter 7. Another method which is adjoint-consistent is the *Local Projection Scheme* (LPS) discussed in [5, 6, 46, 86]. In LPS, we define

$$\widetilde{T} = [\tau_{ij}^{\delta}] \in \mathbb{R}^{n\times n}, \quad \tau_{ij}^{\delta} = \delta \int_{\Omega} \left(\mathbf{w}\cdot\nabla\phi_i - \pi_h(\mathbf{w}\cdot\nabla\phi_i)\right)\left((\mathbf{w}\cdot\nabla\phi_j) - \pi_h(\mathbf{w}\cdot\nabla\phi_j)\right) \tag{2.27}$$

$$\bar{\mathbf{f}} = [f_i] \in \mathbb{R}^n, \qquad f_i = \int_{\Omega} f\phi_i. \tag{2.28}$$

where the stabilization parameter is given by

$$\delta = \begin{cases} 0 & \text{if Pe} < 1, \\ \frac{h}{\|w\|} & \text{if Pe} \geq 1. \end{cases} \tag{2.29}$$

Here π_h represents an L_2-orthogonal projection operator which we will define on patches of our domain. If we consider **Q1** finite elements on an equally spaced mesh, then we can divide into patches of two elements in each direction. On each patch we define $\pi_h(v)$, to be equal to the integral of v over this patch divided by the area of the patch.

For the diffusion operators considered in the previous chapter, we were able to provide bounds on the quadratic form of the stiffness matrix K. This is not the case for $\widetilde{K}$. However, it is known that $\epsilon K + \widetilde{T}$ is positive semi-definite and that N is skew-symmetric [25]. This leads to the positive semi-definiteness of the symmetric part of $\widetilde{K}$, defined as $\frac{1}{2}(\widetilde{K} + \widetilde{K}^T)$ and this result will be used to prove certain results in Chapter 7.

Either method of stabilization results in the series of linear systems described in (2.22) which, as for the heat equation, could be solved in a sequential manner or in an all-at-once fashion. In either case, we will require the solution of a linear system, typically achieved via iterative methods. In the following section, we will describe in more detail the iterative methods which may be utilized for the solution of the systems described here.

2.2 Iterative Methods

Suppose we wish to solve the system

$$A\mathbf{x} = \mathbf{b}, \tag{2.30}$$

where $A \in \mathbb{R}^{n\times n}, \mathbf{b} \in \mathbb{R}^n$ and, as is often the case when solving PDE problems, A is large and sparse. Perhaps the most common methodology for solving linear systems of equations is to use a *direct method*. A direct method computes the solution by

Gaussian elimination, typically through the use of an LU-factorization. For a dense matrix however, the solution requires $\frac{2}{3}n^3$ flops (to highest order), which can be reduced to $2m^2n$ flops for a banded matrix with bandwidth m [43]. For an overview of such methods see [21].

Importantly, direct methods do not maintain the sparsity of A and instead factorization can fill in many entries. Another approach which is regularly used for large, sparse matrices is to use *iterative methods*. These approaches produce a sequence of iterates $\{\mathbf{x}_j\}$ which will, hopefully, converge to the solution. In order to obtain each iterate, the matrix A is only visited using matrix-vector products and, when A is sparse, these are relatively cheap to compute. Additionally, often the solution is only required to be accurate up to a certain tolerance and iterative methods can be terminated when this tolerance is reached, thereby reducing the total work required. As the matrices considered in this thesis will typically be large and sparse, we will utilize only iterative methods for solution of our systems.

However, iterative methods are not always successful in a short amount of time. Therefore, one of the most important practical aspects of iterative methods is the use of *preconditioning*. At a fundamental level, preconditioning aims to transform the system into one which has characteristics which improve the convergence of the iterative method.

Let us consider the invertible matrix P as our preconditioner. The *left-preconditioned system* would be

$$P^{-1}A\mathbf{x} = P^{-1}\mathbf{b}, \tag{2.31}$$

or alternatively P could be applied as a right-preconditioner where

$$AP^{-1}\mathbf{y} = \mathbf{b}, \quad \mathbf{y} = P\mathbf{x}. \tag{2.32}$$

The preconditioned system should now be easier to solve; the process of applying the preconditioner should also be easy compared with solving with A. In order to illustrate this point we can consider the two extreme cases:

- $P = I$: In this case, P is easy to invert but the preconditioned system is unchanged.

- $P = A$: In this case, the system is now solved in one iteration but the preconditioner is just as difficult to invert as A.

Therefore, we hope to find a P that is somehow a compromise between these two alternatives. Preconditioners can also aim to approximate the inverse of A by being the application of a few steps of certain iterative methods. These methods can typically be thought of as the application of a linear operator and, therefore, can be thought of exactly as if they were a matrix P.

We can also frame the role of preconditioners in the context of mappings between spaces. Suppose that the solution vector $\mathbf{x}$ represents an element of the space X and the matrix A is a mapping from X to its dual space X^* or $A : X \to X^*$. Thus, we also have that $\mathbf{b} \in X^*$. If we use an iterative method to solve the original equation (2.30), our right hand side $\mathbf{b}$ and our solution vector $\mathbf{x}$ are in different spaces. However, if we construct a preconditioner P^{-1}, which is a mapping from X^* to X, then for left preconditioning we have that $P^{-1}A : X \to X$. Alternatively, for right preconditioning we have that $AP^{-1} : X^* \to X^*$. In both of these cases, the iterative method is trying to obtain a solution vector which is in the same space as the right hand side vector. An in depth discussion of this approach to the preconditioning of discretizations of elliptic and parabolic operators can be found in [59] or in more generality see [68].

The role of preconditioning is, therefore, undeniably important and the focus of this thesis is to introduce effective preconditioners for iterative methods in order to solve the partial differential equation problems considered. In order to do this, we first introduce each of the iterative methods which will be used.

Two of the methods, namely Chebyshev semi-iteration and Multigrid, will be used as preconditioners. For the outer method, we consider GMRES and BiCGStab for non-symmetric problems while for symmetric problems we utilize MINRES. We also consider LSQR which solves the normal equations of the system. These methods are introduced in the following sections.

2.2.1 Simple Iterations

Perhaps the most elementary iterative method is appropriately known as a *simple iteration* and is also sometimes referred to as a *stationary* or *fixed point method*. This method is based on a matrix splitting of A such that

$$A = P - \Sigma, \tag{2.33}$$

where P is non-singular. We construct as sequence of iterates $\{\mathbf{x}_j\}_{j=1,2,\ldots}$ by the relation

$$P\mathbf{x}_j = \Sigma\mathbf{x}_{j-1} + \mathbf{b}, \quad j = 1, 2, \ldots \tag{2.34}$$

with a suitable initial guess $\mathbf{x}_0$. By simple manipulation it can be shown that $P^{-1}\Sigma = I - P^{-1}A$ and thus we can rewrite (2.34) as

$$\mathbf{x}_j = (I - P^{-1}A)\mathbf{x}_{j-1} + P^{-1}\mathbf{b}. \tag{2.35}$$

The matrix $I - P^{-1}A$ is known as the *iteration matrix*. We define the *spectral radius* of a matrix C, denoted by $\rho(C)$, to be

$$\rho(C) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of} C\}.$$

Convergence properties of this iteration can now be described by the following well known theorem.

Theorem 2.2 [99, Theorem 4.1]. *The iteration* (2.35) *converges for any* $\mathbf{x}_0$ *and* $\mathbf{b}$ *if and only if* $\rho(I - P^{-1}A) < 1$.

In order for a preconditioner, P, to be effective for a simple iteration, it is crucial that it reduces the spectral radius of the iteration matrix. This will be determined by the choice of matrix splitting.

If we let $A = D + L + U$ where D is the diagonal part, L is the strictly lower triangular part and U is the strictly upper triangular part of A then we can define the following possible splittings:

- *Jacobi iteration* ($P = D$): Here P will be easy to invert as it requires only the

straightforward inversion of a diagonal matrix.

- *Relaxed Jacobi iteration* ($P = \omega D$): Here we use a relaxation parameter ω which, if chosen appropriately, can reduce the spectral radius of $I - P^{-1}A$ or result in a more effective smoother for a multigrid process as discussed later in this chapter.

- *Gauss-Seidel iteration* ($P = D + L$): Here we take the lower triangular part of the matrix as the preconditioner.

If our matrix A was a block matrix, each of these methods could be converted to the analogous block version. Then D would refer to the block diagonal matrix and similarly for L and U. We note for block methods to work, we need that all the matrices on the diagonal are invertible. For more discussion of these methods see for example [2, 40].

A simple iteration will converge if $\rho(I - P^{-1}A) < 1$, however, this does not imply anything about the rate of convergence. Let $\mathbf{e}_j$ denote the error at the j-th iteration given by $\mathbf{e}_j = \mathbf{x} - \mathbf{x}_j$. The error at the $(j+1)$-th iteration will be given by

$$\mathbf{e}_{j+1} = (I - P^{-1}A)\mathbf{e}_j$$

and taking norms on both sides we obtain the following inequality,

$$\|\mathbf{e}_{j+1}\| \leq \|I - P^{-1}A\|\|\mathbf{e}_j\|. \tag{2.36}$$

Therefore at each iteration, the norm of the error will be bounded by at least a factor of $\|I - P^{-1}A\|$ times the error at the previous iteration. If $\|I - P^{-1}A\| < 1$ it naturally follows that the error will be reduced at each iteration. The norm here could be any vector induced matrix norm; a common choice is the standard Euclidean norm (or 2-norm). For normal matrices, the 2-norm coincides with the spectral radius and therefore the error would be reduced by a factor of $\rho(I - P^{-1}A)$ at each iteration. However, for non-normal matrices we can have the situation that

$$\rho(I - P^{-1}A) < 1 < \|I - P^{-1}A\|.$$

In this case, the error can grow for a finite number of iterations before ultimately converging. This means, however, that we have no way of knowing the error after a fixed number of iterations; as a consequence, the number of iterations required to achieve a desired accuracy cannot be determined a priori. This is not the case, however, for the following method which makes it ideally suited as a preconditioner.

2.2.1.1 Chebyshev Semi-Iteration

Suppose we have a series of iterates $\{\mathbf{x}_j\}_{j=1,2,\ldots}$ from a simple iteration as described above with iteration matrix $I - P^{-1}A$. It is reasonable to believe that by combining information from all of these iterates we may be able to obtain a better approximation $\mathbf{y}_j$ to the real solution $\mathbf{x}$. Thus we define,

$$\mathbf{y}_j = \sum_{i=1}^{j} \alpha_i \mathbf{x}_i. \tag{2.37}$$

We can think of the coefficients α_i as defining a polynomial p_j of degree j. If $\mathbf{x}_j = \mathbf{x}$ for all j then we should have $\mathbf{y}_j = \mathbf{x}$ so this implies the condition that $\sum_{i=1}^{j} \alpha_i = 1$. This condition could equally be written as requiring $p_j(1) = 1$.

It can be shown that

$$\|\mathbf{y}_j - \mathbf{x}\| \leq \|p_j(I - P^{-1}A)\| \|\mathbf{x} - \mathbf{x}_0\|, \tag{2.38}$$

thus for fast convergence we would like to minimize $\|p_j(I - P^{-1}A)\|$. If we assume that the iteration matrix is symmetric and has eigenvalues λ_i contained in the interval $[a, b]$ where $-1 < a$ and $b < 1$ then we have

$$\|p_j(T)\| = \max_{\lambda_i} |p_j(\lambda_i)| \leq \max_{a \leq \lambda \leq b} |p_j(\lambda)|. \tag{2.39}$$

Thus, in order to ensure fast convergence we require that the polynomial should be small in the region between the extremal eigenvalues of the iteration matrix.

The Chebyshev semi-iteration which we will use is based on the relaxed Jacobi iteration described in the previous section. This method was originally devised by Flanders and Shortley [32] in 1950 but was more extensively analysed by Golub and

Varga [41, 42] in 1961. In this method, the relaxation parameter ω is specifically selected so as to ensure that the eigenvalues of the iteration matrix $I - \omega D^{-1}A$ are symmetric about the origin. If $\lambda(D^{-1}A) \in [\lambda_{min}, \lambda_{max}]$ then this is achieved when,

$$1 - \omega\lambda_{min} = -(1 + \omega\lambda_{max}) \quad \Rightarrow \quad \omega = \frac{2}{\lambda_{max} + \lambda_{min}}. \tag{2.40}$$

The eigenvalues will now be bounded within $[-\rho, \rho]$ where $\rho = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}$. We note that in order to calculate these parameters we required information on the extremal eigenvalues of the preconditioned system.

The polynomial which is minimized over the interval $[-\rho, \rho]$ and also satisfies the requirement that $p_j(1) = 1$ is the *shifted and scaled Chebyshev polynomial* given by

$$p_j(t) = \frac{T_j(\frac{t}{\rho})}{T_j(\frac{1}{\rho})}, \tag{2.41}$$

where $T_j(t) = \cos(j\omega \cos^{-1} t)$. Using trigonometric identities, it can be shown that Chebyshev polynomials can be defined by a three term recurrence formula, namely

$$T_{j+1}(t) = 2tT_j(t) - T_{j-1}(t), \quad j = 1, 2, \ldots, \; t \in [-1, 1], \tag{2.42}$$

which can be used to provide a fast and efficient ways of calculating each of the successive iteration vectors $\mathbf{y}_j$ without explicitly calculating all of the original iterates $\mathbf{x}_j$. Thus we can define the following pseudocode to implement the algorithm.

Algorithm 1 Chebyshev semi-iteration to solve $A\mathbf{x} = \mathbf{b}$, where $\lambda(D^{-1}A) \in [\lambda_{min}, \lambda_{max}]$

$w = \frac{2}{\lambda_{max} + \lambda_{min}}, \rho = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}$
Choose $\mathbf{y}_0, \theta_0 = 1, (\mathbf{y}_{-1} = \mathbf{0})$
for $j = 0, 1, \ldots$ **do**
  $\theta_{j+1} = \left(1 - \frac{\rho\theta^2}{4}\right)^{-1}$
  $\mathbf{z}_j = wD^{-1}(\mathbf{b} - A\mathbf{y}_j)$
  $\mathbf{y}_{j+1} = \theta_{j+1}(\mathbf{z}_j + \mathbf{y}_j - \mathbf{y}_{j-1}) + \mathbf{y}_{j-1}$
end for

The convergence rate is bounded as follows:

$$\frac{\|\mathbf{x}-\mathbf{y}_j\|_2}{\|\mathbf{x}-\mathbf{x}_0\|_2} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^j, \tag{2.43}$$

where κ is the condition number of $D^{-1}A$ [94]. Thus, the number of iterations to obtain a desired accuracy can be determined a priori. In order for this method to be particularly effective we require good estimates of the extremal eigenvalues. However, as we will see in Chapter 5, there exists bounds for Galerkin finite element mass matrices and therefore Chebyshev semi-iteration can be employed as an effective preconditioner in this situation [94, 96, 118]. This method was also shown to be an effective preconditioner when used within the GeneRank algorithm in [10]. More detailed discussion of this method can be found in [40, 94].

2.2.2 Multigrid

The simple iterations discussed in the previous section can be applied to a matrix of any type. However, the matrices we will typically be dealing with in this thesis are discretizations of differential operators and, therefore, they have a very specific structure. *Multigrid* is a method which was specifically developed for problems which have an underlying mesh structure; it has been shown to be an *optimal solver*, meaning that the overall computational work scales linearly with n, so is independent on the mesh parameter h, for elliptic PDE operators.

We will not provide a thorough description of the method here, but typically use it in a 'black box' fashion and refer the interested reader to [15, 25, 49, 119] for more complete derivations. We will outline Geometric Multigrid (GMG) methods in this section and briefly discuss Algebraic Multigrid (AMG) and other variants in Section 2.2.2.1.

The underlying idea of multigrid methods is that a smooth function can be well approximated on a coarse grid. Furthermore, simple iterations of the form (2.35) are shown to quickly reduce the highly oscillatory modes of the error [15, Chapter 2]. Therefore, after a few iterations of a simple iteration, the error is comparatively smooth and can, therefore, be restricted to a coarser grid, where computational work

is much cheaper.

The fundamental elements required for the method are the following:

- **Smoothing operator S:** The first step of the method is to smooth the error through the application of a simple iteration of the form

$$\mathbf{x}_{j+1} = (I - S^{-1}A)\mathbf{x}_j + S^{-1}\mathbf{b}.$$

  Here S is a matrix splitting of A such that $A = S - \Sigma$ as described in (2.33). Post-smoothing is also employed using S^T. Typical examples of the iterations used are the (relaxed) Jacobi and Gauss-Seidel iterations described in Section 2.2.1.

- **Prolongation operator Q:** Also known as an interpolation operator, this transfers the correction term from the coarse grid back onto the fine grid and is just an interpolation between the coarse grid points.

- **Restriction operator R:** Conversely, we require an operator which restricts residuals on the fine grid to the coarse grid. This is typically taken to be $R = Q^T$.

Further discussion on the choices of prolongation and restriction operators can be found for example in [15, Chapter 2] and [25, Chapter 2].

Multigrid is defined as a recursive algorithm; each grid level is restricted to a coarser level until we reach the critical level, l_c, which has sufficiently few points so that the error can be calculated exactly on this level using a direct method. The process of consecutively restricting to the finest level before returning to the fine grid level is referred to as a V-cycle due to its structure. A schematic diagram of a V-cycle is presented in Figure 2.2. Other cycles are possible such as the W-cycle which will not be discussed here. Pseudocode for a GMG V-cycle is provided in Algorithm 2.

An important property of multigrid is that the method can be described as a simple iteration of the form (2.35). If we consider a simple 2-grid algorithm with s steps of pre- and post-smoothing we find that the error at each iteration can be written in the form,

$$\mathbf{e}_{j+1} = (I - S^{-T}A)^s(I - Q\overline{A}^{-1}Q^TA)(I - S^{-1}A)^s\mathbf{e}_j. \tag{2.44}$$

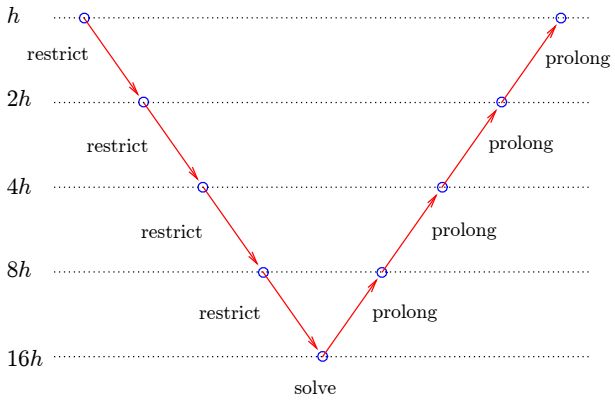


Figure 2.2: Schmematic of a single geometric multigrid V-cycle with 5 levels.

Algorithm 2 Geometric Multigrid V-cycle which can be used as a preconditioner on the system $A\mathbf{x} = \mathbf{b}$.

function `GMG_VCYCLE`$(A, \mathbf{b}, \mathbf{x}, \texttt{level})$
  for $j = 1, \ldots, s$ **do**
    $\mathbf{x} = (I - S^{-1}A)\mathbf{x} + S^{-1}\mathbf{b}$
  end for
  if `level`$= l_c$ **then**
    Solve $A\mathbf{x} = \mathbf{b}$ (using a direct method)
  else
    $\bar{\mathbf{r}} = Q^T(\mathbf{b} - A\mathbf{x})$
    $\bar{A} = Q^T A Q$
    $\bar{\mathbf{e}} =$ `GMG_vcycle`$(\bar{A}, \bar{\mathbf{r}}, \bar{\mathbf{e}}, \texttt{level} + 1)$
    $\mathbf{x} = \mathbf{x} + Q\bar{\mathbf{e}}$
  end if
  for $j = 1, \ldots, s$ **do**
    $\mathbf{x} = (I - S^{-T}A)\mathbf{x} + S^{-T}\mathbf{b}$
  end for
end function

If the iteration matrix has the form $I - P^{-1}A$ this implies that

$$P^{-1} = A^{-1} - (I - S^{-T}A)^s(I - Q\overline{A}^{-1}Q^TA)(I - S^{-1}A)^sA^{-1}, \tag{2.45}$$

where $\overline{A}^{-1}$ is the inverse of the operator reduced to the coarse grid level. As described in the previous section, we can describe the convergence of a simple iteration with respect to a convergence factor ρ. For multigrid as described here, we can make the following statement regarding convergence for a Poisson problem.

Theorem 2.3 [25, Theorem 2.5]. *Given an $\mathcal{H}^2$-regular problem, and a quasi-uniform subdivision of size h in $\mathbb{R}^2$ or $\mathbb{R}^3$, when solving $A\mathbf{x} = \mathbf{b}$ where A is symmetric and using a symmetric ℓ-level multigrid V-cycle with a damped Jacobi smoother, there exists a contraction factor $\rho_\ell < 1$ such that*

$$\|\mathbf{x} - \mathbf{x}_{j+1}\|_A \leq \rho_\ell \|\mathbf{x} - \mathbf{x}_j\|_A, \quad \rho_\ell \leq \rho_\infty := C/(C + s), \tag{2.46}$$

where C is a constant independent of ℓ and h and s is the number of smoothing steps.

This theorem shows that for the typical case of applying multigrid to an elliptic operator on a **Q1** or **P1** grid as described in Section 2.1.1, we will have convergence in the A-norm at a rate independent of h. By thinking of multigrid as a preconditioner with this property, we can extend this result to provide bounds on the eigenvalues of a preconditioned system.

Theorem 2.4 [25, Lemma 4.2]. *If the error of a simple iteration of the form* (2.35) *for the solution of the linear system $A\mathbf{x} = \mathbf{b}$ satisfies*

$$\|\mathbf{x} - \mathbf{x}_{j+1}\|_A \leq \rho \|\mathbf{x} - \mathbf{x}_j\|_A \tag{2.47}$$

with the contraction factor satisfying $\rho < 1$, then

$$1 - \rho \leq \frac{\langle A\mathbf{v}, \mathbf{v}\rangle}{\langle P\mathbf{v}, \mathbf{v}\rangle} \leq 1 + \rho. \tag{2.48}$$

Proof. Since $\mathbf{e}_{j+1} = (I - P^{-1}A)\mathbf{e}_j$ we can write (2.47) as

$$\langle A(I - P^{-1}A)\mathbf{e}_j, (I - P^{-1}A)\mathbf{e}_j\rangle \leq \rho^2\langle A\mathbf{e}_j, \mathbf{e}_j\rangle.$$

Letting $\mathbf{v} = A^{1/2}\mathbf{e}_j$ we have

$$\langle (I - A^{1/2}P^{-1}A^{1/2})\mathbf{v}, (I - A^{1/2}P^{-1}A^{1/2})\mathbf{v}\rangle \leq \rho^2\langle \mathbf{v}, \mathbf{v}\rangle.$$

Taking square roots and rearranging we have

$$-\rho\langle \mathbf{v}, \mathbf{v}\rangle \leq \langle (I - A^{1/2}P^{-1}A^{1/2})\mathbf{v}, \mathbf{v}\rangle \leq \rho\langle \mathbf{v}, \mathbf{v}\rangle$$
$$1 - \rho \leq \frac{\langle (A^{1/2}P^{-1}A^{1/2})\mathbf{v}, \mathbf{v}\rangle}{\langle \mathbf{v}, \mathbf{v}\rangle} \leq 1 + \rho,$$

which gives

$$1 - \rho \leq \frac{\langle A\mathbf{v}, \mathbf{v}\rangle}{\langle P\mathbf{v}, \mathbf{v}\rangle} \leq 1 + \rho$$

for all $\mathbf{v}$. □

Thus, we can see that by using multigrid as a preconditioner in this fashion we will obtain a preconditioned system with eigenvalues bounded about 1. This property will be particularly useful when combined with the Krylov subspace methods discussed later in this chapter.

2.2.2.1 Other Multigrid Methods

The scheme outlined above is an implementation of a geometric multigrid method. There are many other algorithms which use the same basic premise in order to solve a variety of other problems which require additional considerations.

Ramage multigrid: One example of when method alterations may be required, is for the solution of more complex differential operators than simple diffusion. For example a method developed by Ramage in [93] will be used throughout this thesis as a multigrid preconditioner for the convection-diffusion equation. The key aspects of this algorithm are:

- **Coarse grid operator:** Typically, a coarse grid operator is constructed through the use of a scaled Galerkin coarse grid operator so that the coarse grid operator is given by $\overline{A} = RAQ$. In this algorithm the operators are constructed explicitly on each grid level. For solving the convection-diffusion equation as described in Section 2.1.3, this will require constructing the matrices $\widetilde{K}$ and M on each grid level. Consequently, these matrices must also be stored for each grid level throughout the algorithm.

- **Smoothing:** A line Gauss-Seidel splitting will be used to construct the smoother. This involves taking, as a splitting matrix, the block lower triangular part of the matrix $\widetilde{K}$ ordered in different ways. Details of line Gauss-Seidel methods can be found for example in [119]. In order to take account of all possible directions of the wind component, we do this in all possible ways. Thus for two-dimensional problems this equates to two horizontal sweeps of the domain (left → right and right → left) and two vertical (top → bottom and bottom → top). For three dimensional problems this would correspond to 6 smoothing steps.

This method has been used successfully for a variety of problems[25, 86] and will be used throughout this thesis.

Algebraic Multigrid: We will also frequently use *Algebraic Multigrid* (AMG) methods throughout this thesis. These methods are designed to be applied to more general matrices than ones with an explicit grid structure and therefore can be applied as a 'black-box' solver. Instead of constructing operators for each level based on the grid structure, the operators are based entirely on the algebraic structure of the matrix. Since the concept of a mesh is no longer present, we instead can think of nodes i and j as being 'neighbouring' if A_{ij} is non-zero. A smaller set of nodes, analogous to a coarse grid, can then be determined by looking at strongly influencing nodes.

AMG methods are typically based on the methods described by Ruge and Stüben [97] and a more thorough overview can be found in [15, Chapter 8] for example. We will utilize the code `HSL_MI20` within the Harwell Subroutine Library (HSL) as an implementation of AMG [12].

For the preconditioners developed in Chapter 4, we will require approximation to a complex matrix. For this problem, we will use the *Aggregation Based Algebraic Multigrid* (AGMG) method developed by Notay which includes complex matrix capability [74, 77, 78, 79].

Space-Time Multigrid: While we will not explicitly use such methods in this thesis, for time-dependent problems there also exist *space-time multigrid* methods. The basic concept of these methods is to coarsen, not only in the spatial domain, but also in the temporal domain. This is an active area of research particularly due to the parallel-in-time capabilities of such methods. Some key references include [29, 30, 37, 53, 75].

The *parareal algorithm* [65] is a parallel in time method which is of broad interest and is widely researched. As the name suggests, the algorithm aims to solve solve evolutionary problems in real time using parallelization. The method can be thought of as a two-level space-time multigrid method or equally as a multiple shooting method as shown in [30, 38]. The idea of parareal is to construct an initial approximation over the time interval using a coarse propagator. A fine propagator can then approximate the solution on each slice of the time domain in parallel by using the coarse level approximations as the initial conditions. The coarse level approximation is then improved and the algorithm proceeds iteratively.

These methods typically work well on dissipative systems while problems can occur, for example, in wave dominated problem. The effectiveness of the method tends to rely on the choice of coarse solver which needs to both be accurate enough for convergence of the method while also being computationally inexpensive.

The methods developed in this thesis will use preconditioners which are applied using standard spatial multigrid and implemented with a standard Krylov subspace method. Thus, they have the ability to also be used in serial and for some parameters of the optimal control problems considered, they achieve better performance that other existing methods even when implemented in serial.

2.2.3 Non-Symmetric Krylov Subspace Methods

The previous iterative methods discussed, namely Chebyshev semi-iteration and multigrid, are used as preconditioners in our setting rather than solvers. In the next two sections, we discuss the *Krylov subspace methods* which are used as the outer solver for the systems we will consider.

In this thesis, we focus on time-dependent problems which, when discretized and constructed as an all-at-once system, are inherently non-self-adjoint. While effective methods for non-symmetric problems exist, they typically do not have the rigorous convergence bounds that can be provided for their symmetric counterparts. In this section, we examine iterative methods for non-symmetric problems and discuss what, if anything, we can prove regarding their convergence and termination.

In order to introduce these methods, we first need to discuss some preliminaries on Krylov methods in general. As already mentioned, iterative methods are often built on the idea that matrix-vector multiplication is cheap for large sparse matrices. With this in mind, we define the *Krylov subspace* of degree j to be

$$\mathcal{K}_j(A, \mathbf{r}_0) := \text{span}\left\{\mathbf{r}_0, A\mathbf{r}_0, A^2\mathbf{r}_0, \ldots, A^{j-1}\mathbf{r}_0\right\}. \tag{2.49}$$

A Krylov subspace method gives an approximation, $\mathbf{x}_j$, to (2.30) at the j-th iteration of the form,

$$\mathbf{x}_j \in \mathbf{x}_0 + \mathcal{K}_j(A, \mathbf{r}_0), \quad j = 1, 2, \ldots, \tag{2.50}$$

where $\mathbf{x}_0$ is an initial guess and $\mathbf{r}_0$ is the initial residual vector. We note that this is a commonly used abuse of notation and should be thought of as meaning $\mathbf{x}_j - \mathbf{x}_0 \in \mathcal{K}_j(A, \mathbf{r}_0)$. Due to the structure of the Krylov subspace, we note that we can also write the approximation as

$$\mathbf{x}_j = \mathbf{x}_0 + p_{j-1}(A)\mathbf{r}_0, \tag{2.51}$$

where p_{j-1} is a polynomial of degree $j-1$. This structure forms the basis of all the Krylov subspace methods which we will discuss in the following sections.

2.2.3.1 GMRES

One of the most widely used iterative methods for non-symmetric problems is the *Generalized Minimum Residual method* (GMRES) developed by Saad and Schultz [100]. As the name suggests, this method aims to minimize the 2-norm of the residual at each step.

GMRES is based on the *Arnoldi method*, an orthogonal projection method for general non-symmetric matrices. The method uses a modified Gram-Schmidt process to construct an orthogonal basis

$$\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j\}$$

for $\mathcal{K}_j$. We can write this method in matrix form with the use of *upper Hessenberg matrices* (i.e. matrices of zeros except for the subdiagonal and upper triangular part of the matrix). Letting $V_j = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j] \in \mathbb{R}^{n\times j}$, and $H_i \in \mathbb{R}^{i\times i}$ be an upper Hessenberg matrix and $\overline{H}_i = \begin{bmatrix} H_i \\ h_{i+1,i}\mathbf{e}_i^T \end{bmatrix} \in \mathbb{R}^{(i+1)\times i}$ we have

$$AV_i = V_i H_i + h_{i+1,i}[\mathbf{0}, \ldots, \mathbf{0}, \mathbf{v}_{i+1}] = V_{i+1}\overline{H}_i, \quad i = 1, 2, \ldots, j-1, \tag{2.52}$$

where $\mathbf{e}_i$ is the i-th column of the $i \times i$ identity matrix. Choosing $\mathbf{v}_1 = \frac{\mathbf{r}_0}{\|\mathbf{r}_0\|}$, the j-th iterate $\mathbf{x}_j$, corresponds to

$$\mathbf{x}_j = \mathbf{x}_0 + V_j\mathbf{y}_j, \tag{2.53}$$

for some $\mathbf{y}_j \in \mathbb{R}^j$ [25, Section 7.1.1].

We stated that GMRES specifically aims to minimize the 2-norm of the residual and this corresponds to

$$\min_{\mathcal{K}_j(A,\mathbf{r}_0)} \|\mathbf{r}_j\|_2 = \min_{\mathbf{y}_j} \|\|\mathbf{r}_0\|_2\mathbf{e}_1 - \overline{H}_j\mathbf{y}_j\|_2. \tag{2.54}$$

The resulting least squares problem can be solved using a QR factorization which requires only one Givens rotation at each iteration. We note that in polynomial

notation, (2.54) is equivalent to

$$\min_{\mathcal{K}_j(A,\mathbf{r}_0)} \|\mathbf{r}_j\|_2 = \min_{p_j \in \Pi_j, p_j(0)=1} \|p_j(A)\mathbf{r}_0\|_2, \tag{2.55}$$

where Π_j is the set of all polynomials of degree at most j. Preconditioning can be incorporated into the scheme described [98] and for a more detailed derivation of the algorithm we refer the interested reader to [25, 99]. The pseudocode of the algorithm as stated in Algorithm 3 is the right preconditioned version of the method.

Algorithm 3 GMRES algorithms to solve $A\mathbf{x} = \mathbf{b}$, with (right) preconditioner P

```
Choose x_0
Compute r_0 = b − Ax_0, β = ‖r_0‖_2, v_1 = r_0/β
for j = 1, 2, ... do
    Solve P z_j = v_j
    Compute w_{j+1}^{(1)} = A z_j
    for i = 1, 2, ..., j do
        h_{i,j} = ⟨w_{j+1}^{(i)}, v_i⟩
        w_{j+1}^{(i+1)} = w_{j+1}^{(i)} − h_{i,j} v_i
    end for
    h_{j+1,j} = ‖w_{j+1}^{(j+1)}‖_2
    v_{j+1} = w / h_{j+1,j}
    Find y_j such that y_j minimizes ‖β e_1 − H̄_j y_j‖_2
    <test for convergence>
end for
Solve P z_j = V_j y_j
x_j = x_0 + z_j
```

Having described the basics of GMRES implementation, we would like to know what, if anything, we can say about its convergence. As it turns out, convergence properties in general remain elusive. Let us first restrict ourselves to determining what we can say regarding termination, i.e. when will the method find the exact solution?

Theorem 2.5 [58, Theorem 3.1.2]. *Let $A \in \mathbb{R}^{n\times n}$ be non-singular. Then the* GMRES *algorithm will terminate within n iterations with exact arithmetic.*

Proof. The *characteristic polynomial* of A is given by $c(z) = \Pi_{j=1}^{n}(\lambda_j - z)$ where λ_j are the eigenvalues of A for $j = 1, 2, \ldots, n$. The Cayley-Hamilton theorem tells us that

any matrix satisfies its own characteristic polynomial so $c(A) = 0$ and additionally we can see that $c(0) = \det(A) \neq 0$ for non-singular matrices. From (2.55) we have that

$$\frac{\|\mathbf{r}_j\|_2}{\|\mathbf{r}_0\|_2} \leq \min_{p_j \in \Pi_j, p_j(0)=1} \|p_j(A)\|_2. \tag{2.56}$$

If we let $p_n(z) = \frac{c(z)}{c(0)}$ then $p_n \in \Pi_j, p_j(0) = 1$ and $p_n(A) = 0$ and we have constructed a polynomial which satisfies the requirements of the method. Therefore, at iteration n we must have that $\|\mathbf{r}_n\|_2 = 0$ and $\mathbf{x}_n = \mathbf{x}$. □

We now have a guarantee that GMRES will converge within a number of iterations equal to the dimension of the problem, however, we hope that we will be able to converge much sooner than that. We can improve this termination bound if we have information regarding the *minimal polynomial* of the matrix.

Definition 2.1. *The* minimal polynomial *of a matrix A over a field $\mathbf{F}$ is the monic polynomial $m(\cdot)$ of minimal degree such that $m(A) = 0$. The minimal polynomial will always divide the characteristic polynomial.*

Theorem 2.6. *If the minimal polynomial of A has degree k then the* GMRES *algorithm will terminate within k iterations.*

Proof. Similar to the proof of Theorem 2.5, let $p_k(z) = \frac{m(z)}{m(0)}$ so $p_k \in \Pi_k$ and $p_k(0) = 1$. Since, $m(A) = 0$, we have $p_k(A) = 0$ and due to (2.55) we have that $\|\mathbf{r}_k\|_2 = 0$ and $\mathbf{x}_k = \mathbf{x}$. □

We note that for a non-diagonalizable matrix, the minimal polynomial relates to the Jordan blocks of the matrix by the relation

$$m(z) = \Pi_{i=1}^{q} (z - \lambda_i)^{m_i}$$

where $\lambda_i, i = 1, 2, \ldots, q$ are equal to the q distinct eigenvalues of A and m_i is the size of the largest Jordan block with eigenvalue λ_i. We also have that $k = \sum_{i=1}^{q} m_i$.

An analogous termination bound for diagonalizable matrices is as follows.

Theorem 2.7. *If A is diagonalizable with q distinct eigenvalues, then* GMRES *with terminate in at most q iterations.*

Proof. This is a direct consequence of Theorem 2.6 since diagonalizability implies the minimal polynomial will be given by $m(z) = \Pi_{i=1}^{q}(z - \lambda_i)$ with degree q. □

Thus with knowledge of the multiplicity of the eigenvalues of A, we can determine termination of the GMRES algorithm. In general, we do not run the algorithm until we achieve the exact solution, but stop at an iteration when we have converged to a solution within some tolerance of the actual solution. So what can we say about when we will achieve convergence? Bounds for this can be determined for the case when A is diagonalizable.

Theorem 2.8 [25, Theorem 7.1]. *If A is diagonalizable, that is $A = V\Lambda V^{-1}$ where Λ is the diagonal matrix of eigenvalues of A, and V is the matrix whose columns are the eigenvectors, then*

$$\frac{\|\mathbf{r}_j\|_2}{\|\mathbf{r}_0\|_2} \leq \kappa(V) \min_{p_j \in \Pi_j, p_j(0)=1} \max_{\lambda_i} |p_j(\lambda_i)|, \tag{2.57}$$

where $\kappa(V) = \|V\|_2\|V^{-1}\|_2$ is the condition number of V.

Proof. From (2.56) we have that

$$\begin{aligned}\frac{\|\mathbf{r}_j\|_2}{\|\mathbf{r}_0\|_2} &\leq \min_{p_j \in \Pi_j, p_j(0)=1} \|p_j(A)\|_2 = \min_{p_j \in \Pi_j, p_j(0)=1} \|V p_j(\Lambda) V^{-1}\|_2 \\ &\leq \min_{p_j \in \Pi_j, p_j(0)=1} \|V\|_2\|V^{-1}\|_2\|p_j(\Lambda)\|_2 = \min_{p_j \in \Pi_j, p_j(0)=1} \kappa(V) \max_{\lambda_i} |p_j(\lambda_i)|\end{aligned}$$

which is the result. □

Although Theorem 2.8 does provide convergence bounds for diagonalizable matrices, the term $\kappa(V)$ is generally hard to bound and therefore this result is not often practically useful.

Ideally, we would like to be able to make some remarks concerning the convergence of GMRES based solely on the eigenvalues of the preconditioned matrix. In fact, it turns out that not only are we not able to do this, but it is shown by Greenbaum, Ptak, and Strakoš in [44] (and extended in [23]) that given any set of n eigenvalues and any non-increasing convergence curve terminating at or before the n-th iteration, then for a $\mathbf{b}$ there exists a matrix $A \in \mathbb{R}^{n\times n}$ with those eigenvalues and an initial guess

$\mathbf{x}_0$ such that GMRES will give that convergence curve. More negative results than this exist (see for example [22]). Some convergence estimates can be obtained through field of values analysis for certain PDE problems (see for example [67]), analysis of other sets such as ϵ-pseudospectral sets [111], or through transforming the problem to a symmetric one for which symmetric solvers can be used [91]. In general, it remains the case that eigenvalues are used merely as heuristics for convergence as it is certainly often the case that widely spread eigenvalues result in slow convergence for non-symmetric solvers.

2.2.3.2 BiCGStab

While GMRES has the convenient property that it minimizes the 2-norm of the residual at each iteration, it inconveniently requires the construction of a Hessenberg matrix to store the Arnoldi vectors. The number of these vectors will grow in number at each iteration and thus for a large number of iterations, the storage requirements can make GMRES impractical. A possible way to deal with this problem is to use a method called *restarted* GMRES. This method has an upper bound on the number of Arnoldi vectors stored and when this is reached it restarts the algorithm from the current iterate. Convergence of this algorithm can be considerably slower and much of the theory developed for our preconditioners in future chapters would not apply to this algorithm and, therefore, it will not be considered in this thesis.

In order to avoid the potential storage issue of GMRES, we would ideally like a method which has a minimization property but that can be computed using short-term recurrences. A close compromise, is the *Bi-conjugate Gradients method* (BiCG). This method was designed to be analogous to the *Conjugate Gradients* (CG) method, but applicable to general matrices where CG is only applicable to symmetric positive definite matrices. Although this method is not used directly within our methods, we provide a brief introduction to CG here to enable the introduction of the BiCG method.

The Conjugate Gradient method is also a Krylov subspace method which produces iterates of the form

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j, \quad j = 1, 2, \ldots \tag{2.58}$$

where $\mathbf{p}_j$ are search directions which are A-conjugate meaning

$$\mathbf{p}_j^T A \mathbf{p}_i = 0 \text{ for } i \neq j. \tag{2.59}$$

The parameter α_j is chosen such that the A-norm of the error is minimized along the direction $\mathbf{p}_j$. This implies in the symmetric positive definite case that $\|\mathbf{x} - \mathbf{x}_j\|_A$ is minimal for $\mathbf{x}_j \in \mathbf{x}_0 + \mathcal{K}_j(A, \mathbf{r}_0)$.

While BiCG does not have this minimization property, the method produces residuals $\mathbf{r}_j$ and $\tilde{\mathbf{r}}_j$ and search directions $\mathbf{p}_j$ and $\tilde{\mathbf{p}}_j$ such that bi-orthogonality holds:

$$\tilde{\mathbf{r}}_j^T \mathbf{r}_i = 0, \text{ if } i \neq j, \tag{2.60}$$

and bi-conjugacy also holds by

$$\tilde{\mathbf{p}}_j^T A \mathbf{p}_i = 0, \text{ if } i \neq j. \tag{2.61}$$

We note that if A is a symmetric positive definite matrix, BiCG will reduce to exactly the Conjugate Gradient method with $\mathbf{r}_0 = \tilde{\mathbf{r}}_0$. However, in its original form, BiCG has several drawbacks including being prone to breakdown as well as requiring computation with A^T which is not always readily available. Therefore, we will use the stabilized version BiCGStab developed by van der Vorst [114] which has improved behaviour and does not require A^T. The pseudocode for the (left, right or symmetrically) preconditioned method is presented in Algorithm 4. We note that the preconditioned method is equivalent to applying the standard method to the preconditioned system in exact arithmetic.

As stated previously, Gmres can require prohibitive amounts of storage whereas BiCGStab does not have this issue. However, BiCGStab requires two applications of A per iteration while Gmres only required one. Another significant drawback is that since Gmres has a minimization property, we are able to make some statements about termination and, in certain cases, convergence of the iteration while for BiCGStab we have none of these properties. Thus, we will present iteration counts for this method to demonstrate convergence without the storage restrictions

Algorithm 4 BiCGStab algorithm to solve $A\mathbf{x} = \mathbf{b}$, with preconditioner $P = P_1P_2$

Compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$
Choose $\tilde{\mathbf{r}}_0$ such that $\langle \tilde{\mathbf{r}_0}, \mathbf{r}_0 \rangle \neq 0$ e.g. $\tilde{\mathbf{r}}_0 = \mathbf{r}_0$
Set $\rho_0 = \alpha = \omega_0 = 1, \mathbf{v}_0 = \mathbf{p}_0 = \mathbf{0}$
for $j = 1, 2, \ldots$ **do**
  $\rho_j = \langle \tilde{\mathbf{r}}_0, \mathbf{r}_{j-1} \rangle$
  $\beta = (\rho_j/\rho_{j-1})(\alpha/\omega_{j-1})$
  $\mathbf{p}_j = \mathbf{r}_{j-1} + \beta(\mathbf{p}_{j-1} - \omega_{j-1}\mathbf{v}_{j-1})$
  Solve $P\mathbf{y} = \mathbf{p}_j$
  Compute $\mathbf{v}_j = A\mathbf{y}$
  $\alpha = \rho_j / \langle \tilde{\mathbf{r}}_0, \mathbf{v}_j \rangle$
  $\mathbf{h} = \mathbf{x}_{j-1} + \alpha\mathbf{y}$
  <**test for convergence**> if accurate $\mathbf{x}_j = \mathbf{h}$ and quit
  $\mathbf{s} = \mathbf{r}_{j-1} - \alpha\mathbf{v}_j$
  Solve $P\mathbf{z} = \mathbf{s}$
  Compute $\mathbf{t} = A\mathbf{z}$
  $\omega_j = \langle P_1^{-1}\mathbf{t}, P_1^{-1}\mathbf{s} \rangle / \langle P_1^{-1}\mathbf{t}, P_1^{-1}\mathbf{t} \rangle$
  $\mathbf{x}_j = \mathbf{h} + \omega_j\mathbf{z}$
  <**test for convergence**> if accurate then quit
  $\mathbf{r}_j = \mathbf{s} - \omega_j\mathbf{t}$
end for

of Gmres, however none of the convergence theory developed for Gmres will apply.

2.2.4 Symmetric Krylov Subspace Methods

We briefly mentioned the Conjugate Gradients (CG) method in the previous section which solves systems that are symmetric but are also required to be positive definite. For symmetric indefinite systems, as will be the case for the saddle point systems arising from PDE-constrained optimization problems, we will use the Minres algorithm.

2.2.4.1 MINRES

The *Minimal Residual* (Minres) algorithm can be thought of a simplification to Gmres when the matrix A is symmetric, although it was introduced 11 years prior to Gmres by Paige and Saunders in 1975 [83]. As for CG, the Lanczos process [62, 63] is used for symmetric matrices to generate the Krylov subspace but, as is the case

with GMRES, we are searching for vectors which minimize the residual by solving

$$\min_{\mathbf{x}_j \in \mathbf{x}_0 + \mathcal{K}_j(A, \mathbf{r}_0)} \|\mathbf{b} - A\mathbf{x}_j\|_2. \tag{2.62}$$

The advantage of having symmetric matrices for this minimization problem is that the matrix $\overline{H}_i$ is tridiagonal. This enables the use of short term recurrence relations. The pseudocode for the algorithm is presented for reference in Algorithm 5. For thorough derivations of this method we refer the reader to [25, 115].

