University of Reading
School of Mathematical, Physical and Computational Sciences

ON THE PRECONDITIONING FOR WEAK CONSTRAINT FOUR-DIMENSIONAL VARIATIONAL DATA ASSIMILATION

Ieva Daužickaitė

Thesis submitted for the degree of Doctor of Philosophy

December 2021

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Ieva Daužickaitė

Publications

The work in chapters 4, 5, and 6 are strongly based on the following publications:

- Daužickaitė, I., Lawless, A.S., Scott, J.A. and van Leeuwen, P.J. (2021) Randomised preconditioning for the forcing formulation of weak-constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 147(740), 3719 - 3734.

- Daužickaitė, I., Lawless, A.S., Scott, J.A. and van Leeuwen, P.J. (2020) Spectral estimates for saddle point matrices arising in weak constraint four-dimensional variational data assimilation. *Numerical Linear Algebra with Applications*, 27(5): e2313.

- Daužickaitė, I., Lawless, A.S., Scott, J.A. and van Leeuwen, P.J. (2021) On time-parallel preconditioning for the state formulation of incremental weak constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 147(740), 3521 - 3529.

The work presented in these publications was undertaken by Ieva Daužickaitė with coauthors providing guidance and review.

Abstract

Data assimilation is used to obtain an improved estimate (analysis) of the state of a dynamical system by combining a previous estimate with observations of the system. A weak constraint four-dimensional variational assimilation (4D-Var) method accounts for the dynamical model error and is of large interest in numerical weather prediction. The analysis can be approximated by solving a series of large sparse symmetric positive definite (SPD) or saddle point linear systems of equations. The iterative solvers used for these systems require preconditioning for a satisfactory performance. In this thesis, we use randomised numerical methods to construct effective preconditioners that are cheap to construct and apply. We employ a randomised eigenvalue decomposition to construct limited memory preconditioners (LMPs) for a forcing formulation of 4D-Var independently of the previously solved systems. This preconditioning remains effective even if the subsequent systems change significantly. We propose a randomised approximation of a control variable transform technique (CVT) to precondition the SPD system of the state formulation, which preserves potential for a time-parallel model integration. A new way to include the observation information in the approximation of the inverse Schur complement in the block diagonal preconditioner for the saddle point formulations is introduced, namely applying the randomised LMPs. Numerical experiments with idealised systems show that the proposed preconditioners improve the performance of the iterative solvers. We provide theoretical results describing the change of the extreme eigenvalues of the unpreconditioned and preconditioned coefficient matrices when new observations of the dynamical system are added. These show that small positive eigenvalues can cause convergence issues. New eigenvalue bounds for the SPD and saddle point coefficient matrices in the state formulation emphasize their sensitivities to the observations. These results can guide the design of other preconditioners.

Acknowledgements

First of all, my heartfelt thanks goes to my supervisors Dr. Amos Lawless, Prof. Jennifer Scott, and Prof. Peter Jan van Leeuwen. Your enthusiasm, mentoring, and support, which was doubled when times got weird, was a necessary condition for this work to exist and meant a great deal to me. You helped me to improve in many ways.

The EPSRC funded Centre for Doctoral Training in Mathematics of Planet Earth merits a separate love letter for various reasons, including the funding of my studies. Herein I thank its staff for fostering a sense of community, and helping us to grow not only as mathematicians. A separate mention is warranted to Sam Williams and Jan Fillingham for their vast kindness and help in understanding the UK ways. I had a great time with other students, especially the cohort Δ and the ones we have assimilated, whether chatting over a cup of a very strong coffee [1] or going on Italy- and UK-based trips. Many more of these are to be planned and I cherish our friendship immensely. Thank you to