Algorithm 5 MINRES algorithm to solve $A\mathbf{x} = \mathbf{b}$ where A is symmetric and using symmetric positive definite preconditioner P.

Set $\mathbf{v}_0 = \mathbf{w}_0 = \mathbf{w}_1 = \mathbf{0}$
Choose $\mathbf{x}_0$
Compute $\mathbf{v}_1 = \mathbf{b} - A\mathbf{x}_0$
Solve $P\mathbf{z}_1 = \mathbf{v}_1$
Set $\gamma = \sqrt{\langle \mathbf{z}_1, \mathbf{v}_1 \rangle}; \nu = \gamma_1$
Set $s_0 = s_1 = 0; c_0 = c_1 = 1$
for $j = 1, 2, \ldots$ **do**
  $\mathbf{x}_j = \mathbf{z}/\gamma_j$
  $\delta_j = \langle A\mathbf{z}_j, \mathbf{z}_j \rangle$
  $\mathbf{v}_{j+1} = A\mathbf{z}_j - (\delta_j/\gamma_j)\mathbf{v}_j - (\gamma/\gamma_{j-1})\mathbf{v}_{j-1}$
  Solve $P\mathbf{z}_{j+1} = \mathbf{v}_{j+1}$
  $\alpha_0 = c_j\delta_j - c_{j-1}s_j\gamma_j$
  $\alpha_1 = \sqrt{\alpha_0^2 + \gamma_{j+1}^2}$
  $\alpha_2 = s_j\delta_j + c_{j-1}c_j\gamma_j$
  $\alpha_3 = s_{j-1}\gamma_j$
  $c_{j+1} = \alpha_0/\alpha_1; s_{j+1} = \gamma_{j+1}/\alpha_1$
  $\mathbf{w}_{j+1} = (\mathbf{z}_j - \alpha_3\mathbf{w}_{j-1} - \alpha_2\mathbf{w}_j)/\alpha_1$
  $\mathbf{x}_j = \mathbf{x}_{j-1} + c_{j+1}\nu\mathbf{w}_{j+1}$
  $\nu = -s_{j+1}\nu$
  <**test for convergence**>
end for

An important property of MINRES is that it can be applied to any symmetric matrix which is not the case for CG which requires symmetric positive definite systems. This will be an important property for the solution of saddle-point systems which will be generated in the optimal control problems considered in this thesis.

Unlike for non-symmetric systems we can relate convergence of MINRES directly

to the eigenvalues of the system by the relation,

$$\frac{\|\mathbf{r}_j\|_2}{\|\mathbf{r}_0\|_2} \leq \min_{p_j \in \Pi_j, p_j(0)=1} \max_{\lambda_i} |p_j(\lambda_i)| \leq \min_{p_j \in \Pi_j, p_j(0)=1} \max_{z \in [a,b]} |p_j(z)| \tag{2.63}$$

where λ_i are the eigenvalues of A which are contained in the interval $[a, b]$. For the preconditioned algorithm, we obtain the relation

$$\frac{\|\mathbf{r}_j\|_{P^{-1}}}{\|\mathbf{r}_0\|_{P^{-1}}} \leq \min_{p_j \in \Pi_j, p_j(0)=1} \max_{\lambda_i} |p_j(\lambda_i)|, \tag{2.64}$$

where λ_i are the eigenvalues of $P^{-1}A$. We note that for this statement to hold we require that P^{-1} must be able to define a norm and thereby be symmetric positive definite. Thus, we will be restricted to using symmetric positive definite preconditioners for MINRES while the matrix A may be indefinite.

We can see from either statement (2.63) or (2.64) that we can obtain robust error bounds if we know the eigenvalues of A or $P^{-1}A$ respectively. In general, the bounds imply that we would like tightly clustered eigenvalues neither too close nor too far from the origin in order to achieve fast convergence. We can, therefore, have this in mind as we design our preconditioners. This differs from the case of GMRES where such error bounds do not exist.

2.2.5 Normal Equations

We have seen in the previous two sections that there are typically different approaches used for the iterative solution of non-symmetric and symmetric matrices. Moreover, Krylov methods for symmetric matrices such as MINRES, have robust convergence estimates determined solely by eigenvalues, which is not the case for the non-symmetric counterpart GMRES.

An obvious solution to the lack of convergence estimates for non-symmetric matrices is to convert the system to a symmetric one and solve this new system using a symmetric solver.

For the matrix system

$$A\mathbf{x} = \mathbf{b},$$

where A is non-symmetric, we can form the *normal equations* which are given by

$$A^T A\mathbf{x} = A^T\mathbf{b}. \tag{2.65}$$

Solving the system (2.65) can equally be thought of as solving the least squares problem

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|_2. \tag{2.66}$$

There are many methods which incorporate these ideas with existing methods for symmetric matrics such as the Conjugate Gradient (CG) method. These include Conjugate Gradient Normal Equation Error (CGNE), Conjugate Gradients Normal Equation Residual (CGNR), LSQR or LSMR. We will examine the LSQR method in more detail here and discuss the advantages and challenges of successfully using this method. For more detailed discussion of this method and other normal equation based methods, we refer the reader to [40, 115].

2.2.5.1 LSQR

If we were to apply the CG method to the least squares problem in (2.66), we would obtain the so-call CGLS method, however, as pointed out by Paige and Saunders in [84], this method can be unstable particularly for ill-conditioned matrices. For more favourable results the LSQR method, also described in [84], is required. Although mathematically equivalent to CGLS, the LSQR algorithm solves the least squares problem through a standard QR factorization which provides additional numerical stability. The LSQR algorithm is presented for reference in Algorithm 6.

As the LSQR method is mathematically equivalent to applying the Conjugate Gradient method to the normal equations, we need to examine the convergence properties of CG in order to determine convergence of LSQR.

Theorem 2.9 [25, Theorem 2.4]. *After j steps of the Conjugate Gradient method applied to the normal equation in* (2.65)*, the error $\mathbf{e}_{j+1} = \mathbf{x} - \mathbf{x}_{j+1}$ satisfies the bound*

$$\|\mathbf{e}_{j+1}\|_{A^TA} = \|\mathbf{r}_{j+1}\|_2 \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^j \|\mathbf{e}_j\|_{A^TA}, \tag{2.67}$$

Algorithm 6 LSQR algorithm to solve $A\mathbf{x} = \mathbf{b}$.

Set $x_0 = 0, \beta_1 = \|b\|_2, u_1 = \|b\|_2/\beta_1$
Set $v = A^T u_1, \alpha_1 = \|v\|_2, w_1 = v_1 = v/\alpha_1$
Set $\widetilde{\phi_1} = \beta_1, \widetilde{\rho_1} = \alpha_1$
for i = 1,2,... **do**
  $u = Av_i - \alpha_i u_i, \beta_{i+1} = \|u\|_2, u_{i+1} = u/\beta_{i+1}$
  $v = A^T u_{i+1} - \beta_{i+1} v_i, \alpha_{i+1} = \|v\|_2, v_{i+1} = v/\alpha_{i+1}$
  $\rho_i = \sqrt{\widetilde{\rho_i}^2 + \beta_{i+1}^2}$
  $c_i = \widetilde{\rho_i}/\rho_i$
  $s_i = \beta_{i+1}/\rho_i$
  $\theta_{i+1} = s_i \alpha_{i+1}$
  $\widetilde{\rho}_{i+1} = -c_i \alpha_{i+1}$
  $\phi_i = c_i \widetilde{\rho_i}$
  $\widetilde{\phi}_{i+1} = s_i \widetilde{\phi_i}$
  $x_i = x_{i-1} + (\phi_i/\rho_i) w_i$
  $w_{i+1} = v_{i+1} - (\theta_{i+1}/\rho_i) w_i$
  <**test for convergence**>
end for

where κ is the condition number of $A^T A$.

We can see from Theorem 2.9, that convergence of LSQR will depend not on the conditioning of A, but of the conditioning of $A^T A$. Now, the condition number of a matrix A is given by

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2$$

and thus the condition number of $A^T A$ will be given by

$$\kappa(A^T A) = \|A^T A\|_2 \|(A^T A)^{-1}\|_2.$$

However $\|A^T A\|_2 = \sigma_{max}(A^T A) = \lambda_{max}(A^T A)$ since $A^T A$ is clearly symmetric. Furthermore, it is also the case that $\lambda_{max}(A^T A) = \sigma_{max}^2(A) = \|A\|_2^2$. Thus by the relation between eigenvalues and singular values we have that

$$\kappa(A^T A) = \|A^T A\|_2 \|(A^T A)^{-1}\|_2 = \|A\|_2^2 \|A^{-1}\|_2^2 = (\kappa(A))^2.$$

Therefore, if our original matrix A is poorly conditioned, the matrix $A^T A$ will be a power of 2 worse! This fact often means that the LSQR method is not often used

as an iterative solver for poorly conditioned problems unless A is rectangular.

2.2.5.2 Preconditioning the Normal Equations

If the conditioning of the normal equations is preventing the LSQR methods from being effective, then surely preconditioning may be able to solve our problems? In fact, it was shown in Braess and Piesker [13] in 1986 that this is not as straightforward as it might appear.

Consider the matrix

$$A = \begin{bmatrix} 1 & \alpha \\ \alpha & 2\alpha^2 \end{bmatrix},$$

where $\alpha > 1$. A simple choice for a preconditioner of this matrix might be to simply take the diagonal terms. Thus we define our preconditioner P to be

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 2\alpha^2 \end{bmatrix}.$$

If we calculate the eigenvalues of the preconditioned system $P^{-1}A$ we find that they are equal to

$$\frac{2 \pm \sqrt{2}}{2}$$

and therefore are bounded and completely independent of the parameter α. Thus, one might conclude that P is a successful preconditioner for A. Now what about a preconditioner for A^TA? The obvious choice is to choose the matrix P^TP. However if we calculate the eigenvalues of $(P^TP)^{-1}(A^TA)$ we find that they are equal to

$$\frac{4\alpha^4 + 8\alpha^2 + 1 \pm (2\alpha^2 + 1)\sqrt{4\alpha^4 + 12\alpha^2 + 1}}{8\alpha^2}$$

and therefore completely unbounded in the parameter α as $\alpha \to \infty$. Thus, we find that P^TP can be arbitrarily bad as a preconditioner for A^TA even if P is a good preconditioner for A.

Another approach might be to precondition the matrix first and then apply LSQR

to the normal equations of the preconditioned system. However, we can see that the eigenvalues of $(P^{-1}A)^T(P^{-1}A)$ will be identical to those of $(PP^T)^{-1}(AA^T)$ since

$$(P^{-1}A)^T(P^{-1}A) = A^T P^{-T} P^{-1} A$$

and by a similarity transform we have $(PP^T)^{-1}(AA^T)$.

Thus we can see that preconditioning of the normal equations is not as easy as we might have hoped since it is the singular values which determine convergence instead of eigenvalues. It perhaps explains why methods such as LSQR are not more readily used as preconditioning for these methods is simply not well understood. This is due to the fact that it is typically easier to develop preconditioners which reduce the spectral radius of the iteration matrix, but a preconditioner which reduces the spread of the singular values is generally harder to develop. We can make some statements, however, regarding sufficient conditions for when a preconditioner will be effective for the normal equations.

Theorem 2.10 [13]. *For the linear system* $A\mathbf{x} = \mathbf{b}$ *, if all error vectors* $\mathbf{e}_j = \mathbf{x} - \mathbf{x}_j$ *for the simple iteration*

$$\mathbf{x}_{j+1} = (I - P^{-1}A)\mathbf{x}_j + P^{-1}\mathbf{b}$$

satisfy

$$\|\mathbf{e}_{j+1}\|_2 \leq \rho \|\mathbf{e}_j\|_2$$

for some contraction factor $\rho < 1$*, then*

$$(1-\rho)^2 \leq \frac{\mathbf{v}^T AA^T \mathbf{v}}{\mathbf{v}^T PP^T \mathbf{v}} \leq (1+\rho)^2 \quad \textit{for all } \mathbf{v}.$$

Proof. The contraction condition implies that

$$\|(I - P^{-1}A)\mathbf{e}_j\|_2 \leq \rho \|\mathbf{e}_j\|_2.$$

From the properties of the Euclidean norm we have that the smallest possible value

of ρ is given by,

$$\begin{aligned}\rho &= \sup_{\mathbf{e}_j \neq 0} \frac{\|(I - P^{-1}A)\mathbf{e}_j\|_2}{\|\mathbf{e}_j\|_2} = \|I - P^{-1}A\|_2 \\ &= \|(I - P^{-1}A)^T\|_2 = \|I - A^T P^{-T}\|_2 \\ &= \sup_{\mathbf{e}_j \neq 0} \frac{\|(I - A^T P^{-T})\mathbf{e}_j\|_2}{\|\mathbf{e}_j\|_2}.\end{aligned}$$

We can now write the contraction property as

$$\|(I - A^T P^{-T})\mathbf{e}_j\|_2 \leq \rho \|\mathbf{e}_j\|_2$$

which because of the triangle inequality implies

$$(1 - \rho)\|\mathbf{e}_j\|_2 \leq \|A^T P^{-T}\mathbf{e}_j\|_2 \leq (1 + \rho)\|\mathbf{e}_j\|_2.$$

Dividing by $\|\mathbf{e}_j\|$ and squaring gives

$$(1 - \rho)^2 \leq \frac{\langle A^T P^{-T}\mathbf{e}_j, A^T P^{-T}\mathbf{e}_j\rangle}{\langle \mathbf{e}_j, \mathbf{e}_j\rangle} \leq (1 + \rho)^2.$$

Now setting $v = P^{-T}\mathbf{e}_j$ yields

$$(1 - \rho)^2 \leq \frac{\mathbf{v}^T A A^T \mathbf{v}}{\mathbf{v} P P^T \mathbf{v}} \leq (1 + \rho)^2$$

□

We note that in this proof the contraction factor is defined as $\rho = \|I - P^{-1}A\|_2$ and not $\rho(I - P^{-1}A)$. Thus we need the largest singular value of the iteration matrix to be less that one and not just the largest eigenvalue. This is an occasionally misunderstood aspect of this theorem. It was shown in Section 2.2.2 that multigrid can be seen as a simple iteration, so provided that this iteration is contractive in the 2-norm for the original system then it can be effectively used in the normal equation case also.

The concept of preconditioning normal equations also arises in the context of

preconditioning saddle-point problems. Saddle-point problems will arise in the PDE-constrained optimization problems considered in Chapters 5, 6 and 7. A common method for preconditioning these systems is to use preconditioners which contain the Schur complement of the system. It will be shown in these later chapters that this effectively requires a good preconditioner for systems of the form $A^T A$, and the theory detailed above again becomes relevant.

2.3 Solution of Toeplitz Systems

Toeplitz matrices are those whose diagonals have constant entries. Such matrices arise in a variety of areas across applied mathematics including statistics, signal and image processing, control theory and itegral equations and as such efficient solvers for these types of problems have been an active area of research [17]. Toeplitz systems also frequently arise in the study of partial and ordinary differential equations. The context in which we will be examining Toeplitz matrices is through time discretization. This process can be easily demonstrated for the following scalar, linear ordinary differential equation,

$$u_t = \alpha u + f, \quad u(0) = u_0, \tag{2.68}$$

on the time interval $t \in [0, T]$ with constant forcing through time. If we use the θ-method described in Section 2.1.1 with ℓ constant time-steps of size τ, then this corresponds to solving the system,

$$\frac{u^{k+1} - u^k}{\tau} = \theta \alpha u^{k+1} + (1 - \theta)\alpha u^k + f, \quad u^0 = u_0, \tag{2.69}$$

for $k = 0, 1, \ldots, \ell - 1$. If we write this sequence of linear equations in a single matrix, then solution to (2.69) is equivalent to solving,

$$A\mathbf{u} = \mathbf{b}, \tag{2.70}$$

where

$$A \coloneqq \begin{bmatrix} 1-\alpha\theta\tau & & & \\ -1-\alpha(1-\theta)\tau & 1-\alpha\theta\tau & & \\ & \ddots & \ddots & \\ & & -1-\alpha(1-\theta)\tau & 1-\alpha\theta\tau \end{bmatrix}, \tag{2.71}$$

where $\mathbf{b} = [\tau f + (1+\alpha(1-\theta)\tau)u_0, \tau f, \ldots, \tau f]^T$ and $\mathbf{u} = [u^1, u^2, \ldots, u^\ell]^T$. It is evident that A has constant entries on each of the diagonals and is ,therefore, Toeplitz and additionally is lower triangular. This will be the case for other implicit time-stepping schemes just as described for the all-at-once solution to partial differential equations described in Section 2.1.2, however, in the PDE case the all-at-once matrices are block Toeplitz. In Chapter 4 we will introduce block circulant preconditioners for these block Toeplitz systems based on the theory of circulant preconditioning for Toeplitz systems such as the matrix described in (2.71). We describe some of this relevant theory here and for more complete description of the solution of Toeplitz systems we recommend [17, 76].

2.3.1 Circulant Preconditioning

Let $T \in \mathbb{R}^{n\times n}$ be the non-singular Toeplitz matrix and $C \in \mathbb{R}^{n\times n}$ be the non-singular circulant preconditioner given by

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+2} & t_{-n+1} \\ t_1 & t_0 & t_{-1} & & t_{-n+2} \\ \vdots & t_1 & t_0 & \ddots & \vdots \\ t_{n-2} & & \ddots & \ddots & t_{-1} \\ t_{n-1} & t_{n-2} & \cdots & t_1 & t_0 \end{bmatrix}, \quad \text{and} \quad C = \begin{bmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{bmatrix}. \tag{2.72}$$

For Toeplitz systems, circulant matrices have been popular preconditioners, not least because they can be applied quickly using a *fast Fourier transform* (FFT). The

matrix C has the diagonalization

$$C = U\Lambda U^*, \tag{2.73}$$

where, if we denote the Fourier matrix $F = (f_{jk})$ to be $f_{jk} = e^{2(j-1)(k-1)\pi i/n}$, then we have that $U = F/\sqrt{n}$. Also $\Lambda = \mathrm{diag}(F\bar{c})$ where $\bar{c}$ denotes the first column of C. This relationship to the FFT means that the solution of a linear system with a circulant matrix can be performed in $\mathcal{O}(n \log n)$ operations [116].

The idea of preconditioning Toeplitz matrices with a circulant was first introduced independently by Strang in [110] and Olkin in [81]. They proposed the so-called *Strang circulant* which was constructed by taking the central band of T of width $n/2$ and wrapping the entries around to form a circulant. Many other circulant preconditioners have been proposed (see for example [17, 76]). One example is the optimal circulant [18], which minimizes the Frobenius norm distance to the given Toeplitz matrix over all possible circulants. However, although the optimal circulant can be computed via an explicit formula, it does not perform well for certain ill-conditioned matrices. A unifying approach to selecting the best possible circulant preconditioner was proposed in [82].

Theoretical convergence bounds for these types of preconditioners have generally been restricted to symmetric (Hermitian) positive definite Toeplitz matrices. For many existing preconditioners, including the Strang and optimal preconditioners, and for wide classes of Toeplitz matrices, the preconditioned system is given by

$$C^{-1}T = I + R + E,$$

where R has small rank and E small norm. However, for non-symmetric systems this is not sufficient to provide descriptive convergence estimates for standard non-symmetric solvers such as GMRES or BiCGStab. However [91] provides rigorous convergence bounds for nonsymmetric Toeplitz matrices. This is done by reordering

the rows or columns of T by pre- or post-multiplying by the Hankel matrix,

$$Y = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \iddots & & \\ 1 & & & \end{bmatrix}. \tag{2.74}$$

This results in a symmetric (Hankel) system for any Toeplitz matrix and symmetric iterative methods can be applied. We note that it is possible to use LSQR or LSMR [33] to obtain rigorous convergence bounds for non-symmetric Toeplitz matrices, but for scalar Toeplitz problems these methods are typically slower than using symmetrization and MINRES. This type of method will be extended to the time-dependent PDE case in Chapter 4.

CHAPTER 3

Block Diagonal Preconditioners

In Section 2.1, we introduced the concept of solving all time-steps of a time-dependent PDE simultaneously through the construction of an all-at-once system. This immense system consists of a block lower triangular matrix with dimensions $n\ell \times n\ell$, where n is the number of spatial degrees of freedom and ℓ is the number of time-steps. It is, however, never constructed explicitly; rather, all that is required is the spatial operators and a 'recipe' for applying the all-at-once operator.

Large block matrices arise in many areas of mathematics, often as an inherent structure within the problem formulation. For example, in optimal control problems which will be considered later in this thesis, the state, control, and adjoint variables are all required to be solved for resulting in a natural block structure to the linear system. Other block partitioning can be introduced in order to aid in solving the system, for example through the use of domain decomposition methods. Thus, block matrices are a well studied class of problems and many preconditioners have been developed for such systems for a variety of applications.

In this chapter, we investigate the use of a simple block diagonal (or block Jacobi) preconditioner for the all-at-once solution of time-dependent PDE problems[1]. The

[1] A condensed version of the concepts in this chapter can be found in [70].

use of this type of preconditioner is a classical approach, however, we believe that analysis in this context provides a new viewpoint and motivates other preconditioners for PDE-constrained optimization problems which will be discussed in Chapters 6 and 7. We note that this type of preconditioner was used as a smoother within the space-time multigrid approach proposed in [75]. However, as will be shown in this chapter, this preconditioner does not exhibit a significant reduction in the residual norm until the ℓ-th iteration. Thus, a different approach which reduces the error more quickly at the initial iterations may be more appropriate as a smoother.

This chapter is structured as follows. In Section 3.1, we outline the proposed preconditioner and an approximated version for this type of problem. In Section 3.2, we detail convergence analysis for the exact preconditioner and, where convergence analysis is not available, we examine the eigenvalues of the preconditioned system. There are several obvious modifications to this method which will be discussed in more detail in Section 3.3. Further analysis will also be provided for convergence of iterative solution to the normal equations in Section 3.4. Finally, numerical results for all methods will be provided in Section 3.5 and concluding remarks provided in Section 3.6.

3.1 Proposed Approach

The system we are aiming to solve is the all-at-once system for a linear PDE with implicit time-stepping as introduced in Section 2.1. We will not restrict ourselves to a particular time-stepping scheme and instead look at the system for a general k-step method. For example, θ-methods are a one-step method while the BDF2 system in (2.17) corresponds to $k = 2$. Other higher order methods could equally be considered. Thus, we consider solving the system $\mathcal{A}\mathbf{x} = \mathbf{b}$ where,

$$\mathcal{A} := \begin{bmatrix} A_0 & & & & & \\ A_1 & A_0 & & & & \\ \vdots & \ddots & \ddots & & & \\ A_k & & \ddots & \ddots & & \\ & \ddots & & A_1 & A_0 & \\ & & A_k & \cdots & A_1 & A_0 \end{bmatrix} \in \mathbb{R}^{n\ell \times n\ell}. \tag{3.1}$$

In order to solve such an immense system we will use iterative methods and focus on effective preconditioning to improve performance. The preconditioner that we study here is the block Jacobi preconditioner given by,

$$\mathcal{P} := \begin{bmatrix} A_0 & & & \\ & A_0 & & \\ & & \ddots & \\ & & & A_0 \end{bmatrix} \in \mathbb{R}^{n\ell \times n\ell}. \tag{3.2}$$

In order to invert $\mathcal{P}$, we need to invert a total of ℓ matrices of dimension $n \times n$; however, simply solving $\mathcal{A}$ by block forward substitution would also require a total of ℓ matrix inversions. These matrix inversions could be completed using a multigrid process such that if each time-step required a maximum of r V-cycles the total amount of work would be equivalent to $r\ell$ V-cycles. In practice, for the heat equation with constant time-step size τ equal to the grid size $h = 2^{-5}$, $r = 5$ V-cycles of an algebraic multigrid method were observed to be required at each step to obtain a solution with relative residual less than 10^{-6} using the solution to the previous time-step as the starting guess in each case.

Rather than complete the inversion of $\mathcal{P}$ exactly, we consider an approximate preconditioner $\widehat{\mathcal{P}}$ which uses a multigrid process to approximately complete each inversion.

We define,

$$\widehat{\mathcal{P}}^{-1} := \begin{bmatrix} A_{0(MG)} & & & \\ & A_{0(MG)} & & \\ & & \ddots & \\ & & & A_{0(MG)} \end{bmatrix}, \tag{3.3}$$

where $A_{0(MG)}$ denotes the application of one V-cycle of multigrid to the matrix A_0. The total work to apply this approximate preconditioner is now ℓ V-cycles.

Both the 'exact' preconditioner $\mathcal{P}$ defined in (3.2) and the approximate preconditioner $\widehat{\mathcal{P}}$ in (3.3) have block diagonal structure. This structure is inherently parallelizable as each inversion or V-cycle can be completed on ℓ separate processors. Thus for $\widehat{\mathcal{P}}$, the ℓ V-cycles could theoretically be completed in the same time as one V-cycle if access to ℓ processors was available.

We note an application of the preconditioner $\mathcal{P}$ corresponds to solving multiple systems with the same coefficient matrix but with multiple right-hand sides and methods have been developed which speed up this process such as that suggested in [66]. These methods could be included in an implementation of our approach but we have not considered such effects here.

We have demonstrated that this style of preconditioner can be efficiently applied, however, it remains to predict the convergence properties of an appropriate preconditioned iterative method. In order to do this, we will examine the convergence of the exact preconditioner $\mathcal{P}$ before looking at the approximated version.

3.2 Convergence Analysis

Due to the non-symmetric nature of the system, eigenvalues alone are not sufficient to determine convergence of a non-symmetric Krylov subspace method such as GMRES (as discussed in Section 2.2.3). However, in this particular case, we are able to exactly determine the maximum number of GMRES iterations required when $\mathcal{P}$ is used as the preconditioner.

Theorem 3.1. *For $\mathcal{A}$ and $\mathcal{P}$ as defined in* (3.1) *and* (3.2)*, the eigenvalues of the pre-*

conditioned system $\mathcal{T} = \mathcal{P}^{-1}\mathcal{A}$ *are all equal to 1. Furthermore, the minimal polynomial of* $\mathcal{T}$ *is,*

$$\rho(\mathcal{T}) = (\mathcal{T} - I_{n\ell})^m, \tag{3.4}$$

where $m \leq \ell$ *and* $I_{n\ell}$ *is the identity matrix of dimension* $n\ell \times n\ell$*. Therefore an appropriate Krylov subspace method will terminate in at most* ℓ *iterations.*

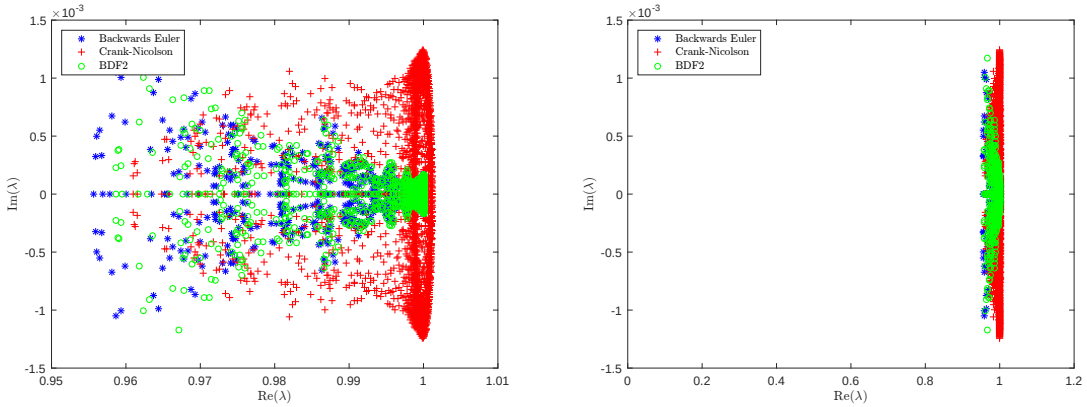
Proof. As $\mathcal{P}^{-1}$ is simply a block diagonal matrix with A_0 on the diagonal, the preconditioned system is therefore equal to,

$$\mathcal{T} = \begin{bmatrix} I_n & & & & & \\ J_1 & I_n & & & & \\ \vdots & \ddots & I_n & & & \\ J_k & & \ddots & \ddots & & \\ & \ddots & & J_1 & I_n & \\ & & J_k & \cdots & J_1 & I_n \end{bmatrix}, \tag{3.5}$$

where $J_i = A_0^{-1}A_i$. Thus, $\mathcal{T} - I_{n\ell}$ is a strictly block lower triangular matrix and simple computation shows that $(\mathcal{T} - I_{n\ell})^\ell = 0$. It is possible, however, for some $m \leq \ell$ that $(\mathcal{T} - I_{n\ell})^m = 0$ thus we only have that the minimal polynomial has a maximum degree of ℓ. It therefore follows that a non-symmetric Krylov method such as GMRES will converge in a maximum of ℓ iterations as shown in Theorem 2.6. □

We note that although a minimal polynomial of degree $m < \ell$ is possible, in practice it is likely that $m = \ell$ and thus GMRES would converge in exactly ℓ iterations in exact arithmetic.

By comparison, the approximate version of the preconditioner $\widehat{\mathcal{P}}$ will not have this same Jordan structure as these structures are well-known to be unstable to small perturbations. Therefore, we will not be able to determine such precise convergence estimates for $\widehat{\mathcal{P}}$. Simply as a heuristic, we examine the eigenvalues of the approximate preconditioned system to show that they are indeed very close to those of the exact preconditioner. Figure 3.1 shows the eigenvalues of a small system with various time-stepping methods. It is evident that the eigenvalues are clustered extremely close around one on the real axis.

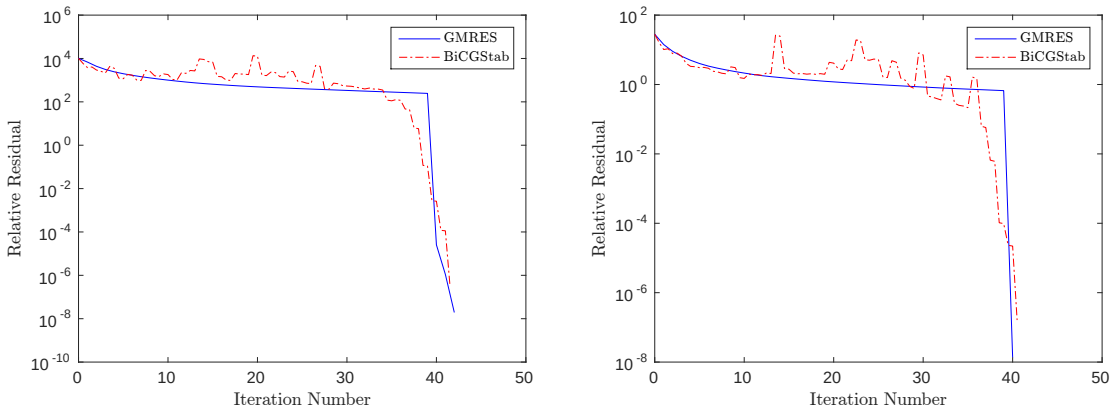


(a) (b)

Figure 3.1: Eigenvalues of $\widehat{\mathcal{P}}^{-1}\mathcal{A}$ with $n = 289, \ell = 10$, and $\tau = 0.1$ for different time-stepping schemes. Figure (a) provides a detailed view of the eigenvalues while (b) shows the same eigenvalues on a larger axis in order to demonstrate how closely the eigenvalues are clustered about 1 on the real axis.

Although the eigenvalues remain nicely clustered around one, it was the minimal polynomial, or equivalently the Jordan structure of the preconditioned system which was used to predict convergence and this structure is no longer guaranteed using $\widehat{\mathcal{P}}$, however, we hope that similar properties may be observed. Figure 3.2 shows the relative residual for the GMRES and BiCGSTAB methods using $\widehat{\mathcal{P}}$ as a preconditioner. For both methods, little convergence is seen until the ℓ-th iteration, at which point sudden decrease in the residual is seen. This suggests that although $\widehat{\mathcal{P}}^{-1}\mathcal{A}$ does not have the same Jordan structure as $\mathcal{P}^{-1}\mathcal{A}$, this 'nearby' eigenvalue degeneracy affects the convergence of the iterative method. This is perhaps suggested by the work on GMRES convergence for perturbed systems found in [102].

We have shown that using a block Jacobi preconditioner exactly results in termination in ℓ iterations and that the multigrid approximation exhibits a similar behaviour. We noted that $\widehat{\mathcal{P}}$ could be parallelized over time resulting in each application of the preconditioner being able to be performed in the same amount of time as one V-cycle. Thus, if termination occurs at approximately the ℓ-th iteration, the total time to convergence is equivalent to ℓ V-cycles. If we compare this to a block forward substitution method which takes $r\ell$ V-cycles we are theoretically able to obtain overall speed-up by a factor of r.



(a) Heat Equation (b) Convection-Diffusion Equation

Figure 3.2: Convergence of $\widehat{\mathcal{P}}^{-1}\mathcal{A}$ with $n = 289, \ell = 40$, and $\tau = 1/40$ using GMRES and BiCGStab methods. Rapid convergence is seen for both problems at the ℓ-th iteration.

3.3 Method Modifications

This method is very straightforward in its approach, however, there are several relatively simple alterations which could be made to potentially alter performance. These modifications may be beneficial particularly if the number of processors available is less than the number of time-steps ℓ.

3.3.1 Block Modification

By dropping all subdiagonals of $\mathcal{A}$ to form the preconditioner $\mathcal{P}$, we lose all information of how the system propagates through time. One obvious modification to the proposed method is to retain some of this information so that the preconditioner is a better approximation to the original system. One way this could be achieved is by retaining 'blocks' of the original system and solving each of these separately. Taking blocks containing b time-steps we can write our new preconditioner as a block matrix with ℓ/b blocks of dimension $nb \times nb$. We note that if ℓ is not divisible by b then there will be one smaller block.

Thus we define this new 'block' preconditioner $\mathcal{P}_B$ as

$$\mathcal{P}_B := \left.\begin{bmatrix} B_0 & & & \\ & B_0 & & \\ & & \ddots & \\ & & & \overline{B_0} \end{bmatrix}\right\} \lceil \ell/b \rceil \text{ blocks of size } nb \tag{3.6}$$

where $B_0 \in \mathbb{R}^{nb \times nb}$ is

$$B_0 := \left.\begin{bmatrix} A_0 & & & \\ A_1 & A_0 & & \\ \vdots & & \ddots & \\ A_b & \cdots & A_1 & A_0 \end{bmatrix}\right\} b \text{ blocks of size } n, \tag{3.7}$$

and $\overline{B_0}$ contains a number of blocks equal to the remainder of ℓ/b if ℓ is not divisible by b. We note that if $b > k$, B will be banded with only k subdiagonals. The corresponding approximate preconditioner $\widehat{\mathcal{P}}_B$ is defined by,

$$\widehat{\mathcal{P}}_B^{-1} := \begin{bmatrix} B_{0(MG)} & & & \\ & B_{0(MG)} & & \\ & & \ddots & \\ & & & \overline{B}_{0(MG)} \end{bmatrix}, \tag{3.8}$$

where $B_{0(MG)}$ represents approximating the inverse of B_0 with one V-cycle for each matrix inversion. As with the block Jacobi method, we will first examine the convergence behaviour of the exact preconditioner before showing that a multigrid approximation exhibits similar behaviour.

Theorem 3.2. *For $\mathcal{A}$ and $\mathcal{P}_B$ as defined in* (3.1) *and* (3.6) *respectively with $2b > k$, the eigenvalues of the preconditioned system $\mathcal{T} = \mathcal{P}_B^{-1}\mathcal{A}$ are all equal to one. Furthermore, the minimal polynomial is given by*

$$\rho(\mathcal{T}) = (\mathcal{T} - I_{n\ell})^{\lceil \ell/b \rceil}, \tag{3.9}$$

and therefore a Krylov subspace method will terminate in at most $\lceil \ell/b \rceil$ *iterations.*

Proof. By dividing $\mathcal{A}$ into blocks of the appropriate size also, we have that

$$\mathcal{A} = \begin{bmatrix} B_0 & & & \\ B_1 & B_0 & & \\ & \ddots & \ddots & \\ & & \overline{B_1} & \overline{B_0} \end{bmatrix}, \quad \text{and thus} \quad \mathcal{P}_B^{-1}\mathcal{A} = \begin{bmatrix} I_{bn} & & & \\ J & I_{bn} & & \\ & \ddots & \ddots & \\ & & \bar{J} & I_{rn} \end{bmatrix},$$

where $J = B_0^{-1}B_1$ and $\bar{J} = \overline{B_0}^{-1}\overline{B_1}$. Analogous to Theorem 3.1, $(\mathcal{P}_B^{-1}\mathcal{A} - I_{n\ell})$ will be strictly lower triangular, however we now have that $(\mathcal{P}_B^{-1}\mathcal{A} - I_{n\ell})^{\lceil \ell/b \rceil} = 0$ as there are only $\lceil \ell/b \rceil$ blocks. □

If $2b < k$, the proof follows exactly as before but with an additional subdiagonal of B_i in $\mathcal{A}$, however, we are not typically thinking of such a large value of k.

Although we have succeeded in reducing the number of GMRES iterations for convergence, we now have to do more work in a sequential manner at each iteration. The inversion of each block B_0 requires b inversions performed sequentially and we can only parallelize inversion of the block diagonal matrix $\mathcal{P}_B$ over $\lceil \ell/b \rceil$ processors. Thus the ℓ inversions required to solve a system with $\mathcal{P}_B$ can now only be performed in the same amount of sequential time as b inversions rather than a single inversion as with $\mathcal{P}$.

We can also compute this modification approximately by replacing every inversion with the action of a single multigrid V-cycle. Again this could only be calculated in the same amount of time as b V-cycles when parallelized over time. However, if we obtain convergence in $\lceil \ell/b \rceil$ iterations, with parallelization over time we will once again require a total time essentially equivalent to ℓ V-cycles.

3.3.2 Subdiagonal Modification

The previous modification attempted to retain more information from $\mathcal{A}$ within the preconditioner and the following modification attempts to do this also. In this method, however, we retain more information from the inverse of $\mathcal{A}$.

Since $\mathcal{A}$ is block lower triangular and block Toeplitz, simple calculation shows that $\mathcal{A}^{-1}$ will also be of this form. Knowing this, it is evident that the blocks on the diagonal of $\mathcal{A}^{-1}$ will be A_0^{-1}. Thus, $\mathcal{P}^{-1}$ is equal to the block diagonal of $\mathcal{A}^{-1}$. However, we can also see that for all $k \geq 1$, the first subdiagonal of $\mathcal{A}^{-1}$ is equal to $-A_0^{-1}A_1A_0^{-1}$. Thus we can consider a preconditioner which also retains the first subdiagonal of $\mathcal{A}^{-1}$.

We therefore define the exact preconditioner $\mathcal{P}_S$ and the corresponding multigrid approximated version $\widehat{\mathcal{P}}_S$ to be

$$\mathcal{P}_S^{-1} = \begin{bmatrix} A_0^{-1} & & & \\ -A_0^{-1}A_1A_0^{-1} & A_0^{-1} & & \\ & \ddots & \ddots & \\ & & -A_0^{-1}A_1A_0^{-1} & A_0^{-1} \end{bmatrix} \tag{3.10}$$

and

$$\widehat{\mathcal{P}}_S^{-1} = \begin{bmatrix} A_{0(MG)} & & & \\ -A_{0(MG)}A_1A_{0(MG)} & A_{0(MG)} & & \\ & \ddots & \ddots & \\ & & -A_{0(MG)}A_1A_{0(MG)} & A_{0(MG)} \end{bmatrix}. \tag{3.11}$$

In order to compute $\mathbf{y} = \mathcal{P}_S^{-1}\mathbf{x}$, we first calculate $\mathbf{w_i} = A_0^{-1}\mathbf{x_i}$ for $i = 1, 2, \ldots, \ell$ and store these vectors. We then find $\mathbf{z_i} = -A_0^{-1}A_1\mathbf{w_i}$ for $i = 1, 2, \ldots, \ell - 1$. Finally, $\mathbf{y_1} = \mathbf{w_1}$ and $\mathbf{y_i} = \mathbf{w_{i-1}} + \mathbf{z_i}$ for $i = 2, \ldots, \ell$. Therefore, in total we need to perform $2\ell - 1 \approx 2\ell$ matrix inversions. Correspondingly, the total work in applying $\widehat{\mathcal{P}}_S^{-1}$ would correspond to 2ℓ V-cycles.

Theorem 3.3. *For $\mathcal{A}$ and $\mathcal{P}_S^{-1}$ as defined in* (3.1) *and* (3.10) *respectively, the eigenvalues of the preconditioned system $\mathcal{T} = \mathcal{P}_S^{-1}\mathcal{A}$ are all equal to one. Furthermore, the minimal polynomial is given by*

$$\rho(\mathcal{T}) = (\mathcal{T} - I_{n\ell})^{\lceil \ell/2 \rceil}, \tag{3.12}$$

and therefore a Krylov subspace method will terminate in at most $\lceil \ell/2 \rceil$ *iterations.*

Proof. The preconditioned system $\mathcal{T}$ is given by

$$\mathcal{P}_S^{-1}\mathcal{A} = \begin{bmatrix} I_n & & & & & & \\ 0 & I_n & & & & & \\ J_1 & \ddots & I_n & & & & \\ \vdots & J_1 & \ddots & I_n & & & \\ J_{k-1} & & \ddots & \ddots & \ddots & & \\ & \ddots & & J_1 & 0 & I_n & \\ & & J_{k-1} & \cdots & J_1 & 0 & I_n \end{bmatrix},$$

where $J_i = (-1)(A_0^{-1}A_1)^2 + A_0A_{i+1}$ for $i = 1, \ldots, k-1$. Therefore $\mathcal{P}_S^{-1}\mathcal{A} - I_{n\ell}$ will be strictly lower triangular and the first subdiagonal will also be zero matrices. It is evident that $(\mathcal{T} - I_{n\ell})^{\lceil \ell/2 \rceil} = 0$ since each power introduces two additional subdiagonals of zero matrices. □

Similarly to the block modification, we see that although each application of $\mathcal{P}_S$ takes twice the amount of sequential time as $\mathcal{P}$, the convergence is achieved in half the number of iterations. Thus with parallelization over time, we again see convergence in a total time roughly equivalent to ℓ V-cycles.

3.3.3 Comparison of Methods

Both of these method modifications have retained additional information from the original system $\mathcal{A}$ which results in them being more expensive to apply, however, they achieve convergence in a smaller number of time-steps. Figure 3.3 presents the convergence of both the Block and Subdiagonal methods for a small problem. For the Block method, blocks of size $b = 2$ were used so that both methods were predicted to converge at approximately ℓ/b steps which, in this case, is equal to 40. For both methods, GMRES exhibits rapid convergence within 40 iterations as predicted, while BiCGStab does show some convergence prior to this, in particular for $\widehat{\mathcal{P}}_B$.

For a fully parallelized implementation, it is also worth noting that $\widehat{\mathcal{P}}_S$ would have additional communication costs as information need to be passed between processors

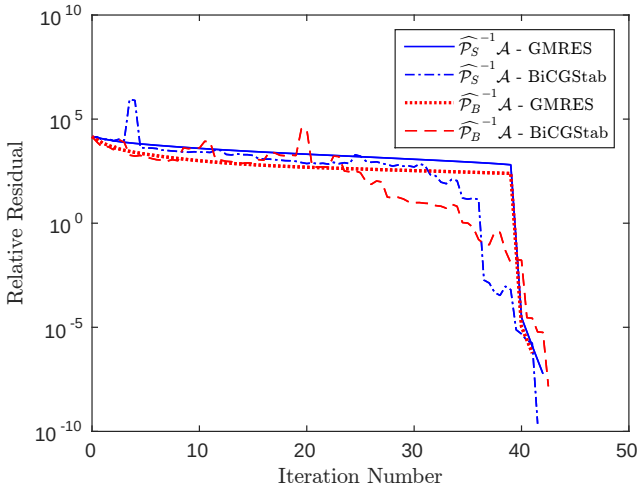


Figure 3.3: Convergence of $\widehat{\mathcal{P}}_S^{-1}\mathcal{A}$ and $\widehat{\mathcal{P}}_B^{-1}\mathcal{A}$ with $n = 289, \ell = 80$, and $\tau = 1/80$ using GMRES and BiCGStab methods. Blocks of size 2 were used for the Block method and as predicted, rapid convergence is seen for both problems at the $\ell/2$-th iteration.

while each block of $\widehat{\mathcal{P}}_B$ can be computed completely independently on separate processors. Also when parallelized, both of these preconditioners can be applied in the same amount of time as 2ℓ V-cycles. If we applied two V-cycles rather than one as proposed for the original block diagonal approximation $\widehat{\mathcal{P}}$ in (3.3), we would still only achieve convergence at approximately the ℓ-th iteration. Thus for the same amount of time per iteration, it would be much more effective to use these method modifications than apply more V-cycles and achieve a more accurate approximation. Furthermore, if less than ℓ processors were available, then $\widehat{\mathcal{P}}_B$ enables fewer processors to be used while still achieving convergence in the same time.

3.4 Normal Equations

For non-symmetric systems, construction of the normal equations is one approach which allows the rigorous convergence theory of symmetric solvers to be applied. As discussed in Section 2.2.5.1, LSQR is mathematically equivalent to applying Conjugate Gradients to the normal equations and thus examining the eigenvalues of the normal equation provides convergence analysis for this method. The normal equations also arise in Schur complement based preconditioners for saddle point problems

which will be discussed in much greater detail in Chapter 5.

The normal equations of the preconditioned system are given by,

$$\mathcal{T}^T\mathcal{T} = (\mathcal{P}^{-1}\mathcal{A})^T(\mathcal{P}^{-1}\mathcal{A}).$$

We note that the eigenvalues of $\mathcal{T}^T\mathcal{T}$ are equivalent to the square of the singular values of $\mathcal{T}$. Firstly, we have the structure of $\mathcal{T}$ from (3.5) and we can clearly see that $\mathcal{T} = I_{n\ell} + \mathcal{J}$, where

$$\mathcal{J} = \begin{bmatrix} 0 & & & & & \\ J_1 & 0 & & & & \\ \vdots & \ddots & 0 & & & \\ J_k & & \ddots & \ddots & & \\ & \ddots & & J_1 & 0 & \\ & & J_k & \cdots & J_1 & 0 \end{bmatrix}. \tag{3.13}$$

Thus, we can think of $\mathcal{T}$ as a perturbation to the identity matrix. The following theorem from Weyl [104, 120] provides bounds for the singular values of additive matrix perturbations.

Theorem 3.4. *If $\widetilde{A} = A + E \in \mathbb{R}^{m\times m}$, then*

$$|\sigma_i(\widetilde{A}) - \sigma_i(A)| \leq \|E\|_2, \quad \forall i = 1, \ldots, m. \tag{3.14}$$

Corollary 3.1. *Let σ be a singular value and λ an eigenvalue of $\mathcal{T}^T\mathcal{T}$. Then,*

$$\max\{0, 1 - \|\mathcal{J}\|_2\} \leq \quad \sigma \quad \leq 1 + \|\mathcal{J}\|_2 \tag{3.15}$$

$$\max\left\{0, (1 - \|\mathcal{J}\|_2)^2\right\} \leq \quad \lambda \quad \leq (1 + \|\mathcal{J}\|_2)^2. \tag{3.16}$$

If $\|\mathcal{J}\|_2 \leq 1$, we are able to bound the eigenvalues of the normal equations in an interval away from zero. For the simple case of Backwards Euler applied to the heat equation on regular grids, this condition is obtained and can be proved by using eigenvalue bounds on the Galerkin finite element matrices since in this case we have

that the mass and stiffness matrices commute, i.e. $KM = MK$. However, this theory does not apply to more complex time-stepping schemes or variable time-steps and is only presented in order to demonstrate that convergence guarantees for the normal equations can be difficult to obtain.

Theorem 3.5. *Let $\mathcal{A}_{BE}$ be the all-at-once system for the heat equation on a regular grid using the Backwards Euler time-stepping as defined in* (3.1) *with $A_0 = M + \tau K$ and $A_1 = -M$, and assuming that M and K commute. Let $\mathcal{P}_{BE}$ be the associated preconditioner as defined in* (3.2)*. The preconditioned system $\mathcal{T}_{BE} = \mathcal{P}_{BE}^{-1}\mathcal{A}_{BE} = I_{n\ell} + \mathcal{J}_{BE}$ where $\mathcal{T}_{BE}$ and $\mathcal{J}_{BE}$ are defined in* (3.5) *and* (3.13) *respectively with $k = 1$. For the constants c_2 and d_1 as defined in* (2.12) *and* (2.13) *we have*

$$\|\mathcal{J}_{BE}\|_2 \leq \frac{1}{1 + \tau\frac{d_1}{c_2}} \tag{3.17}$$

Proof. We know $\|\mathcal{J}_{BE}\|_2 = \sigma_{max}(\mathcal{J}_{BE}) = \sqrt{\lambda_{max}(\mathcal{J}_{BE}^T\mathcal{J}_{BE})}$ and

$$\mathcal{J}_{BE}^T\mathcal{J}_{BE} = \begin{bmatrix} J_1^TJ_1 & & & \\ & \ddots & & \\ & & J_1^TJ_1 & \\ & & & 0 \end{bmatrix},$$

where $J_1 = -(M+\tau K)^{-1}M = -(I_n+\tau M^{-1}K)^{-1}$. Now this implies that $\sqrt{\lambda_{max}(\mathcal{J}_{BE}^T\mathcal{J}_{BE})} = \sqrt{\lambda_{max}(J_1^TJ_1)} = \sigma_{max}(J_1)$. Now, we have assumed that the problem is on a regular grid and so $KM = KM$. Therefore $J_1 = (I_n + \tau M^{-1}K)^{-1}$ is symmetric and we have that $\sigma_{max}(J_1) = \lambda_{max}(J_1)$. Thus,

$$\begin{aligned}\lambda_{max}(J_1) &= \frac{1}{\lambda_{min}(I + \tau M^{-1}K)} \\ &= \frac{1}{1 + \tau\lambda_{min}(M^{-1}K)}\end{aligned}$$

and from (2.12) and (2.13) we have that

$$\frac{d_1}{c_2} \leq \frac{\mathbf{v}^T K \mathbf{v}}{\mathbf{v}^T M \mathbf{v}},$$

for all $\mathbf{v} \in \mathbb{R}^{n \times n}$ and the result follows. □

Corollary 3.2. *The eigenvalues of $\mathcal{T}_{BE}^T \mathcal{T}_{BE}$ are contained within the interval*

$$\left[1 - \frac{1}{1 + \tau \frac{d_1}{c_2}}, 1 + \frac{1}{1 + \tau \frac{d_1}{c_2}} \right]. \tag{3.18}$$

Proof. Combining the results from Corollary 3.1 and Theorem 3.5 gives the result. □

We have now shown that the eigenvalues of the normal equations for this simple heat equation example are bounded in a interval around one, independent of both n and ℓ and only mildly dependent on the time-step size parameter τ. However, these results do not apply to more complex time-stepping schemes or variable time-steps and therefore are limited in their applicability.

3.5 Numerical Results

We present iteration numbers here which are based on an implementation of the simple block diagonal preconditioning method described in Section 3.1 within the IFISS [26, 27, 103] framework. Our implementation of GMRES with no restarting was from the IFISS package. As GMRES can require prohibitive amounts of storage (and growing work) if many iterations are required, we also completed the calculations with the built-in Matlab implementation of BICGSTAB. Both methods were stopped with an absolute residual tolerance of 10^{-6}. The finite element discretization used **Q1** finite elements over the domain $\Omega = [0,1] \times [0,1]$ for the heat equation and $\Omega = [-1,1] \times [-1,1]$ for the convection-diffusion problem. A random starting vector was used for all methods.

3.5.1 Heat Equation

We firstly present results for the solution of the heat equation as described in (2.1). For the multigrid preconditioner, we used the Harwell Subroutine Library AMG preconditioner implementation, HSL_MI20 [12] employed as a 'black box'. This is a standard implementation of Ruge-Stüben algebraic multigrid method [97]. As per the default settings, a Gauss-Seidel smoother was used with two pre- and two post-smoothing steps.

We consider two test problems; one with smooth initial data and a second with random initial conditions. For the first test problem, we consider the Backward Euler method in time with both constant and variable time-step sizes. For the second problem, we consider Crank-Nicolson and BDF2 methods. These methods have not been chosen specifically for each problem but rather just to demonstrate that a variety of time-stepping methods can be used.

3.5.1.1 Smooth Test Problem

Our first example is defined by the initial conditions,

$$u_0 = x(x-1)y(y-1),$$

with no external forcing (i.e. $f = 0$). The iteration numbers are summarized in Table 3.1. For the first part of the table, the solution on the time domain (0,5] is computed and the final solution has values of order 10^{-10} at the final time. As the time-step is reduced along with the grid size, the total number of time-steps ℓ increases. Thus we expect the total number of iterations to increase also, as is seen in the results. In the second part of the table we give results where we maintain a constant number of time-steps, therefore as τ is reduced the final time, $T = \ell\tau$ is also reduced.

In Table 3.2, we present results with variable time-steps. The time-step sizes were chosen to double at each time-step. The range of time-step size τ_i is listed in the table.

Table 3.1: **Heat Equation, smooth test problem, Backward Euler:** Number of iterations for given grid and step sizes. The first part shows doubling of ℓ for halving of h while the second part shows fixed ℓ for varying h.

h	τ	ℓ	DoF	GMRES	BiCGSTAB
2^{-3}	2^{-3}	40	3240	40	37
2^{-4}	2^{-4}	80	23120	80	77
2^{-5}	2^{-5}	160	174240	160	161
2^{-3}	2^{-3}	40	3240	40	37
2^{-4}	2^{-4}	40	11560	40	39
2^{-5}	2^{-5}	40	43560	43	43
2^{-6}	2^{-6}	40	169000	45	45
2^{-7}	2^{-7}	40	665640	47	45

Table 3.2: **Heat Equation, smooth test problem, variable step size:** Number of iterations for given grid and step sizes. Computations were all run until a final time $T = 5$.

h	$[\tau_{min},\tau_{max}]$	ℓ	DoF	GMRES	BiCGSTAB
2^{-3}	$[2^{-6},2^{2}]$	9	729	9	5
2^{-4}	$[2^{-7},2^{2}]$	10	2890	10	6
2^{-5}	$[2^{-8},2^{2}]$	11	11979	12	7
2^{-6}	$[2^{-9},2^{2}]$	12	50700	13	8
2^{-7}	$[2^{-10},2^{2}]$	13	216333	14	9
2^{-8}	$[2^{-11},2^{2}]$	14	924686	15	11

3.5.1.2 Non-Smooth Test Problem

The second example was defined with random initial data taking values from a uniform distribution on $[0, 10]$. In order to demonstrate that other implicit time-stepping schemes are also effective, Crank-Nicolson was used to discretize in time and the results are summarized in Table 3.3 below. In Table 3.4, the 2-step Backwards Differentiation Formula (BDF2) method is used for the time-stepping. It is again evident that the iteration numbers are approximately equal to the number of time-steps and independent of h. However, for both of these time-stepping methods, we see higher iteration numbers than for the previous smooth test-problem.

Table 3.3: **Heat Equation, non-smooth test problem, Crank-Nicolson:** Number of iterations for given grid and step sizes. The first part shows doubling of ℓ for halving of h while the second part shows fixed ℓ for varying h.

h	τ	ℓ	DoF	GMRES	BiCGSTAB
2^{-3}	2^{-5}	40	3240	49	48
2^{-4}	2^{-6}	80	23120	90	95
2^{-5}	2^{-7}	160	174240	175	199
2^{-3}	2^{-5}	40	3240	49	48
2^{-4}	2^{-6}	40	11560	52	50
2^{-5}	2^{-7}	40	43560	53	53
2^{-6}	2^{-8}	40	169000	56	53
2^{-7}	2^{-9}	40	665640	57	54

Table 3.4: **Heat Equation, non-smooth test problem, BDF2:** Number of iterations for given grid and step sizes. The first part shows doubling of ℓ for halving of h while the second part shows fixed ℓ for varying h.

h	τ	ℓ	DoF	GMRES	BiCGSTAB
2^{-3}	2^{-3}	40	3240	45	48
2^{-4}	2^{-4}	80	23120	87	90
2^{-5}	2^{-5}	160	174240	169	190
2^{-3}	2^{-3}	40	3240	45	48
2^{-4}	2^{-4}	40	11560	49	50
2^{-5}	2^{-5}	40	43560	51	52
2^{-6}	2^{-6}	40	169000	53	55
2^{-7}	2^{-7}	40	665640	55	58

3.5.2 Convection-Diffusion Equation

In this section we solve the convection-diffusion equation as defined in (2.18). For the multigrid approximation the Ramage modified geometric multigrid [93] described in Section 2.2.2.1 and implemented in IFISS was used. Two pre- and two post-smoothing steps were used with four-directional line Gauss-Seidel smoothing.

In each of the problems, ϵ was set equal to 1/200 so the maximum mesh Peclet number ranged between approximately 46 for $h = 2^{-3}$ and 3 for $h = 2^{-7}$. Streamline-Upwind Petrov Galerkin (SUPG) stabilization was used throughout.

3.5.2.1 Variable Vertical Wind

The first test problem considered is taken as Example 6.1.2 from [25]. The wind is vertical and is described by $\mathbf{w} = (0, 1 + (x+1)^2/4)$ and thus increases from left to right. Dirichlet boundary conditions apply to the inflow and side boundaries and u is set equal to 1 on the inflow boundary and decreases quadratically to 0 on the right wall and cubically on the left wall. The vector $\mathbf{u}_0$ is specified as zero except for the boundary conditions. Again, the first half of the table shows results calculated for a time domain of (0,5] with $\tau = h$, while the lower half shows results with a constant number of time-steps. We again see that the number of iterations remains approximately equal to the number of time-steps and independent of the mesh size. There is some increase in iteration numbers evident when the BiCGSTAB method was used.

3.5.2.2 Double Glazing Problem

The second convection-diffusion test problem is given by Example 6.1.4 in [25] and is known as the double glazing problem. This is because it is a simple model for temperature in a cavity with recirculating wind and an external wall which is 'hot'. The wind is described by $\mathbf{w} = (2y(1-x^2), -2x(1-y^2))$. Dirichlet boundary conditions are imposed everywhere on the boundary with $\mathbf{u} = 1$ on the boundary where $x = 1$ and zero on all other boundaries. The vector $\mathbf{u}_0$ was zero everywhere except the boundaries where is satisfies the boundary conditions. The iteration numbers remain

Table 3.5: **Convection-diffusion, variable vertical wind.** Number of iterations for given grid and step sizes. The first part shows doubling of ℓ for halving of h while the second part shows fixed ℓ for varying h.

h	τ	ℓ	DoF	GMRES	BiCGSTAB
2^{-3}	2^{-3}	40	3240	40	38
2^{-4}	2^{-4}	80	23120	80	79
2^{-5}	2^{-5}	160	174240	160	164
2^{-4}	2^{-4}	160	46240	160	142
2^{-5}	2^{-5}	160	174240	160	166
2^{-6}	2^{-6}	160	676000	161	172
2^{-7}	2^{-7}	160	2662560	161	175

Table 3.6: **Convection-diffusion, double glazing problem.** Number of iterations for given grid and step sizes. The first part shows doubling of ℓ for halving of h while the second part shows fixed ℓ for varying h.

h	τ	ℓ	DoF	GMRES	BiCGSTAB
2^{-3}	2^{-3}	40	3240	40	41
2^{-4}	2^{-4}	80	23120	80	78
2^{-5}	2^{-5}	160	174240	160	167
2^{-4}	2^{-4}	160	46240	160	182
2^{-5}	2^{-5}	160	174240	160	167
2^{-6}	2^{-6}	160	676000	175	176
2^{-7}	2^{-7}	160	2662560	165	174

roughly independent of the mesh size.

3.6 Summary

In this chapter, we considered a simple solver for the solution of the all-at-once system for a time-dependent PDE-problem. The simple approach was to use a block Jacobi preconditioner used within a standard non-symmetric solver. If the preconditioner is applied exactly, it was shown that the preconditioned system has a minimal polynomial with degree at most equal to the number of time-steps ℓ. Thus, a Krylov subspace method will terminate in at most ℓ iterations.

In practice, the preconditioner was not applied exactly but approximated through

the use of a multigrid process. Our minimal polynomial theory does not apply to this system, however, we found that the approximate system also exhibited sudden drop-off at the ℓ-th iteration. Thus, a nearby Jordan form seemed to affect the performance of the iterative method; this of itself is an interesting numerical observation.

Due to the block diagonal structure of the proposed preconditioner, it is embarrassingly parallelizable over time. If we were to parallelize the preconditioner over ℓ processors, we would theoretically be able to compute the solution in a total time equivalent to ℓ V-cycles, a speed-up from a sequential approach which would take $r\ell$ V-cycles where r is the average number of V-cycles to solve the system exactly. Method modifications were also proposed to see the effect of retaining additional information in the preconditioner, and we see a conservation of the total number of sequential V-cycles. Some benefits may be achieved, for example, if there were less than ℓ processors available, then the block modification may be advantageous.

In Chapters 6 and 7, this preconditioner will form the basis of an approximation to the Schur complement of a control system. In that formulation, we will want to approximate a term of the form $\mathcal{A}^T\mathcal{A}$ and therefore it is advantageous to consider performance for the normal equations. For simple, heat equation systems we are able to provide eigenvalue bounds for the normal equation which are independent of both the problem size parameters n and ℓ.

The approach examined in this chapter exhibits convergence completely independent of the spatial degrees of freedom n, while it is highly dependent on the number of time-steps. In the following section we propose a circulant based preconditioner where convergence is the complete opposite; while there may be mild dependence on n, iterations are completely independent of ℓ.

CHAPTER 4

Circulant Based Preconditioners

As described in the previous chapters, we can write a time-dependent problem such as the heat equation in an all-at-once manner so we solve one large, block system rather than a sequence of smaller systems. In this chapter, we develop an alternative preconditioner for this problem which utilizes the block Toeplitz structure of the matrix[1].

4.1 Motivation and Model Problem

We begin by considering the heat equation described in (2.1) with non-constant forcing f, however, for the time being, we will restrict ourselves to considering only the Backwards Euler method for time-stepping. Using this method, we only have one block sub-diagonal and the overall system can be described by,

[1]This chapter is based on the report which is Ref. [69]. In particular, Lemma's 4.1 and 4.2 and Theorem 4.7 were completed by Jennifer Pestana as part of that report.

$$\mathcal{A}_{BE}\mathbf{x} := \begin{bmatrix} A & & & \\ B & A & & \\ & \ddots & \ddots & \\ & & B & A \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_\ell \end{bmatrix} = \begin{bmatrix} M\mathbf{u}_0 + \tau\mathbf{f}_1 \\ \tau\mathbf{f}_2 \\ \vdots \\ \tau\mathbf{f}_\ell \end{bmatrix} := \mathbf{b}, \tag{4.1}$$

where $A = M + \tau K \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $B = -M \in \mathbb{R}^{n\times n}$ is symmetric negative definite. We note that $\mathcal{A}_{BE}$ is now an immense $n\ell \times n\ell$ matrix; the construction of $\mathcal{A}_{BE}$ only requires copies of A and B and is never done explicitly.

The matrix $\mathcal{A}_{BE}$ is clearly block Toeplitz, however, we note that it does not have Toeplitz blocks. A summary of preconditioning methods for block Toeplitz matrices which do have Toeplitz blocks (often referred to as BTTB matrices) can be found in [17, Chapter 5]. As we saw in Section 2.3, circulant matrices are one effective method for preconditioning Toeplitz matrices and we want to be able to extend this concept to the block Toeplitz case considered here. Although there are several types of circulants which can be used in this manner, we will focus on the Strang circulant described in Section 2.3.1.

Let us therefore consider the block Strang circulant of $\mathcal{A}_{BE}$ as a preconditioner for the matrix $\mathcal{A}_{BE}$. As $\mathcal{A}_{BE}$ is lower triangular with just one subdiagonal, the corresponding Strang circulant simply consists of wrapping the subdiagonal entry B around to create the block circulant matrix given by

$$\mathcal{P}_{BE} := \begin{bmatrix} A & & & B \\ B & A & & \\ & \ddots & \ddots & \\ & & B & A \end{bmatrix}. \tag{4.2}$$

In order to describe the preconditioned system, we make the observation that $\mathcal{P}_{BE}$ is a rank n perturbation of $\mathcal{A}_{BE}$ since $\mathcal{P}_{BE} = \mathcal{A}_{BE} + E_1 B E_\ell^T$, where $E_i = e_i \otimes I_n \in \mathbb{R}^{n\ell\times n}$ with e_i denoting the i-th column of the $\ell \times \ell$ identity matrix I_ℓ. We can now examine the eigenvalues of the preconditioned system.

Theorem 4.1. *The preconditioned system is equal to* $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE} = I_{n\ell} - \mathcal{A}_{BE}^{-1}E_1 Z^{-1}E_\ell^T$, *which is a rank* n *perturbation to the identity matrix* $I_{n\ell} \in \mathbb{R}^{n\ell\times n\ell}$, *where* $Z = B^{-1} + (\mathcal{A}_{BE}^{-1})_{\ell-1}$ *and* $(\mathcal{A}_{BE}^{-1})_{\ell-1} = E_\ell^T\mathcal{A}_{BE}^{-1}E_1$. *Furthermore,* $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ *has* $n(\ell-1)$ *eigenvalues equal to 1 and* n *eigenvalues equal to the eigenvalues of* $I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1}$.

Proof. Writing $\mathcal{P}_{BE} = \mathcal{A}_{BE} + E_1 B E_\ell^T$, then by the Sherman-Morrison-Woodbury formula we have that

$$\mathcal{P}_{BE}^{-1} = (\mathcal{A}_{BE} + E_1 B E_\ell^T)^{-1} = \mathcal{A}_{BE}^{-1} - \mathcal{A}_{BE}^{-1}E_1(B^{-1} + E_\ell^T\mathcal{A}_{BE}^{-1}E_1)^{-1}E_\ell^T\mathcal{A}_{BE}^{-1},$$

and thus,

$$\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE} = I_{n\ell} - \mathcal{A}_{BE}^{-1}E_1(B^{-1} + E_\ell^T\mathcal{A}_{BE}^{-1}E_1)^{-1}E_\ell^T. \tag{4.3}$$

Since $\mathcal{A}_{BE}^{-1}E_1(B^{-1} + E_\ell^T\mathcal{A}_{BE}^{-1}E_1)^{-1}E_\ell^T$ is of rank n, this shows that the preconditioned system is a rank n perturbation of the identity. Noting that the inverse of $\mathcal{A}_{BE}$ will also be block lower triangular and block Toeplitz, and letting $Z = B^{-1} + E_\ell^T\mathcal{A}_{BE}^{-1}E_1$, then we have that

$$\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE} = I_{n\ell} - \mathcal{A}_{BE}^{-1}E_1 Z^{-1}E_\ell^T = \begin{bmatrix} I_n & & & -(\mathcal{A}_{BE}^{-1})_0 Z^{-1} \\ & I_n & & -(\mathcal{A}_{BE}^{-1})_1 Z^{-1} \\ & & \ddots & \vdots \\ & & I_n & -(\mathcal{A}_{BE}^{-1})_{\ell-2}Z^{-1} \\ & & & I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1} \end{bmatrix},$$

from which we can easily see that the eigenvalues of $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ are $n(\ell-1)$ copies of 1 as well as the n eigenvalues of $I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1}$. □

In fact, we can further describe the eigenvalues of $I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1}$ in terms of the matrices A and B.

Theorem 4.2. *If* μ *is an eigenvalue of* $B^{-1}A$ *then* $\dfrac{\mu^\ell}{\mu^\ell + (-1)^{\ell-1}}$ *is an eigenvalue of* $I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1}$.

Proof. Firstly, a simply inductive argument can be used to show that

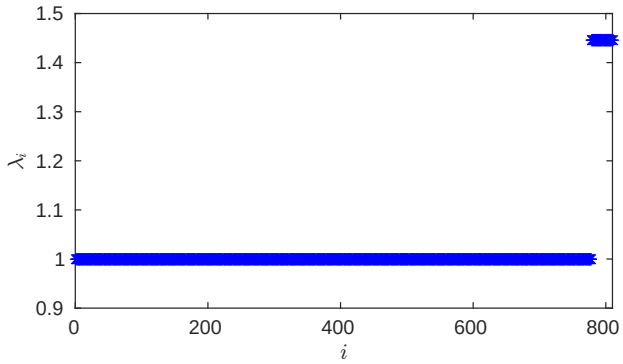


Figure 4.1: The eigenvalues of $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ with $n = 81$ and $\ell = 10$. There are 32 eigenvalues with a value approximately 1.445.

$(\mathcal{A}_{BE}^{-1})_{k-1} = (-1)^{k-1}(A^{-1}B)^{k-1}A^{-1}$ for all $k = 1, \ldots, \ell$. Thus we have that

$$\begin{aligned} I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1} &= I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}(B^{-1} + (\mathcal{A}_{BE}^{-1})_{\ell-1})^{-1} \\ &= I_n - \left[B^{-1}(\mathcal{A}_{BE}^{-1})_{\ell-1}^{-1} + I_n\right]^{-1} \\ &= I_n - \left[B^{-1}(-1)^{\ell-1}A(B^{-1}A)^{\ell-1} + I_n\right]^{-1} \\ &= I_n - \left[(-1)^{\ell-1}(B^{-1}A)^{\ell} + I_n\right]^{-1}. \end{aligned}$$

Finally if μ is an eigenvalue of $B^{-1}A$ then there exists a nonzero vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$\begin{aligned} & B^{-1}A\mathbf{x} = \mu\mathbf{x} \\ \Leftrightarrow \quad & (-1)^{\ell-1}(B^{-1}A)^{\ell}\mathbf{x} = (-1)^{\ell-1}\mu^{\ell}\mathbf{b}x \\ \Leftrightarrow \quad & \left[(-1)^{\ell-1}(B^{-1}A)^{\ell} + I_n\right]^{-1}\mathbf{x} = \frac{1}{(-1)^{\ell-1}\mu^{\ell} + 1}\mathbf{x} \\ \Leftrightarrow \quad & I_n - [(-1)^{\ell-1}(B^{-1}A)^{\ell} + I_n]^{-1}\mathbf{x} = \frac{\mu^{\ell}}{\mu^{\ell} + (-1)^{\ell-1}}\mathbf{x}, \end{aligned}$$

which completes the proof. □

This shows that, although $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ has n eigenvalues not equal to one, if μ is large then these eigenvalues can cluster very close to one. In the case of the heat equation, we see that the largest eigenvalues of $B^{-1}A$ grow with h^{-2} where h is the grid size and therefore we see extremely clustered eigenvalues in practice. Figure 4.1 shows the eigenvalues of $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ for a small system.

We will now show that $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ is diagonalizable.

Theorem 4.3. *The matrix $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ is diagonalizable.*

Proof. Now, $B^{-1}A = -(I_n + \tau M^{-1}K)$ with M and K both symmetric positive definite. From the proof of Theorem 4.2 we have that

$$(\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1} = \left[(-1)^{2\ell-1}(I_n + \tau M^{-1}K)^{\ell} + I_n\right]^{-1} = \left[I_n - (I_n + \tau M^{-1}K)^{\ell}\right]^{-1},$$

which is diagonalizable and has real, negative eigenvalues. Thus, $I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1}$ is diagonalizable, and has eigenvalues that are real and larger than one.

Let $I_n - (\mathcal{A}_{BE}^{-1})_{\ell-1}Z^{-1}$ have diagonalization $V^{-1}DV$. Then $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ has the diagonalization $\mathcal{V}^{-1}\mathcal{D}\mathcal{V}$, where

$$\mathcal{V} = \begin{bmatrix} V_0 & I & & & \\ V_1 & \mathbf{0} & I & & \\ \vdots & & \ddots & \ddots & \\ V_{\ell-2} & & & \mathbf{0} & I \\ V & & & & \mathbf{0} \end{bmatrix}, \quad \text{and} \quad \mathcal{D} = \begin{bmatrix} D & & & & \\ & I & & & \\ & & I & & \\ & & & \ddots & \\ & & & & I \end{bmatrix} \tag{4.4}$$

with $V_i = (\mathcal{A}_{BE}^{-1})_i Z^{-1} V (I_n - D)^{-1}$. □

Since $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ is diagonalizable, then combined with the result from Theorem 4.1, we know that $\mathcal{P}_{BE}^{-1}\mathcal{A}_{BE}$ has exactly $n+1$ distinct eigenvalues and thus GMRES will terminate in at most $n+1$ iterations. However, for more complex time-stepping schemes it is not always obvious that the preconditioned system is diagonalizable, as will be discussed in Section 4.1.3. Furthermore, Theorem 4.3 will not necessarily be applicable if the preconditioner is applied approximately, such as with a multigrid method.

Although we have now demonstrated that the preconditioned system has a number of non-unit eigenvalues independent of the number of time-steps ℓ (but dependent on h), the circulant preconditioner we have proposed is, in principle, just as difficult to invert as the original matrix $\mathcal{A}$. In order to demonstrate an easy, and indeed parallelizable, method of inverting $\mathcal{P}$ we will now consider the matrices in Kronecker product notation.

4.1.1 Kronecker Product Form

The block structure of the matrices considered allows us to describe them in Kronecker product form as

$$\mathcal{A}_{BE} = I_\ell \otimes A + \Sigma \otimes B \tag{4.5}$$

$$\mathcal{P}_{BE} = I_\ell \otimes A + C \otimes B, \tag{4.6}$$

where

$$\Sigma = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & & & 1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}, \tag{4.7}$$

and I_ℓ is the identity matrix of dimension $\ell \times \ell$. Since C is a circulant matrix, as described in Section 2.3.1 we can define the diagonalization $C = U\Lambda U^*$ and easily apply C or the inverse of C to a vector using the Fast Fourier Transform (FFT). We let the diagonal entries of Λ be denoted by λ_i, and note that in general they are complex. Furthermore, for this very specific circulant, the eigenvalues are in fact the ℓ roots of unity, so that $\lambda_i = e^{2\pi i/\ell}$ where $i = 0, 1, \ldots \ell - 1$.

The Kronecker product has the property that $(W \otimes X)(Y \otimes Z) = (WY \otimes XZ)$. Using this, and the fact that U is unitary, we are able to rewrite the preconditioner $\mathcal{P}_{BE}$ as

$$\begin{aligned} \mathcal{P}_{BE} &= I_\ell \otimes A + C \otimes B \\ &= UU^* \otimes A + U\Lambda U^* \otimes B \\ &= (U \otimes I_n)(U^* \otimes A + \Lambda U^* \otimes B) \\ &= (U \otimes I_n)[I_\ell \otimes A + \Lambda \otimes B](U^* \otimes I_n) \end{aligned}$$

and therefore, noting that $(U \otimes I_n)^{-1} = (U^* \otimes I_n)$ we have

$$\mathcal{P}_{BE}^{-1} = (U \otimes I_n)[I_\ell \otimes A + \Lambda \otimes B]^{-1}(U^* \otimes I_n). \tag{4.8}$$

A similar formulation was used in [51] for a semi-circulant preconditioner.

Applying the inverse of $\mathcal{P}_{BE}$ to a vector requires us to multiply by $U \otimes I_n$ and $U^* \otimes I_n$ and invert the block diagonal matrix $I_\ell \otimes A + \Lambda \otimes B$. In order to apply $U \otimes I_n$, we can first apply a column and row permutation that allows us to instead multiply by the block diagonal matrix $I_n \otimes U$, which has n blocks of size $\ell \times \ell$. Following this multiplication, we must reverse the row and column permutation. Since the required permutation, which is a simple reordering of the spatial and temporal degrees of freedom, is known in advance, multiplication by $U \otimes I_n$ or $U^* \otimes I_n$ can be applied easily. Due to the block diagonal structure present here, some parallelization may also be possible.

The matrix $I_\ell \otimes A + \Lambda \otimes B$ is block diagonal and therefore can be inverted in parallel over ℓ processors. This matrix is complex symmetric and a method such as a complex algebraic multigrid, for example, the AGMG method [73, 74, 77, 78], could be used to approximately perform this step.

4.1.2 Simultaneous Diagonalization

For our formulation of the heat equation, the blocks A and B in (4.1) are symmetric. Let us assume for the moment that A and B also commute, which would be the case if a regular grid is used. Then, as a result, the blocks can be simultaneously diagonalized. The property allows us to further simplify the manner in which we can apply $\mathcal{P}_{BE}$.

If we let $A = X\Phi X^T$ and $B = X\Psi X^T$ then we have

$$\mathcal{P}_{BE}^{-1} = (U \otimes I_n)(I_\ell \otimes X)[I_\ell \otimes \Phi + \Lambda \otimes \Psi]^{-1}(I_\ell \otimes X^T)(U^* \otimes I_n).$$

Now in order to apply the inverse of $I_\ell \otimes A + \Lambda \otimes B$, we first need to apply $(I_\ell \otimes X^T)$, which is a block diagonal matrix and could be applied over ℓ separate processors. We then invert $I_\ell \otimes \Phi + \Lambda \otimes \Psi$, which is diagonal and therefore trivial, before applying $(I_\ell \otimes X)$, which is again block diagonal. Thus when we have this property, the application of a circulant preconditioner becomes much cheaper as the only matrix which requires inversion is diagonal.

If we use a finite element formulation to discretize (2.1) then M and K are simultaneously diagonalizable if we use a uniform square grid. For finite difference methods, the finite element mass matrix is replaced by the identity matrix and therefore will always commute with the diffusion operator K. We note that for the Dirichlet problem discretized by finite elements with uniform grids we are able to compute the diagonalization using sine transforms as we now describe for the problem in two dimensions.

For the x and y directions respectively, the i-th element of the j-th normalized eigenvector is given by $v_x(i,j) = \sqrt{\frac{2}{n_x+1}} \sin\left(\frac{ij\pi}{n_x+1}\right)$, $v_y(i,j) = \sqrt{\frac{2}{n_y+1}} \sin\left(\frac{ij\pi}{n_y+1}\right)$, where n_x is the number of interior nodes in the x-direction and n_y is the number of interior nodes in the y-direction. We construct $X_x \in \mathbb{R}^{n_x \times n_x}$ and $X_y \in \mathbb{R}^{n_y \times n_y}$ by embedding each matrix within an identity matrix such that:

$$X_x = I_{n_x+2}, \quad X_x(2:n_x, 2:n_x) = v_x \tag{4.9}$$

$$X_y = I_{n_y+2}, \quad X_y(2:n_y, 2:n_y) = v_y. \tag{4.10}$$

We now form the two-dimensional eigenvectors X by the simple relation $X = X_x \otimes X_y$. As a result, we can apply X to a vector using discrete sine transforms.

We will now examine the effect more complex time-stepping schemes have on the system.

4.1.3 Multi-Step Methods

For simplicity, we discretized the heat equation from (2.1) using a Backward Euler time-stepping scheme. However other implicit time-stepping schemes could also be used. In this section, we describe how the ideas in the previous sections can be extended to a k-step scheme where $\mathcal{A}$ has k subdiagonals.

Define $\mathcal{A}$ to be the following $n\ell \times n\ell$ block lower triangular Toeplitz matrix formed of ℓ blocks of $n \times n$ matrices with $k \leq \ell - 1$ subdiagonals, and define $\mathcal{P}$ to be corre-

sponding Strang circulant:

$$\mathcal{A} := \begin{bmatrix} A_0 & & & & & \\ A_1 & A_0 & & & & \\ \vdots & \ddots & \ddots & & & \\ A_k & & \ddots & \ddots & & \\ & \ddots & & A_1 & A_0 & \\ & & A_k & \cdots & A_1 & A_0 \end{bmatrix}, \quad \mathcal{P} := \begin{bmatrix} A_0 & & A_k & \cdots & A_2 & A_1 \\ A_1 & A_0 & & & & A_2 \\ \vdots & \ddots & \ddots & & \ddots & \vdots \\ A_k & & \ddots & \ddots & & A_k \\ & \ddots & & A_1 & A_0 & \\ & & A_k & \cdots & A_1 & A_0 \end{bmatrix}. \tag{4.11}$$

Define $\Sigma_i \in \mathbb{R}^{\ell\times\ell}$ to be the Toeplitz matrix of zeros except for 1s on the i-th subdiagonal and C_i to be the corresponding Strang circulant with 1s on the i-th subdiagonal and the $(\ell - i)$-th superdiagonal. Note that C defined in (4.7) would correspond to C_1.

By simple computation we can observe that $C_i = (C_1)^i$, and therefore if we diagonalize $C_1 = U\Lambda U^*$ then $C_i = (C_1)^i = (U\Lambda U^*)^i = U\Lambda^i U^*$.

Again $\mathcal{A}$ and $\mathcal{P}$ can be written in Kronecker form as

$$\mathcal{A} = I_\ell \otimes A_0 + \sum_{i=1}^{k} \Sigma_i \otimes A_i, \tag{4.12}$$

$$\mathcal{P} = I_\ell \otimes A_0 + \sum_{i=1}^{k} C_i \otimes A_i = I_\ell \otimes A_0 + \sum_{i=1}^{k} U\Lambda^i U^* \otimes A_i. \tag{4.13}$$

We can make additional simplifications if all A_i commute with each other and are therefore simultaneously diagonalizable. This will occur for any time-stepping method if the spatial operators K and M commute. We thus assume that we have the diagonalizations $A_i = X\Delta_i X^T$, X orthogonal. We can now write that

$$\mathcal{P} = I_\ell \otimes A_0 + \sum_{i=1}^{k} U\Lambda^i U^* \otimes A_i = (U \otimes I_n) \begin{bmatrix} G_1 & & & \\ & G_2 & & \\ & & \ddots & \\ & & & G_\ell \end{bmatrix} (U^* \otimes I_n) = (U \otimes I_n)\mathcal{G}(U^* \otimes I_n), \tag{4.14}$$

where $\mathcal{G} = \text{diag}(G_1, \ldots, G_\ell)$ and

$$G_j = A_0 + \sum_{i=1}^{k} \lambda_j^i A_i = X \left(\Delta_0 + \sum_{i=1}^{k} \lambda_j^i \Delta_i \right) X^T := X \mathtt{g}_j X^T. \tag{4.15}$$

Furthermore, $\mathcal{G} = (I_\ell \otimes X)\text{diag}(\mathtt{g}_1, \ldots, \mathtt{g}_\ell)(I_\ell \otimes X^T)$ where $(I_\ell \otimes X)$ and $(I_\ell \otimes X^T)$ are block diagonal and $\text{diag}(\mathtt{g}_1, \ldots, \mathtt{g}_\ell)$ is diagonal. The point here is that even for multi-step methods, with simultaneous diagonalization of the spatial operators we can apply the inverse of the preconditioner $\mathcal{P}$ using only multiplication with block diagonal matrices and the inversion of a diagonal matrix, which are all extremely cheap to apply.

We note that the preconditioner $\mathcal{P}$ is now a rank nk perturbation to $\mathcal{A}$. Consequently, the preconditioned system $\mathcal{P}^{-1}\mathcal{A}$ will have at most nk eigenvalues not equal to one and if it is diagonalizable, then GMRES would converge in at most $nk+1$ steps. However, without simultaneous diagonalization of all blocks A_i, this property is not obvious. If simultaneous diagonalization is not possible then an optimal method, such as algebraic multigrid, can be used to approximately solve with $\mathcal{P}$ in an efficient manner. However, robust convergence bounds for GMRES would not be able to be proved in this case.

4.2 Symmetrized System

Although we have been able to describe the eigenvalues of the preconditioned system and have shown that the number of non-unit eigenvalues is independent of the number of time-steps, we require diagonalizability of the preconditioned system in order to determine the convergence of non-symmetric solvers such as GMRES. If we do not have this diagonalizability, provided our spatial operators are symmetric and using the ideas developed in [91], we are able to propose a method which rewrites our system as a symmetric one. This then allows us to use eigenvalue analysis to determine convergence estimates.

As stated earlier, the matrix $\mathcal{A}$ in (4.11) is block Toeplitz with symmetric blocks. We note that we can symmetrize any matrix of this type by pre- or post-multiplication

with the following block Hankel matrix,

$$\mathcal{Y} := \begin{bmatrix} & & I_n \\ & \iddots & \\ I_n & & \end{bmatrix} = Y \otimes I_n, \quad \text{where} \quad Y := \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}. \tag{4.16}$$

Pre- or post-multiplication by $\mathcal{Y}$ will symmetrize any block Toeplitz matrix with symmetric blocks, however in general $\mathcal{YA}$ does not equal $\mathcal{AY}$; for this we would additionally require $\mathcal{A}$ to be symmetric. If we wish to solve the system of equations $\mathcal{A}\mathbf{x} = \mathbf{b}$ then we can solve the equations

$$(\mathcal{YA})\mathbf{x} = \mathcal{Y}\mathbf{b} \text{ or } \mathcal{AY}\mathbf{y} = \mathbf{b},\ \mathbf{y} = \mathcal{Y}\mathbf{x}. \tag{4.17}$$

However, unlike for the original system we are able to use iterative methods for symmetric systems for which much better convergence estimates exist. We also note that Y and $\mathcal{Y}$ are involutory and thus $\mathcal{Y}^{-1} = \mathcal{Y}$ and $Y^{-1} = Y$.

In order to use a symmetric matrix solver such as MINRES we require a symmetric positive definite preconditioner. Our original block Strang circulant preconditioner $\mathcal{P}$ defined in (4.14) is certainly not symmetric and therefore will not be suitable. In [91] it was suggested to use the absolute value of the circulant preconditioner and we extend this idea here. Therefore we define the absolute value preconditioner $|\mathcal{P}|$ to be

$$\begin{aligned} |\mathcal{P}| &= (\mathcal{P}^*\mathcal{P})^{1/2} \\ &= [(U \otimes I_n)\mathcal{G}^*(U^* \otimes I_n)(U \otimes I_n)\mathcal{G}(U^* \otimes I_n)]^{1/2} \\ &= [(U \otimes I_n)\mathcal{G}^*\mathcal{G}(U^* \otimes I_n)]^{1/2} \\ &= (U \otimes I_n)|\mathcal{G}|(U^* \otimes I_n) \\ &= (U \otimes I_n)(I_\ell \otimes X)\begin{bmatrix} |\mathsf{g}_1| & & \\ & \ddots & \\ & & |\mathsf{g}_\ell| \end{bmatrix}(I_\ell \otimes X^T)(U^* \otimes I_n), \end{aligned} \tag{4.18}$$

where g_j is the diagonal $n \times n$ matrix in (4.15) and $|\mathsf{g}_j|$ is its elementwise absolute

value. We note $|\mathcal{P}|$ is symmetric positive definite and therefore can be used in MINRES with the symmetric form of the equation (4.17).

4.2.1 Eigenvalue Analysis

We have now described a symmetric positive definite preconditioner for the symmetrized system (4.17) to be implemented with MINRES. Since eigenvalues provide robust convergence bounds for MINRES, unlike for GMRES, we now wish to determine the eigenvalues of the preconditioned system $|\mathcal{P}|^{-1}\mathcal{Y}\mathcal{A}$. In order to do this, we first need to show that $|\mathcal{P}|$ is symmetric with symmetric blocks and is also block Toeplitz.

Theorem 4.4. *The absolute value preconditioner $|\mathcal{P}|$ as defined in* (4.18) *is block Toeplitz and is symmetric with symmetric blocks.*

Proof. By definition we have that

$$|\mathcal{P}| = (U \otimes I_n)(I_\ell \otimes X)diag(|\mathsf{g}_1|, \ldots, |\mathsf{g}_\ell|)(I_\ell \otimes X^T)(U^* \otimes I_n)$$

where $\mathsf{g}_i = \Delta_0 + \sum_{i=1}^{k} \lambda_j^i \Delta_i$. Since $((U \otimes I_n)(I_\ell \otimes X))^* = (I_\ell \otimes X^T)(U^* \otimes I_n)$, the pre- and post-multiplication by these matrices will preserve the symmetry of the diagonal matrix in the middle and therefore $|\mathcal{P}|$ is Hermitian. Furthermore, since $|\mathcal{P}| = (\mathcal{P}\mathcal{P}^T)^{1/2}$ with $\mathcal{P}$ real, then $|\mathcal{P}|$ is also real, and therefore symmetric. To determine the symmetry of each block we need to look more closely.

Firstly, let

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1\ell} \\ u_{21} & u_{22} & & \\ \vdots & & \ddots & \vdots \\ u_{\ell 1} & & \cdots & u_{\ell\ell} \end{bmatrix} \text{ and } U^* = \begin{bmatrix} \overline{u_{11}} & \overline{u_{21}} & \cdots & \overline{u_{\ell 1}} \\ \overline{u_{12}} & \overline{u_{22}} & & \\ \vdots & & \ddots & \vdots \\ \overline{u_{1\ell}} & & \cdots & \overline{u_{\ell\ell}} \end{bmatrix}.$$

By expanding out the matrices in the definition of $|\mathcal{P}|$ from (4.18), we find that if we

let $|\mathcal{P}|_{jk}$ denote the (j,k) block of $|\mathcal{P}|$ of size $n \times n$, then we have

$$|\mathcal{P}|_{jk} = \sum_{m=1}^{\ell} u_{jm} X |\mathsf{g}|_m \overline{u_{km}} X^T = \sum_{m=1}^{\ell} u_{jm} \overline{u_{km}} X |\mathsf{g}|_m X^T.$$

Since it is evident that the matrix $X|\mathsf{g}|_m X^T$ will be symmetric for all m then all $|\mathcal{P}|_{jk}$ will be a sum of symmetric matrices and therefore also symmetric.

To prove that $|\mathcal{P}|$ is also block Toeplitz we need to look at the definition of each u_{jk}. Now, we know that the columns of U are the eigenvectors of a circulant matrix. Thus, $u_{jk} = f_{jk}/\sqrt{\ell}$ where $f_{jk} = e^{2(j-1)(k-1)\pi i/\ell}$. For $|\mathcal{P}|$ to be block Toeplitz we need that $|\mathcal{P}|_{jk} = |\mathcal{P}|_{(j+1)(k+1)}$ for all $j,k \in \{1,\ldots,\ell-1\}$ as well as that $|\mathcal{P}|_{jj} = |\mathcal{P}|_{kk}$ for all $j,k \in \{1,\ldots,\ell\}$.

Firstly, $|\mathcal{P}|_{jk} = \sum_{m=1}^{\ell} u_{jm}\overline{u_{km}} X|\mathsf{g}|_m X^T$ and

$$u_{jm}\overline{u_{km}} = \frac{1}{\ell} e^{2(j-1)(m-1)\pi i/\ell} e^{-2(k-1)(m-1)\pi i/\ell} = \frac{1}{\ell} e^{2(j-k)(m-1)\pi i/\ell}.$$

Similarly, $|\mathcal{P}|_{(j+1)(k+1)} = \sum_{m=1}^{\ell} u_{(j+1)m}\overline{u_{(k-1)m}} X|\mathsf{g}|_m X^T$, with

$$u_{(j+1)m}\overline{u_{(k+1)m}} = \frac{1}{\ell} e^{2(j-k)(m-1)\pi i/\ell}.$$

Thus $|\mathcal{P}|_{jk} = |\mathcal{P}|_{(j+1)(k+1)}$, so all off diagonals have constant blocks. We finally just need to prove that the diagonal blocks are also constant. This is easy to see, as $u_{jm}\overline{u_{jm}} = 1$ for all $j \in \{1,\ldots,\ell\}$. □

In our eigenvalue analysis, it will prove useful to relate $\mathcal{P}$ in (4.11) and $|\mathcal{P}|$ in (4.18) without using the square root of $\mathcal{P}^T\mathcal{P}$. To do this we introduce the orthogonal matrix $\widetilde{\mathcal{P}}$ where

$$\widetilde{\mathcal{P}} = (U \otimes I_n)(I_\ell \otimes X) \begin{bmatrix} \operatorname{sgn}(\mathsf{g}_1) & & & \\ & \operatorname{sgn}(\mathsf{g}_2) & & \\ & & \ddots & \\ & & & \operatorname{sgn}(\mathsf{g}_\ell) \end{bmatrix} (I_\ell \otimes X^T)(U^* \otimes I_n), \quad (4.19)$$

and $\operatorname{sgn}(\mathsf{g}_j) = \mathsf{g}_j/|\mathsf{g}_j|$ is defined elementwise. Then we have that

$$|\mathcal{P}| = \widetilde{\mathcal{P}}\mathcal{P}. \tag{4.20}$$

We will also use the matrix $\widetilde{\mathcal{P}}^{1/2}$ defined in the usual fashion as

$$\widetilde{\mathcal{P}}^{1/2} = (U \otimes I_n)(I_\ell \otimes X)\operatorname{diag}\left((\operatorname{sgn}(\mathsf{g}_1))^{1/2}, \ldots, (\operatorname{sgn}(\mathsf{g}_\ell))^{1/2}\right)(I_\ell \otimes X^T)(U^* \otimes I_n). \tag{4.21}$$

Theorem 4.5. *The matrices $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{P}}^{1/2}$ as defined in* (4.19) *and* (4.21) *are block Toeplitz and symmetric with symmetric blocks.*

Proof. As with $|\mathcal{P}|$, it is evident that left multiplication with $(U \otimes I_n)(I_\ell \otimes X)$ and right multiplication with $((U \otimes I_n)(I_\ell \otimes X))^*$ maintains the symmetry of the diagonal matrix and so $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{P}}^{1/2}$ are clearly symmetric. We see that $\widetilde{\mathcal{P}}_{jk} = \sum_{m=1}^{\ell} u_{jm}\overline{u_{km}} X \operatorname{sgn}(\mathsf{g}_m) X^T$ and once again this shows that since each block is the sum of Hermitian matrices and is real, then each block is symmetric. Since the coefficients u_{jm} and $\overline{u_{km}}$ are exactly the same as for $|\mathcal{P}|$, the proof of the block Toeplitz structure of $\widetilde{\mathcal{P}}$ follows exactly as in the proof of Theorem 4.4. The same logic also applies for $\widetilde{\mathcal{P}}^{1/2}$. □

We will now use these matrices to help us examine the eigenvalues of the preconditioned system $|\mathcal{P}|^{-1}\mathcal{Y}\mathcal{A}$. It is evident that these eigenvalues will be the same as the eigenvalues of the matrix $|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{A}|\mathcal{P}|^{-1/2}$ by a similarity transform. The matrix $\mathcal{Y}$ of (4.16) comprises of ℓ blocks, and we denote $\mathcal{Y}_k$ to be the corresponding matrix with k blocks.

Theorem 4.6. *Define $E_i = e_i \otimes I_n \in \mathbb{R}^{n\ell \times n}$ with e_i denoting the i-th column of the $\ell \times \ell$ identity matrix I_ℓ. Let $\Phi = [E_1, \ldots E_k] \in \mathbb{R}^{n\ell \times nk}$, $\Psi = [E_{\ell-k+1}, \ldots E_\ell] \in \mathbb{R}^{n\ell \times nk}$ and*

$$W = \begin{bmatrix} A_k & \ldots & A_2 & A_1 \\ & A_k & & A_2 \\ & & \ddots & \vdots \\ & & & A_k \end{bmatrix} \in \mathbb{R}^{nk \times nk}.$$

Then for $|\mathcal{P}|$ *and* $\mathcal{A}$ *as defined as in* (4.18) *and* (4.11) *respectively,*

$$|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{A}|\mathcal{P}|^{-1/2} = Q - Z\Theta Z^T,$$

where $Q = \widetilde{\mathcal{P}}\mathcal{Y}$ *is symmetric and orthogonal,* $Z = |\mathcal{P}|^{-1/2}\Psi S \in \mathbb{R}^{n\ell\times nk}$ *has full rank and the symmetric matrix* $\mathcal{Y}_k W \in \mathbb{R}^{nk\times nk}$ *has the eigenvalue decomposition* $\mathcal{Y}_k W = S\Theta S^T$*.*

Proof. Firstly we see that since the circulant is a rank nk perturbation of $\mathcal{A}$, we can write $\mathcal{P} = \mathcal{A} + \Phi W \Psi^T$. Thus, $\mathcal{A} = \mathcal{P} - \Phi W \Psi^T$ and we have

$$|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{A}|\mathcal{P}|^{-1/2} = |\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{P}|\mathcal{P}|^{-1/2} - |\mathcal{P}|^{-1/2}\mathcal{Y}\Phi W\Psi^T|\mathcal{P}|^{-1/2}.$$

Since both $\mathcal{P}$ and $\widetilde{\mathcal{P}}$ are block Toeplitz with symmetric blocks, pre- or post-multiplication with $\mathcal{Y}$ will result in a symmetric matrix. Furthermore, block circulant matrices will commute provided that the blocks commute. Since we have assumed that all A_i commute we have that $\mathcal{P}$ and $|\mathcal{P}|$ also commute. Thus,

$$|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{P}|\mathcal{P}|^{-1/2} = \mathcal{Y}|\mathcal{P}|^{-1/2}\mathcal{P}|\mathcal{P}|^{-1/2} = \mathcal{Y}\mathcal{P}|\mathcal{P}|^{-1/2}|\mathcal{P}|^{-1/2} = \mathcal{Y}\widetilde{\mathcal{P}}^T = \widetilde{\mathcal{P}}\mathcal{Y} = Q.$$

Now $\mathcal{Y}\Phi = \mathcal{Y}[E_1 \dots E_k] = [E_\ell \dots E_{\ell-k+1}] = \Psi\mathcal{Y}_k$. Thus,

$$|\mathcal{P}|^{-1/2}\mathcal{Y}\Phi W\Psi^T|\mathcal{P}|^{-1/2} = |\mathcal{P}|^{-1/2}\Psi\mathcal{Y}_k W\Psi^T|\mathcal{P}|^{-1/2} = (|\mathcal{P}|^{-1/2}\Psi S)\Theta(|\mathcal{P}|^{-1/2}\Psi S)^T.$$

Since $|\mathcal{P}|$, Ψ and S have full rank, $Z = |\mathcal{P}|^{-1/2}\Psi S$ has rank nk. □

Lemma 4.1. *The matrix* Q *has the same eigenvalues as* $\mathcal{Y}$*, which has* $n\lfloor \ell/2 \rfloor$ *eigenvalues equal to -1 and* $n\lceil \ell/2 \rceil$ *eigenvalues equal to 1. The eigenvectors of* $\mathcal{Y}$ *corresponding to the -1 eigenvalues are the columns of* $E_j - E_{\ell-j+1}$ *for* $j = 1, \dots, \lfloor \ell/2 \rfloor$*. Similarly the eigenvectors corresponding to the 1 eigenvalues are the columns of* $E_j + E_{\ell-j+1}$ *for* $j = 1, \dots, \lfloor \ell/2 \rfloor$*. If* ℓ *is odd the remaining* n *unit eigenvalues have the eigenvectors* $E_{\lceil \ell/2 \rceil}$*.*

Proof. Firstly we want to show that Q and $\mathcal{Y}$ are similar, and therefore have the same eigenvalues. We have shown that $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{P}}^{1/2}$ are block Toeplitz with Hermitian

blocks and thus, $\mathcal{Y}$ will symmetrize them. Since $\widetilde{\mathcal{P}}$ is orthogonal,

$$Q = \widetilde{\mathcal{P}}\mathcal{Y} = \widetilde{\mathcal{P}}^{1/2}\widetilde{\mathcal{P}}^{1/2}\mathcal{Y} = \widetilde{\mathcal{P}}^{1/2}\mathcal{Y}(\widetilde{\mathcal{P}}^{1/2})^T = \widetilde{\mathcal{P}}^{1/2}\mathcal{Y}\widetilde{\mathcal{P}}^{-1/2}.$$

Therefore Q and $\mathcal{Y}$ will have the same eigenvalues.

It is left to determine the eigenvalues of $\mathcal{Y}$. Firstly we note that $\mathcal{Y}E_j = E_{\ell-j+1}$. Therefore we have

$$\mathcal{Y}(E_j - E_{\ell-j+1}) = E_{\ell-j+1} - E_j = -(E_j - E_{\ell-j+1}),$$

so -1 will be an eigenvalue associated with an eigenvector equal to one of the columns of $(E_j - E_{\ell-j+1})$. Similarly, the columns of

$$\mathcal{Y}(E_j + E_{\ell-j+1}) = E_{\ell-j+1} + E_j$$

give the form of the eigenvectors corresponding to unit eigenvalues. If ℓ is odd then for $j = \lceil \ell/2 \rceil$ we have

$$\mathcal{Y}(E_{\lceil \ell/2 \rceil} + E_{\ell-\lceil \ell/2 \rceil+1}) = E_{\ell-\lceil \ell/2 \rceil+1} + E_{\lceil \ell/2 \rceil},$$

so that the remaining n eigenvalues are 1. □

Lemma 4.2. *The eigenvectors of Q are of the form $\widetilde{\mathcal{P}}^{1/2}\mathbf{z}$, where for eigenvalues equal to 1, $\mathbf{z} = \mathcal{Y}\mathbf{z}$ and for eigenvalues equal to -1, $\mathbf{z} = -\mathcal{Y}\mathbf{z}$.*

Proof. As previously noted, $\widetilde{\mathcal{P}}^{1/2}$ is orthogonal and symmetrized by $\mathcal{Y}$. Therefore we have

$$Q\widetilde{\mathcal{P}}^{1/2}\mathbf{z} = \widetilde{\mathcal{P}}\mathcal{Y}\widetilde{\mathcal{P}}^{1/2}\mathbf{z} = \widetilde{\mathcal{P}}^{1/2}\widetilde{\mathcal{P}}^{1/2}\widetilde{\mathcal{P}}^{-1/2}\mathcal{Y}\mathbf{z} = \widetilde{\mathcal{P}}^{1/2}\mathcal{Y}\mathbf{z} = \widetilde{\mathcal{P}}^{1/2}\mathbf{z}.$$

The negative case follows in a similar way since when $\mathbf{z} = -\mathcal{Y}\mathbf{z}$, then

$$Q\widetilde{\mathcal{P}}^{1/2}\mathbf{z} = \widetilde{\mathcal{P}}^{1/2}\mathcal{Y}\mathbf{z} = -\widetilde{\mathcal{P}}^{1/2}\mathbf{z},$$

which concludes the proof. □

Theorem 4.7. *Assuming that* $\widetilde{\mathcal{P}}^{1/2} \in \mathbb{R}^{n\ell\times n\ell}$ *and* $\lceil \ell/2 \rceil < \ell - k$*, the geometric multiplicity of the eigenvalue 1 of* $|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{A}|\mathcal{P}|^{-1/2}$ *is at least* $n(\ell-k) - n\lfloor \ell/2 \rfloor$*, while the geometric multiplicity of the eigenvalue -1 is at least* $n(\ell-k) - n\lceil \ell/2 \rceil$*. This leaves at most* $2nk$ *eigenvalues that are not* ± 1*.*

Proof. Define $\Psi_c = [E_1, \ldots, E_{\ell-k}] \in \mathbb{R}^{n\ell\times n(\ell-k)}$, so that $[\Psi_c, \Psi] = I_{n\ell}$. Since $Z = |\mathcal{P}|^{-1/2}\Psi S$, the columns of $Z_c = |\mathcal{P}|^{1/2}\Psi_c$ will be orthogonal to the columns of Z. Therefore for any $\mathbf{y} \in \mathbb{R}^{n\ell}$,

$$|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{A}|\mathcal{P}|^{-1/2}Z_c\mathbf{y} = QZ_c\mathbf{y} - Z\Theta Z^T Z_c\mathbf{y} = QZ_c\mathbf{y}.$$

Now if $Z_c\mathbf{y}$ is an eigenvector of Q, then Q has an eigenvalue equal to ± 1 by Lemma 4.1, and so

$$|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{A}|\mathcal{P}|^{-1/2}Z_c\mathbf{y} = QZ_c\mathbf{y} = \pm Z_c\mathbf{y}.$$

Thus $Z_c\mathbf{y}$ is an eigenvector of $|\mathcal{P}|^{-1/2}\mathcal{Y}\mathcal{A}|\mathcal{P}|^{-1/2}$ with eigenvalue ± 1. From Lemma 4.2 we have that the eigenvectors of Q must be of the form $\widetilde{\mathcal{P}}^{1/2}\mathbf{z}$, where $\mathbf{z} = \pm\mathcal{Y}\mathbf{z}$, so we now need to determine the form of $Z_c\mathbf{y}$ such that it is an eigenvector of Q.

Looking firstly at the positive eigenvalues of Q, we want to have $Z_c\mathbf{y} = \widetilde{\mathcal{P}}^{1/2}\mathbf{z}$, where $\mathbf{z} = \mathcal{Y}\mathbf{z}$. Thus,

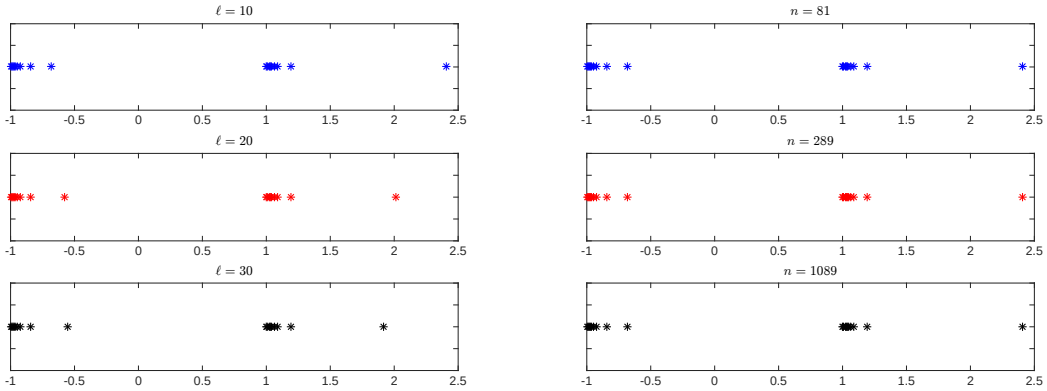
$$\mathbf{z} = \widetilde{\mathcal{P}}^{-1/2}Z_c\mathbf{y} = \widetilde{\mathcal{P}}^{-1/2}|\mathcal{P}|^{1/2}\Psi_c\mathbf{y} = \mathcal{P}^{1/2}\Psi_c\mathbf{y},$$

and since $\mathbf{z} = \mathcal{Y}\mathbf{z}$,

$$\mathcal{P}^{1/2}\Psi_c\mathbf{y} = \mathcal{Y}\mathcal{P}^{1/2}\Psi_c\mathbf{y},$$

which implies that $\mathbf{y} \in \text{null}((I_{n\ell} - \mathcal{Y})\mathcal{P}^{1/2}\Psi_c)$. Therefore, the number of eigenvectors associated with unit eigenvalues is equal to the dimension d_+ of the nullspace of $(I_{n\ell} - \mathcal{Y})\mathcal{P}^{1/2}\Psi_c$. Now, $d_+ = n\ell - \text{rank}((I_{n\ell} - \mathcal{Y})\mathcal{P}^{1/2}\Psi_c)$. We know that $\text{rank}(I_{n\ell} - \mathcal{Y}) = n\lfloor \ell/2 \rfloor$, and since Ψ_c and $\mathcal{P}^{1/2}$ have full rank, it follows that $\text{rank}(\mathcal{P}^{1/2}\Psi_c) = n(\ell-k)$. For any matrices A and B we have that

$$\text{rank}(AB) \leq \min(\text{rank}(A), \text{rank}(B)),$$



(a) (b)

Figure 4.2: Eigenvalues of the preconditioned system $|\mathcal{P}|^{-1}\mathcal{Y}\mathcal{A}$ for varying number of time-steps and varying mesh sizes. In the (a), $n = 81$, and in the (b) $\ell = 10$. In all cases $\tau = 0.1$.

so it follows that $\text{rank}((I_{n\ell} - \mathcal{Y})\mathcal{P}^{1/2}\Psi_c) \leq \min(n\lfloor \ell/2 \rfloor, n(\ell - k))$. We have assumed that $\lceil \ell/2 \rceil < \ell - k$, and so $d_+ \geq (\ell - k)n - \lfloor \ell/2 \rfloor n$.

Using an analogous argument for the negative eigenvalues we find that $d_- \geq (\ell - k)n - \lceil \ell/2 \rceil n$. Thus we have shown that the eigenvalue 1 has geometric multiplicity at least $n(\ell - k) - n\lfloor \ell/2 \rfloor$ and the eigenvalue -1 has multiplicity at least $n(\ell - k) - n\lceil \ell/2 \rceil$. This means that there are at most $2nk$ eigenvalues that are not equal to ± 1. □

Having shown that the preconditioned system has at most $2nk$ eigenvalues that are not ± 1, we know that MINRES will converge in at most $2nk + 2$ steps, which is independent of the number of time-steps, although we not that it is not independent of n. In practice, we do not see nearly this many steps, as the eigenvalues that are not ± 1 are also closely clustered in our numerical experiments for the heat equation. Figure 4.2a shows the eigenvalues of the preconditioned system $|\mathcal{P}|^{-1}\mathcal{Y}\mathcal{A}$ for the same grid sizes with varying numbers of time-steps. We can see that the eigenvalues remain extremely well clustered for each of the values of ℓ.

In Figure 4.2b we also show the eigenvalues of the preconditioned system for a fixed number of time-steps and various spatial grid sizes. It is evident that although the eigenvalues become more spread out as n increases, the eigenvalues remain well

clustered, with only one cluster of eigenvalues away from ±1.

4.3 Non-Symmetric Systems

Throughout the previous section, we have assumed that all A_i are symmetric, as without this property $\mathcal{Y}$ would not symmetrize the system. This property was used as a way to obtain rigorous convergence estimates by using a symmetric method. However, for cases where A_i are not symmetric, we can also form the normal equations by solving the system using LSQR. We note that we could also use this method when A_i are symmetric. We now analyse the eigenvalues of the normal equations of the preconditioned system in order to determine the convergence of LSQR.

Theorem 4.8. *The matrix $(\mathcal{P}^{-1}\mathcal{A})^T(\mathcal{P}^{-1}\mathcal{A})$ has $n(\ell-2k)$ eigenvalues equal to 1, kn eigenvalues less than or equal to 1, and nk eigenvalues greater than or equal to 1.*

Proof. Let $\mathcal{P} = \mathcal{A}+\Phi W\Psi^T$ where $\Phi = [E_1, \dots E_k] \in \mathbb{R}^{n\ell\times nk}$, $\Psi = [E_{\ell-k+1}, \dots E_\ell] \in \mathbb{R}^{n\ell\times nk}$ and

$$W = \begin{bmatrix} A_k & \dots & A_2 & A_1 \\ & A_k & & A_2 \\ & & \ddots & \vdots \\ & & & A_k \end{bmatrix} \in \mathbb{R}^{nk\times nk}.$$

Using the Sherman-Morrison-Woodbury formula, we have

$$\begin{aligned} \mathcal{P}^{-1} &= \mathcal{A}^{-1} - \mathcal{A}^{-1}\Phi(W^{-1} + \Psi^T\mathcal{A}^{-1}\Phi)^{-1}\Psi^T\mathcal{A}^{-1} \\ \mathcal{P}^{-1}\mathcal{A} &= I_{n\ell} - \mathcal{A}^{-1}\Phi(W^{-1} + \Psi^T\mathcal{A}^{-1}\Phi)^{-1}\Psi^T \\ &= I_{n\ell} - \mathcal{A}^{-1}\Phi Z^{-1}\Psi^T, \end{aligned}$$

where $Z = W^{-1} + \Psi^T\mathcal{A}^{-1}\Phi \in \mathbb{R}^{nk\times nk}$. We can now see that $Z^{-1}\Psi^T = \begin{bmatrix} 0 & Z^{-1} \end{bmatrix} \in \mathbb{R}^{nk\times n\ell}$, and consequently $\Phi Z^{-1}\Psi^T = \begin{bmatrix} 0 & Z^{-1} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\ell\times n\ell}$. If we partition $\mathcal{A}^{-1}$ into blocks we

have

$$\mathcal{A}^{-1} = \left[\begin{array}{ccc:ccc} \mathcal{A}_0^{-1} & & & & & \\ \vdots & \ddots & & & & \\ \mathcal{A}_k^{-1} & \cdots & \mathcal{A}_0^{-1} & & & \\ \hdashline \mathcal{A}_{k+1}^{-1} & \cdots & \mathcal{A}_1^{-1} & \mathcal{A}_0^{-1} & & \\ \vdots & \ddots & \vdots & \vdots & \mathcal{A}_0^{-1} & \\ \mathcal{A}_\ell^{-1} & \cdots & \mathcal{A}_{\ell-k+1}^{-1} & \mathcal{A}_{\ell-k}^{-1} & \cdots & \mathcal{A}_0^{-1} \end{array}\right] := \begin{bmatrix} \mathcal{A}_{11}^{-1} & 0 \\ \mathcal{A}_{21}^{-1} & \mathcal{A}_{22}^{-1} \end{bmatrix}, \tag{4.22}$$

where $\mathcal{A}_{11}^{-1} \in \mathbb{R}^{nk\times nk}$, $\mathcal{A}_{21}^{-1} \in \mathbb{R}^{n(\ell-k)\times nk}$, and $\mathcal{A}_{22}^{-1} \in \mathbb{R}^{n(\ell-k)\times n(\ell-k)}$. We can now write that

$$\mathcal{P}^{-1}\mathcal{A} = I_{n\ell} - \mathcal{A}^{-1}\Phi Z^{-1}\Psi^T = I_{n\ell} - \begin{bmatrix} 0 & \mathcal{A}_{11}^{-1}Z^{-1} \\ 0 & \mathcal{A}_{21}^{-1}Z^{-1} \end{bmatrix},$$

and thus

$$\begin{aligned}(\mathcal{P}^{-1}\mathcal{A})^T(\mathcal{P}^{-1}\mathcal{A}) &= \begin{bmatrix} I_{n(\ell-k)} & \mathbf{0} \\ -Z^{-T}\mathcal{A}_{11}^{-T} & I_{nk} - Z^{-T}\mathcal{A}_{21}^{-T} \end{bmatrix}\begin{bmatrix} I_{n(\ell-k)} & -\mathcal{A}_{11}^{-1}Z^{-1} \\ \mathbf{0} & I_{nk} - \mathcal{A}_{21}^{-1}Z^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I_{n(\ell-k)} & -\mathcal{A}_{11}^{-1}Z^{-1} \\ -Z^{-T}\mathcal{A}_{11}^{-T} & (I_{nk} - Z^{-T}\mathcal{A}_{21}^{-T})(I_{n(\ell-k)} - \mathcal{A}_{21}^{-1}Z^{-1}) \end{bmatrix}.\end{aligned}$$

From here we can see that the upper $n(\ell-k)$ principle submatrix is the identity and we can use the Cauchy Interlacing theorem (see for example Chapter 10 of [85]) to relate the eigenvalues of $(\mathcal{P}^{-1}\mathcal{A})^T(\mathcal{P}^{-1}\mathcal{A})$ to the eigenvalues of the identity. The theorem tells us that if we let λ_i be the i-th eigenvalue of $(\mathcal{P}^{-1}\mathcal{A})^T(\mathcal{P}^{-1}\mathcal{A})$ with $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_{n\ell}$, then

$$\lambda_i \le 1 \le \lambda_{i+nk}, \tag{4.23}$$

which gives that the eigenvalues λ_1 to λ_{nk} must be less than or equal to 1, the eigenvalues λ_{nk+1} to $\lambda_{n(\ell-k)}$ must be equal to 1 and eigenvalues $\lambda_{n(\ell-k)+1}$ to $\lambda_{n\ell}$ must be greater than or equal to 1. □

Now since $|\mathcal{P}|^2 = \mathcal{P}^T\mathcal{P} = \mathcal{P}\mathcal{P}^T$, we have

$$\begin{aligned}(\mathcal{P}^{-1}\mathcal{A})^T(\mathcal{P}^{-1}\mathcal{A}) &= \mathcal{A}^T\mathcal{P}^{-T}\mathcal{P}^{-1}\mathcal{A} = \mathcal{A}^T(\mathcal{P}\mathcal{P}^T)^{-1}\mathcal{A} \\ &= \mathcal{A}^T(|\mathcal{P}|)^{-2}\mathcal{A} = (|\mathcal{P}|^{-1}\mathcal{A})^T(|\mathcal{P}|^{-1}\mathcal{A}).\end{aligned}$$

Therefore the eigenvalues of the normal equations when using either $\mathcal{P}$ or $|\mathcal{P}|$ as the preconditioner are the same. We also note that $\mathcal{A}^T(|\mathcal{P}|)^{-2}\mathcal{A}$ has the same eigenvalues as $\mathcal{Y}\mathcal{A}(|\mathcal{P}|)^{-2}\mathcal{A}\mathcal{Y}$, since this is a similarity transform with $\mathcal{Y}^{-1} = \mathcal{Y}$. It follows that the eigenvalues of $(|\mathcal{P}|^{-1}\mathcal{A}\mathcal{Y})^T(|\mathcal{P}|^{-1}\mathcal{A}\mathcal{Y})$ are the same as the eigenvalues of $(\mathcal{P}^{-1}\mathcal{A})^T(\mathcal{P}^{-1}\mathcal{A})$, and that the singular values of $|\mathcal{P}|^{-1}\mathcal{A}\mathcal{Y}$ are the same as those of $\mathcal{P}^{-1}\mathcal{A}$. However, we specifically looked at the normal equations for non-symmetric problems which could not be symmetrized in this manner.

Therefore, we have again shown that using a block circulant based preconditioner results in a number of non-unit eigenvalues independent of the number of time-steps. However, the values of the non-unit eigenvalues can depend on both the number of time-steps ℓ and the number of spatial degrees of freedom n.

4.4 Numerical Results

In this section, we present numerical results for an implementation of the method described in the previous sections within the IFISS [26, 27, 103] framework. Our implementation of GMRES was the standard Matlab implementation and did not allow restarting. We also used the standard Matlab implementations of MINRES, LSQR and BICGSTAB. All methods were stopped with a relative residual tolerance of 10^{-6} and used a random initial guess. The finite element discretization used **Q1** finite elements over the domain $\Omega = [0,1]\times[0,1]$ for the heat equation and $\Omega = [-1,1]\times[-1,1]$ for the convection-diffusion equation. For the algebraic multigrid preconditioner, we used AGMG [73, 74, 77, 78] with default settings, which can be applied to complex matrices. Note that for use with GMRES, we employ $\mathcal{P}_{MG}$ and not $|\mathcal{P}_{MG}|$ (which would, in this case, be awkward if not impossible to compute). We have no rate of convergence guarantees for this approximate non-symmetric solver, but we observe

rapid convergence as seen in Tables 4.1, 4.2 and 4.3. These observations are perhaps not a complete surprise given the supporting rigorous theory in the corresponding symmetric case.

4.4.1 Heat Equation

Our first example is the heat equation as defined in (2.1) with the initial conditions

$$u_0 = x(x-1)y(y-1),$$

with no external forcing (i.e. $f = 0$). We used both the Backward Euler and the 2-step Backward Differentiation Formula (BDF2) for the time-stepping method, with time-step size equal to $\tau = 1/\ell$.

The results presented in Table 4.1 are for the Backward Euler time-stepping method and show that for all methods, iteration numbers are essentially independent of the number of time-steps. Although iteration numbers do increase as the mesh size is decreased, and therefore n increases, the increase is fairly minimal particularly for the GMRES iterations. Mesh independent convergence is observed for GMRES with the approximate preconditioner $\mathcal{P}_{MG}$.

Similar results are observed for the BDF2 method (see Table 4.2), with iteration counts for GMRES and MINRES with $|\mathcal{P}|$ approximately independent of the number of time-steps. Here there does appear to be a mild dependence on the number of time-steps when $\mathcal{P}_{MG}$ is applied.

Table 4.1: Iteration numbers for the heat equation using the Backward Euler method with varying values of grid size and time-step number.

h	ℓ	DoF	GMRES $\mathcal{P}^{-1}\mathcal{A}$	GMRES $\mathcal{P}_{MG}^{-1}\mathcal{A}$	MINRES $\lvert\mathcal{P}\rvert^{-1}\mathcal{Y}\mathcal{A}$	LSQR $\lvert\mathcal{P}\rvert^{-1}\mathcal{A}$
2^{-3}	20	1620	3	3	14	12
	40	3240	3	3	15	16
	60	4860	3	3	17	17
	80	6480	3	3	17	19
	100	8100	3	3	17	21
2^{-4}	20	5780	3	10	14	12
	40	11560	3	11	16	16
	60	17340	3	11	17	17
	80	23120	3	12	17	19
	100	28900	3	12	20	22
2^{-5}	20	21780	4	11	19	19
	40	43560	4	12	21	26
	60	65340	4	12	22	30
	80	87120	4	12	24	36
	100	108900	4	12	24	41
2^{-6}	20	84500	4	11	19	19
	40	169000	4	11	23	26
	60	253500	4	11	23	31
	80	338000	4	11	25	36
	100	422500	4	11	25	41

Table 4.2: Iteration numbers for the heat equation using the BDF2 method with varying values of grid size and time-step number.

h	ℓ	DoF	GMRES $\mathcal{P}^{-1}\mathcal{A}$	GMRES $\mathcal{P}_{MG}^{-1}\mathcal{A}$	MINRES $\lvert\mathcal{P}\rvert^{-1}\mathcal{Y}\mathcal{A}$	LSQR $\lvert\mathcal{P}\rvert^{-1}\mathcal{A}$
2^{-3}	20	1620	3	8	18	16
	40	3240	3	8	22	21
	60	4860	3	8	22	25
	80	6480	3	9	22	28
	100	8100	3	11	23	32
2^{-4}	20	5780	3	13	20	17
	40	11560	3	15	22	22
	60	17340	3	15	23	25
	80	23120	3	16	23	29
	100	28900	3	18	25	32
2^{-5}	20	21780	4	14	25	27
	40	43560	4	16	27	41
	60	65340	4	16	28	52
	80	87120	4	16	29	60
	100	108900	4	20	31	71
2^{-6}	20	84500	4	14	25	28
	40	169000	4	15	27	41
	60	253500	4	15	28	52
	80	338000	4	16	30	62
	100	422500	4	21	32	74

4.4.2 Convection-Diffusion Equation

The convection-diffusion test problem is known as the double glazing problem and was also used as a test problem in the previous chapter. The wind is described by $\mathbf{w} = (2y(1-x^2), -2x(1-y^2))$. Dirichlet boundary conditions are imposed everywhere on the boundary, with $u = 1$ on the boundary where $x = 1$ and zero on all other boundaries. The initial vector $\mathbf{u}_0$ was zero everywhere except the boundaries where it satisfies the boundary conditions. Streamline-Upwind Petrov Galerkin (SUPG) stabilization [16] was used to stabilize the system. For this problem, we used Backward Euler time-stepping and ran to a final time of 5, and so the time-step size was $\tau = 5/\ell$.

As this is a non-symmetric system and the spatial operators do not commute, we were not able to use the simultaneous diagonalization method described in Section 4.1.2. However, we were still able to apply the absolute value preconditioner, although this did require computing ℓ diagonalizations. We therefore also used the AGMG preconditioner with both the Gmres and BiCGStab methods. BiCGStab was used as Gmres without restarting can result in the storage of a large number of Arnoldi vectors which can become problematic for a large number of iterations and BiCGStab avoids this issue.

In Table 4.3, we can see iteration numbers for the non-symmetric solver, which are independent of the number of time-steps and essentially also independent of the grid size. For the LSQR method, although we are able to prove that the number of non-unit eigenvalues of the normal equations is independent of ℓ we see that the values taken by the outlying eigenvalues can become large as ℓ increases; we therefore see that the number of LSQR iterations grows quite rapidly. There is essentially no growth in the number of iterations for the Gmres and BiCGStab methods to which our analysis does not apply.

4.5 Conclusions

We have presented a method of preconditioning an all-at-once system of evolutionary equations based on circulant methods for Toeplitz matrices. For symmetric systems, such as the heat equation, on a regular grid we can use simultaneous diagonalization

Table 4.3: Iteration numbers for the convection-diffusion equation with varying values of grid size and time-step number (* indicates iterations above the maximum of 200).

h	ℓ	DoF	GMRES $\mathcal{P}^{-1}\mathcal{A}$	GMRES $\mathcal{P}_{MG}^{-1}\mathcal{A}$	BiCGSTAB $\mathcal{P}_{MG}^{-1}\mathcal{A}$	LSQR $\lvert\mathcal{P}\rvert^{-1}\mathcal{A}$
2^{-3}	20	1620	12	12	10	86
	40	3240	12	12	10	156
	60	4860	12	12	10	*
	80	6480	13	13	10	*
	100	8100	13	13	10	*
2^{-4}	20	5780	14	14	10	106
	40	11560	15	15	10	192
	60	17340	15	15	10	*
	80	23120	15	16	10	*
	100	28900	15	16	10	*
2^{-5}	20	21780	16	15	10	107
	40	43560	16	16	10	*
	60	65340	17	16	11	*
	80	87120	17	17	10	*
	100	108900	17	17	10	*
2^{-6}	20	84500	16	15	16	106
	40	16900	17	16	16	*
	60	253500	17	17	14	*
	80	338000	18	17	16	*
	100	422500	18	18	15	*

to efficiently apply a block circulant or its absolute value as a preconditioner. We can also rewrite the system as a symmetric one through the use of a block Hankel matrix. This allows us to use MINRES and to provide an eigenvalue analysis, which guarantees convergence in a maximum number of iterations independent of the number of time-steps. In practice, we observe much better convergence even than predicted by this eigenvalue analysis.

For non-symmetric systems, we can also provide eigenvalue analysis for the preconditioned normal equations. Although convergence estimates were able to be determined for LSQR, this method appears to be of little use in practice. Conversely, while we were not able to provide any theory for the multigrid approximated precon-

ditioners used with either GMRES or BiCGStab, these performed well practically.

For both symmetric and non-symmetric systems an algebraic multigrid process can also be employed to approximate the preconditioner; this provides an inexpensive alternative. Although we cannot prove convergence bounds when AMG is used in this way, we nevertheless see promising results for both symmetric and non-symmetric spatial operators with our approach. Conceptually, there seems no barrier to applying any of these preconditioning approaches in parallel over time.

4.5.1 Comparison of Block Diagonal and Circulant Based Preconditioners

In the previous two chapters, we have presented approaches to preconditioning the all-at-once system for the heat and convection-diffusion problems. Both preconditioners should allow some degree of parallelization over time due to the presence of block diagonal structures. In Chapter 3, the preconditioner itself is block diagonal and therefore a matrix-vector product could theoretically be completed separately on ℓ processors. For the circulant based preconditioner presented in this chapter, parallelization is less straight-forward since the preconditioner needs to be decomposed into a product of Kronecker products. Furthermore, the multigrid approximation used in this chapter must be able to be applied to a complex matrix, which is perhaps less desirable than being able to utilize a straightforward (real) geometric multigrid algorithm.

Iteration counts using the multigrid approximation of the block diagonal preconditioner from Chapter 3 scale approximately linearly with the number of time-steps ℓ. In contrast, the iteration counts using the circulant based method approximated using algebraic multigrid, only grow slightly with increases in ℓ and n. Thus for large ℓ, the circulant method may be favourable. We also note that with parallelization, the block diagonal preconditioner only achieves a constant factor speed-up in comparison to a traditional sequential method. Although easily implemented, the amount of speed-up achieved may not be worthwhile for small systems.

A significant drawback of the circulant based method is that it is only applicable to linear, constant coefficient problems with constant time-step size. Furthermore,

only the multigrid approximated methods seemed to provide a practical approach for solution of the non-symmetric convection-diffusion problem. However, we believe that the exploitation of the block Toeplitz structure for the types of systems and the ability to convert these systems to symmetric ones which are solvable with MINRES, is an interesting and novel approach worthy of further investigation. Both of these methods show a great deal of promise when used within a preconditioner for the much larger optimal control problems. It is in this context that we will further develop these methods in the following chapters.

CHAPTER 5

Background to Optimal Control

Up to this point, we have studied the solution of linear, time-dependent constant-coefficient partial differential equations. These types of problems are at the core of applied mathematics as they can be used to model an array of physical processes in areas as diverse as biology, engineering, and economics. However, while PDEs can determine the behaviour of a system under some known forces, what if we instead have a desired behaviour and are seeking a description of the forces which might cause it? This can be formulated as an optimization problem whereby we seek to find a forcing which results in the closest approximation to our desired outcome. However, we are constrained by the PDE which governs the relation between forcing and state. This is therefore known as a *PDE-constrained optimization* or *optimal control* problem.

Mathematically, the problem that we will consider can be formulated as

$$\min_{y,u} \frac{1}{2}\|y - \widehat{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)} \tag{5.1}$$

subject to

$$\begin{aligned} \mathcal{L}y &= u, \quad \text{in } \Omega \\ y &= g, \quad \text{on } \partial\Omega, \end{aligned} \tag{5.2}$$

and is often referred to as a 'tracking-type' problem. In this formulation, y is the *state variable* while $\widehat{y}$ is the known *desired state*. Naturally, the term $\|y-\widehat{y}\|^2_{L_2(\Omega)}$ will be small when y is close to the desired behaviour. The *control variable* u affects the state through the application of the differential operator $\mathcal{L}$ with appropriate boundary conditions. The second term in (5.1) is the penalization term $\frac{\beta}{2}\|u\|^2_{L_2(\Omega)}$, without which the problem would be in general ill-posed in the sense of Hadamard. An optimal control problem is well-posed in the sense of Hadamard if it has a unique solution with continuous dependence on the problem parameters. By including this term in the functional we ensure that the L_2-norm of u remains bounded and u has better regularity properties.

The parameter $\beta > 0$, which is known as the *regularization parameter* or *Tikhonov parameter* determines the extent that the penalization is enforced and therefore has important physical properties. If β is relatively large, then the amount of energy in the control variable u will be kept small and therefore the state may differ from that of the desired state. Conversely, if β is small, more energy is allowed in the control and therefore the resulting state is likely to be very similar to the desired one. In our work, the similarity of the state and desired state are measured by the L_2-norm, as is the size of the control variable penalization.

Arising in a variety of industrial applications, these tracking-type optimal control problems have been the subject of intense research. Some examples of areas of research include but are not limited to: chemical reaction processes [4, 87, 122], data assimilation for weather forecasting [31, 64], the Monge-Kantorovich mass transfer problem [7, 9], flow control [47], medical applications [1, 60], pattern formation [107] and financial applications [24]. An example of another type of optimal control, which is not of the tracking type, are drag minimizations problems [20, 35, 39]. For a more thorough overview of these optimal control problems we recommend [11, 25, 112].

5.1 Poisson Control Problem

The focus of this thesis is time-dependent problems and this includes time-dependent optimal control problems. However before these can be discussed, we will illustrate the basic concepts of PDE-constrained optimization through the simpler example of a time-independent problem. The distributed Poisson control problem can be stated as follows:

$$\min_{y,u} \frac{1}{2}\|y - \widehat{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)} \tag{5.3}$$

subject to

$$\begin{aligned} -\nabla^2 y &= u, \quad \text{in } \Omega \\ y &= g, \quad \text{on } \partial\Omega. \end{aligned} \tag{5.4}$$

This is referred to as a distributed control problem as the control is applied over the whole domain Ω. We note this is not always the case and subdomain problems can be developed for a range of applications. Additionally, we have assumed fixed Dirichlet boundary conditions, however, these types of problems can also be formulated as a Neumann boundary control problem. A description of the problem formulation for such systems can be found in [86] although throughout this thesis we will focus on the distributed control problems described in (5.3–5.4).

As with all control problems, we have two options of how to form the linear system which describes the problem. The first is to discretize the problem spatially and then to optimize the resulting discrete system; this is known as the *discretize-then-optimize* approach. The other alternative, known as the *optimize-then-discretize* approach, is to optimize the continuous system first and then discretize the resulting equations.

Importantly, the two methods do not always yield the same result. It is naturally desirable that the two methods coincide so there is no confusion as to the 'correct' solution. To achieve this, care must be taken with certain choices such as stabilization strategies in convection-diffusion problems and time integration methods for time-dependent problems discussed later in this chapter.

In the simple example introduced here, the optimize-then-discretize approach will

yield a symmetric system, however, this is not always the case for more complex problems. For example, in time-dependent problems discussed later in this chapter the choice of time discretization scheme can break this symmetry. However, the discretize-then-optimize method guarantees symmetry due to its construction. As we have seen in Chapter 2, iterative methods for symmetric systems have certain advantages over methods for general systems and therefore it is often advantageous to maintain a symmetric system. Furthermore, this seems more natural in the context of optimization which typically yields symmetric systems. For these reasons, we will follow the discretize-then-optimize method approach, however, we refer the reader to [86] and [94] for more detailed discussion of the comparison between the two approaches.

5.1.1 Discretization and Optimization

To discretize the problem we first consider the constraint equation, which in this case is the Poisson equation. Using the Galerkin finite element method described in Section 2.1.1, we take the following approximations to the state and control variables,

$$y_h = \sum_{j=1}^{n} Y_j \phi_j + \sum_{j=n+1}^{n+n_\partial} Y_j \phi_j, \quad u_h = \sum_{j=1}^{n} U_j \phi_j + \sum_{j=n+1}^{n+n_\partial} U_j \phi_j. \tag{5.5}$$

The weak formulation of the constraint equation (5.4), is to find $y \in \mathcal{H}_0^1(\Omega)$ and $u \in L_2(\Omega)$ such that

$$\int_\Omega \nabla y \cdot \nabla v = \int_\Omega uv, \quad \forall v \in \mathcal{H}_{E_0}^1(\Omega). \tag{5.6}$$

The discrete approximation is thus to find $y_h \in V_E^h$ and $u_h \in V_E^h$ such that

$$\int_\Omega \nabla y_h \cdot \nabla v_h = \int_\Omega u_h v_h, \quad \forall v_h \in V_0^h. \tag{5.7}$$

Finally, substituting in (5.5) we obtain the equations

$$\sum_{j=1}^{n} Y_j \int_\Omega \nabla \phi_j \cdot \nabla \phi_i + \sum_{j=n+1}^{n+n_\partial} Y_j \int_\Omega \nabla \phi_j \cdot \nabla \phi_i = \sum_{j=1}^{n} U_j \int_\Omega \phi_j \phi_i + \sum_{j=n+1}^{n+n_\partial} U_j \int_\Omega \phi_j \phi_i, \tag{5.8}$$

for $i = 1, \ldots, n$. Writing in matrix form using the definitions of the mass and stiffness matrices from (2.11), we obtain

$$K\mathbf{y} = M\mathbf{u} + \mathbf{g}, \tag{5.9}$$

where $\mathbf{g}$ incorporates the boundary conditions. We note that both $\mathbf{y}$ and $\mathbf{u}$ are unknowns and therefore there is no single solution to this problem. However, this equation only represents the constraint within the larger optimization problem.

Using the same finite element approximation we turn to the objective function (5.3) where, after substituting in (5.5), we obtain

$$\begin{aligned} \|y_h - \widehat{y}\|^2_{L_2(\Omega)} &= \int_\Omega (y_h - \widehat{y})^2 = \int_\Omega y_h^2 - 2y_h\widehat{y} + \widehat{y}^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{y_i}\mathbf{y_j} \int_\Omega \phi_i\phi_j - 2\sum_{j=1}^{n} \mathbf{y_j} \int_\Omega \phi_j\widehat{y} + \int_\Omega \widehat{y}^2 \\ &= \mathbf{y}^T M\mathbf{y} - 2\mathbf{y}^T\mathbf{b} + C \end{aligned} \tag{5.10}$$

where C is a constant. Combining (5.9) and (5.10), the discrete version of the control problem can be stated as

$$\begin{aligned} &\min_{\mathbf{y},\mathbf{u}} \frac{1}{2}\mathbf{y}^T M\mathbf{y} - \mathbf{y}^T\mathbf{b} + \frac{\beta}{2}\mathbf{u}^T M\mathbf{u} \\ &\text{such that } K\mathbf{y} = M\mathbf{u} + \mathbf{g}, \end{aligned} \tag{5.11}$$

where, since C is constant, it has been ignored for the optimization problem. Now that the problem has been discretized, it remains to find the optimality conditions for the system. We form the Lagrangian which combines the objective function with the constraint equation and is given by

$$\mathcal{L}(\mathbf{y}, \mathbf{u}, \mathbf{p}) := \frac{1}{2}\mathbf{y}^T M\mathbf{y} - \mathbf{y}^T\mathbf{b} + \frac{\beta}{2}\mathbf{u}^T M\mathbf{u} + \mathbf{p}^T(K\mathbf{y} - M\mathbf{u} - \mathbf{g}). \tag{5.12}$$

Here $\mathbf{p}$ corresponds to the vector $(P_1, \ldots, P_n)^T$ resulting from the finite element dis-

cretization of the adjoint variable p given by

$$p_h = \sum_{j=1}^{n} P_j \phi_j + \sum_{j=n+1}^{n+n_\partial} P_j \phi_j. \tag{5.13}$$

The conditions for stationarity of $\mathcal{L}$ are obtained by differentiating the Lagrangian with respect to the three variables $\mathbf{y}, \mathbf{u}$ and $\mathbf{p}$ and setting each of the derivatives equal to zero. Beginning with the state variable $\mathbf{y}$ we obtain

$$\frac{\partial \mathcal{L}}{\partial Y_i} = 0 \Rightarrow \quad M\mathbf{y} - \mathbf{b} + K^T\mathbf{p} = \mathbf{0} \tag{5.14}$$

which is referred to as the *adjoint equation*. Next we have

$$\frac{\partial \mathcal{L}}{\partial U_i} = 0 \Rightarrow \quad \beta M\mathbf{u} - M\mathbf{p} = \mathbf{0}, \tag{5.15}$$

known as the *gradient equation* and finally,

$$\frac{\partial \mathcal{L}}{\partial P_i} = 0 \Rightarrow \quad K\mathbf{y} - M\mathbf{u} - \mathbf{g} = \mathbf{0}, \tag{5.16}$$

which is called the *state equation*, and is exactly the discretization of the Poisson problem.

Writing these equations in block matrix form results in the following system,

$$A\mathbf{x} := \begin{bmatrix} M & 0 & K^T \\ 0 & \beta M & -M^T \\ K & -M & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{g} \end{bmatrix}, \tag{5.17}$$

which is often referred to as the *Karush-Kuhn-Tucker* (KKT) system. We note that the resulting matrix A is in the form of a *saddle-point* matrix. This type of matrix occurs in many applications and several methods exist to exploit the structure to obtain more efficient solves for this specific type of problem. This will be discussed in more detail in the following section.

5.1.2 Preconditioning for Saddle Point Problems

A symmetric saddle point system can be written as

$$A\mathbf{x} := \begin{bmatrix} \Phi & \Psi^T \\ \Psi & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} = \mathbf{b}, \tag{5.18}$$

where $\Phi \in \mathbb{R}^{m\times m}$ is symmetric and invertible and $\Psi \in \mathbb{R}^{p\times m}$, $p \leq m$ has full row rank. Solving saddle-point systems is a large area of research and many methods have been developed to obtain numerical solutions of these systems. In addition to the optimal problem introduced here, saddle point problems also arise in mixed finite element methods (see for example [25]) as well as interior point methods for other optimization problems. For a comprehensive overview of such methods we recommend [8].

We will focus on developing preconditioners for iterative methods in order to solve the linear system. Since the saddle point system described is symmetric but indefinite, MINRES could be used as an iterative solver for this system. However, as discussed in Section 2.2.4.1, MINRES requires that any preconditioner used must be symmetric and positive definite. Thus, we will restrict our possible preconditioners to those which satisfy this condition.

A widely used preconditioner for saddle point problems of this form is the following:

$$P := \begin{bmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & S \end{bmatrix}, \tag{5.19}$$

where

$$S := \Psi\Phi^{-1}\Psi^T \tag{5.20}$$

is the (negative) *Schur complement* of the system. In order to guarantee the non-singularity of S, we additionally require that Φ is symmetric, positive definite. This preconditioner is optimal as, under exact arithmetic, a Krylov subspace method will converge in a constant number of steps. This was proved by Murphy, Golub, and

Wathen in [72] and Kuznetsov in [61].

Theorem 5.1 [89, Proposition 1]. *For a saddle point matrix A as defined in* (5.18) *and the preconditioner P as defined in* (5.19)*, then if the preconditioned system $T = P^{-1}A$ is non-singular, it will be diagonalizable with three distinct eigenvalues equal to*

$$1, \frac{1+\sqrt{5}}{2}, \quad \textit{and} \quad \frac{1-\sqrt{5}}{2}. \tag{5.21}$$

This result is significant as it shows that any appropriate Krylov subspace method will converge in at most three iterations. However, this is not the only Schur complement based preconditioner which can achieve rapid convergence. Block triangular preconditioners, which include the Uzawa method, are also widely used and studied, however, as they are not symmetric these methods cannot be applied with MINRES and use the rigorous convergence theory this provides. Constraint preconditioning is another possibility in which the preconditioner is also in saddle-point form [57]. However, for the purposes of developing parallelizable methods, block diagonal preconditioners have obvious advantages. Therefore, the preconditioner described in (5.19) will form the basis of the methods considered here.

The matrix P is called an ideal preconditioner since it achieves convergence in a constant number of iterations, however, in practice, it would be difficult to apply. If we consider applying P^{-1} we need to compute Φ^{-1} and $(\Psi\Phi^{-1}\Psi^T)^{-1}$, which is comparable to the amount of work required to invert A by direct elimination. Therefore, practically there is no advantage to using P as a preconditioner in its current form.

To achieve practical preconditioners, we will need to approximate. One way this could be achieved is by replacing the Schur complement S with an expression which is more easily applied. Additionally, instead of computing the inverses of each of the matrices exactly, we could approximate the inverse through a few steps of a separate iterative method, such as multigrid. We will use a combination of both of these approaches as detailed in the following sections.

5.1.3 Schur Complement Approximations

As we have stated, the exact Schur complement is infeasible to use exactly, therefore, we would like to find a suitable replacement which is easier to apply. But how do we determine if the new preconditioner is still effective? Suppose we have a new approximate preconditioner formed by replacing the Schur complement in the (2,2) block of P with $\widetilde{S}$. Let us call this $\widetilde{P}$ given by

$$\widetilde{P} = \begin{bmatrix} \Phi & \mathbf{0} \\ \mathbf{0} & \widetilde{S} \end{bmatrix}. \tag{5.22}$$

What do the eigenvalues of the preconditioned system $\widetilde{T} = \widetilde{P}^{-1}A$ look like now? The eigenvalue problem is of the form,

$$\widetilde{T}\mathbf{x} = \lambda\mathbf{x}$$
$$\begin{bmatrix} I & \Phi^{-1}\Psi^T \\ \widetilde{S}^{-1}\Psi & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \lambda\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix},$$

which gives us

$$\mathbf{x}_1 + \Phi^{-1}\Psi^T\mathbf{x}_2 = \lambda\mathbf{x}_1$$
$$\widetilde{S}^{-1}\Psi\mathbf{x}_1 = \lambda\mathbf{x}_2.$$

One possible solution occurs if $\widetilde{S}^{-1}\Psi\mathbf{x}_1 = 0$ and consequently, $\mathbf{x}_2 = 0$ and $\lambda = 1$. If $\lambda \neq 1$, then

$$-(1-\lambda)\mathbf{x}_1 = \Phi^{-1}\Psi^T\mathbf{x}_2$$
$$-(1-\lambda)\widetilde{S}^{-1}\Psi\mathbf{x}_1 = \widetilde{S}^{-1}\Psi\Phi^{-1}\Psi^T\mathbf{x}_2$$
$$-(1-\lambda)\lambda\mathbf{x}_2 = \widetilde{S}^{-1}S\mathbf{x}_2$$

and we therefore have that $\mathbf{x}_2$ is an eigenvalue of $\widetilde{S}^{-1}S$. By left multiplying by $\mathbf{x}_2^T$

and dividing by $\mathbf{x}_2^T\mathbf{x}_2$ we have,

$$-\lambda(1-\lambda) = \frac{\mathbf{x}_2^T\widetilde{S}^{-1}S\mathbf{x}_2}{\mathbf{x}_2^T\mathbf{x}_2},$$

and if we let $s = \frac{\mathbf{x}_2^T\widetilde{S}^{-1}S\mathbf{x}_2}{\mathbf{x}_2^T\mathbf{x}_2}$, then

$$\lambda = \frac{1 \pm \sqrt{1+4s}}{2}.$$

Since $\mathbf{x}_2$ is an eigenvalue of $\widetilde{S}^{-1}S$ we can bound the values of s by

$$s_{min} = \lambda_{min}(\widetilde{S}^{-1}S) \leq \frac{\mathbf{x}_2^T\widetilde{S}^{-1}S\mathbf{x}_2}{\mathbf{x}_2^T\mathbf{x}_2} \leq \lambda_{max}(\widetilde{S}^{-1}S) = s_{max}$$

and we have the following result.

Theorem 5.2. *Let $\widetilde{S}$ be a symmetric positive definite Schur complement approximation with the eigenvalues of $\widetilde{S}^{-1}S$ contained in the interval $[s_{min}, s_{max}]$. For a saddle-point system of the form* (5.18) *and a preconditioner of the form* (5.22) *then the eigenvalues of $\widetilde{T} = \widetilde{P}^{-1}A$ satisfy either*

$$\lambda = 1,$$

$$\frac{1+\sqrt{1+4s_{min}}}{2} \leq \lambda \leq \frac{1+\sqrt{1+4s_{max}}}{2},$$

or

$$\frac{1-\sqrt{1+4s_{max}}}{2} \leq \lambda \leq \frac{1-\sqrt{1+4s_{min}}}{2}.$$

This tells us that, provided that the Schur complement approximation $\widetilde{S}$ is spectrally close to S, the approximate preconditioner will have bounded eigenvalues.

Let us return to the optimal control system described in (5.18). We can write the system in saddle point form as

$$A = \begin{bmatrix} \Phi & \Psi^T \\ \Psi & \mathbf{0} \end{bmatrix}, \quad \text{where} \quad \Phi = \begin{bmatrix} M & \mathbf{0} \\ \mathbf{0} & \beta M \end{bmatrix}, \quad \text{and} \quad \Psi = \begin{bmatrix} K & -M \end{bmatrix}. \tag{5.23}$$

The Schur complement for this system is

$$S = KM^{-1}K^T + \frac{1}{\beta}M. \tag{5.24}$$

If we were able to invert this Schur complement exactly, the preconditioner from (5.19) would achieve convergence in exactly 3 steps of an appropriate Krylov subspace method. However, due to the additive structure of the Schur complement in (5.24), there is no simple expression for the inverse of S.

Therefore, suppose we want to form a Schur complement approximation that is more easily invertible. Since the additive nature of the two terms causes the problems, a simple solution might be to drop one term altogether. Provided that β is not too small, the second term in the expression could be considered smaller than the first term as it is higher order in h. Thus, the simplification introduced in [95] is to replace S with the approximation given by

$$S_1 := KM^{-1}K^T. \tag{5.25}$$

We will refer to this approach as the 'dropping' strategy. The inverse of S_1 is equal to $K^{-T}MK^{-1}$ and therefore can be applied easily so long as there is a method for computing or approximating solutions to systems with coefficient matrix K. From Theorem 5.2, to ensure the effectiveness of a new preconditioner we would like that S_1 is spectrally close to the original Schur complement S. This property is described by the following eigenvalue problem,

$$\begin{aligned} S_1^{-1}S\mathbf{x} &= \lambda\mathbf{x} \\ (KM^{-1}K^T)^{-1}(KM^{-1}K^T + \frac{1}{\beta}M)\mathbf{x} &= \lambda\mathbf{x} \\ (KM^{-1}K^T)^{-1}M\mathbf{x} &= \beta(\lambda-1)\mathbf{x}. \end{aligned}$$

For the Poisson control problem we have that $K = K^T$ and therefore the left hand side is equal to $(K^{-1}M)^2$, so if we let $\nu^2 = \beta(\lambda - 1)$ then the above is equivalent to

the eigenvalue problem

$$K^{-1}M\mathbf{x} = \nu\mathbf{x}$$
$$\nu = \frac{\mathbf{x}^T M\mathbf{x}}{\mathbf{x}^T K\mathbf{x}}.$$

As discussed in Section 2.1.1 we have bounds for the Galerkin finite element matrices for a general d dimensional problem, which tell us

$$\begin{aligned} \frac{c_1 h^d}{d_2 h^{d-2}} \leq & \frac{\mathbf{x}^T M\mathbf{x}}{\mathbf{x}^T K\mathbf{x}} & \leq \frac{c_2 h^d}{d_1 h^d} \\ \frac{c_1 h^2}{d_2} \leq & \nu & \leq \frac{c_2}{d_1}, \end{aligned}$$

thus

$$\begin{aligned} \left(\frac{c_1 h^2}{d_2}\right)^2 \leq & \beta(\lambda - 1) & \leq \left(\frac{c_2}{d_1}\right)^2 \\ 1 + \frac{1}{\beta}\left(\frac{c_1 h^2}{d_2}\right)^2 \leq & \lambda & \leq 1 + \frac{1}{\beta}\left(\frac{c_2}{d_1}\right)^2. \end{aligned}$$

Although the bounds for the eigenvalues of $S_1^{-1}S$ are not completely independent of h, we can see that as $h \to 0$, the eigenvalues remain bounded away from zero. If S_1 was substituted for $\widetilde{S}$ in (5.19), we therefore might expect that the performance of the preconditioner does not deteriorate significantly for small h. However, we do see that if β is small, the eigenvalues can be no longer clustered about one. This led to the desire to develop preconditioners which are more robust to changes in the regularization parameter β, in particular for small values of β.

The so called 'matching strategy' introduced by Pearson and Wathen in [89], attempted to achieve this robustness by balancing the $\frac{1}{\beta}M$ term with $KM^{-1}K^T$. They proposed the following Schur complement approximation

$$S_2 := \left(K + \frac{1}{\sqrt{\beta}}M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}}M\right)^T, \tag{5.26}$$

which when expanded is equal to

$$S_2 = KM^{-1}K^T + \frac{1}{\beta}M + \frac{1}{\sqrt{\beta}}K + \frac{1}{\sqrt{\beta}}K^T = S + \frac{1}{\sqrt{\beta}}K + \frac{1}{\sqrt{\beta}}K^T. \tag{5.27}$$

Thus, we see that we have two additional terms from the actual Schur complement, but these terms behave like $\frac{1}{\sqrt{\beta}}$ which does not tend to infinity as quickly as $\frac{1}{\beta}$ when $\beta \to 0$. It may be expected that the eigenvalues of $S_2^{-1}S$ are simply more closely clustered than $S_1^{-1}S$ for small β, but we instead find that the eigenvalues of are bounded completely independently of β.

Theorem 5.3 [89]. *Let λ be an eigenvalue of $S_2^{-1}S$. Then,*

$$\frac{1}{2} \leq \lambda \leq 1. \tag{5.28}$$

Proof. The relevant generalized Rayleigh quotient is

$$\frac{\mathbf{v}^T(KM^{-1}K^T + \frac{1}{\beta}M)\mathbf{v}}{\mathbf{v}^T(K + \frac{1}{\sqrt{\beta}}M)M^{-1}(K + \frac{1}{\sqrt{\beta}}M)^T\mathbf{v}}. \tag{5.29}$$

Noting that M is symmetric positive definite, we can define $\mathbf{a} = M^{-1/2}K^T\mathbf{v}$ and $\mathbf{b} = \frac{1}{\sqrt{\beta}}M^{-1/2}M\mathbf{v} = \frac{1}{\sqrt{\beta}}M^{1/2}\mathbf{v}$. Thus we can write (5.29) as

$$\frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}.$$

At this point we introduce a simple quadratic inequality which will be used many times in the remainder of this thesis. If we consider the term $(\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b})$ we can see that this is always greater than or equal to zero as the term is quadratic. Thus,

$$(\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b}) \geq 0 \tag{5.30}$$

$$\Leftrightarrow \quad \mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b} \geq \mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a}. \tag{5.31}$$

Using the above inequality we have

$$\frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} = \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}+\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a}} \geq \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b})} = \frac{1}{2}.$$

Noting that $\mathbf{a}^T\mathbf{a} \geq 0$ and $\mathbf{b}^T\mathbf{b} \geq 0$ as they are also both quadratic and again using the inequality (5.31) we find

$$\frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} \leq \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a}} \leq 1$$

which gives the result. □

Thus, the 'matching' strategy provides a Schur complement approximation which is completely robust to variation in β, making it a particularly strong candidate for a preconditioner. Furthermore, as we will see throughout this thesis, the quadratic inequalities introduced here are able to be used to provide theoretical bounds for several extensions to this preconditioning concept.

5.1.4 Preconditioner Approximations

We previously stated that in order to obtain an effective and practical preconditioner we could approximate the Schur complement and additionally, we could also only approximately apply P. Having studied the former by examining the dropping and matching strategies, we now examine approximate application of the proposed preconditioners. The two Schur complement variants discussed in the previous section leads to the preconditioners P_1 and P_2 for the system A in (5.17), defined as

$$P_1 := \begin{bmatrix} M & & \\ & \beta M & \\ & & S_1 \end{bmatrix}, \quad \text{and} \quad P_2 := \begin{bmatrix} M & & \\ & \beta M & \\ & & S_2 \end{bmatrix},$$

where S_1 and S_2 are defined by 5.25 and (5.26) respectively. Let us now consider how we actually apply these preconditioners. Both are block diagonal so we can consider each block separately. The (1,1) and (2,2) blocks of both preconditioners consist of (a

constant times) a mass matrix, therefore, we desire an efficient way to approximate the inverse of a mass matrix.

One approach might be to take a fixed number of steps of an iterative method such as one discussed in Section 2.2. A Krylov method such as Conjugate Gradients may seem appealing, however, this method is necessarily nonlinear and therefore the theory of linear preconditioning cannot hold. As discussed in [118], a particularly effective alternative is to apply a fixed number of steps of the (linear) Chebyshev semi-iteration discussed in Section 2.2.1.1.

There are several advantages to this approach. Firstly, we noted in Section 2.2.1.1 that for the Chebyshev semi-iteration to be effective, we required bounds for the eigenvalues of the iteration matrix. For finite element mass matrices we have tight bounds for their eigenvalues [117, 118] and therefore the method can be applied effectively. Additionally, we are able to determine a priori the number of steps required to achieve a certain accuracy [96]. Perhaps one of the largest advantages is the comparative cheapness of this method. At each iteration, the main work is only a single sparse matrix-vector multiply. For these reasons, we will use a fixed number of Chebyshev semi-iterations to approximate the action of the inverse of the mass matrices.

This deals with the first two blocks of P_1 and P_2 and we now need to consider the (3,3) block. Both Schur complement approximations contain the stiffness matrix, which is more of a computational challenge than the mass matrix. However, for elliptic PDEs, multigrid methods have been shown to be very effective in approximating the differential operator. Such methods were discussed in Section 2.2.2 and are widely used for such problems.

For more complex differential operators such as convection-diffusion, we need to be more careful with our choice of approximation. However, several multigrid methods have been developed to produce good approximations to these non-symmetric systems. In particular, the Ramage multigrid [93] described in Section 2.2.2.1, as well as algebraic multigrid operators such as AGMG [73, 74, 77, 78, 79], can be effectively be used as 'black-box' solvers for such problems.

5.2 Time-Dependent Optimal Control Problems

While steady-state optimal control problems form an active area of research, the remainder of this thesis will focus on time-dependent PDE-constrained optimization problems. These problems pose a significant computational challenge as we are now required to solve for the state, control, and adjoint variables at each time-step. However, much of the theory from the previous sections can be transferred to the time-dependent case, giving us a solid foundation from which to base our methods.

In Chapters 3 and 4 we saw that all-at-once methods can be used effectively to solve time-dependent PDE problems. We wish to combine these approaches with the Schur complement based preconditioners developed in the previous section.

5.2.1 Heat Control Problem

We begin by considering the time-dependent counterpart to the Poisson control problem, namely the heat control problem which can be stated as follows,

$$\begin{aligned} \min_{y,u} \mathcal{J}(y,u) &= \frac{1}{2}\int_0^T \|y-\widehat{y}\|^2_{L_2(\Omega)}\,dt + \frac{\beta}{2}\int_0^T \|u\|^2_{L_2(\Omega)}\,dt \\ \text{such that} \quad y_t - \nabla^2 y &= u, \quad \text{for } (\mathbf{x},t) \in \Omega \times [0,T] \\ y &= f, \quad \text{on } \partial\Omega, \\ y &= y_0, \quad \text{at } t = 0. \end{aligned} \tag{5.32}$$

For this problem, the state is required to be close to the desired state at all times. It is also possible to construct a problem were the state only needs to be near to the desired state at the final time-step, in which case the first term of the functional $\mathcal{J}(y,u)$ is only integrated over the spatial domain and not the time-domain as well. Often the solution to these 'final-time' problems, only contain energy in the control variable near the final time-step and therefore the only interesting behaviour occurs in the final part of the temporal domain. For this reason, we only focus on the all-times case but more details of final-time problem formulations can be found in [86, 109].

5.2.2 Discretization and Optimization

As with the Poisson control problem, we will form the linear system using a discretize-then-optimize approach. However, we now require both spatial and temporal discretization schemes. For the spatial domain, we will again use the finite element method discussed in Section 2.1.1. For the time discretization, we want to use a numerically stable implicit finite difference scheme, therefore, we will use the Backwards Euler approach throughout. The trapezoidal rule is used to integrate in time.

We note that the constraint is the heat equation, which was discussed in Section 2.1. We will use the all-at-once approach as we did for the forward problem. This results in the constraint equation being given by,

$$\mathcal{K}\mathbf{y} - \tau\mathcal{M}\mathbf{u} = \mathbf{d}, \tag{5.33}$$

where

$$\mathcal{M} := \begin{bmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \end{bmatrix}, \quad \text{and} \quad \mathcal{K} := \begin{bmatrix} M+\tau K & & & \\ -M & M+\tau K & & \\ & \ddots & \ddots & \\ & & -M & M+\tau K \end{bmatrix}. \tag{5.34}$$

The vector $\mathbf{d} = [M\mathbf{y}_0 + \mathbf{g}, \mathbf{g}, \ldots, \mathbf{g}]^T$ incorporates the boundary and initial data where we have assumed that the boundary data remains constant at all time-steps. The state and control are represented in the vectors $\mathbf{y} = [\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^\ell]^T$ and $\mathbf{u} = [\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^\ell]^T$ respectively, where once again we have divided the time domain into ℓ time-steps of constant size τ.

We now turn to the objective function $\mathcal{J}(y, u)$. Following a very similar method to the time-independent case, the objective function can be discretized to form,

$$\frac{\tau}{2}\mathbf{y}^T\mathcal{M}_{1/2}\mathbf{y} - \tau\mathbf{y}^T\mathbf{b} + C + \frac{\beta\tau}{2}\mathbf{u}^T\mathcal{M}_{1/2}\mathbf{u}, \tag{5.35}$$

where

$$\mathcal{M}_{1/2} := \begin{bmatrix} \frac{1}{2}M & & & & \\ & M & & & \\ & & \ddots & & \\ & & & M & \\ & & & & \frac{1}{2}M \end{bmatrix}, \tag{5.36}$$

and $\mathbf{b} = \left[\frac{1}{2}\mathbf{b}^1, \mathbf{b}^2, \ldots, \frac{1}{2}\mathbf{b}^\ell\right]^T$. The $\frac{1}{2}$ term in the $(1,1)$ and (ℓ,ℓ) blocks of $\mathcal{M}_{1/2}$ result from the use of the trapezoidal rule to integrate over time.

The discretized minimization problem can now be written as follows, once again ignoring the constant C,

$$\begin{aligned} \min_{\mathbf{y},\mathbf{u}} \quad & \frac{\tau}{2}\mathbf{y}^T\mathcal{M}_{1/2}\mathbf{y} - \tau\mathbf{y}^T\mathbf{b} + \frac{\beta\tau}{2}\mathbf{u}^T\mathcal{M}_{1/2}\mathbf{u} \\ \text{such that} \quad & \mathcal{K}\mathbf{y} - \tau\mathcal{M}\mathbf{u} = \mathbf{d}. \end{aligned}$$

The associated Lagrangian is given by

$$\mathcal{L} = \frac{\tau}{2}\mathbf{y}^T\mathcal{M}_{1/2}\mathbf{y} - \tau\mathbf{y}^T\mathbf{b} + \frac{\beta\tau}{2}\mathbf{u}^T\mathcal{M}_{1/2}\mathbf{u} + \mathbf{p}^T(\mathcal{K}\mathbf{y} - \tau\mathcal{M}\mathbf{u} - \mathbf{d}), \tag{5.37}$$

where once again $\mathbf{p}$ represents the adjoint variable. As with the time-independent case, the optimality conditions are obtained by differentiating $\mathcal{L}$ with respect to each of the variables and setting these derivatives equal to zero. This results in the following equations which are again known as the state, gradient and adjoint equations respectively:

$$\begin{aligned} \tau\mathcal{M}_{1/2}\mathbf{y} + \mathcal{K}^T\mathbf{p} &= \tau\mathbf{b}, \\ \beta\tau\mathcal{M}_{1/2}\mathbf{u} - \tau\mathcal{M}\mathbf{p} &= \mathbf{0}, \\ \mathcal{K}\mathbf{y} - \tau\mathcal{M}\mathbf{u} &= \mathbf{d}. \end{aligned}$$

Forming the KKT system results in the saddle point system

$$\mathcal{A}\mathbf{x} := \begin{bmatrix} \tau\mathcal{M}_{1/2} & \mathbf{0} & \mathcal{K}^T \\ \mathbf{0} & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M} \\ \mathcal{K} & -\tau\mathcal{M} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}, \tag{5.38}$$

which will be non-singular for all non-zero values of β and τ.

There are several interesting aspects of this system. Firstly, it is of immense dimension. While the time-independent problems resulted in a linear system with dimensions $3n \times 3n$, the matrix $\mathcal{A}$ is of dimension $3n\ell \times 3n\ell$. Importantly, however, it is never constructed explicitly, rather only copies of the mass and stiffness matrices (or functions which compute their action on a vector) need to be stored when an iterative method is used. However, vectors of length $3n\ell$ are required to be stored and we therefore make the assumption that this amount of memory is available to us, which is potentially a restriction to this method.

We also note that the matrix $\mathcal{K}$ is exactly the all-at-once discretization of the heat equation. Therefore, within the larger optimal control problem, we require the solution to the heat equation and its adjoint. In Chapters 3 and 4 we discussed possible methods for solving such all-at-once systems and therefore we may be able to employ such methods within the solution to the time-dependent optimal control problems.

5.2.3 Preconditioning for Time-Dependent Problems

As with the time-independent case, preconditioning will play an important role for solving the linear system (5.38) in a reasonable amount of computational time. As we again have a saddle-point system, we can use the analogous preconditioners for

the time-dependent case. Thus, we define the preconditioner

$$\mathcal{P} := \begin{bmatrix} \tau\mathcal{M}_{1/2} & & \\ & \beta\tau\mathcal{M}_{1/2} & \\ & & \widetilde{\mathcal{S}} \end{bmatrix}, \tag{5.39}$$

where $\widetilde{\mathcal{S}}$ will be a Schur complement approximation. For this problem we find that the Schur complement is given by

$$\mathcal{S} = \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}. \tag{5.40}$$

As before, Theorem 5.2 tells us that we require $\widetilde{\mathcal{S}}$ to be spectrally close to $\mathcal{S}$ in order to obtain clustered eigenvalues of the preconditioned system and consequently rapid convergence of MINRES. Once again, we base our Schur complement approximations on those developed for the time-independent case. The time-dependent equivalent to the dropping strategy is given by

$$\mathcal{S}_1 = \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T. \tag{5.41}$$

Proving eigenvalues bounds for the time-dependent case is slightly more involved than the steady-state case as, while for the Poisson control problem $K = K^T$, here $\mathcal{K} \neq \mathcal{K}^T$. If we look at the generalized Rayleigh quotient we have

$$R_1 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_1\mathbf{v}} = \frac{\mathbf{v}^T\left(\frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\right)\mathbf{v}}{\mathbf{v}^T\frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}} = 1 + \frac{\tau^2}{\beta}\frac{\mathbf{v}^T\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\mathbf{v}}{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}}. \tag{5.42}$$

Now, looking specifically at the remaining Rayleigh quotient we see that

$$\frac{\mathbf{v}^T\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\mathbf{v}}{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{M}^2\mathbf{v}}{\mathbf{v}^T\mathcal{K}\mathcal{K}^T\mathbf{v}}\frac{\mathbf{w}^T\mathcal{M}_{1/2}^{-1}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}\frac{\mathbf{x}^T\mathbf{x}}{\mathbf{x}^T\mathcal{M}_{1/2}^{-1}\mathbf{x}}, \tag{5.43}$$

by letting $\mathbf{w} = \mathcal{M}\mathbf{v}$ and $\mathbf{x} = \mathcal{K}^T\mathbf{v}$. Since $\mathcal{M}_{1/2}$ and $\mathcal{M}^2$ are both block diagonal, we are able to determine bounds on these terms based solely on the Galerkin finite

element bounds for the mass and stiffness matrices. Thus it only remains to determine eigenvalue bounds for the matrix $\mathcal{K}\mathcal{K}^T$. We saw in Section 3.4 that we are able to bound eigenvalues of the normal equations using the Weyl inequalities for the singular values of perturbed matrices (see [104, 120]). We can bound the eigenvalues of $\mathcal{K}\mathcal{K}^T$, or equivalently the singular values of $\mathcal{K}$, by considering the subdiagonal blocks as a perturbation. If we let $\mathcal{K} + \widehat{\Sigma} = \widehat{\mathcal{K}}$ where

$$\widehat{\Sigma} := \begin{bmatrix} \mathbf{0} & & & \\ M & \mathbf{0} & & \\ & \ddots & \ddots & \\ & & M & \mathbf{0} \end{bmatrix}, \tag{5.44}$$

then

$$|\sigma_i(\widehat{\mathcal{K}}) - \sigma_i(\mathcal{K})| \leq \|\widehat{\Sigma}\|.$$

Since $\|\widehat{\Sigma}\| = \lambda_{max}(M) \leq c_2 h^d$ and $(c_1 + \tau d_1)h^d \leq \sigma_i(\widehat{\mathcal{K}}) \leq c_2 h^d + \tau d_2$, then by using Theorem 3.4 we obtain the following bound,

$$(c_1 + \tau d_1 - c_2)^2 h^{2d} \leq \lambda(\mathcal{K}\mathcal{K}^T) \leq (2c_2 h^d + \tau d_2)^2, \tag{5.45}$$

where the constants are defined by the following relations for the Galerkin finite element matrices:

$$c_1 h^d \leq \frac{\mathbf{v}^T M \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq c_2 h^d, \tag{5.46}$$

$$d_1 h^d \leq \frac{\mathbf{v}^T K \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq d_2 h^{d-2}. \tag{5.47}$$

Thus, using (5.43) we obtain

$$\begin{aligned}
\frac{\lambda_{min}(\mathcal{M}^2)}{\lambda_{max}(\mathcal{K}\mathcal{K}^T)}\frac{\lambda_{min}(\mathcal{M}_{1/2}^{-1})}{\lambda_{max}(\mathcal{M}_{1/2}^{-1})} &\leq \frac{\mathbf{v}^T\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\mathbf{v}}{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}} \leq \frac{\lambda_{max}(\mathcal{M}^2)}{\lambda_{min}(\mathcal{K}\mathcal{K}^T)}\frac{\lambda_{max}(\mathcal{M}_{1/2}^{-1})}{\lambda_{min}(\mathcal{M}_{1/2}^{-1})} \\
\frac{(c_1h^d)^2}{(2c_2h^d+\tau d_2)^2}\frac{c_1h^d}{c_2h^d} &\leq \frac{\mathbf{v}^T\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\mathbf{v}}{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}} \leq \frac{(c_2h^d)2}{(c_1+\tau d_1-c_2)^2h^{2d}}\frac{c_2h^d}{c_1h^d} \\
\frac{(c_1h^d)^2}{(2c_2h^d+\tau d_2)^2}\frac{c_1}{c_2} &\leq \frac{\mathbf{v}^T\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\mathbf{v}}{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}} \leq \frac{c_2^2}{(c_1+\tau d_1-c_2)^2}\frac{c_2}{c_1}.
\end{aligned}$$

As with the steady-state case, we have that the lower bound is dependent on h, however, we find that the eigenvalues of $\mathcal{S}_1^{-1}\mathcal{S}$ as given in (5.42) remain bounded away from zero as h approaches zero. Combining these inequalities with the original Rayleigh quotient definition in (5.42) we obtain the following theorem.

Theorem 5.4. *For $\mathcal{S}$ and $\mathcal{S}_1$ as defined in* (5.40) *and* (5.41) *respectively and the constants c_1, c_2, d_1, and d_2 defined in* (5.46) *and* (5.47) *then,*

$$1+\frac{\tau^2}{\beta}\frac{(c_1h^d)^2}{(2c_2h^d+\tau d_2)^2}\frac{c_1}{c_2} \leq \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_1\mathbf{v}} \leq 1+\frac{\tau^2}{\beta}\frac{c_2^2}{(c_1+\tau d_1-c_2)^2}\frac{c_2}{c_1}. \tag{5.48}$$

From here we turn to our second Schur complement approximation, the time-dependent matching strategy preconditioner given by

$$\mathcal{S}_2 = \frac{1}{\tau}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T. \tag{5.49}$$

Proving eigenvalue bounds for the matching strategy turns out to use exactly the same strategy as the time-independent case. However, we first need to recall the positive definiteness of the matrices involved.

Lemma 5.1 [88, Theorem 1]. *For $\mathbf{a} = \mathcal{M}_{1/2}^{-1/2}\mathcal{K}^T\mathbf{v}$ and $\mathbf{b} = \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1/2}^{-1/2}\mathcal{M}\mathbf{v}$ we have,*

$$\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a} > 0. \tag{5.50}$$

Proof. Since $\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a} = \mathbf{v}^T(\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{M}+\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{K})\mathbf{v}$, showing that this is positive is equivalent to showing that $\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{M}+\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T$ is positive definite. Now, if we let $\Delta = \mathcal{M}\mathcal{M}_{1/2}^{-1}$ we can see that Δ will be a block diagonal matrix where the diagonal

blocks Δ_i consist of either 1 or $(1/2)^{-1}$ times the identity matrix I_n. Thus,

$$\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{M} + \mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T = \mathcal{K}\Delta + \Delta\mathcal{K}^T. \tag{5.51}$$

If we let $L = M + \tau K$, and noting that M and K are symmetric and that each Δ_i, being a scalar multiple of the identity matrix, will commute with any matrix, we have that

$$\mathcal{K}\Delta + \Delta\mathcal{K}^T = \begin{bmatrix} 2L\Delta_1 & -\Delta_1 M & & & \\ -\Delta_1 M & 2L\Delta_2 & -\Delta_2 M & & \\ & \ddots & \ddots & \ddots & \\ & & -\Delta_{\ell-2}M & 2L\Delta_{\ell-1} & -M\Delta_{\ell-1} \\ & & & -\Delta_{\ell-1}M & 2L\Delta_\ell \end{bmatrix}. \tag{5.52}$$

By straightforward manipulation we have,

$$\begin{aligned} \mathbf{v}^T(\mathcal{K}\Delta + \Delta\mathcal{K}^T)\mathbf{v} &= 2\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i M\mathbf{v}_i + 2\tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i K\mathbf{v}_i - \sum_{i=1}^{\ell-1}\mathbf{v}_i^T\Delta_i M\mathbf{v}_{i+1} - \sum_{i=1}^{\ell-1}\mathbf{v}_{i+1}^T\Delta_i M\mathbf{v}_i \\ &= 2\tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i K\mathbf{v}_i + 2\sum_{i=1}^{\ell-1}(\mathbf{v}_i - \mathbf{v}_{i+1})^T\Delta_i M(\mathbf{v}_i - \mathbf{v}_{i+1}) \\ &\quad + \mathbf{v}_1^T\Delta_1 M\mathbf{v}_1 + \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_\ell. \end{aligned}$$

Since this expression is a sum of quadratic forms of positive definite matrices it is greater than zero which gives the result. □

Theorem 5.5. *For $\mathcal{S}$ and $\mathcal{S}_2$ as defined in* (5.40) *and* (5.49) *then,*

$$\frac{1}{2} \le R_2 \le 1, \tag{5.53}$$

where

$$R_2 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}. \tag{5.54}$$

Proof. The generalized Rayleigh quotient is given by

$$R_2 = \frac{\mathbf{v}^T(\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau^2}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1/2}\mathcal{M})\mathbf{v}}{\mathbf{v}^T(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M})\mathcal{M}_{1/2}^{-1}(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M})^T\mathbf{v}}. \tag{5.55}$$

Noting that $\mathcal{M}_{1/2}$ is symmetric and positive definite and letting $\mathbf{a} = \mathcal{M}_{1/2}^{-1/2}\mathcal{K}^T\mathbf{v}$ and $\mathbf{b} = \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1/2}^{-1/2}\mathcal{M}\mathbf{v}$, we can write (5.55) as

$$R_2 = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}.$$

Using the quadratic inequality from (5.31) we have,

$$\frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b})} = \frac{1}{2}.$$

Also, from Lemma 5.1 we have that $\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a} > 0$ and thus

$$\frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} \leq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a}} \leq 1$$

which gives the result. □

This theorem shows that $\mathcal{S}_2$ will be spectrally close to $\mathcal{S}$ for all parameter values which is a significant result. This will lead to preconditioners with well-clustered eigenvalues of the preconditioned system. In particular, the matching strategy based preconditioners can be used for small values of β, a regime in which the dropping strategy did not perform well. As discussed further in Chapter 7, this preconditioner has also been shown to be effective for convection-diffusion control problems [90].

5.3 Summary

In this section, we provided an overview of the concept of PDE-constrained optimization. The problem arises in a wide range of applications. For a long time, methods were focussed on steady-state problems as time-dependent problems simply posed too great of a computational challenge. In the first part of this chapter, we outlined

methods for solving a time-independent optimal control problem and demonstrated how Schur complement approximations can be developed to form effective preconditioners for the resulting saddle-point systems. The dropping and matching strategies were particularly focussed on.

In the latter part of the chapter, it was demonstrated that all of the steady-state theory can be extended to the time-dependent case. As computational capabilities have increased, research is increasingly focussed on these types of problems as they become solvable in a practical amount of time if, crucially, effective preconditioners and solvers can be developed.

As with any time-dependent PDE problem, parallelization is inherently difficult. However, with the scale of time-dependent optimal control problems, parallelization is imperative to achieve speed-up of the existing methods. In the following chapters, we aim to combine the parallelizable preconditioners for the forward problem developed in Chapters 3 and 4, with the preconditioners discussed in this chapter for optimal control problems.

CHAPTER 6

Heat Control Problem

It was shown in the previous chapter that Schur complement based preconditioners can form part of an effective solution approach for the all-at-once system of time-dependent optimal control problems. The two Schur complement based preconditioners we examined in more detail were the dropping strategy [95, 109] and the matching strategy [86, 88, 89].

Any such solution method requires approximation of the forward problem (and the backwards problem) which, in the context of the heat control problem, is the all-at-once formulation of the heat equation. Preconditioners for the forward problem were developed in Chapters 3 and 4 and we now want to see if these ideas can be transferred to the control problem context. We also note that for the formulation considered here, the backwards problem corresponds to the transpose of the discretized forward problem. The enables us to use the transpose of the block diagonal and circulant preconditioners to precondition the backwards problem. However, as noted in Section 2.2.5.1, just because a preconditioner is effective for the forward problem does not guarantee good performance when used within the normal equations or, as in this case, the Schur complement.

For the heat control problem, we showed in the previous chapter that by using

a discretization-then-optimization approach, we can obtain the linear system (5.38), which we reproduce for convenience here:

$$\mathcal{A}\mathbf{x} := \begin{bmatrix} \tau\mathcal{M}_{1/2} & \mathbf{0} & \mathcal{K}^T \\ \mathbf{0} & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M} \\ \mathcal{K} & -\tau\mathcal{M} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathbf{b} \\ 0 \\ \mathbf{d} \end{bmatrix}. \tag{6.1}$$

The preconditioners we focussed on were of the form

$$\mathcal{P} = \begin{bmatrix} \tau\mathcal{M}_{1/2} & & \\ & \beta\tau\mathcal{M}_{1/2} & \\ & & \widetilde{\mathcal{S}} \end{bmatrix}, \tag{6.2}$$

where $\widetilde{\mathcal{S}}$ was an approximation to the Schur complement

$$\mathcal{S} = \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}. \tag{6.3}$$

A summary of the two approaches considered is as follows:

- Dropping Strategy: This approach ignores the final term and approximates the Schur complement with

  $$\mathcal{S}_1 = \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T. \tag{6.4}$$

  We note this strategy generally performs well only if the regularization parameter β is relatively large.

- Matching Strategy: By attempting to achieve more robustness to the regularization parameter β, this method approximates the Schur complement by

  $$\mathcal{S}_2 = \frac{1}{\tau}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T, \tag{6.5}$$

  and has the property that the eigenvalues of $\mathcal{S}_2^{-1}\mathcal{S}$ are bounded between $1/2$ and 1 and, therefore, completely independent of all problem parameters.

While we can approximately apply each of these Schur complement approximations

using multigrid processes, the parallel capabilities of such preconditioners are inherently limited due to the lower diagonal structure of the forward problem $\mathcal{K}$. However, two parallelizable preconditioners for $\mathcal{K}$ were introduced in Chapters 3 and 4. These can be summarized as follows:

- Block Diagonal Variation: Here we simply preconditioned the forward problem $\mathcal{K}$ with its block diagonal

$$\widehat{\mathcal{K}} := \begin{bmatrix} M+\tau K & & & \\ & M+\tau K & & \\ & & \ddots & \\ & & & M+\tau K \end{bmatrix}. \tag{6.6}$$

  Even when approximated with a multigrid process, termination of a Krylov method applied to $\mathcal{K}$ with $\widehat{\mathcal{K}}$ as a preconditioner was seen at approximately ℓ iterations, regardless of the grid parameter h. Given the block diagonal structure, this approach is inherently parallelizable over time.

- Circulant Variation: Here we based preconditioners on the corresponding block Strang circulant matrix of $\mathcal{K}$ given by

$$\overline{\mathcal{K}} := \begin{bmatrix} M+\tau K & & & & -M \\ -M & M+\tau K & & & \\ & \ddots & \ddots & & \\ & & -M & M+\tau K \end{bmatrix}. \tag{6.7}$$

  It was shown that only n eigenvalues of $\overline{\mathcal{K}}^{-1}\mathcal{K}$ are not equal to 1, and subsequently this method was numerically shown to converge in a number of iterations with little dependence on n or ℓ. It is also possible, through the use of Kronecker products, to apply the preconditioner in a manner which could also be parallelized over time as shown in Section 4.1.1.

Our goal in this chapter is to combine these approaches to achieve effective pre-

conditioners for time-dependent optimal control problems, which could also be parallelized. This is achieved by replacing $\mathcal{K}$ within $\mathcal{S}_1$ and $\mathcal{S}_2$ with the parallelizable variation $\widehat{\mathcal{K}}$ or $\overline{\mathcal{K}}$. This chapter will be structured as follows. In Section 6.1 we will investigate the dropping strategy with the forward problem replaced first by the block diagonal matrix $\widehat{\mathcal{K}}$, and then by the block circulant matrix $\overline{\mathcal{K}}$. We repeat this process for the matching strategy in Section 6.2, before providing a summary of all of the eigenvalue bounds obtained in Section 6.3. Since all problems considered here are symmetric, these eigenvalue bounds lead directly to convergence estimates for preconditioned iterative methods. Numerical results are presented in Section 6.4 and concluding remarks given in Section 6.5.

6.1 Dropping Strategy Based Preconditioners

6.1.1 Block Diagonal Variation

By combining the Schur complement approximation $\mathcal{S}_1$ with the block diagonal forward problem approximation $\widehat{\mathcal{K}}$ we obtain a new Schur complement approximation $\widehat{\mathcal{S}}_1$ which is given by

$$\widehat{\mathcal{S}}_1 := \frac{1}{\tau}\widehat{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\widehat{\mathcal{K}}^T. \tag{6.8}$$

This approach is cheapest to apply because it will require fewer matrix-vector multiplications than $\mathcal{S}$ as the subdiagonal blocks of $\mathcal{K}$ are not used. However more importantly, $\widehat{\mathcal{S}}_1$ is completely block diagonal and could be computed on ℓ separate processors. This parallelism over time could result in a significant speed-up as, theoretically, $\widehat{\mathcal{S}}_1$ could be applied ℓ times faster than $\mathcal{S}_1$.

It remains to be shown, however, that this approximation yields an effective preconditioner in the context of the heat control problem. We showed in Theorem 5.2 that the eigenvalues of the preconditioned system will be bounded provided that the Schur complement approximation is spectrally close to $\mathcal{S}$. In order to demonstrate this, we wish to bound the eigenvalues of $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$. To do this we consider the Rayleigh quotient

$$\widehat{R}_1 := \frac{\mathbf{v}^T \mathcal{S} \mathbf{v}}{\mathbf{v}^T \widehat{\mathcal{S}}_1 \mathbf{v}} = \frac{\mathbf{v}^T \left(\frac{1}{\tau} \mathcal{K} \mathcal{M}_{1/2}^{-1} \mathcal{K}^T + \frac{\tau}{\beta} \mathcal{M} \mathcal{M}_{1/2}^{-1} \mathcal{M} \right) \mathbf{v}}{\mathbf{v}^T \frac{1}{\tau} \widehat{\mathcal{K}} \mathcal{M}_{1/2}^{-1} \widehat{\mathcal{K}}^T \mathbf{v}}. \tag{6.9}$$

If we define,

$$\widehat{\Sigma} := \begin{bmatrix} \mathbf{0} & & & \\ M & \mathbf{0} & & \\ & \ddots & \ddots & \\ & & M & \mathbf{0} \end{bmatrix}, \tag{6.10}$$

so that $\mathcal{K} = \widehat{\mathcal{K}} - \widehat{\Sigma}$ and also let

$$\begin{aligned} \mathbf{a} &:= \mathcal{M}_{1/2}^{-1/2} \mathcal{K}^T \mathbf{v}, & \mathbf{b} &:= \frac{\tau}{\sqrt{\beta}} \mathcal{M}_{1/2}^{-1/2} \mathcal{M}^T \mathbf{v}, \\ \widehat{\mathbf{c}} &:= \mathcal{M}_{1/2}^{-1/2} \widehat{\Sigma}^T \mathbf{v}, & \widehat{\mathbf{d}} &:= \mathcal{M}_{1/2}^{-1/2} \widehat{\mathcal{K}}^T \mathbf{v}, \end{aligned} \tag{6.11}$$

then we can write (6.9) in two ways as

$$\widehat{R}_1 = \frac{\mathbf{a}^T \mathbf{a} + \mathbf{b}^T \mathbf{b}}{(\mathbf{a} + \widehat{\mathbf{c}})^T (\mathbf{a} + \widehat{\mathbf{c}})} \quad \text{or} \quad \widehat{R}_1 = \frac{(\widehat{\mathbf{d}} - \widehat{\mathbf{c}})^T (\widehat{\mathbf{d}} - \widehat{\mathbf{c}}) + \mathbf{b}^T \mathbf{b}}{\widehat{\mathbf{d}}^T \widehat{\mathbf{d}}}. \tag{6.12}$$

Due to the similarity between $\mathcal{M}$ and $\mathcal{M}_{1/2}$ as before we define

$$\Delta := \mathcal{M} \mathcal{M}_{1/2}^{-1} = blkdiag(\Delta_1, \Delta_2, \ldots, \Delta_\ell), \tag{6.13}$$

where $\Delta_i = \alpha_i I_n$, with $\alpha_1 = \alpha_\ell = 2$, and $\alpha_i = 1$ for $i = 2, \ldots, \ell - 1$.

In order to construct overall bounds for $\widehat{R}_1$, we first require several inequalities.

Lemma 6.1. *For* $\mathbf{b}$ *and* $\widehat{\mathbf{c}}$ *as defined in* (6.11) *and the constants* c_1 *and* c_2 *as described in* (5.46)*, then*

$$\widehat{\mathbf{c}}^T \widehat{\mathbf{c}} \le C \frac{\beta}{\tau^2} \mathbf{b}^T \mathbf{b}, \tag{6.14}$$

where $C = \frac{2c_2}{c_1}$.

Proof. By definition

$$\frac{\widehat{\mathbf{c}}^T \widehat{\mathbf{c}}}{\frac{\beta}{\tau^2} \mathbf{b}^T \mathbf{b}} = \frac{\mathbf{v}^T \widehat{\Sigma} \mathcal{M}_{1/2}^{-1} \widehat{\Sigma}^T \mathbf{v}}{\mathbf{v}^T \mathcal{M} \mathcal{M}_{1/2}^{-1} \mathcal{M} \mathbf{v}},$$

and by examining each of these matrices we find that

$$\widehat{\Sigma}\mathcal{M}_{1/2}^{-1}\widehat{\Sigma}^T = blkdiag(\mathbf{0}, \Delta_1 M, \ldots, \Delta_{\ell-1} M)$$

and

$$\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M} = blkdiag(\Delta_1 M, \Delta_2 M, \ldots, \Delta_\ell M).$$

As both of these matrices are block diagonal, their eigenvalues will consist of the eigenvalues of each of their diagonal blocks. Using bounds for the eigenvalues of the Galerkin finite element matrices presented in (5.46), we have

$$\frac{\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}}{\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}} \leq \frac{\lambda_{max}(\widehat{\Sigma}\mathcal{M}_{1/2}^{-1}\widehat{\Sigma}^T)}{\lambda_{min}(\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M})} \leq \frac{\alpha_{max}c_2h^d}{\alpha_{min}c_1h^d} = \frac{2c_2}{c_1} := C, \tag{6.15}$$

by noting that $\alpha_{max} = 2$ and $\alpha_{min} = 1$ which proves the result. □

Lemma 6.2. *For* $\mathbf{b}$ *and* $\widehat{\mathbf{d}}$ *as defined in* (6.11) *and the constants* c_1, c_2, *and* d_1 *as described in* (5.46) *and* (5.47)*, then*

$$\frac{\mathbf{b}^T\mathbf{b}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} \leq \frac{\tau^2}{\beta}\frac{\widetilde{C}}{(c_1+\tau d_1)^2}, \tag{6.16}$$

where $\widetilde{C} = \frac{c_2^3}{c_1}$.

Proof. From the definitions, and by letting $\mathbf{y} = \mathcal{M}\mathbf{v}$ and $\mathbf{w} = \widehat{\mathcal{K}}\mathbf{v}$ we have,

$$\frac{\mathbf{b}^T\mathbf{b}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} = \frac{\tau^2}{\beta}\frac{\mathbf{v}^T\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\mathbf{v}}{\mathbf{v}^T\widehat{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\widehat{\mathcal{K}}\mathbf{v}} = \frac{\tau^2}{\beta}\frac{\mathbf{y}^T\mathcal{M}_{1/2}^{-1}\mathbf{y}}{\mathbf{y}^T\mathcal{M}^{-2}\mathbf{y}}\frac{\mathbf{w}^T\widehat{\mathcal{K}}^{-2}\mathbf{w}}{\mathbf{w}^T\mathcal{M}_{1/2}^{-1}\mathbf{w}}.$$

Once again, we only have block diagonal matrices to consider, so we obtain

$$\begin{aligned}\frac{\mathbf{b}^T\mathbf{b}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} &\leq \frac{\tau^2}{\beta}\frac{\lambda_{max}(\mathcal{M}_{1/2}^{-1})}{\lambda_{min}(\mathcal{M}^{-2})}\frac{\lambda_{max}(\widehat{\mathcal{K}}^{-2})}{\lambda_{min}(\mathcal{M}_{1/2}^{-1})} = \frac{\tau^2}{\beta}\frac{\lambda_{max}(\mathcal{M}_{1/2})}{\lambda_{min}(\mathcal{M}_{1/2})}\frac{\lambda_{max}(\mathcal{M}^2)}{\lambda_{min}(\widehat{\mathcal{K}}^2)} \\ &\leq \frac{\tau^2}{\beta}\frac{c_2h^d}{c_1h^d}\frac{(c_2h^d)^2}{((c_1+\tau d_1)h^d)^2} = \frac{\tau^2}{\beta}\frac{\widetilde{C}}{(c_1+\tau d_1)^2},\end{aligned}$$

which gives the result. □

We are now in a position to provide full bounds for the Rayleigh quotient $\widehat{R}_1$

from (6.9).

Theorem 6.1. *Let* $\widehat{R}_1$ *be as defined in* (6.9) *and the constants* c_1, c_2, *and* d_1 *as described in* (5.46) *and* (5.47). *If* $C\frac{\beta}{\tau^2} < 1$, *then*

$$\frac{1}{2} < \widehat{R}_1 \leq 2 + \frac{\widetilde{C}}{(c_1 + \tau d_1)^2}\left(2C + \frac{\tau^2}{\beta}\right), \tag{6.17}$$

otherwise,

$$\frac{1}{2C}\frac{\tau^2}{\beta} \leq \widehat{R}_1 \leq 2 + \frac{\widetilde{C}}{(c_1 + \tau d_1)^2}\left(2C + \frac{\tau^2}{\beta}\right). \tag{6.18}$$

Proof. Firstly, let us consider the upper bound. From the definition of $\widehat{R}_1$, the simple quadratic inequalities described in (5.31), and the bound in Lemma 6.2 we obtain

$$\begin{aligned}
\widehat{R}_1 &= \frac{(\widehat{\mathbf{d}} - \widehat{\mathbf{c}})^T(\widehat{\mathbf{d}} - \widehat{\mathbf{c}}) + \mathbf{b}^T\mathbf{b}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} \leq \frac{2\widehat{\mathbf{d}}^T\widehat{\mathbf{d}} + 2\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} + \frac{\mathbf{b}^T\mathbf{b}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} \\
&= 2 + \frac{2\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} + \frac{\mathbf{b}^T\mathbf{b}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} \leq 2 + \left(2C\frac{\beta}{\tau^2} + 1\right)\frac{\mathbf{b}^T\mathbf{b}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}} \\
&\leq 2 + \frac{\tau^2}{\beta}\frac{\widetilde{C}}{(c_1 + \tau d_1)^2}\left(2C\frac{\beta}{\tau^2} + 1\right),
\end{aligned}$$

which gives the result. For the lower bound, we use the alternative definition of $\widehat{R}_1$ in (6.12) and the bound from Lemma 6.1 to obtain,

$$\widehat{R}_1 = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a} + \widehat{\mathbf{c}})^T(\mathbf{a} + \widehat{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a} + \widehat{\mathbf{c}}^T\widehat{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2\mathbf{a}^T\mathbf{a} + 2C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}}.$$

Now if $C\frac{\beta}{\tau^2} < 1$, then

$$\widehat{R}_1 > \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b})} = \frac{1}{2},$$

while if $C\frac{\beta}{\tau^2} \geq 1$, we have

$$\widehat{R}_1 \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2C\frac{\beta}{\tau^2}(\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b})} = \frac{1}{2C}\frac{\tau^2}{\beta},$$

which gives the final result. □

Thus, we have shown that the eigenvalues of $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$ are bounded independently

of the mesh size h and consequently independent of the spatial degrees of freedom n. Additionally, the eigenvalues are independent of the number of time-steps ℓ. The bounds do depend on the problem parameters β and τ, however, this is to be expected since the eigenvalues of $\mathcal{S}_1^{-1}\mathcal{S}$ were also dependent of these parameters. However, we note that if $\frac{\tau^2}{\beta}$ is close to one, then the eigenvalues will be bounded within constants of order one.

6.1.2 Circulant Based Variation

Analogous to the previous section, we can replace the forward problem $\mathcal{K}$ in the dropping strategy approximation with the block circulant matrix $\overline{\mathcal{K}}$ from (6.7). Using the Kronecker product method outlined in Section 4.1.1, the main work in applying this preconditioner can also be performed over ℓ separate processors, allowing significant speed-up to be achieved via parallelization.

In order to determine eigenvalue bounds for the preconditioned system using appropriate Rayleigh quotient, we first define

$$\overline{\Sigma} = \begin{bmatrix} & & M \\ & & \\ & & \end{bmatrix}, \tag{6.19}$$

so that $\overline{\mathcal{K}} = \mathcal{K} - \overline{\Sigma}$. Our new Schur complement approximation is given by

$$\overline{\mathcal{S}}_1 := \frac{1}{\tau}\overline{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\overline{\mathcal{K}}^T. \tag{6.20}$$

The eigenvalues of $\overline{\mathcal{S}}_1^{-1}\mathcal{S}$ can be examined by bounding the generalized Rayleigh quotient given by

$$\overline{R}_1 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{S}}_1\mathbf{v}} = \frac{\mathbf{v}^T(\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau^2}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M})\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\overline{\mathcal{K}}^T\mathbf{v}}, \tag{6.21}$$

and by letting

$$\bar{\mathbf{c}} = \mathcal{M}_{1/2}^{-1/2}\overline{\Sigma}\mathbf{v}, \tag{6.22}$$

we obtain

$$\overline{R}_1 := \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}-\bar{\mathbf{c}})^T(\mathbf{a}-\bar{\mathbf{c}})}. \tag{6.23}$$

Lemma 6.3. *For* $\mathbf{b}$ *and* $\bar{\mathbf{c}}$ *as defined in* (6.11) *and* (6.22) *respectively, then*

$$\bar{\mathbf{c}}^T\bar{\mathbf{c}} \leq C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}. \tag{6.24}$$

Proof. Similarly to the proof of Lemma 6.1, we want to look more closely at the matrices which appear in $\bar{\mathbf{c}}^T\bar{\mathbf{c}}$ and $\mathbf{b}^T\mathbf{b}$. Using the definition of Δ in (6.13) we again see that,

$$\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M} = blkdiag(\Delta_1 M, \Delta_2 M, \ldots, \Delta_\ell M)$$

however for the circulant case, we have

$$\overline{\Sigma}\mathcal{M}_{1/2}^{-1}\overline{\Sigma}^T = blkdiag(\mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}, \Delta_\ell M).$$

Since each of these matrices are block diagonal we can obtain the following bound

$$\frac{\bar{\mathbf{c}}^T\bar{\mathbf{c}}}{\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}} \leq \frac{\lambda_{max}(\overline{\Sigma}\mathcal{M}_{1/2}^{-1}\overline{\Sigma}^T)}{\lambda_{min}(\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M})} \leq \frac{\alpha_\ell c_2 h^d}{\alpha_{min} c_1 h^d} = \frac{2c_2}{c_1} = C, \tag{6.25}$$

which proves the result. □

Theorem 6.2. *For* $\overline{R}_1$ *defined in* (6.23)*, if* $C\frac{\beta}{\tau^2} < 1$ *then*

$$\frac{1}{2} \leq \overline{R}_1, \tag{6.26}$$

and otherwise,

$$\frac{1}{2\left(C\frac{\beta}{\tau^2}\right)} \leq \overline{R}_1. \tag{6.27}$$

Proof. Using Lemma 6.3 we have that,

$$\overline{R}_1 = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}-\bar{\mathbf{c}})^T(\mathbf{a}-\bar{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a} + \bar{\mathbf{c}}^T\bar{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a} + C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b})}.$$

Now if $C\frac{\beta}{\tau^2} < 1$,

$$\overline{R}_1 > \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b})} = \frac{1}{2}, \tag{6.28}$$

while if $C\frac{\beta}{\tau^2} \geq 1$, then

$$\overline{R}_1 \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2C\frac{\beta}{\tau^2}(\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b})} = \frac{1}{2C\frac{\beta}{\tau^2}}, \tag{6.29}$$

which gives the result. □

We have thus obtained a lower bound for $\overline{R}_1$; it is more challenging to obtain upper bounds and our arguments here are more heuristic. We first make the following observations. Since

$$\widehat{R}_1 = \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{S}}_1\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_1\mathbf{v}}\frac{\mathbf{v}^T\mathcal{S}_1\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{S}}_1\mathbf{v}}, \tag{6.30}$$

and we obtained bounds for $\dfrac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_1\mathbf{v}}$ in Theorem 5.4, we only need to look specifically at $\dfrac{\mathbf{v}^T\mathcal{S}_1\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{S}}_1\mathbf{v}}$. Now,

$$\frac{\mathbf{v}^T\mathcal{S}_1\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{S}}_1\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\overline{\mathcal{K}}^T\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{K}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{K}}\overline{\mathcal{K}}^T\mathbf{v}}\frac{\mathbf{w}^T\mathcal{M}_{1/2}^{-1}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}\frac{\mathbf{y}^T\mathbf{y}}{\mathbf{y}^T\mathcal{M}_{1/2}^{-1}\mathbf{v}},$$

where $\mathbf{w} = \mathcal{K}^T\mathbf{v}$ and $\mathbf{y} = \overline{\mathcal{K}}^T\mathbf{v}$. We note that $\dfrac{\mathbf{v}^T\mathcal{K}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{K}}\overline{\mathcal{K}}^T\mathbf{v}}$ will be bounded by the extremal eigenvalues of $(\overline{\mathcal{K}}^{-1}\mathcal{K})^T(\overline{\mathcal{K}}^{-1}\mathcal{K})$ which were examined in Section 4.3. Although eigenvalues bounds were not obtained, we did determine that $n(\ell-2)$ eigenvalues are equal to 1, $n\ell$ are less than 1 and ℓn are greater than 1. This does not imply that the same eigenvalue distribution holds for $\overline{\mathcal{S}}^{-1}\mathcal{S}$, however it does give us some indication that there may only be a relatively small number of outlying eigenvalues. This is confirmed numerically in Figure 6.1.

Now if we return to the original quotient,

$$\frac{\mathbf{v}^T\mathcal{S}_1\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{S}}_1\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\bar{\mathcal{K}}^T\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\left(\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T - \bar{\Sigma}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T - \mathcal{K}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T + \bar{\Sigma}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T\right)\mathbf{v}}$$
$$= \frac{1}{1 + \dfrac{\mathbf{v}^T\left(-\bar{\Sigma}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T - \mathcal{K}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T + \bar{\Sigma}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T\right)\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\bar{\mathcal{K}}^T\mathbf{v}}}.$$

We now need to examine the remaining Rayleigh quotient. The numerator contains three matrices given by

$$\bar{\Sigma}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T = \begin{bmatrix} & \Delta_\ell(M+\tau K) \\ & \end{bmatrix}, \quad \mathcal{K}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T = \begin{bmatrix} & \\ \Delta_\ell(M+\tau K) & \end{bmatrix},$$

and

$$\bar{\Sigma}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T = \begin{bmatrix} \Delta_\ell M & \\ & \end{bmatrix}.$$

Since each of these matrices only contains one non-zero block, the Rayleigh quotient can be simplified by writing the numerator as

$$\begin{aligned}
&\mathbf{v}^T\left(-\bar{\Sigma}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T - \mathcal{K}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T + \bar{\Sigma}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T\right)\mathbf{v} \\
&\quad = -\mathbf{v}_1^T\Delta_\ell(M+\tau K)\mathbf{v}_\ell - \mathbf{v}_\ell^T\Delta_\ell(M+\tau K)\mathbf{v}_1 + \mathbf{v}_1^T\Delta_\ell M\mathbf{v}_1 \\
&\quad = (\mathbf{v}_1-\mathbf{v}_\ell)^T\Delta_\ell(M+\tau K)(\mathbf{v}_1-\mathbf{v}_\ell) - \mathbf{v}_1^T\Delta_\ell(M+\tau K)\mathbf{v}_1 \\
&\qquad - \mathbf{v}_\ell^T\Delta_\ell(M+\tau K)\mathbf{v}_\ell + \mathbf{v}_1^T\Delta_\ell M\mathbf{v}_1,
\end{aligned} \tag{6.31}$$

which is the sum of quadratic forms, so each term can be bounded by its eigenvalues.

Thus, we have

$$\begin{aligned}
&\frac{\mathbf{v}^T\left(-\bar{\Sigma}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T - \mathcal{K}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T + \bar{\Sigma}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T\right)\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \\
&\quad \geq \lambda_{min}(\Delta_\ell(M+\tau K)) - 2\lambda_{max}(\Delta_\ell(M+\tau K)) + \lambda_{min}(\Delta_\ell M) \\
&\quad \geq 2\left((c_1+\tau d_1)h^d - 2(c_2h^d+\tau d_2) + c_1h^d\right), \qquad (6.32)
\end{aligned}$$

noting that $\alpha_\ell = 2$. If we now look at the denominator and let $\mathbf{w} = \mathcal{K}^T\mathbf{v}$, we can write

$$\frac{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{K}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\mathbf{v}}\frac{\mathbf{w}^T\mathcal{M}_{1/2}^{-1}\mathbf{w}}{\mathbf{w}^T\mathbf{w}} \leq \lambda_{max}(\mathcal{K}\mathcal{K}^T)\lambda_{max}(\mathcal{M}_{1/2}^{-1}).$$

From (5.45) we have upper bounds on $\lambda(\mathcal{K}\mathcal{K}^T)$ and since $\mathcal{M}_{1/2}^{-1}$ is block diagonal we can use the bounds in (5.46) to bound $\lambda_{max}(\mathcal{M}_{1/2}^{-1})$. Thus we have

$$\frac{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}}{\mathbf{v}^T\mathbf{v}} \leq \frac{(2c_2h^d+\tau d_2)^2}{c_1h^d}, \qquad (6.33)$$

and combining (6.32) and (6.33) we have,

$$\begin{aligned}
&\frac{\mathbf{v}^T\left(-\bar{\Sigma}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T - \mathcal{K}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T + \bar{\Sigma}\mathcal{M}_{1/2}^{-1}\bar{\Sigma}^T\right)\mathbf{v}}{\mathbf{v}^T\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T\mathbf{v}} \\
&\geq \quad \frac{2\left((c_1+\tau d_1)h^d - 2(c_2h^d+\tau d_2) + c_1h^d\right)c_1h^d}{(2c_2h^d+\tau d_2)^2}.
\end{aligned}$$

We can see that this bound is not h independent, however to highest order both the numerator and denominator are $O(h^{2d})$, hence overall this term should balance at $O(1)$. Indeed this implies that $\dfrac{\mathbf{v}^T\mathcal{S}_1\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{S}}_1\mathbf{v}}$ should be bounded from above by a constant of order 1. Combining with the result from Theorem 5.4 that

$$\frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_1\mathbf{v}} \leq 1 + \frac{\tau^2}{\beta}\frac{\widetilde{C}}{(c_1+\tau d_1-c_2)^2}$$

we therefore predict that the maximal eigenvalues of $\bar{\mathcal{S}}_1^{-1}\mathcal{S}$ will behave approximately like $\frac{\tau^2}{\beta}$. This will be numerically verified later in Section 6.3.

In order to achieve eigenvalues clustered about 1, from the lower bound described

in Theorem 6.2 we know that we would ideally like $\frac{\beta}{\tau^2}$ to be a small value. However, we predict the upper bound will behave in the opposite way and therefore we ideally would like $\frac{\beta}{\tau^2}$ to take a large value to ensure that the largest eigenvalue remains close to 1. These contrasting requirements imply that, regardless of parameter values, we may not be able to achieve tightly clustered eigenvalues for the circulant variation to the dropping preconditioner. However, as we previously stated, we predict that only a small number of eigenvalues may take large values, so we hope that performance does not greatly deteriorate.

6.2 Matching Strategy Based Preconditioners

6.2.1 Block Diagonal Variation

Just as we did for the dropping strategy, we can define new variations to the matching strategy preconditioners. For the block diagonal approach we define

$$\widehat{\mathcal{S}}_2 := \frac{1}{\tau}\left(\widehat{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\widehat{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T. \tag{6.34}$$

We note that this approximation was suggested as a possible parallelizable approach in [86] but no analysis or numerical experiments were conducted. Once again the eigenvalues of $\widehat{\mathcal{S}}_2^{-1}\mathcal{S}$ can be bounded by examining the appropriate generalized Rayleigh quotient. Since clean eigenvalue bounds have already been obtained for $\mathcal{S}_2^{-1}\mathcal{S}$, we find that it is easiest to represent the Rayleigh quotient in a multiplicative way. Thus we let,

$$\widehat{R}_2 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\widehat{\mathcal{S}}_2\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}\frac{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}{\mathbf{v}^T\widehat{\mathcal{S}}_2\mathbf{v}} := \widehat{R}_{2A}\widehat{R}_{2B}. \tag{6.35}$$

We know that $\widehat{R}_{2A}$ is bounded between 1/2 and 1 from Theorem 5.5. Thus it only remains to determine bounds for $\widehat{R}_{2B}$. Using the values defined in (6.11) we have

$$\begin{aligned}\widehat{R}_{2B} &= \frac{\mathbf{v}^T(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M})\mathcal{M}_{1/2}^{-1}(\mathcal{K}+\frac{\tau}{\sqrt{\beta}}\mathcal{M})^T\mathbf{v}}{\mathbf{v}^T(\widehat{\mathcal{K}}+\frac{\tau}{\sqrt{\beta}}\mathcal{M})\mathcal{M}_{1/2}^{-1}(\widehat{\mathcal{K}}+\frac{\tau}{\sqrt{\beta}}\mathcal{M})^T\mathbf{v}} \\ &= \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{(\mathbf{a}+\widehat{\mathbf{c}}+\mathbf{b})^T(\mathbf{a}+\widehat{\mathbf{c}}+\mathbf{b})} = \frac{(\widehat{\mathbf{d}}-\widehat{\mathbf{c}}+\mathbf{b})^T(\widehat{\mathbf{d}}-\widehat{\mathbf{c}}+\mathbf{b})}{(\widehat{\mathbf{d}}+\mathbf{b})^T(\widehat{\mathbf{d}}+\mathbf{b})}. \end{aligned} \tag{6.36}$$

The manner in which we determine bounds for $\widehat{R}_{2B}$ is very similar to the way in which bounds were obtained for $\widehat{R}_{2A}$. As such, we require the positivity of $\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a}$ as proved in Lemma 5.1 and additionally we require bounds for terms involving $\widehat{\mathbf{d}}$.

Lemma 6.4. *For* $\widehat{\mathbf{d}}$ *and* $\mathbf{b}$ *as defined in* (6.11)*, then*

$$\widehat{\mathbf{d}}^T\widehat{\mathbf{d}} \le \frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}, \quad \textit{and} \quad \widehat{\mathbf{d}}^T\mathbf{b}+\mathbf{b}^T\widehat{\mathbf{d}} \le 2\frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b}.$$

Proof. Firstly,

$$\begin{aligned}\widehat{\mathbf{d}}^T\widehat{\mathbf{d}} &= \mathbf{v}^T\widehat{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\widehat{\mathcal{K}}^T\mathbf{v} = \sum_{i=1}^{\ell}\mathbf{v}_i^T(M+\tau K)\Delta_i M^{-1}(M+\tau K)^T\mathbf{v}_i \\ &= \sum_{i=1}^{\ell}\mathbf{v}_i^T M\Delta_i\mathbf{v}_i + \sum_{i=1}^{\ell}\mathbf{v}_i^T(\tau K\Delta_i+\tau\Delta_i K^T+\tau^2 KM^{-1}K^T)\mathbf{v}_i \\ &= \frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}+\sum_{i=1}^{\ell}\mathbf{v}_i^T(\tau K\Delta_i+\tau\Delta_i K^T+\tau^2 KM^{-1}K^T)\mathbf{v}_i \ge \frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b},\end{aligned}$$

since $K\Delta_i, \Delta_i K^T$, and $KM^{-1}K^T$ are all positive definite, so the final summation term will always be positive. For the second bound we only examine $\widehat{\mathbf{d}}^T\mathbf{b}$ as Δ and $\widehat{\mathcal{K}}$ are both block diagonal and symmetric, we have that $\widehat{\mathbf{d}}^T\mathbf{b} = \mathbf{v}^T\widehat{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\mathcal{M}\mathbf{v} = \mathbf{v}^T\widehat{\mathcal{K}}\Delta\mathbf{v} = \mathbf{v}^T\Delta\widehat{\mathcal{K}}^T\mathbf{v} = \mathbf{b}^T\widehat{\mathbf{d}}$. Using the positive definiteness of $K\Delta_i$ we have,

$$\begin{aligned}\widehat{\mathbf{d}}^T\mathbf{b} &= \mathbf{v}^T\widehat{\mathcal{K}}\Delta\mathbf{v} = \sum_{i=1}^{\ell}\mathbf{v}_i^T(M+\tau K)\Delta_i\mathbf{v}_i = \sum_{i=1}^{\ell}\mathbf{v}_i^T M\Delta_i\mathbf{v}_i + \sum_{i=1}^{\ell}\mathbf{v}_i^T\tau K\Delta_i\mathbf{v}_i \\ &= \frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b}+\sum_{i=1}^{\ell}\mathbf{v}_i^T\tau K\Delta_i\mathbf{v}_i \ge \frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b},\end{aligned}$$

which gives the result. □

We are now in a position to state complete bounds for the term $\widehat{R}_{2B}$ and subsequently $\widehat{R}_2$ as a whole.

Lemma 6.5. *For $\widehat{R}_{2B}$ as defined in* (6.35)*, then*

$$\frac{1}{2\left(1+C\frac{\beta}{\tau^2}\right)} \leq R_{2B} < 2+2C. \tag{6.37}$$

Proof. Concentrating first on the upper bound and using the simple quadratic inequalities described in (5.31) and the results from Lemmas 6.1 and 6.2 we obtain,

$$\begin{aligned}
\widehat{R}_{2B} &= \frac{(\widehat{\mathbf{d}}+\mathbf{b}-\widehat{\mathbf{c}})^T(\widehat{\mathbf{d}}+\mathbf{b}-\widehat{\mathbf{c}})}{(\widehat{\mathbf{d}}+\mathbf{b})^T(\widehat{\mathbf{d}}+\mathbf{b})} \\
&\leq \frac{2(\widehat{\mathbf{d}}+\mathbf{b})^T(\widehat{\mathbf{d}}+\mathbf{b})+2\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}}{(\widehat{\mathbf{d}}+\mathbf{b})^T(\widehat{\mathbf{d}}+\mathbf{b})} = 2+\frac{2\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}+\mathbf{b}^T\widehat{\mathbf{d}}+\widehat{\mathbf{d}}^T\mathbf{b}+\mathbf{b}^T\mathbf{b}} \\
&\leq 2+\frac{2C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}}{\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}+2\frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b}+\mathbf{b}^T\mathbf{b}} = 2+\frac{2C\frac{\beta}{\tau^2}}{(\frac{\sqrt{\beta}}{\tau}+1)^2} \\
&< 2+2C,
\end{aligned}$$

since $\frac{\frac{\beta}{\tau^2}}{(\frac{\sqrt{\beta}}{\tau}+1)^2} < 1$. In order to obtain the lower bound, we define the quotient as

$$\begin{aligned}
\widehat{R}_{2B} &= \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{(\mathbf{a}+\widehat{\mathbf{c}}+\mathbf{b})^T(\mathbf{a}+\widehat{\mathbf{c}}+\mathbf{b})} \geq \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{2\left((\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})+\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}\right)} \\
&\geq \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{2\left((\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})+\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}\right)} = \frac{1}{2\left(1+\frac{\beta}{\tau^2}\frac{\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}\right)}
\end{aligned}$$

which gives the result because by Lemma 5.1 we have that $\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a}$ is positive and therefore $\frac{\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} \leq 1$. □

Theorem 6.3. *The Rayleigh quotient $\widehat{R}_2$ defined in* (6.35) *is bounded by,*

$$\frac{1}{4\left(1+C\frac{\beta}{\tau^2}\right)} \leq \widehat{R}_2 < 2+2C. \tag{6.38}$$

Proof. We know that $\widehat{R}_2 = \widehat{R}_{2A}\widehat{R}_{2B}$ so if we have that $\widehat{R}_{2A(min)} \leq \widehat{R}_{2A} \leq \widehat{R}_{2A(max)}$ and

$\widehat{R}_{2B(min)} \leq \widehat{R}_{2B} \leq \widehat{R}_{2B(max)}$ then it follows that

$$\widehat{R}_{2A(min)}\widehat{R}_{2B(min)} \leq \widehat{R}_2 \leq \widehat{R}_{2A(max)}\widehat{R}_{2B(max)}.$$

By using the bounds obtained in Theorem 5.5 and Lemma 6.5, we obtain the extra factor of $1/2$ on the lower bound from the bounds on $\widehat{R}_{2A}$ and thus achieve the result. □

We have shown that for the block diagonal variation of the matching strategy approximation $\widehat{\mathcal{S}}_2$, the eigenvalues of $\widehat{\mathcal{S}}_2^{-1}\mathcal{S}$ have an upper bound independent of all problem parameters. A lower bound is also proved, which is independent of the mesh parameter h but does depend on the regularization parameter β and the time-step size τ.

6.2.2 Circulant Based Variation

Moving to the circulant variation we define the new Schur complement approximation as

$$\bar{\mathcal{S}}_2 := \frac{1}{\tau}\left(\overline{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\overline{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T. \tag{6.39}$$

Throughout our analysis of the matching strategy preconditioner, we have been able to provide eigenvalue bounds by simply using quadratic inequalities. We can continue this procedure even for the circulant variation, even though this was not possible for the dropping strategy. In this case, in addition to requiring the positivity of $\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a}$ we also require $\bar{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\bar{\mathbf{d}}$ to be positive, where

$$\bar{\mathbf{d}} = \mathcal{M}_{1/2}^{-1/2}\overline{\mathcal{K}}^T\mathbf{v}. \tag{6.40}$$

.

Lemma 6.6. *For* $\bar{\mathbf{d}} = \mathcal{M}_{1/2}^{-1/2}\overline{\mathcal{K}}^T\mathbf{v}$ *and* $\mathbf{b} = \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1/2}^{1/2}\mathcal{M}^T\mathbf{v}$*, then*

$$\bar{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\bar{\mathbf{d}} > 0. \tag{6.41}$$

Proof. As defined in (6.13) we have that $\Delta = \mathcal{M}_{1/2}^{-1}\mathcal{M}$ so $\bar{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\bar{\mathbf{d}} = \mathbf{v}^T(\bar{\mathcal{K}}\Delta + \Delta\bar{\mathcal{K}}^T)\mathbf{v}$. Letting $L = M + \tau K$ we have

$$\bar{\mathcal{K}}\Delta + \Delta\bar{\mathcal{K}} = \begin{bmatrix} 2L\Delta_1 & -\Delta_1 M & & & -M\Delta_\ell \\ -\Delta_1 M & 2L\Delta_2 & -\Delta_2 M & & \\ & \ddots & \ddots & \ddots & \\ & & -\Delta_{\ell-2}M & 2L\Delta_{\ell-1} & -M\Delta_{\ell-1} \\ -\Delta_\ell M & & & -\Delta_{\ell-1}M & 2L\Delta_\ell \end{bmatrix}. \tag{6.42}$$

Noting that $\Delta_1 = \Delta_\ell$ since $\alpha_1 = \alpha_\ell$ we can write $\bar{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\bar{\mathbf{d}}$ in the following way,

$$\begin{aligned} \mathbf{v}^T(\bar{\mathcal{K}}\Delta + \Delta\bar{\mathcal{K}}^T)\mathbf{v} &= 2\sum_{i=1}^{\ell}\mathbf{v}_i^T L\Delta_i\mathbf{v}_i - \sum_{i=1}^{\ell-1}\mathbf{v}_{i+1}^T M\Delta_i\mathbf{v}_i - \sum_{i=1}^{\ell-1}\mathbf{v}_i^T\Delta_i M\mathbf{v}_{i+1} \\ &\quad - \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_1 - \mathbf{v}_1^T M\Delta_\ell\mathbf{v}_\ell \\ &= 2\tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i K\mathbf{v}_i + \sum_{i=1}^{\ell-1}(\mathbf{v}_i - \mathbf{v}_{i+1})^T M\Delta_i(\mathbf{v}_i - \mathbf{v}_{i+1}) \\ &\quad + \mathbf{v}_1^T\Delta_1 M\mathbf{v}_1 + \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_\ell - \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_1 - \mathbf{v}_1^T\Delta_\ell M\mathbf{v}_\ell \\ &= 2\tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i K\mathbf{v}_i + \sum_{i=1}^{\ell-1}(\mathbf{v}_i - \mathbf{v}_{i+1})^T M\Delta_i(\mathbf{v}_i - \mathbf{v}_{i+1}) \\ &\quad + \mathbf{v}_1^T\Delta_1 M\mathbf{v}_1 + (\mathbf{v}_1 - \mathbf{v}_\ell)^T\Delta_\ell M(\mathbf{v}_1 - \mathbf{v}_\ell). \end{aligned}$$

Since this is the sum of positive definite terms we therefore have the result. □

With this additional positive definiteness result, we can provide bounds for the Rayleigh quotient $\bar{R}_2$, which is defined as

$$\bar{R}_2 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{S}}_2\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}\frac{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{S}}_2\mathbf{v}} := \bar{R}_{2A}\bar{R}_{2B}. \tag{6.43}$$

Theorem 6.4. *For $\bar{R}_2$ as defined in* (6.43),

$$\frac{1}{4(1+\frac{\beta}{\tau^2})} \le \bar{R}_2 \le 2\left(1 + \frac{\beta}{\tau^2}\right). \tag{6.44}$$

Proof. As in the block diagonal case, we only need to look at the term $\overline{R}_{2B}$ since $\overline{R}_{2A}$ is bounded between 1/2 and 1 from Theorem 5.5. We can write $\overline{R}_{2B}$ in two ways as

$$\overline{R}_{2B} = \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{(\mathbf{a}+\mathbf{b}+\bar{\mathbf{c}})^T(\mathbf{a}+\mathbf{b}+\bar{\mathbf{c}})} \quad \text{or} \quad \overline{R}_{2B} = \frac{(\bar{\mathbf{d}}+\bar{\mathbf{c}}+\mathbf{b})^T(\bar{\mathbf{d}}+\bar{\mathbf{c}}+\mathbf{b})}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})}, \tag{6.45}$$

with $\mathbf{a}$ and $\mathbf{b}$ as defined in (6.11), $\bar{\mathbf{c}}$ in (6.22), and $\bar{\mathbf{d}}$ in (6.40). Using the first definition we obtain,

$$\begin{aligned}\overline{R}_{2B} &= \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{(\mathbf{a}+\mathbf{b}-\bar{\mathbf{c}})^T(\mathbf{a}+\mathbf{b}-\bar{\mathbf{c}})} \geq \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{2(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})+2\bar{\mathbf{c}}^T\bar{\mathbf{c}}} \\ &= \frac{1}{2(1+\frac{\bar{\mathbf{c}}^T\bar{\mathbf{c}}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})})} \geq \frac{1}{2(1+C\frac{\beta}{\tau^2}\frac{\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})})},\end{aligned}$$

and from Lemma 5.1 we have that $\mathbf{a}^T\mathbf{b}+\mathbf{b}^T\mathbf{a} > 0$ and therefore $\frac{\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^2(\mathbf{a}+\mathbf{b})} \leq 1$, which gives the lower bound result. For the upper bound we write,

$$\begin{aligned}\overline{R}_{2B} &= \frac{(\bar{\mathbf{d}}+\mathbf{b}-\bar{\mathbf{c}})^T(\bar{\mathbf{d}}+\mathbf{b}-\bar{\mathbf{c}})}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \leq \frac{2(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})+2\bar{\mathbf{c}}^T\bar{\mathbf{c}}}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \\ &= 2+\frac{2\bar{\mathbf{c}}^T\bar{\mathbf{c}}}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \leq 2+\frac{2C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \leq 2+C\frac{\beta}{\tau^2},\end{aligned}$$

by using Lemma 6.6 to know that $\dfrac{\mathbf{b}^T\mathbf{b}}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \leq 1$. Finally, multiplying each side by the bounds for $\overline{R}_{2A}$ from Theorem 5.5 the extra factor of 1/2 in the lower bound is obtained. □

These bounds tell us that the eigenvalues of $\overline{\mathcal{S}}_2^{-1}\mathcal{S}$ will be tightly clustered about 1 provided that $\frac{\beta}{\tau^2}$ is small. This is precisely the region for which the matching strategy was designed. Thus, although this variant is no longer completely robust to changes in β and τ as the approximation $\mathcal{S}_2$ was, we do predict good performance in the regions where $\mathcal{S}_1$ and associated variants, perform poorly.

6.3 Summary of Eigenvalue Bounds

The four preconditioners developed throughout this chapter each have their own advantages and disadvantages. In order to clarify this, the bounds proved in the previous sections are summarized in Table 6.1. In brackets, below each bound is shown the value for $\frac{\beta}{\tau^2}$ which results in eigenvalues tightly clustered around 1. We see that for the dropping strategy approximation $\mathcal{S}_1$, the lower and upper bounds for both variants have competing requirements for the value of $\frac{\beta}{\tau^2}$ in order to achieve clustered eigenvalues. Thus, it is likely that the eigenvalues will be most well clustered when $\frac{\beta}{\tau^2} = 1$, which is a particularly strict requirement.

We recall that $\mathcal{S}_1$ was most effective as an approximation when $\frac{\beta}{\tau^2}$ was large. However, when $\frac{\beta}{\tau^2}$ is small, and thus $\frac{\tau^2}{\beta}$ is large, the second term in the Schur complement $\mathcal{S} = \frac{1}{\tau}\left(\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K} + \frac{\tau^2}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}\right)$, becomes dominant. Thus in this regime, any variations we make to $\mathcal{K}$ in the first term become less significant overall. Therefore, we might expect any $\widetilde{\mathcal{S}}$ with an approximation to $\mathcal{K}$ to perform better when $\frac{\beta}{\tau^2}$ is small. This can perhaps explain the competing constraints for our dropping strategy variations.

Analogously, while the original matching strategy performed equally well regardless of the size of $\frac{\beta}{\tau^2}$, our variations do introduce a preference for $\frac{\beta}{\tau^2}$ to be small. However, with their additional parallel capabilities, these variations can perform competitively for all values of β as will be shown in the numerical results section.

To further illustrate how spectrally close the approximations are to the original Schur complements, the eigenvalues of $\widetilde{\mathcal{S}}^{-1}\mathcal{S}$ for each approximation $\widetilde{\mathcal{S}}$ are shown in Figures 6.1 and 6.2. In each column of figures, the value of β decreases while the value of τ remains fixed at 10^{-2}. As predicted by the theoretical bounds, $\widehat{\mathcal{S}}_1$ and $\overline{\mathcal{S}}_1$ are spectrally closest to $\mathcal{S}$ when β and τ^2 are relatively well balanced. We can also see that the upper eigenvalues behave approximately like $\frac{\beta}{\tau^2}$. Furthermore, as alluded to in Section 6.1.2, only a relatively small number of eigenvalues of $\overline{\mathcal{S}}_1^{-1}\mathcal{S}_1$ approach the upper bound while most remain fairly small. We note that these figures are plotted on a log scale for clarity.

For $\widehat{\mathcal{S}}_2$ and $\overline{\mathcal{S}}_2$, as predicted we see the best behaviour when $\beta \ll 1$. Furthermore,

for very small β we see that eigenvalues are bounded between 1/2 and 1, just as the eigenvalues of $\mathcal{S}_2^{-1}\mathcal{S}$ were. Therefore, in this regime we expect we may see performance similar to $\mathcal{S}_2$ for both our variants $\widehat{\mathcal{S}}_2$ and $\bar{\mathcal{S}}_2$. However, our variants have the potential to be parallelized over time which could allow them to gain a significant advantage over the sequentially applied $\mathcal{S}_2$.

These eigenvalue plots illustrate the how spectrally close our proposed preconditioners are to the original preconditioning strategy they are based on and confirm the theoretically predicted bounds. Additionally, they show clustering of the eigenvalues. For example, for $\bar{\mathcal{S}}_1$ we see that only a small number of eigenvalues are very large and therefore may not dramatically reduce the performance of the preconditioner.

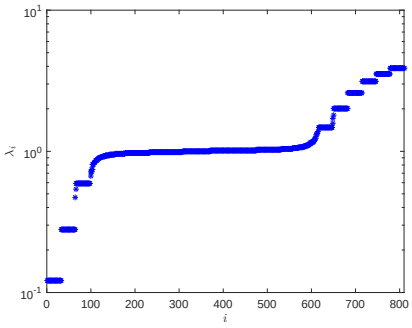
Table 6.1: Summary of eigenvalue bounds for $\widetilde{\mathcal{S}}^{-1}\mathcal{S}$ for each considered Schur complement approximation $\widetilde{\mathcal{S}}$.

	Block Diagonal ($\widehat{\mathcal{S}}$)		Circulant ($\bar{\mathcal{S}}$)	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound
$\mathcal{S}_1$	$\frac{1}{2}$ or $\frac{1}{2C\frac{\beta}{\tau^2}}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	$2 + \frac{\widetilde{C}}{(c_1+\tau d_1)^2}\left(2C + \frac{\tau^2}{\beta}\right)$ $\left(\frac{\beta}{\tau^2} \gg 1\right)$	$\frac{1}{2}$ or $\frac{1}{2C\frac{\beta}{\tau^2}}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	$1 + O(\frac{\tau^2}{\beta})$ $\left(\frac{\beta}{\tau^2} \gg 1\right)$
$\mathcal{S}_2$	$\frac{1}{4\left(1+C\frac{\beta}{\tau^2}\right)}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	$2 + 2C$ -	$\frac{1}{4\left(1+C\frac{\beta}{\tau^2}\right)}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	$2 + 2C\frac{\tau^2}{\beta}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$

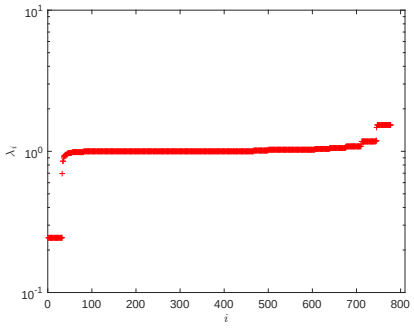
6.4 Numerical Results

In this section, we provide numerical results to demonstrate the effectiveness of all of the considered preconditioners. All the finite element matrices were computed using the IFISS [26, 27, 103] framework.

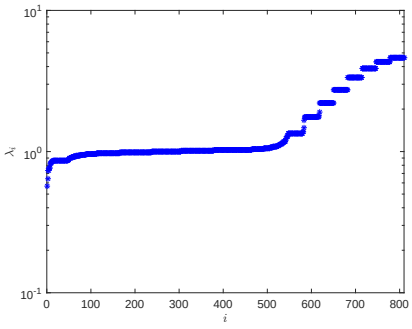
All of the preconditioners used are of the form described in (6.2) with each different version of the Schur complement approximations $\widetilde{\mathcal{S}}$. Thus, all preconditioners have common $(1,1)$ and $(2,2)$ blocks which are formed solely of mass matrices. These are approximately inverted by 10 iterations of the linear Chebyshev semi-iteration as described in Section 2.2.1.1. To compute the multigrid approximations for $\mathcal{S}_1, \mathcal{S}_2, \widehat{\mathcal{S}}_1$,



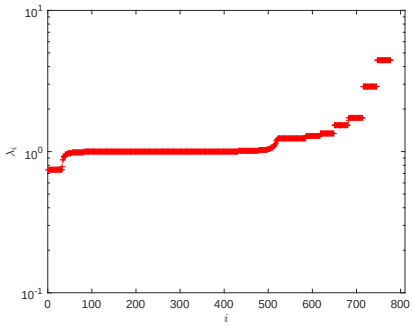
(a) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



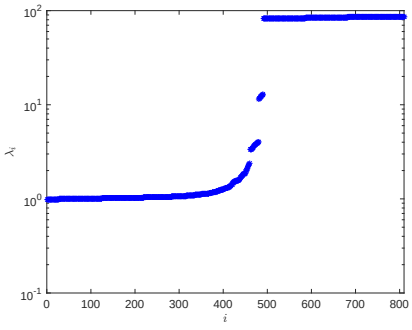
(b) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



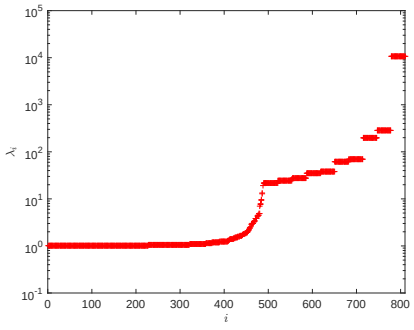
(c) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



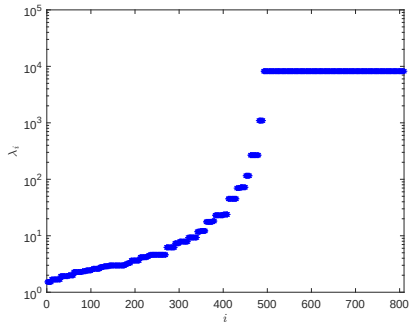
(d) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



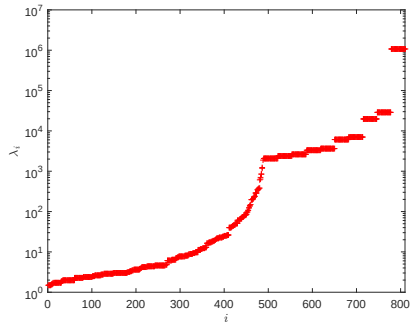
(e) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$



(f) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$

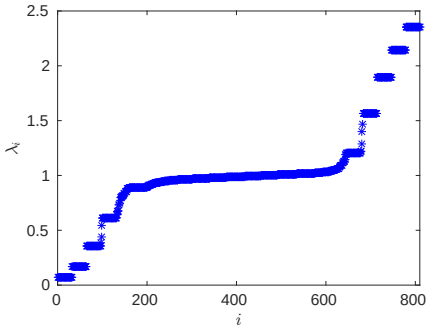


(g) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$

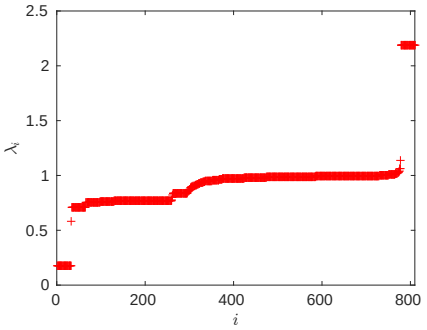


(h) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$

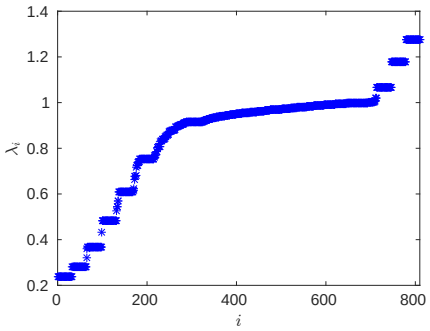
Figure 6.1: Eigenvalues of $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$ and $\overline{\mathcal{S}}_2^{-1}\mathcal{S}$ for different β values and $\tau = 0.1, h = 2^{-3}$ and $\ell = 10$ (shown in log scale).



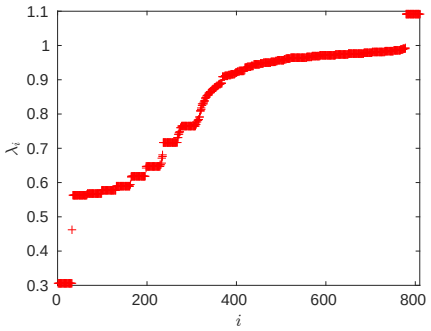
(a) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



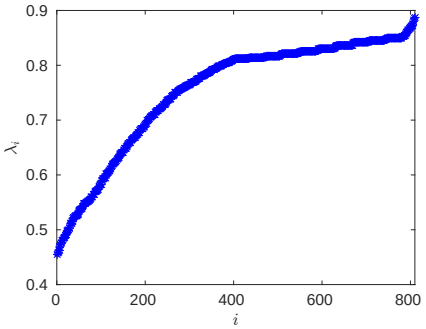
(b) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



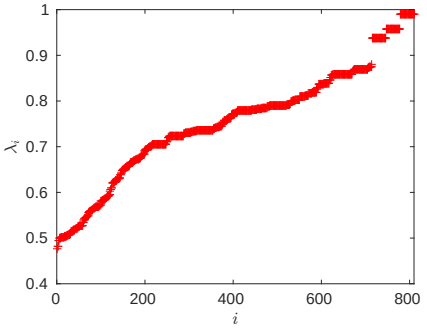
(c) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



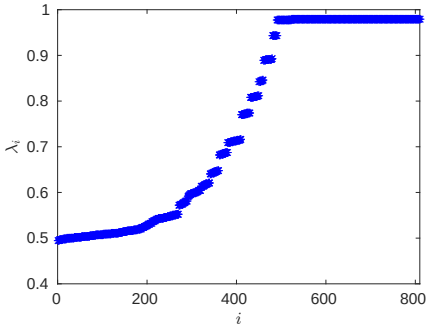
(d) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



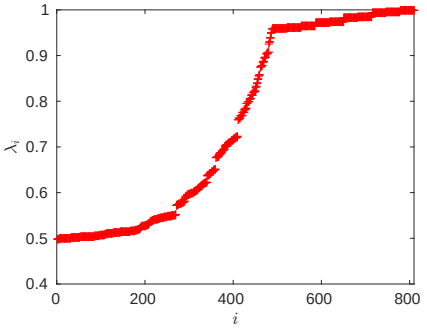
(e) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$



(f) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$



(g) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$



(h) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$

Figure 6.2: Eigenvalues of $\widehat{\mathcal{S}}_2^{-1}\mathcal{S}$ and $\overline{\mathcal{S}}_2^{-1}\mathcal{S}$ for different β and τ values with $h = 2^{-3}$ and $\ell = 10$.

and $\widehat{\mathcal{S}}_2$ we used 2 V-cycles of the Harwell Subroutine Library AMG code HSL_MI20. For the circulant preconditioner, as the matrix requiring approximation is complex, we used the algebraic multigrid code AGMG [73, 74, 77, 78]. For the AGMG preconditioner, we found that we required additional V-cycles in order to achieve sufficiently accurate approximation for our preconditioners to be effective. This is perhaps due to the matrix which is being solved being complex for which the method was not designed. Thus, for the circulant variation, the preconditioner was applied in the manner described in Chapter 4 with 5 V-cycles of AGMG to approximately invert each block of the matrix $\mathcal{G}$ defined in (4.15). We used the standard Matlab implementation of MINRES to solve the linear system and all solutions were solved to a tolerance of 10^{-4}.

The heat control problem we solved was described by

$$\begin{aligned} \min_{y,u} \quad & \int_0^T \|y-\hat{y}\|^2_{L_2(\Omega)}\,dt + \frac{\beta}{2}\int_0^T \|u\|^2_{L_2(\Omega)}\,dt \\ \text{such that} \quad & y_t - \nabla^2 y = u, \quad \text{for } (\mathbf{x},t) \in \Omega \times [0,T], \\ & \widehat{y} = 64t\sin(2\pi((x_1-0.5)^2+(x_2-0.5)^2)), \\ & y = 0, \quad \text{on } \partial\Omega, \\ & y(\mathbf{x},0) = 0, \end{aligned}$$

with $\mathbf{x} = [x_1, x_2]^T$ and $\Omega = [0,1]\times[0,1]$. Figure 6.3 shows the state and control variable at a specific time t.

In Table 6.2, we present iteration counts when using the dropping preconditioner $\mathcal{P}_1$ and our variations $\widehat{\mathcal{P}}_1$ and $\overline{\mathcal{P}}_1$. In the first part of the table, we examine the behaviour of the preconditioners as the mesh parameter h decreases. We can see that both the original preconditioner $\mathcal{P}_1$, as well as the block diagonal variant $\widehat{\mathcal{P}}_1$, maintain approximately mesh independent iteration counts while the iterations of $\overline{\mathcal{P}}_1$ do increase significantly as h decreases. In the second part of the table, we examine dependence on the number of time-steps, ℓ. Both $\widehat{\mathcal{P}}_1$ and $\overline{\mathcal{P}}_1$ require more iterations as ℓ increases; we see that both exhibit iteration counts which appear to plateau as ℓ becomes large. However, at large values of β it appears that $\overline{\mathcal{P}}_1$ is not as affected

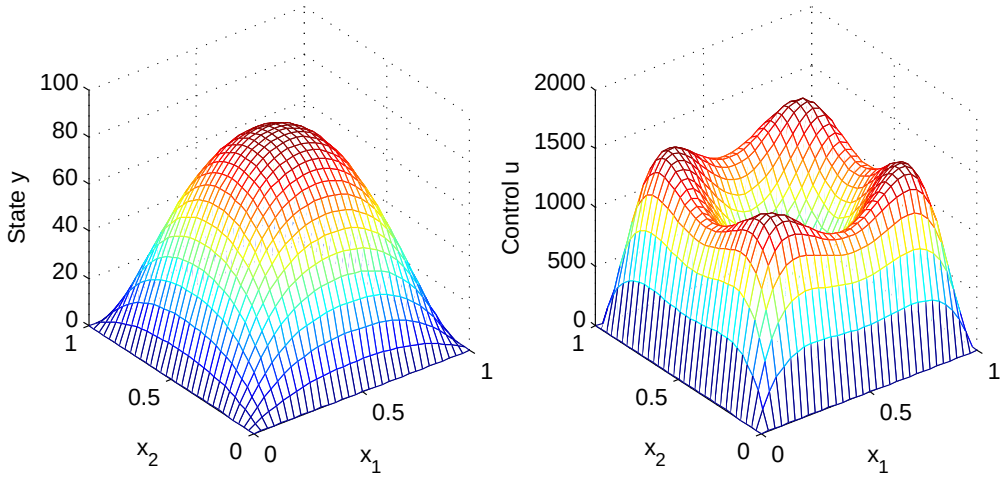


Figure 6.3: Computed solution to the heat control problem using IFISS at time $t = 0.2$, with $h = 2^{-5}$, $\beta = 10^{-1}$

by increases in ℓ compared with $\widehat{\mathcal{P}}_1$. The opposite appears to occur when β is small, however, where the iteration counts using $\widehat{\mathcal{P}}_1$ remain much more stable than those of $\overline{\mathcal{P}}_1$. While we obtained a similar type of opposing result for the forward problem in Chapters 3 and 4, we might have predicted $\overline{\mathcal{P}}_1$ to scale better with changes in ℓ than $\widehat{\mathcal{P}}_1$ but this only appears to be the case for large β. It is not immediately clear to us why this should occur.

Furthermore, as predicted by the theory, all of the preconditioners have significantly reduced performance for very small values of β. However, it is noted that $\widehat{\mathcal{P}}_1$ performed better at the median value $\beta = 10^{-4}$ than at either other value of β considered. This is most likely because since $\tau = 10^{-2}$, when $\beta = 10^{-6}$, we have that $\frac{\beta}{\tau^2} = 1$. Another surprising result was that for $\beta = 10^{-6}$, we find that $\widehat{\mathcal{P}}_1$ outperforms the original preconditioner $\mathcal{P}_1$. This is an important result as it indicates that $\widehat{\mathcal{P}}_1$ may be an effective choice of preconditioner for some parameter ranges, even when parallel computations are not available.

The results for $\mathcal{P}_2, \widehat{\mathcal{P}}_2$, and $\overline{\mathcal{P}}_2$ are presented in Table 6.3. We see that $\widehat{\mathcal{P}}_2$ has almost no dependence on h and, while the iteration counts do increase with ℓ when $\beta = 10^{-2}$, at smaller values of β the iteration counts remain relatively constant. The variant $\overline{\mathcal{P}}_2$ does increase iterations numbers as h is decreased for $\beta = 10^{-2}$ but again

Table 6.2: Iteration counts for MINRES for various values of regularization parameter with $\tau = 0.01$. (* denotes when iterations numbers exceeded 500)

h	ℓ	DoF	$\beta = 10^{-2}$			$\beta = 10^{-4}$			$\beta = 10^{-6}$		
			$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$	$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$	$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$
2^{-4}	20	28,900	12	70	42	34	44	44	213	105	222
2^{-5}	20	65,340	12	72	44	34	44	45	229	114	242
2^{-6}	20	253,500	12	72	54	36	44	67	258	117	*
2^{-7}	20	998,460	12	72	58	36	44	71	242	117	*
2^{-8}	20	3,962,940	12	72	96	36	44	98	244	117	*
2^{-5}	20	65,340	12	72	44	35	44	45	229	114	242
2^{-5}	40	130,680	14	104	45	45	46	56	319	120	328
2^{-5}	60	196,020	15	130	46	52	48	62	388	124	391
2^{-5}	80	261,360	15	148	46	58	48	68	440	126	444
2^{-5}	100	326,700	15	153	46	63	50	74	482	128	486

we see much more stable iteration counts for small values of β. The iteration counts for $\overline{\mathcal{P}}_2$ remain approximately independent of ℓ for all values of β.

Examining the behaviour for decreasing β, we see that all our variants perform better with small β, moreover when $\beta = 10^{-6}$ we see almost identical iteration numbers with the original preconditioner $\mathcal{P}_2$. In fact $\overline{\mathcal{P}}_2$ outperforms the original preconditioner $\mathcal{P}_2$ when $\beta = 10^{-6}$. However, with parallel capabilities included our preconditioners could be applied more quickly than $\mathcal{P}_2$. Thus for small β, even with limited parallel capability, there is no advantage to using the full preconditioner.

6.5 Conclusions

Throughout this chapter, we have shown that the preconditioning techniques developed in Chapters 3 and 4 for the forward problem can be successfully extended to develop preconditioners for the heat control problem. Furthermore, in this context we were able to obtain rigorous eigenvalue bounds for the preconditioned systems.

We have demonstrated that variations of the dropping strategy preconditioner have competing requirements on the value of $\frac{\beta}{\tau^2}$ to maintain well clustered eigenvalues and therefore only perform well when this value is relatively near one. However, while

Table 6.3: Iteration counts for MINRES for various values of regularization parameter β with $\tau = 0.01$.

h	ℓ	DoF	$\beta = 10^{-2}$			$\beta = 10^{-4}$			$\beta = 10^{-6}$		
			$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$	$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$	$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$
2^{-4}	20	17,340	16	70	40	19	36	26	23	22	18
2^{-5}	20	65,340	17	71	42	20	40	26	24	25	19
2^{-6}	20	253,500	17	72	48	21	40	26	23	23	20
2^{-7}	20	998,460	18	71	54	23	36	28	27	27	20
2^{-8}	20	3,962,940	16	71	74	23	34	28	27	27	20
2^{-5}	20	65,340	17	72	42	20	40	26	24	25	19
2^{-5}	40	130,680	19	102	44	21	36	28	19	19	20
2^{-5}	60	196,020	21	130	46	22	38	28	20	20	20
2^{-5}	80	261,360	21	147	46	22	38	28	20	20	21
2^{-5}	100	326,700	21	152	46	22	38	29	20	21	22

the variations of the matching strategy were predicted to achieve the smallest iteration numbers for small $\frac{\beta}{\tau^2}$, even for larger values of β the preconditioner behaves extremely well. Taking into account the considerable parallel in time capabilities of both the block diagonal and circulant variations, these matching strategy based approaches could be highly effective.

CHAPTER 7

Convection-Diffusion Control Problem

The convection-diffusion equation describes many important physical applications; one example might be the spread of a contaminant under the influence of flow in the containing fluid, as well as diffusive effects. Here we focus on the convection-diffusion control problem rather that just the forward problem. An example where this may be required, may be finding the most effective way to apply a mitigating substance so as to achieve desired concentrations of a contaminant.

While this type of problem is of great interest, it also poses a significant computational challenge. One key consideration is that care is required in general for non-self-adjoint problems such as convection-diffusion, to ensure that the discretize-then-optimize and optimize-then-discretize approaches obtain the same linear system. For example, the widely used Streamline Upwind Petrov-Galerkin (SUPG) stabilization method [16] does not, in general, satisfy the property that the forward and adjoint problems commute. However, we are able to utilize the Local Projection Stabilization (LPS) method [5, 6], which is adjoint consistent in order to solve this problem.

With any such stabilization method, it still remains the case that the differential operator will be non-self-adjoint. As we used the symmetry of the stiffness matrix several times to produce eigenvalue bounds for the heat control problem, this may

present us with further challenges. We note that the overall optimality system remains symmetric with an adjoint consistent approach. Therefore, eigenvalues can be used to provide convergence estimates for MINRES for these control problems. All of the preconditioners discussed previously, utilize multigrid processes for practical approximate application. Again for the convection-diffusion case, we need to take extra care to ensure that the multigrid process used can accurately approximate convection-diffusion operators.

Before we discuss the numerical solution to these problems, we will first spend some time illustrating the derivation of the linear systems in Section 7.1. As for the heat control problems considered in Chapter 6, we will examine block diagonal and circulant based variations to the dropping and matching preconditioning strategies. These approaches will be considered in Sections 7.2 and 7.3 respectively. Numerical results are provided in Section 7.5 before giving concluding remarks in Section 7.6.

7.1 Problem Derivation

In this chapter we consider a time-dependent convection-diffusion control problem of the form

$$\begin{aligned} \min_{y,u} \mathcal{J}(y,u) &= \frac{1}{2}\int_0^T \|y-\widehat{y}\|^2_{L_2(\Omega)}\,dt + \frac{\beta}{2}\int_0^T \|u\|^2_{L_2(\Omega)}\,dt \\ \text{such that } y_t - \epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y &= u, \quad \text{for } (\mathbf{x},t) \in \Omega\times[0,T] \\ y &= f, \quad \text{on } \partial\Omega, \\ y &= y_0, \quad \text{at } t=0, \end{aligned} \tag{7.1}$$

where, as before, y denotes the state variable, $\widehat{y}$ is some known desired state, u is the control variable, and $\beta > 0$ is the regularization parameter. Also present is the diffusivity parameter $\epsilon > 0$ and the divergence free wind vector $\mathbf{w}$. We assume that the convection term is more dominant than the diffusion, as is typically the case for real world applications, and thus $\epsilon \ll \|\mathbf{w}\|$. Problems in which diffusion dominates can generally be treated as for heat conduction.

We can see that this formulation is identical to that of the heat control problem

except that the constraint equation is now the time-dependent convection-diffusion equation rather than the heat equation. We showed in Section 2.1.3 that the convection-diffusion equation can be discretized using Galerkin finite elements and Backwards Euler time-stepping to obtain

$$\mathcal{K}\mathbf{y} - \tau\mathcal{M}\mathbf{u} = \mathbf{d}, \tag{7.2}$$

where, similar to the heat equation, we have

$$\mathcal{M} := \begin{bmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \end{bmatrix}, \quad \text{and} \quad \mathcal{K} := \begin{bmatrix} M + \tau\widetilde{K} & & & \\ -M & M + \tau\widetilde{K} & & \\ & \ddots & \ddots & \\ & & -M & M + \tau\widetilde{K} \end{bmatrix}, \tag{7.3}$$

however we now have the term

$$\widetilde{K} = \epsilon K + N + \widetilde{T}. \tag{7.4}$$

Here K is the standard Galerkin stiffness matrix, N represents the convection term and $\widetilde{T}$ is the stabilization term. As mentioned earlier, we need to use an adjoint consistent stabilization method in order to ensure that the linear systems obtained by discretizing first or by optimizing first, will coincide. Therefore, we will use the LPS stabilization method [5, 6] throughout this chapter to maintain consistency between the two approaches. Thus, we obtain the KKT system,

$$\mathcal{A}\mathbf{x} := \begin{bmatrix} \tau\mathcal{M}_{1/2} & \mathbf{0} & \mathcal{K}^T \\ \mathbf{0} & \beta\tau\mathcal{M}_{1/2} & -\tau\mathcal{M} \\ \mathcal{K} & -\tau\mathcal{M} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \tau\mathbf{b} \\ \mathbf{0} \\ \mathbf{d} \end{bmatrix}. \tag{7.5}$$

As with the heat control problem, our preconditioning strategies are based on Schur complement approximations. The Schur complement of the optimality system in (7.5) is given by

$$\mathcal{S} := \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T + \frac{\tau^2}{\beta}\mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{M}. \tag{7.6}$$

As before, the two main approximation strategies upon which we will build our preconditioners are the dropping strategy, given by

$$\mathcal{S}_1 := \frac{1}{\tau}\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T, \tag{7.7}$$

and the matching strategy, given by

$$\mathcal{S}_2 := \frac{1}{\tau}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\mathcal{K} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T. \tag{7.8}$$

Each of our Schur complement approximations will be formed by substituting either the block diagonal or circulant approximations to the forward problem for $\mathcal{K}$ in (7.7) and (7.8). As we saw in Chapter 3 and 4, finding effective preconditioners for the convection-diffusion problem is often more challenging than for problems with only self-adjoint elliptic operators. We nonetheless attempt to provide eigenvalue bounds for the preconditioned optimality system on which MINRES convergence bounds are based.

7.2 Dropping Strategy Based Preconditioners

7.2.1 Block Diagonal Variation

The preconditioners developed in this section will be based on the dropping strategy Schur complement approximation $\mathcal{S}_1$ described in (7.7). The block diagonal variation will approximate the forward problem $\mathcal{K}$ from (7.3) by only the block diagonal terms. Thus we define

$$\widehat{\mathcal{K}} := \begin{bmatrix} M + \tau\widetilde{K} & & & \\ & M + \tau\widetilde{K} & & \\ & & \ddots & \\ & & & M + \tau\widetilde{K} \end{bmatrix}, \tag{7.9}$$

and we note that, unlike for the heat control problem, $\widehat{\mathcal{K}}$ is no longer symmetric since $\widetilde{K} = \epsilon K + N + \widetilde{T}$ is not symmetric. We do still maintain the relation $\mathcal{K} = \widehat{\mathcal{K}} - \widehat{\Sigma}$ where

$$\widehat{\Sigma} := \begin{bmatrix} \mathbf{0} & & & \\ M & \mathbf{0} & & \\ & \ddots & \ddots & \\ & & M & \mathbf{0} \end{bmatrix}. \tag{7.10}$$

Using this block diagonal approximation instead of $\mathcal{K}$ we can form a new Schur complement approximation given by

$$\widehat{\mathcal{S}}_1 := \frac{1}{\tau}\widehat{\mathcal{K}}\mathcal{M}_{1/2}\widehat{\mathcal{K}}^T. \tag{7.11}$$

In order to provide bounds of the generalized Rayleigh quotient of $\mathcal{S}$ and $\widehat{\mathcal{S}}_1$ we define the following terms:

$$\begin{aligned} \mathbf{a} &:= \mathcal{M}_{1/2}^{-1/2}\mathcal{K}^T\mathbf{v}, & \mathbf{b} &:= \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1/2}^{-1/2}\mathcal{M}^T\mathbf{v}, \\ \widehat{\mathbf{c}} &:= \mathcal{M}_{1/2}^{-1/2}\widehat{\Sigma}^T\mathbf{v}, & \widehat{\mathbf{d}} &:= \mathcal{M}_{1/2}^{-1/2}\widehat{\mathcal{K}}^T\mathbf{v}. \end{aligned} \tag{7.12}$$

The appropriate Rayleigh quotient $\widehat{R}_1$ is defined as

$$\widehat{R}_1 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\widehat{\mathcal{S}}_1\mathbf{v}} = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\widehat{\mathbf{c}})^T(\mathbf{a}+\widehat{\mathbf{c}})}. \tag{7.13}$$

For the heat control problem, in order to prove eigenvalue bounds for the corresponding preconditioner, we required two results. Firstly, in Lemma 6.1 it was shown that $\widehat{\mathbf{c}}^T\widehat{\mathbf{c}} \leq C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}^T$, and secondly in Lemma 6.2 it was shown that $\mathbf{b}^T\mathbf{b}$ was less than $\widehat{\mathbf{d}}^T\widehat{\mathbf{d}}$ times a constant factor independent of the mesh parameter h. Simply by looking at the definition of $\widehat{\mathbf{c}}$ and $\mathbf{b}$, we can see that these terms only involve mass matrices and are thus identical to the terms present in the heat control problem. Thus, the bound proved in Lemma 6.1 continues to hold in the convection-diffusion setting.

Unfortunately, we are not so lucky for the latter bound. For this bound, we explicitly required bounds for the eigenvalues of K, and as we do not have such

bounds for $\widetilde{K}$, we were not able to prove an analogous result. This is currently a deficiency in our analysis. However, this result is only required for the upper bound on $\widehat{R}_1$, so we are still able to make the following statement regarding the lower bound.

Theorem 7.1. *Let $\widehat{R}_1$ be as defined in* (7.13)*. If $C\frac{\beta}{\tau^2} < 1$, then*

$$\frac{1}{2} < \widehat{R}_1, \tag{7.14}$$

otherwise,

$$\frac{1}{2C}\frac{\tau^2}{\beta} \leq \widehat{R}_1. \tag{7.15}$$

Proof. By definition of $\widehat{R}_1$ and using the bound from Lemma 6.1 we can obtain

$$\widehat{R}_1 = \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}+\widehat{\mathbf{c}})^T(\mathbf{a}+\widehat{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a} + \widehat{\mathbf{c}}^T\widehat{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2\mathbf{a}^T\mathbf{a} + 2C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}}. \tag{7.16}$$

As per the proof of Theorem 6.1, if $C\frac{\beta}{\tau^2} < 1$, then $\widehat{R}_1 > \frac{1}{2}$, while if $C\frac{\beta}{\tau^2} \geq 1$, we have

$$\widehat{R}_1 \geq \frac{\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b}}{2C\frac{\beta}{\tau^2}(\mathbf{a}^T\mathbf{a} + \mathbf{b}^T\mathbf{b})} = \frac{1}{2C}\frac{\tau^2}{\beta},$$

which gives the final result. □

We can see that provided $\frac{\beta}{\tau^2}$ is sufficiently small, the smallest eigenvalue of $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$ will be relatively close to one but we are not able to make any statements regarding the size of the largest eigenvalues. This will be examined numerically in Section 7.4.

7.2.2 Circulant Based Variation

The circulant based variation is based upon forming the block Strang circulant of the forward problem $\mathcal{K}$. We denote this by $\overline{\mathcal{K}}$ as defined by

$$\overline{\mathcal{K}} := \begin{bmatrix} M+\tau\widetilde{K} & & & -M \\ -M & M+\tau\widetilde{K} & & \\ & \ddots & \ddots & \\ & & -M & M+\tau\widetilde{K} \end{bmatrix}. \tag{7.17}$$

By substituting $\overline{\mathcal{K}}$ in for $\mathcal{K}$ in the Schur complement approximation in (7.7) we obtain our new approximation

$$\overline{\mathcal{S}}_1 := \frac{1}{\tau}\overline{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\overline{\mathcal{K}}^T. \tag{7.18}$$

If we define

$$\bar{\mathbf{c}} := \mathcal{M}_{1/2}^{-1/2}\overline{\Sigma}^T\mathbf{v}, \tag{7.19}$$

then the relevant Rayleigh quotient is defined as

$$\overline{R}_1 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\overline{\mathcal{S}}_1\mathbf{v}} = \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{(\mathbf{a}-\bar{\mathbf{c}})^T(\mathbf{a}-\bar{\mathbf{c}})}. \tag{7.20}$$

For the analogous preconditioner in the heat control context, we were only able to obtain rigorous lower eigenvalue bounds. These lower bounds also hold here since, similar to the block diagonal variant, the lower bounds only required the relation between $\bar{\mathbf{c}}^T\bar{\mathbf{c}}$ and $\mathbf{b}^T\mathbf{b}$ proved in Lemma 6.3. All of these terms are exactly the same in the convection-diffusion case and therefore Lemma 6.3 still holds.

Theorem 7.2. *Let $\overline{R}_1$ be as defined in* (7.20)*, then if $C\frac{\beta}{\tau^2} < 1$*

$$\frac{1}{2} \leq \overline{R}_1, \tag{7.21}$$

otherwise,

$$\frac{1}{2\left(C\frac{\beta}{\tau^2}\right)} \leq \overline{R}_1. \tag{7.22}$$

Proof. This proof follows exactly in the same manner as Theorem 6.2 by using Lemma 6.3 to obtain,

$$\overline{R}_1 = \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{(\mathbf{a}-\bar{\mathbf{c}})^T(\mathbf{a}-\bar{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a}+\bar{\mathbf{c}}^T\bar{\mathbf{c}})} \geq \frac{\mathbf{a}^T\mathbf{a}+\mathbf{b}^T\mathbf{b}}{2(\mathbf{a}^T\mathbf{a}+C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b})},$$

where if $C\frac{\beta}{\tau^2} < 1$, we have that $\overline{R}_1 > \frac{1}{2}$, while if $C\frac{\beta}{\tau^2} \geq 1$, then $\overline{R}_1 \geq \frac{1}{2C\frac{\beta}{\tau^2}}$. □

Once again we are not able to determine upper bounds on the eigenvalues of $\overline{\mathcal{S}}_1^{-1}\mathcal{S}_1$, however, these will be examined numerically in Section 7.4.

7.3 Matching Strategy Based Preconditioners

7.3.1 Block Diagonal Variation

The matching strategy has been widely used for steady convection-diffusion control problems as rigorous eigenvalue bounds were obtained in [90]. These lead to robust convergence estimates unlike for the dropping strategy for which there are no known bounds in the convection-diffusion context.

By substituting the block diagonal approximation $\widehat{\mathcal{K}}$ from (7.9) into $\mathcal{S}_2$ in (7.8) we obtain a new approximation

$$\widehat{\mathcal{S}}_2 := \frac{1}{\tau}\left(\widehat{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\widehat{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T. \tag{7.23}$$

We wish to obtain bounds for the generalized Rayleigh quotient $\widehat{R}_2$ given by

$$\widehat{R}_2 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\widehat{\mathcal{S}}_2\mathbf{v}}. \tag{7.24}$$

We maintain the positive definiteness of $\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a}$ in the case of the convection-diffusion control problem as shown in the following Lemma which very much follows the analogous proof for the heat control problem.

Lemma 7.1. *For* $\mathbf{a} = \mathcal{M}_{1/2}^{-1/2}\mathcal{K}^T\mathbf{v}$ *and* $\mathbf{b} = \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1/2}^{-1/2}\mathcal{M}\mathbf{v}$ *we have,*

$$\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a} > 0. \tag{7.25}$$

Proof. By definition $\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a} = \mathbf{v}^T(\mathcal{K}\mathcal{M}_{1/2}^{-1}\mathcal{M} + \mathcal{M}\mathcal{M}_{1/2}^{-1}\mathcal{K}^T)\mathbf{v} = \mathbf{v}^T(\mathcal{K}\Delta + \Delta\mathcal{K}^T)\mathbf{v}$, where Δ is as defined in (6.13). If we let $L = M + \tau\widetilde{K}$ and note that the scaled identity

Δ_i will commute with any matrix, we have that

$$\mathcal{K}\Delta+\Delta\mathcal{K}^T = \begin{bmatrix} L\Delta_1+\Delta_1L^T & -\Delta_1M & & & \\ -\Delta_1M & L\Delta_2+\Delta_2L^T & -\Delta_2M & & \\ & \ddots & \ddots & \ddots & \\ & & -\Delta_{\ell-2}M & L\Delta_{\ell-1}+\Delta_{\ell-1}L^T & -M\Delta_{\ell-1} \\ & & & -\Delta_{\ell-1}M & L\Delta_\ell+\Delta_\ell L^T \end{bmatrix}. \tag{7.26}$$

By straightforward manipulation we have

$$\begin{aligned}\mathbf{v}^T(\mathcal{K}\Delta+\Delta\mathcal{K}^T)\mathbf{v} &= \sum_{i=1}^{\ell}\mathbf{v}_i^T(L\Delta_i+\Delta_iL^T)\mathbf{v}_i - \sum_{i=1}^{\ell-1}\mathbf{v}_i^T\Delta_iM\mathbf{v}_{i+1} - \sum_{i=1}^{\ell-1}\mathbf{v}_{i+1}^T\Delta_iM\mathbf{v}_i \\ &= \tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i(\widetilde{K}+\widetilde{K}^T)\mathbf{v}_i + \sum_{i=1}^{\ell-1}(\mathbf{v}_i-\mathbf{v}_{i+1})^T\Delta_iM(\mathbf{v}_i-\mathbf{v}_{i+1}) + \mathbf{v}_1^T\Delta_1M\mathbf{v}_1 \\ &\quad + \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_\ell.\end{aligned}$$

We know from Section 2.1.3 that $\widetilde{K}+\widetilde{K}^T$ is positive semi-definite, and M is positive definite so that the resulting sum of quadratic forms is greater than zero. This gives the result. □

In order to prove eigenvalue bounds, we also require an analogous proof to Lemma 6.2 which relates terms involving $\widehat{\mathbf{d}}$ and $\mathbf{b}$.

Lemma 7.2. *For* $\widehat{\mathbf{d}}$ *and* $\mathbf{b}$ *as defined in* (7.12)*, then*

$$\widehat{\mathbf{d}}^T\widehat{\mathbf{d}} \le \frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}, \quad \textit{and} \quad \widehat{\mathbf{d}}^T\mathbf{b}+\mathbf{b}^T\widehat{\mathbf{d}} \le 2\frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b}, \quad \forall\mathbf{v}\neq 0.$$

Proof. As with the proof of Lemma 6.2 we have,

$$\begin{aligned}\widehat{\mathbf{d}}^T\widehat{\mathbf{d}} &= \mathbf{v}^T\widehat{\mathcal{K}}\mathcal{M}_{1/2}^{-1}\widehat{\mathcal{K}}^T\mathbf{v} = \sum_{i=1}^{\ell}\mathbf{v}_i^T(M+\tau\widetilde{K})\Delta_iM^{-1}(M+\tau\widetilde{K})^T\mathbf{v}_i \\ &= \sum_{i=1}^{\ell}\mathbf{v}_i^TM\Delta_i\mathbf{v}_i + \sum_{i=1}^{\ell}\mathbf{v}_i^T\big(\tau\widetilde{K}\Delta_i+\tau\Delta_i\widetilde{K}^T+\tau^2\widetilde{K}M^{-1}\widetilde{K}^T\big)\mathbf{v}_i \\ &= \frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b} + \tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i(\widetilde{K}+\widetilde{K}^T)\mathbf{v}_i + \tau^2\sum_{i=1}^{\ell}\mathbf{v}_i^T\widetilde{K}M^{-1}\widetilde{K}^T\mathbf{v}_i \ge \frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b},\end{aligned}$$

since $\widetilde{K} + \widetilde{K}^T$ is positive semi-definite as is $\widetilde{K}M^{-1}\widetilde{K}^T$. For the second bound we use positive semi-definiteness of $\widetilde{K} + \widetilde{K}^T$ and the commutativity of Δ_i to find

$$\begin{aligned}
\widehat{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\widehat{\mathbf{d}} &= \mathbf{v}^T(\widehat{\mathcal{K}}\Delta + \Delta\widehat{\mathcal{K}}^T)\mathbf{v} \\
&= \sum_{i=1}^{\ell} \mathbf{v}_i^T(M + \tau\widetilde{K})\Delta_i\mathbf{v}_i + \sum_{i=1}^{\ell} \mathbf{v}_i^T\Delta_i(M + \tau\widetilde{K}^T)\mathbf{v}_i \\
&= 2\sum_{i=1}^{\ell} \mathbf{v}_i^T M\Delta_i\mathbf{v}_i + \tau\sum_{i=1}^{\ell} \mathbf{v}_i^T(\widetilde{K} + \widetilde{K})\Delta_i\mathbf{v}_i \\
&= \frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b} + \sum_{i=1}^{\ell} \mathbf{v}_i^T\tau K\Delta_i\mathbf{v}_i \geq \frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b},
\end{aligned}$$

which gives the result. □

Finally we can bound the overall Rayleigh quotient $\widehat{R}_2$.

Theorem 7.3. *For $\widehat{R}_2$ as defined in* (7.24)*,*

$$\frac{1}{4\left(1 + C\frac{\beta}{\tau^2}\right)} \leq \widehat{R}_2 < 2 + 2C. \tag{7.27}$$

Proof. As previously seen, we define $\widehat{R}_2$ to be

$$\widehat{R}_2 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}\frac{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}{\mathbf{v}^T\widehat{\mathcal{S}}_2\mathbf{v}} := \widehat{R}_{2A}\widehat{R}_{2B}, \tag{7.28}$$

where $\widehat{R}_{2A}$ is bounded between 1/2 and 1 by Theorem 5.5. Using the results from Lemma 6.1 and 7.2 we see that

$$\begin{aligned}
\widehat{R}_{2B} &= \frac{(\widehat{\mathbf{d}} + \mathbf{b} - \widehat{\mathbf{c}})^T(\widehat{\mathbf{d}} + \mathbf{b} - \widehat{\mathbf{c}})}{(\widehat{\mathbf{d}} + \mathbf{b})^T(\widehat{\mathbf{d}} + \mathbf{b})} \\
&\leq \frac{2(\widehat{\mathbf{d}} + \mathbf{b})^T(\widehat{\mathbf{d}} + \mathbf{b}) + 2\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}}{(\widehat{\mathbf{d}} + \mathbf{b})^T(\widehat{\mathbf{d}} + \mathbf{b})} = 2 + \frac{2\widehat{\mathbf{c}}^T\widehat{\mathbf{c}}}{\widehat{\mathbf{d}}^T\widehat{\mathbf{d}} + \mathbf{b}^T\widehat{\mathbf{d}} + \widehat{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}} \\
&\leq 2 + \frac{2C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}}{\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b} + 2\frac{\sqrt{\beta}}{\tau}\mathbf{b}^T\mathbf{b} + \mathbf{b}^T\mathbf{b}} = 2 + \frac{2C\frac{\beta}{\tau^2}}{(\frac{\sqrt{\beta}}{\tau} + 1)^2} \\
&< 2 + 2C,
\end{aligned}$$

since $\frac{\frac{\beta}{\tau^2}}{(\frac{\sqrt{\beta}}{\tau}+1)^2} < 1$. The lower bound follows exactly as in Lemma 6.5 by

$$\widehat{R}_{2B} = \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{(\mathbf{a}+\widehat{\mathbf{c}}+\mathbf{b})^T(\mathbf{a}+\widehat{\mathbf{c}}+\mathbf{b})} \geq \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{2\left((\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b}) + \frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}\right)}$$
$$\geq \frac{1}{2\left(1 + C\frac{\beta}{\tau^2}\frac{\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}\right)} \geq \frac{1}{2\left(1 + C\frac{\beta}{\tau^2}\right)}$$

which, when combined with the extra factor of $1/2$ from $\widehat{R}_{2A}$, gives the result. □

We have now bounded the eigenvalues of $\widehat{\mathcal{S}}_2^{-1}\mathcal{S}$ in the exact same manner as with the heat control problem. This is significant as convection-diffusion control problems tend to be significantly harder to solve than heat control problems, but the eigenvalue bounds achieved directly lead to convergence estimates for MINRES, just as they did for the heat control problem.

7.3.2 Circulant Based Variation

The final Schur complement approximation considered uses the circulant variant within the matching strategy and is defined by

$$\bar{\mathcal{S}}_2 := \frac{1}{\tau}\left(\bar{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)\mathcal{M}_{1/2}^{-1}\left(\bar{\mathcal{K}} + \frac{\tau}{\sqrt{\beta}}\mathcal{M}\right)^T, \tag{7.29}$$

where $\bar{\mathcal{K}}$ is as defined in (7.17). In order to bound the eigenvalues of $\bar{\mathcal{S}}_2^{-1}\mathcal{S}$ we consider the Rayleigh quotient,

$$\bar{R}_2 := \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{S}}_2\mathbf{v}} = \frac{\mathbf{v}^T\mathcal{S}\mathbf{v}}{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}\frac{\mathbf{v}^T\mathcal{S}_2\mathbf{v}}{\mathbf{v}^T\bar{\mathcal{S}}_2\mathbf{v}} := \bar{R}_{2A}\bar{R}_{2B}. \tag{7.30}$$

In order to bound this term we need to define

$$\widehat{\mathbf{d}} = \mathcal{M}_{1/2}^{-1/2}\bar{\mathcal{K}}^T\mathbf{v}, \tag{7.31}$$

and prove the positivity of $\bar{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\bar{\mathbf{d}}$. The proof of this result for the convection-diffusion problem is given in the following Lemma.

Lemma 7.3. *For* $\bar{\mathbf{d}} = \mathcal{M}_{1/2}^{-1/2}\overline{\mathcal{K}}^T\mathbf{v}$ *and* $\mathbf{b} = \frac{\tau}{\sqrt{\beta}}\mathcal{M}_{1/2}^{1/2}\mathcal{M}^T\mathbf{v}$, *then*

$$\bar{\mathbf{d}}^T\mathbf{b} + \mathbf{b}^T\bar{\mathbf{d}} > 0. \tag{7.32}$$

Proof. Letting $L = M + \tau\widetilde{K}$ and noting that $\Delta_1 = \Delta_\ell$ we have

$$\begin{aligned}
\mathbf{v}^T(\overline{\mathcal{K}}\Delta + \Delta\overline{\mathcal{K}}^T)\mathbf{v} &= 2\sum_{i=1}^{\ell}\mathbf{v}_i^T L\Delta_i\mathbf{v}_i - \sum_{i=1}^{\ell-1}\mathbf{v}_{i+1}^T M\Delta_i\mathbf{v}_i - \sum_{i=1}^{\ell-1}\mathbf{v}_i^T\Delta_i M\mathbf{v}_{i+1} \\
&\quad - \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_1 - \mathbf{v}_1^T M\Delta_\ell\mathbf{v}_\ell \\
&= \tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i(\widetilde{K} + \widetilde{K}^T)\mathbf{v}_i + \sum_{i=1}^{\ell-1}(\mathbf{v}_i - \mathbf{v}_{i+1})^T M\Delta_i(\mathbf{v}_i - \mathbf{v}_{i+1}) \\
&\quad + \mathbf{v}_1^T\Delta_1 M\mathbf{v}_1 + \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_\ell - \mathbf{v}_\ell^T\Delta_\ell M\mathbf{v}_1 - \mathbf{v}_1^T\Delta_\ell M\mathbf{v}_\ell \\
&= \tau\sum_{i=1}^{\ell}\mathbf{v}_i^T\Delta_i(\widetilde{K} + \widetilde{K}^T)\mathbf{v}_i + \sum_{i=1}^{\ell-1}(\mathbf{v}_i - \mathbf{v}_{i+1})^T M\Delta_i(\mathbf{v}_i - \mathbf{v}_{i+1}) \\
&\quad + \mathbf{v}_1^T\Delta_1 M\mathbf{v}_1 + (\mathbf{v}_1 - \mathbf{v}_\ell)^T\Delta_\ell M(\mathbf{v}_1 - \mathbf{v}_\ell).
\end{aligned}$$

Since this is the sum of positive definite or semi-definite terms we have the result. □

Theorem 7.4. *For* $\overline{R}_2$ *as defined in* (7.30),

$$\frac{1}{4(1+\frac{\beta}{\tau^2})} \le \overline{R}_2 \le 2\left(1 + \frac{\beta}{\tau^2}\right). \tag{7.33}$$

Proof. As in the block diagonal case, we only focus on $\overline{R}_{2B}$ since $1/2 \le \overline{R}_{2A} \le 1$ from Theorem 5.5. We see that we can write $\overline{R}_{2B}$ as,

$$\begin{aligned}
\overline{R}_{2B} &= \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{(\mathbf{a}+\mathbf{b}-\bar{\mathbf{c}})^T(\mathbf{a}+\mathbf{b}-\bar{\mathbf{c}})} \ge \frac{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})}{2(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b}) + 2\bar{\mathbf{c}}^T\bar{\mathbf{c}}} \\
&= \frac{1}{2(1+\frac{\bar{\mathbf{c}}^T\bar{\mathbf{c}}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})})} \le \frac{1}{2(1+C\frac{\beta}{\tau^2}\frac{\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})})},
\end{aligned}$$

and from Lemma 7.1 we have that $\mathbf{a}^T\mathbf{b} + \mathbf{b}^T\mathbf{a} > 0$ and therefore $\frac{\mathbf{b}^T\mathbf{b}}{(\mathbf{a}+\mathbf{b})^T(\mathbf{a}+\mathbf{b})} \le 1$ which

gives the lower bound result. For the upper bound we have,

$$\begin{aligned}\overline{R}_{2B} &= \frac{(\bar{\mathbf{d}}+\mathbf{b}-\bar{\mathbf{c}})^T(\bar{\mathbf{d}}+\mathbf{b}-\bar{\mathbf{c}})}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \leq \frac{2(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})+2\bar{\mathbf{c}}^T\bar{\mathbf{c}}}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \\ &\leq 2+\frac{2C\frac{\beta}{\tau^2}\mathbf{b}^T\mathbf{b}}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \leq 2+C\frac{\beta}{\tau^2}.\end{aligned}$$

using Lemma 7.3 to know that $\dfrac{\mathbf{b}^T\mathbf{b}}{(\bar{\mathbf{d}}+\mathbf{b})^T(\bar{\mathbf{d}}+\mathbf{b})} \leq 1$. Finally, we multiply each side by the bounds for $\overline{R}_{2A}$ from Theorem 5.5 to obtain the extra factor of $1/2$ in the lower bound. □

7.4 Summary of Eigenvalue Bounds

As with the heat control problem, we have summarized the eigenvalues bounds obtained for the preconditioners developed in the previous sections. For the dropping strategy based preconditioners, only lower bounds were obtained for the eigenvalues of the preconditioned system. However, in Figure 7.1 the eigenvalues of $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$ and $\overline{\mathcal{S}}_1^{-1}\mathcal{S}$ are presented for various values of β. We can see that for small values of β the largest eigenvalues of both $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$ and $\overline{\mathcal{S}}_1^{-1}\mathcal{S}$ grow significantly, in particular for the circulant variation.

As we were not theoretically able to provide a complete set of bounds for these eigenvalues, these plots provide additional evidence for the performance of our preconditioners. For example, although an upper bound was not able to be provided for the eigenvalues of $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$ or $\overline{\mathcal{S}}_1^{-1}\mathcal{S}$, we can see clearly from the plots that the largest eigenvalues appear to grow like $\frac{\tau^2}{\beta}$. In the case of $\bar{\mathcal{S}}_1$, there is only relatively few extremely large eigenvalues so performance may not deteriorate as much as could be expected.

For the matching strategy, just as for the heat control problem we are able to provide eigenvalues bounds for each of the preconditioners. Plots of the eigenvalues for various values of β for $\widehat{\mathcal{S}}_2^{-1}\mathcal{S}$ and $\overline{\mathcal{S}}_2^{-1}\mathcal{S}$ are presented in Figure 7.2. It is evident that for small values of β we see extremely well-clustered eigenvalues; for both block diagonal and circulant variations with $\frac{\beta}{\tau^2} = 10^{-4}$ we see eigenvalues between 0.8 and

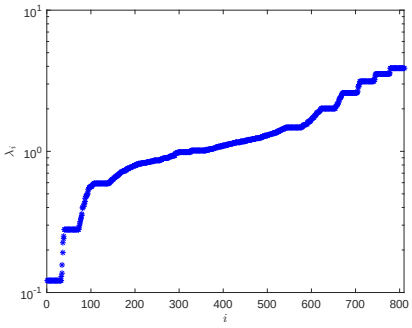
Table 7.1: Summary of eigenvalue bounds for $\widetilde{\mathcal{S}}^{-1}\mathcal{S}$ for each considered Schur complement approximation $\widetilde{\mathcal{S}}$ for the convection-diffusion control problem.

	Block Diagonal ($\widehat{\mathcal{S}}$)		Circulant ($\bar{\mathcal{S}}$)	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound
$\mathcal{S}_1$	$\frac{1}{2}$ or $\frac{1}{2C\frac{\beta}{\tau^2}}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	- -	$\frac{1}{2}$ or $\frac{1}{2C\frac{\beta}{\tau^2}}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	- -
$\mathcal{S}_2$	$\frac{1}{4\left(1+C\frac{\beta}{\tau^2}\right)}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	$2+2C$ -	$\frac{1}{4\left(1+C\frac{\beta}{\tau^2}\right)}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$	$2+2C\frac{\tau^2}{\beta}$ $\left(\frac{\beta}{\tau^2} \ll 1\right)$

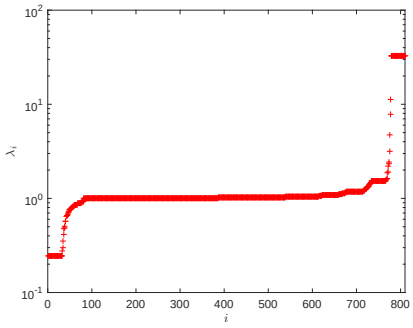
1 and therefore predict the preconditioners to perform well in this regime.

7.5 Numerical Results

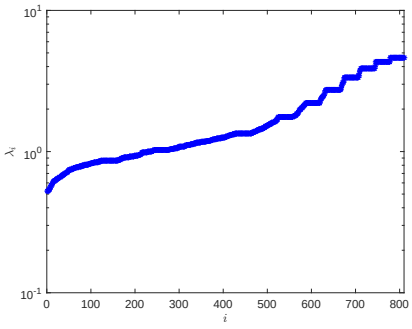
In this section, we provide numerical results for each of the proposed preconditioners in order to demonstrate the effectiveness of each method. We will again use the Incompressible Flow and Iterative Solver Software (IFISS) framework [103] to compute our results and we discretize the variables state variable y, the control variable u and the adjoint variable p using **Q1** finite elements. As we are now considering convection-diffusion operators, we use 2 V-cycles of the Ramage multigrid [93] to approximately invert the convection-diffusion operators in $\mathcal{S}_1, \mathcal{S}_2, \widehat{\mathcal{S}}_1$, and $\widehat{\mathcal{S}}_2$. We again use 5 V-cycles of AGMG [73, 74, 77, 78] to approximately invert the circulant based variants $\bar{\mathcal{S}}_1$ and $\bar{\mathcal{S}}_2$. The mass matrices which form the (1,1) and (2,2) blocks of all our preconditioners are approximated by 10 iterations of the linear Chebyshev semi-iteration. For the convection-diffusion control problem, we will consider two different problems which are described in detail below.



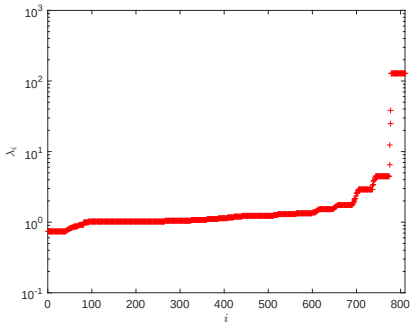
(a) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



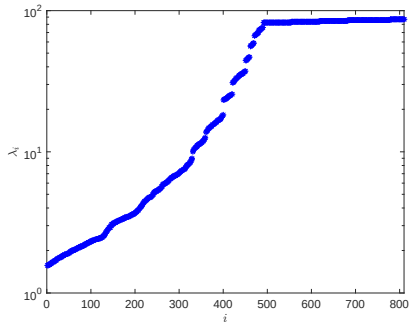
(b) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



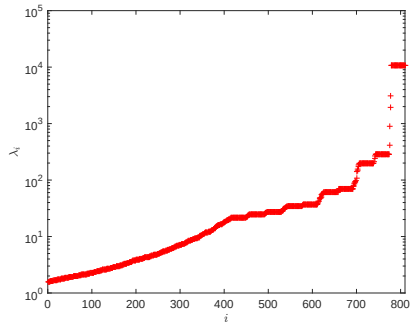
(c) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



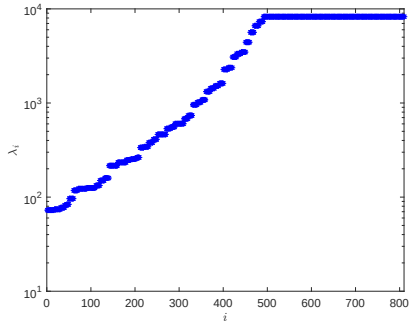
(d) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



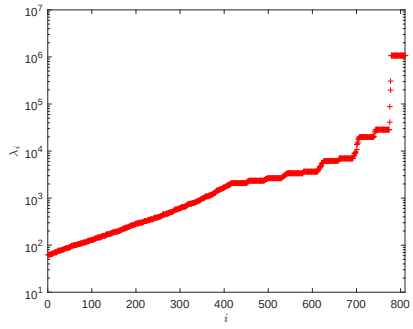
(e) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$



(f) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$

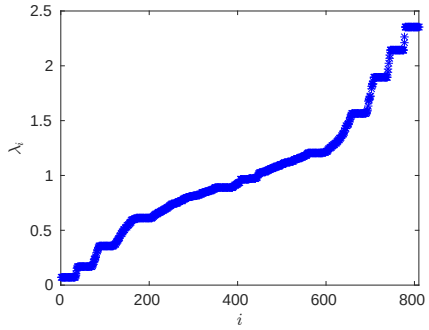


(g) $\lambda_i(\widehat{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$

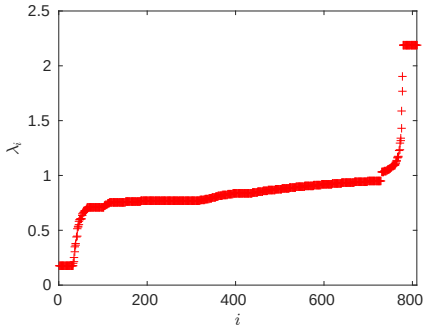


(h) $\lambda_i(\overline{\mathcal{S}}_1^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$

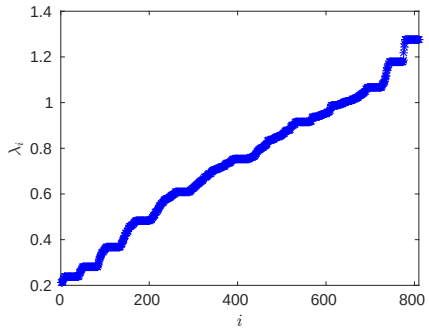
Figure 7.1: Eigenvalues of $\widehat{\mathcal{S}}_1^{-1}\mathcal{S}$ and $\overline{\mathcal{S}}_1^{-1}\mathcal{S}$ for different β values with $\tau = 0.1, h = 2^{-3}$ and $\ell = 10$ (shown in log scale).



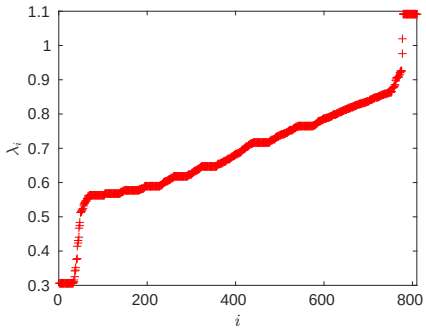
(a) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



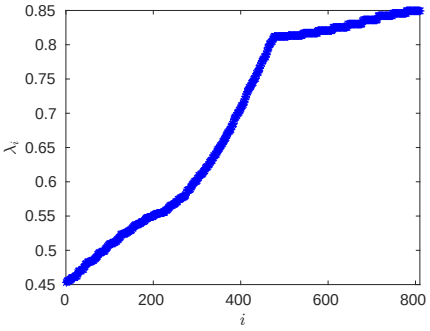
(b) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-1}, \frac{\beta}{\tau^2} = 10$



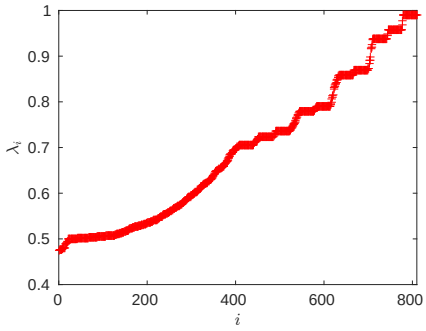
(c) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



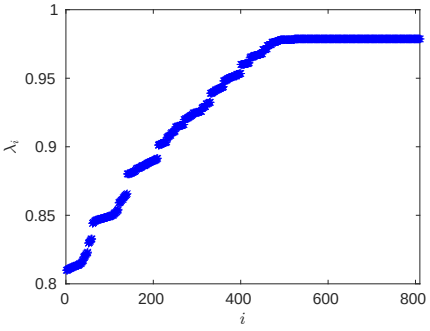
(d) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-2}, \frac{\beta}{\tau^2} = 1$



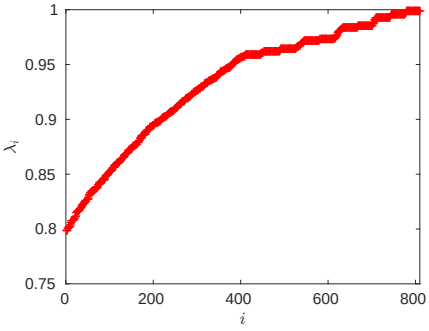
(e) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$



(f) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-4}, \frac{\beta}{\tau^2} = 10^{-2}$

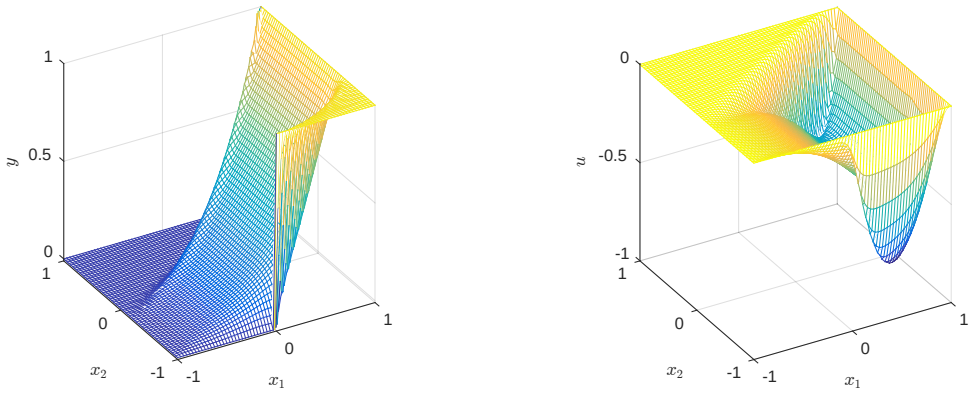


(g) $\lambda_i(\widehat{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$



(h) $\lambda_i(\overline{\mathcal{S}}_2^{-1}\mathcal{S}), \beta = 10^{-6}, \frac{\beta}{\tau^2} = 10^{-4}$

Figure 7.2: Eigenvalues of $\widehat{\mathcal{S}}_2^{-1}\mathcal{S}$ and $\overline{\mathcal{S}}_2^{-1}\mathcal{S}$ for different β values with $\tau = 0.1, h = 2^{-3}$ and $\ell = 10$.



(a) State y (b) Control u

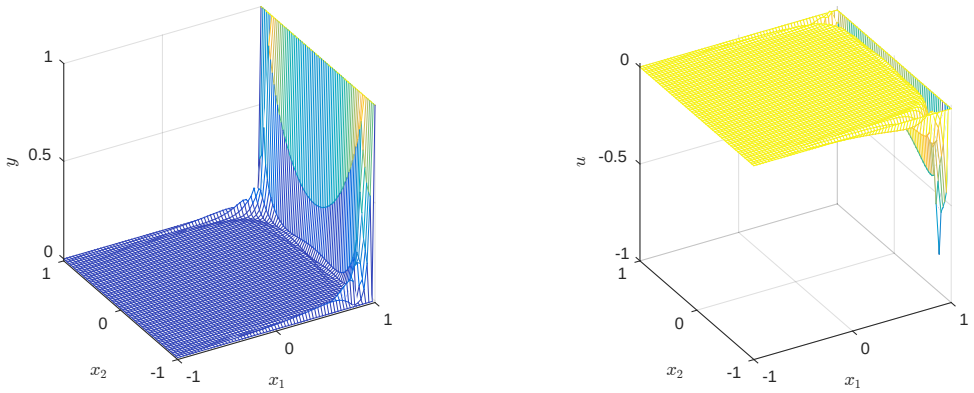
Figure 7.3: Solutions to Problem 1 at time $t = 0.2$ with $\epsilon = \frac{1}{200}$ and $\beta = 0.01$

Problem 1

The first problem is defined on the domain $\Omega = [-1, 1] \times [-1, 1]$ and is given by,

$$\begin{aligned}
&\min_{y,u} \frac{1}{2}\int_0^T \|y - \widehat{y}\|^2_{L_2(\Omega)}\, dt + \frac{\beta}{2}\int_0^T \|u\|^2_{L_2(\Omega)}\, dt \\
&\quad \text{such that} \quad y_t - \epsilon\nabla^2 y + \mathbf{w}\cdot\nabla y = u, \quad \text{in } \Omega \times [0, T] \\
&\quad \hat{y} = 0, \\
&\quad y(\mathbf{x}, 0) = y_0 = \tanh\left(100(x_2 + 1)\right)\left(1 + \tanh\left(100(x_1 - 1)\right)\right) \\
&\qquad\qquad + \frac{1}{2}\left(1 - \tanh\left(100(x_2 + 1)\right)\right)\left(1 + \tanh(100x_1)\right) \\
&\quad y = y_0, \quad \text{on } \partial\Omega
\end{aligned}$$

where $\mathbf{w} = [\sin\frac{\pi}{6}, \cos\frac{\pi}{6}]^T$. The initial condition y_0 is selected so that y is very close to 1 on $[0, 1] \times \{-1\}$ and exponentially tends to 0 elsewhere in the domain. The boundary conditions also satisfy this property. This type of constant wind problem has been considered for the time-independent control problem in [25, 86, 94]. A plot of the solution for both the state y and control u at a given point in time is presented in Figure 7.3. The iteration numbers for this problem for each of the preconditioners is presented in Tables 7.2 and 7.3.



(a) State y (b) Control u

Figure 7.4: Solutions to Problem 2 at time $t = 0.2$ with $\epsilon = \frac{1}{200}$ and $\beta = 0.01$

Problem 2

Also defined on the domain $\Omega = [-1, 1] \times [-1, 1]$, the second problem considered is control version of the *double glazing problem* considered in Section 3.5.2 and is given by,

$$\begin{aligned}
\min_{y,u} \frac{1}{2} \int_0^T \|y - \widehat{y}\|^2_{L_2(\Omega)} \, dt + \frac{\beta}{2} \int_0^T \|u\|^2_{L_2(\Omega)} \, dt \\
\text{such that} \quad y_t - \epsilon \nabla^2 y + \mathbf{w} \cdot \nabla y = u, \quad \text{in } \Omega \times [0, T] \\
y(\mathbf{x}, 0) = y_0 = 1 + \tanh\left(100(x - 1)\right) \\
y = y_0, \quad \text{on } \partial\Omega \\
\hat{y} = 0,
\end{aligned}$$

where $\mathbf{w} = [2x_2(1-x_1^2), -2x_1(1-x_2^2)]^T$. Once again we have the property that y_0 is very close to 1 on $\{1\} \times [-1, 1]$ and exponentially tends to 0 elsewhere in the domain. This problem has been considered for the time-independent control problem in [25, 86, 94] and for the time-dependent control problem in [106]. A plot of the solution for both the state y and control u at a given point in time is presented in Figure 7.4. The iteration numbers for this problem for each of the preconditioners is presented in Tables 7.4 and 7.5.

For both problems, we can see that the dropping strategy does not perform as

Table 7.2: MINRES iteration counts for Problem 1 with dropping strategy based preconditioners with various values of regularization parameter β and a constant $\tau = 0.01$. (* denotes when iterations numbers exceeded 500)

$\frac{h}{2}$	ℓ	DoF	$\beta = 10^{-2}$			$\beta = 10^{-4}$			$\beta = 10^{-6}$		
			$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$	$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$	$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$
2^{-4}	20	17,340	13	100	120	69	27	104	295	49	478
2^{-5}	20	65,340	13	102	147	79	27	135	*	81	*
2^{-6}	20	253,500	11	106	*	72	29	313	*	87	*
2^{-7}	20	998,460	15	90	*	111	27	*	*	43	*
2^{-8}	20	3,962,940	15	92	*	171	27	*	*	47	*
2^{-5}	20	65,340	13	102	120	79	27	104	*	81	*
2^{-5}	40	130,680	17	122	118	121	29	119	*	84	*
2^{-5}	60	196,020	21	131	116	145	29	134	*	87	*
2^{-5}	80	261,360	25	139	118	192	29	148	*	87	*
2^{-5}	100	326,700	27	143	119	212	29	163	*	89	*

well for small values of β as for the heat control problem. Surprisingly, we do see that $\widehat{\mathcal{P}}_1$ performs better at small values of β than the original preconditioner $\mathcal{P}_1$ and it is unclear why this should occur. As with the heat control problem, we also see the best performance of $\widehat{\mathcal{P}}_1$ when $\beta = 10^{-4}$.

For the matching strategy based preconditioners, when $\beta = 10^{-6}$ we see very little difference between iteration numbers for the original preconditioner $\mathcal{P}_2$ and the two parallelizable variants $\widehat{\mathcal{P}}_2$ and $\overline{\mathcal{P}}_2$. Again for both of these variants we see little dependence on both the mesh parameter h or the number of time-steps ℓ. For larger values of β, both variants have an increased number of iterations from the original preconditioner $\mathcal{P}_2$, however, in the presence of parallelism over time, speed-up for this method could still be achieved by using either $\widehat{\mathcal{P}}_2$ or $\overline{\mathcal{P}}_2$.

7.6 Conclusion

In this chapter, we considered the time-dependent convection-diffusion optimal control problem. There are many additional challenges which must be addressed for solution of this problem when compared with the heat control problem considered in

Table 7.3: MINRES iteration counts for Problem 1 with matching strategy based preconditioners with various values of regularization parameter β and a constant $\tau = 0.01$.

$\frac{h}{2}$	ℓ	DoF	$\beta = 10^{-2}$			$\beta = 10^{-4}$			$\beta = 10^{-6}$		
			$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$	$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$	$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$
2^{-4}	20	17,340	11	98	78	11	19	15	9	7	9
2^{-5}	20	65,340	11	100	76	11	20	16	9	9	9
2^{-6}	20	253,500	11	103	98	11	21	20	11	10	11
2^{-7}	20	998,460	11	88	95	11	20	15	7	5	7
2^{-8}	20	3,962,940	11	90	94	11	21	15	9	7	9
2^{-5}	20	65,340	11	100	76	11	20	16	9	9	9
2^{-5}	40	130,680	12	120	76	11	21	17	9	9	10
2^{-5}	60	196,020	13	130	77	11	21	17	11	9	11
2^{-5}	80	261,360	13	137	78	11	21	17	11	9	11
2^{-5}	100	326,700	13	140	78	11	21	17	11	9	11

Table 7.4: MINRES iteration counts for Problem 2 with dropping strategy based preconditioners with various values of regularization parameter β and a constant $\tau = 0.01$ (* denotes when iterations numbers exceeded 500)

$\frac{h}{2}$	ℓ	DoF	$\beta = 10^{-2}$			$\beta = 10^{-4}$			$\beta = 10^{-6}$		
			$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$	$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$	$\mathcal{P}_1$	$\widehat{\mathcal{P}}_1$	$\overline{\mathcal{P}}_1$
2^{-4}	20	17,340	13	78	248	81	24	277	329	35	*
2^{-5}	20	65,340	13	81	231	93	25	257	*	51	*
2^{-6}	20	253,500	13	82	460	99	25	434	*	54	*
2^{-7}	20	998,460	13	84	*	105	25	*	*	53	*
2^{-8}	20	3,962,940	13	86	*	11	27	*	*	47	*
2^{-5}	20	65,340	13	81	231	93	25	257	*	51	*
2^{-5}	40	130,680	25	98	211	161	25	268	*	55	*
2^{-5}	60	196,020	31	114	201	225	27	294	*	57	*
2^{-5}	80	261,360	35	118	201	278	27	327	*	57	*
2^{-5}	100	326,700	41	120	200	329	27	347	*	59	*

Table 7.5: MINRES iteration counts for Problem 2 with matching strategy based preconditioners with various values of regularization parameter β and a constant $\tau = 0.01$.

$\frac{h}{2}$	ℓ	DoF	$\beta = 10^{-2}$			$\beta = 10^{-4}$			$\beta = 10^{-6}$		
			$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$	$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$	$\mathcal{P}_2$	$\widehat{\mathcal{P}}_2$	$\overline{\mathcal{P}}_2$
2^{-4}	20	17,340	9	77	72	9	17	13	7	5	7
2^{-5}	20	65,340	9	80	72	10	18	13	7	7	7
2^{-6}	20	253,500	9	80	116	11	19	21	7	7	9
2^{-7}	20	998,460	9	82	102	11	19	29	9	7	9
2^{-8}	20	3,962,940	9	84	94	11	20	15	9	7	9
2^{-5}	20	65,340	9	80	72	10	18	13	7	7	7
2^{-5}	40	130,680	10	96	73	11	19	13	7	7	7
2^{-5}	60	196,020	11	110	74	11	19	15	7	7	7
2^{-5}	80	261,360	11	114	76	11	19	15	7	7	9
2^{-5}	100	326,700	11	116	76	11	19	15	9	7	9

the previous chapter. Firstly, we require an adjoint-consistent stabilization method in order to obtain an accurate solution for the convection-diffusion forward problem and ensure that the discretize-then-optimize and optimize-then-discretize systems coincide. By using the local projection stabilization methods we are able to achieve this.

As we do not have eigenvalue bounds for the convection-diffusion operator, we are unable to use estimates to obtain overall bounds for the preconditioned systems. We can, however, use the positive-semi-definiteness of the symmetric part of $\widetilde{K}$ to obtain analogous results to the heat control problem. In fact, the eigenvalue bounds obtained for the matching strategy based preconditioners are identical to those obtained for the heat control problem. This is perhaps due to the strength of the matching strategy preconditioner and its applicability to many different problems.

Up until relatively recently, convection-diffusion control problems were considered a significant numerical challenge for applied mathematicians, while the time-dependent case was at the limit of numerical capabilities. Here we have presented four potential preconditioners which all have the ability to be parallelized over time and thus, for various parameter regimes, all offer potentially significant speed-up over

sequential methods. In particular, the variations presented here perform best for small values of β and thus lend themselves to be introduced as part of the matching strategy based preconditioners. For the case of small β we see little or no difference between the iteration counts of $\mathcal{S}_2$ and our variants $\widehat{\mathcal{S}}_2$ and $\overline{\mathcal{S}}_2$. However our variants could be completed in parallel over time and theoretically could be computed up to ℓ times faster.

CHAPTER 8

Conclusion

In this thesis, we have examined the solution of time-dependent PDE problems through the use of preconditioned iterative methods. We have focussed on solution of the all-at-once system, whereby we solve for all time-steps simultaneously in a large, block matrix system. By using the all-at-once system, we were able to develop preconditioners which have the ability to be parallelized over time. One method which achieved this was simply using the block diagonal of the system as a preconditioner. Using this block Jacobi style preconditioner is by no means an original approach, however, the use in this context and the accompanying analysis of minimal polynomials and achieving termination bounds for an appropriate iterative method, are new. Furthermore, we were able to show that even when the preconditioner is applied approximately, we achieve a similar convergence despite our minimal polynomial theory not applying. This phenomenon, whereby the Jordan form of a nearby matrix is influencing the convergence behaviour is, in our opinion, worthy of further study.

By taking advantage of the block Toeplitz structure of the all-at-once system, in Chapter 4 we developed block circulant preconditioners, which also have the potential to be parallelized over time. Despite being a non-symmetric problem, we were able to

determine that the total number of distinct eigenvalues of the preconditioned system was independent of the number of time-steps ℓ. Thus, an appropriate Krylov subspace method would terminate in at most a number of steps independent of ℓ.

It was the preconditioners developed for the forward problem in these chapters which motivated the preconditioners for time-dependent optimal control problems. The proposed preconditioners were based on block diagonal Schur complement based preconditioners and, as such, were able to be implemented within MINRES. By replacing the all-at-once system of the forward problem with either the block diagonal or circulant preconditioner (and the transpose of these matrices for the backward problem) within the two Schur complement approximations we were able to obtain four new preconditioning strategies. All of these strategies could be implemented in parallel over time and, therefore, could theoretically achieve significant speed-up to the existing preconditioners considered. Approaches based on the matching strategy were the most robust and, in general, the parallelizable preconditioners performed best for small values of the regularization parameter β. In fact for small values of β, there was little or no difference in iteration counts between the existing preconditioners and the parallelizable variations. Thus, in this parameter regime, there is no advantage to using the original sequential preconditioners even in the absence of parallel implementation. We note that preconditioners do not always perform well for the normal equations, even if they are effective for the original system. It is, therefore, noteworthy that our preconditioners are proven to work well within both the forward problem and optimal control settings.

As our parallelizable in time preconditioners are simply used within an existing Krylov subspace method, implementation is relatively straightforward and does not rely on an understanding of more complex parallel solvers. Thus, we hope that this allows this type of preconditioner to be appealing to many researchers.

The preconditioners developed in this thesis could easily be extended to other time-dependent problems which can be formulated in an all-at-once manner. Another interesting area where this theory may be of use in the solution of non-linear time-dependent PDEs and this would certainly be an interesting continuation of the themes developed in this thesis.

References

[1] S. R. Arridge, *Optical Tomography in Medical Imaging*, Inverse Problems, 15 (1999), pp. R41–R93.

[2] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1996.

[3] A. T. Barker and M. Stoll, *Domain Decomposition in Time for PDE-Constrained Optimization*, Computer Physics Communications, 197 (2015), pp. 136–143.

[4] W. Barthel, C. John, and F. Tröltzsch, *Optimal Boundary Control of a System of Reaction Diffusion Equations*, ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift fr Angewandte Mathematik und Mechanik, 90 (2010), pp. 966–982.

[5] R. Becker and M. Braack, *A Finite Element Pressure Gradient Stabilization for the Stokes Equations Based on Local Projections*, CALCOLO, 38 (2001), pp. 173–199.

[6] R. Becker and B. Vexler, *Optimal Control of the Convection-Diffusion Equation Using Stabilized Finite Element Methods*, Numerische Mathematik, 106 (2007), pp. 349–367.

[7] J.-D. Benamou and Y. Brenier, *A Computational Fluid Mechanics Solution to the Monge-Kantorovich Mass Transfer Problem*, Numerische Mathematik, 84 (2000), pp. 375–393.

[8] M. Benzi, G. H. Golub, and J. Liesen, *Numerical Solution of Saddle Point Problems*, Acta Numerica, 14 (2005), pp. 1–137.

[9] M. Benzi, E. Haber, and L. Taralli, *A Preconditioning Technique for a Class of PDE-Constrained Optimization Problems*, Advances in Computational Mathematics, 35 (2011), pp. 149–173.

[10] M. Benzi and V. Kuhlemann, *Chebyshev Acceleration of the GeneRank Algorithm*, Electonic Transactions on Numerical Analysis, 40 (2013), pp. 311–320.

[11] A. Borzi and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2012.

[12] J. Boyle, M. D. Mihajlovic, and J. A. Scott, *HSL_MI20: An Efficient AMG Preconditioner for Finite Element Problems in 3D*, International Journal for Numerical Methods in Engineering, 82 (2010), pp. 64–98.

[13] D. Braess and P. Peisker, *On the Numerical Solution of the Biharmonic Equation and the Role of Squaring Matrices for Preconditioning*, IMA Journal of Numerical Analysis, 6 (1986), pp. 393–404.

[14] S. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, vol. 15 of Texts in Applied Mathematics, Springer-Verlag New York, 3rd ed., 2008.

[15] W. Briggs, V. Henson, and S. McCormick, *A Multigrid Tutorial: Second Edition*, Society for Industrial and Applied Mathematics, 2000.

[16] A. N. Brooks and T. J. Hughes, *Streamline Upwind/Petrov-Galerkin Formulation for Convection Dominated Flows with Particular Emphasis on the*

Incompressible Navier-Stokes Equations, Computer Methods in Applied Mechanics and Engineering, 32 (1982), pp. 199–259.

[17] R. Chan and X. Jin, *An Introduction to Iterative Toeplitz Solvers*, Society for Industrial and Applied Mathematics, 2007.

[18] T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM Journal on Scientific and Statistical Computing, 9 (1988), pp. 766–771.

[19] S. S. Collis and M. Heinkenschloss, *Analysis of the Streamline Upwind/Petrov Galerkin Method Applied to the Solution of Optimal Control Problems*, CAAM TR02-01, (2002).

[20] M. Desai and K. Ito, *Optimal controls of navier–stokes equations*, SIAM Journal on Control and Optimization, 32 (1994), pp. 1428–1446.

[21] I. Duff, A. Erisman, and J. Reid, *Direct Methods for Sparse Matrices*, Numerical Mathematics and Scientific Computation Series, Oxford Science Publications, 1989.

[22] J. Duintjer Tebbens and G. Meurant, *Any Ritz Value Behavior Is Possible for Arnoldi and for GMRES*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 958–978.

[23] ——, *Prescribing the Behavior of Early Terminating GMRES and Arnoldi Iterations*, Numerical Algorithms, 65 (2014), pp. 69–90.

[24] H. Egger and H. W. Engl, *Tikhonov Regularization Applied to the Inverse Problem of Option Pricing: Convergence Analysis and Rates*, Inverse Problems, 21 (2005), pp. 1027–1045.

[25] H. Elman, D. J. Silvester, and A. J. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 2nd ed., 2014.

[26] H. C. Elman, A. Ramage, and D. J. Silvester, *Algorithm 866: IFISS, a Matlab Toolbox for Modelling Incompressible Flow*, ACM Transactions on Mathematical Software, 33 (2007), pp. 2–14.

[27] H. C. Elman, A. Ramage, and D. J. Silvester, *IFISS: A Computational Laboratory for Investigating Incompressible Flow Problems*, SIAM Review, 56 (2014), pp. 261–273.

[28] M. Emmett and M. Minion, *Toward an Efficient Parallel in Time Method for Partial Differential Equations*, Communications in Applied Mathematics and Computational Science, 7 (2012), pp. 105–132.

[29] R. Falgout, S. Friedhoff, T. Kolev, S. MacLachlan, J. Schroder, and S. Vandewalle, *Multigrid Methods with Space-Time Concurrency*. Submitted, 2016.

[30] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel Time Integration with Multigrid*, SIAM Journal on Scientific Computing, 36 (2014), pp. C635–C661.

[31] M. Fisher, J. Nocedal, Y. Trémolet, and S. J. Wright, *Data Assimilation in Weather Forecasting: A Case Study in PDE-Constrained Optimization*, Optimization and Engineering, 10 (2009), pp. 409–426.

[32] D. Flanders and G. Shortley, *Numerical Determination of Fundamental Modes*, Journal of Applied Physics, 21 (1950), pp. 1326–1352.

[33] D. C.-L. Fong and M. Saunders, *LSMR: An Iterative Algorithm for Sparse Least-Squares Problems*, SIAM Journal of Scientific Computing, 33 (2011), pp. 2950–2971.

[34] T.-P. Fries, H. G. Matthies, et al., *A Review of Petrov-Galerkin Stabilization Approaches and an Extension to Meshfree Methods*, Technische Universitat Braunschweig, Brunswick, (2004).

[35] A. V. Fursikov, M. D. Gunzburger, and L. Hou, *Boundary value problems and optimal boundary control for the navier–stokes system: the two-dimensional case*, SIAM Journal on Control and Optimization, 36 (1998), pp. 852–894.

[36] M. J. Gander, Y.-L. Jiang, and R.-J. Li, *Domain Decomposition Methods in Science and Engineering XX*, vol. 60 of Lecture Notes in Computational Science and Engineering, 2013, ch. Parareal Schwarz Waveform Relaxation Methods, pp. 45–46.

[37] M. J. Gander and M. Neumüller, *Analysis of a New Space-Time Parallel Multigrid Algorithm for Parabolic Problems.* to appear in SIAM Journal of Scientific Computing, 2016.

[38] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing, 29 (2007), pp. 556–578.

[39] M. B. Giles and N. A. Pierce, *An introduction to the adjoint approach to design*, Flow, Turbulence and Combustion, 65 (2000), pp. 393–415.

[40] G. Golub and C. Van Loan, *Matrix Computations*, Matrix Computations, Johns Hopkins University Press, 4th ed., 2012.

[41] G. Golub and R. Varga, *Chebyshev Semi-Iteration Methods, Successive Over-Relaxation Iterative Methods, and Second Order Richardson Iterative Methods, Part 1*, Numerische Mathematik, 3 (1961), pp. 147–156.

[42] ——, *Chebyshev Semi-Iteration Methods, Successive Over-Relaxation Iterative Methods, and Second Order Richardson Iterative Methods, Part 2*, Numerische Mathematik, 3 (1961), pp. 157–168.

[43] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Society for Industrial and Applied Mathematics, 1997.

[44] A. Greenbaum, V. Ptak, and Z. Strakoš, *Any Nonincreasing Convergence Curve is Possible for GMRES*, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 465–469.

[45] A. Griewank and A. Walther, *Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation*, ACM Transactions on Mathematical Software (TOMS), 26 (2000), pp. 19–45.

[46] J.-L. Guermond, *Stabilization of Galerkin Approximations of Transport Equations by Subgrid Modeling*, Modélisation Mathématique et Analyse Numérique, 33 (1999), pp. 1293–1316.

[47] M. Gunzberger, *Perspectives in Flow Control and Optimization*, SIAM, 1987.

[48] S. Güttel, *Domain Decomposition Methods in Science and Engineering XX*, vol. 91 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2013, ch. A Parallel Overlapping Time-Domain Decomposition Method for ODEs, pp. 459–466.

[49] W. Hackbusch, *Multi-Grid Methods and Applications*, vol. 4 of Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, 1985.

[50] M. Heinkenschloss and D. Leykekhman, *Local Error Estimates for SUPG Solutions of Advection-Dominated Elliptic Linear-Quadratic Optimal Control Problems*, SIAM Journal on Numerical Analysis, 47 (2010), pp. 4607–4638.

[51] L. Hemmingsson, *A Semi-Circulant Preconditioner for the Convection-Diffusion Equation*, Numerische Mathematik, 81 (1998), pp. 211–248.

[52] M. Hinze, *A variational discretization concept in control constrained optimization: the linear-quadratic case*, Computational Optimization and Applications, 30 (2005), pp. 45–61.

[53] G. Horton and S. Vandewalle, *A Space-Time Multigrid Method for Parabolic Partial Differential Equations*, SIAM Journal of Scientific Computing, 16 (1995), pp. 848–864.

[54] T. Hughes and A. Brooks, *A Multi-Dimensioal Upwind Scheme with no Crosswind Diffusion*, vol. 34, ASME, 1979, pp. 19–35.

[55] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press, 2nd ed., 2008. Cambridge Books Online.

[56] B. S. Jovanović and E. Süli, *Analysis of Finite Difference Schemes*, vol. 46 of Springer Series in Computational Mathematics, Springer-Verlag London, 2014.

[57] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint Preconditioning for Indefinite Linear Systems*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1300–1317.

[58] C. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, 1995.

[59] R. C. Kirby, *From functional analysis to iterative methods*, SIAM review, 52 (2010), pp. 269–293.

[60] M. V. Klibanov and T. R. Lucas, *Numerical Solution of a Parabolic Inverse Problem in Optical Tomography Using Experimental Data*, SIAM Journal on Applied Mathematics, 59 (1999), pp. 1763–1789.

[61] Y. Kuznetsov, *Efficient Iterative Solvers for Elliptic Finite Element Problems on Nonmatching Grids*, Russ. J. Numer. Anal. Math. Modelling, 10 (1995), pp. 187–211.

[62] C. Lanczos, *An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators*, Journal of Research of the National Bureau of Standards, 45 (1950), pp. 255–282.

[63] ——, *Solution of Systems of Linear Equations by Minimized Iterations*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 33–53.

[64] A. Lawless, *Large Scale Inverse Problems: Computational Methods and Applications in the Earth Sciences*, vol. 13 of Radon Series on Computational and

Applied Mathematics, De Gruyter, 2013, ch. Variational data assimilation for very large environmental problems, pp. 55–90.

[65] J. Lions, Y. Maday, and G. Turinici, *A 'Parareal' in Time Discretization of PDEs*, Comptes Rendus de l'Academie des Sciences Series I Mathematics, 332 (2001), pp. 661–668.

[66] X. Liu, E. Chow, K. Vaidyanathan, and M. Smelyanskiy, *Improving the Performance of Dynamical Simulations via Multiple Right-Hand Sides*, in Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International, IEEE, 2012, pp. 36–47.

[67] D. Loghin and A. J. Wathen, *Analysis of Preconditioners for Saddle-Point Problems*, SIAM Journal of Scientific Computing, 25 (2004), pp. 2029–2049.

[68] K.-A. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numerical Linear Algebra with Applications, 18 (2011), pp. 1–40.

[69] E. McDonald, J. Pestana, and A. Wathen, *Preconditioning and Iterative Solution of All-at-Once Systems for Evolutionary Partial Differential Equations*, Tech. Rep. 2, University of Oxford, 2016.

[70] E. McDonald and A. Wathen, *A Simple Proposal for Parallel Computation Over Time of an Evolutionary Process with Implicit Time Stepping.* To Appear in Proceedings of ENUMATH 2015, the 11th European Conference on Numerical Mathematics and Advanced Applications, 2016.

[71] K. Morton, *Numerical Solution of Convection-Diffusion Problems*, no. 12 in Applied Mathematics and Mathematical Computation, London: Chapman & Hall, 1996.

[72] M. F. Murphy, G. H. Golub, and A. Wathen, *A Note on Preconditioning for Indefinite Linear Systems*, SIAM Journal of Scientific Computing, 21 (2000), pp. 1969–1972.

[73] A. Napov and Y. Notay, *Aggregation-Based Algebraic Multigrid for Convection-Diffusion Equations*, SIAM Journal of Scientific Computing, 34 (2012), pp. A2288–A2316.

[74] ——, *An Algebraic Multigrid Method with Guaranteed Convergence Rate*, SIAM Journal of Scientific Computing, 34 (2012), pp. A1079–A1109.

[75] M. Neumüller, *Space-Time Methods: Fast Solvers and Applications*, PhD thesis, Graz University of Technology, June 2013.

[76] M. K. Ng, *Iterative Methods for Toeplitz Systems*, Oxford University Press, Oxford, UK, 2004.

[77] Y. Notay, *AGMG Software and Documentation; see* `http://homepages.ulb.ac.be/~ynotay/AGMG`.

[78] Y. Notay, *An Aggregation-Based Algebraic Multigrid Method*, Electronic Transactions on Numerical Analysis, 37 (2010), pp. 123–146.

[79] Y. Notay, *Aggregation-Based Algebraic Multigrid for Convection-Diffusion Equations*, SIAM Journal of Scientific Computing, 34 (2012), pp. A2288–A2316.

[80] J. Ockendon, *Applied Partial Differential Equations*, Oxford Texts in Applied and Engineering Mathematics, Oxford University Press, 2003.

[81] J. A. Olkin, *Linear and Nonlinear Deconvolution Problems*, PhD thesis, Rice University, 1986.

[82] I. Oseledets and E. Tyrtyshnikov, *A Unifying Approach to the Construction of Circulant Preconditioners*, Linear Algebra and its Applications, 418 (2006), pp. 435–449.

[83] C. Paige and M. Saunders, *Solution of Sparse Indefinite Systems of Linear Equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

[84] C. C. Paige and M. A. Saunders, *LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares*, ACM Transactions on Mathematical Software, 8 (1982), pp. 43–71.

[85] B. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA, USA, classics ed., 1998.

[86] J. W. Pearson, *Fast Iterative Solvers for PDE-constrained Optimization Problems*, DPhil Thesis, University of Oxford, 2013.

[87] J. W. Pearson and M. Stoll, *Fast Iterative Solution of Reaction-Diffusion Control Problems Arising from Chemical Processes*, SIAM Journal on Scientific Computing, 35 (2013), pp. B987–B1009.

[88] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-Robust Preconditioners for Time-Dependent PDE-Constrained Optimization Problems*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 1126–1152.

[89] J. W. Pearson and A. J. Wathen, *A New Approximation of the Schur Complement in Preconditioners for PDE-Constrained Optimization*, Numerical Linear Algebra with Applications, 19 (2012), pp. 816–829.

[90] ——, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, Electronic Transactions on Numerical Analysis, 40 (2013), pp. 294–310.

[91] J. Pestana and A. J. Wathen, *A Preconditioned MINRES Method for Nonsymmetric Toeplitz Matrices*, SIAM Journal on Matrix Analysis and Applications, 36 (2015), pp. 273–288.

[92] A. M. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer Publishing Company, Incorporated, 1st ed. 1994. 2nd printing ed., 2008.

[93] A. Ramage, *A Multigrid Preconditioner for Stabilised Discretisation of Advection-Diffusion Problems*, Journal of Computational and Applied Mathematics, 110 (1999), pp. 187–203.

[94] T. Rees, *Preconditioning Iterative Methods for PDE Constrained Optimization*, DPhil Thesis, University of Oxford, 2010.

[95] T. Rees, H. S. Dollar, and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM Journal of Scientific Computing, 32 (2008), pp. 271–298.

[96] T. Rees, M. Stoll, and A. Wathen, *All-At-Once Preconditioning in PDE-Constrained Optimization*, Kybernetika, 46 (2010), pp. 341–360.

[97] J. Ruge and K. Stüben, *Multigrid Methods*, Society for Industrial Applied Mathematics, Philadelphia, Pennsylvania, 1987, ch. Algebraic Multigrid.

[98] Y. Saad, *A Flexible Inner-Outer Preconditioned GMRES Algorithm*, SIAM Journal of Scientific Computing, 14 (1993), pp. 461–469.

[99] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd ed., 2003.

[100] Y. Saad and M. H. Schultz, *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*, SIAM Journal of Scientific Computing, 7 (1986), pp. 856–869.

[101] S. Salsa, *Partial Differential Equations in Action: From Modelling to Theory*, Springer International Publishing, 2nd edition ed., 2015.

[102] J. A. Sifuentes, M. Embree, and R. B. Morgan, *GMRES Convergence for Perturbed Coefficient Matrices, with Application to Approximate Deflation Preconditioning*, SIAM Journal on Matrix Analysis and Applications, 34 (2013), pp. 1066–1088.

[103] D. Silvester, H. Elman, and A. Ramage, *Incompressible Flow and Iterative Solver Software (IFISS) Version 3.3*, October 2013. `http://www.manchester.ac.uk/ifiss/`.

[104] G. W. Stewart, *Perturbation theory for the singular value decomposition*, Tech. Rep. CS-TR 2539, University of Maryland, College Park, College Park, MDm USA, 1990.

[105] M. Stoll, *One-Shot Solution of a Time-Dependent Time-Periodic PDE-Constrained Optimization Problem*, IMA Journal of Numerical Analysis, 34 (2014), pp. 1554–1577.

[106] M. Stoll and T. Breiten, *A Low-Rank in Time Approach to PDE-Constrained Optimization*, SIAM Journal on Scientific Computing, 37 (2015), pp. B1–B29.

[107] M. Stoll, J. W. Pearson, and P. K. Maini, *Fast Solvers for Optimal Control Problems from Pattern Formation*, Journal of Computational Physics, 304 (2016), pp. 27–45.

[108] M. Stoll and A. Wathen, *All-At-Once Solution of Time-Dependent Stokes Control*, Journal of Computational Physics, 232 (2013), pp. 498–515.

[109] M. Stoll and A. J. Wathen, *All-At-Once Solution of Time-Dependent PDE-Constrained Optimization Problems*, Tech. Rep. Technical Report 10/47, Oxford Centre for Collaborative Applied Mathematics, 2010.

[110] G. Strang, *A Proposal for Toeplitz Matrix Calculations*, Studies in Applied Mathematics, 74 (1986), pp. 171–176.

[111] L. N. Trefethen and M. Embree, *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*, Princeton University Press, 2005.

[112] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, American Mathematical Society, 2010.

[113] S. Ulbrich, *Real-Time PDE-Constrained Optimization*, Society for Industrial and Applied Mathematics, 2007, ch. Generalized SQP Methods with Parareal Time-Domain Decomposition for Time-Dependent PDE-Constrained Optimization, pp. 145–168.

[114] H. van der Vorst, *Bi-CGSTAB: A Fast and Smoothly Convergent Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*, SIAM Journal of Scientific Computing, 13 (1992), pp. 631–644.

[115] H. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2003.

[116] C. Van Loan, *Compuational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, PA, USA, 1992.

[117] A. Wathen, *Realistic Eigenvalue Bounds for the Galerkin Mass Matrix*, IMA Journal of Numerical Analysis, 7 (1987), pp. 449–457.

[118] A. Wathen and T. Rees, *Chebyshev Semi-Iteration in Preconditioning for Problems Including the Mass Matrix*, Electronic Transactions on Numerical Analysis, 34 (2009), pp. 125–135.

[119] P. Wesseling, *An Introduction to Multigrid Methods*, R.T. Edwards, 2004.

[120] H. Weyl, *Das Asymptotische Verteilungsgesetz der Eigenwerte Linearer Partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung)*, Mathematische Annalen, 71 (1912), pp. 441–479.

[121] F. Yilmaz and B. Karasözen, *An All-At-Once Approach for the Optimal Control of the Unsteady Burgers Equation*, Journal of Computational and Applied Mathematics, 259, Part B (2014), pp. 771–779.

[122] H. Yücel, M. Heinkenschloss, and B. Karasözen, *Distributed Optimal Control of Diffusion-Convection-Reaction Equations using Discontinuous Galerkin Methods*, in Numerical Mathematics and Advanced Applications 2011, Springer Berlin Heidelberg, 2013, pp. 389–397.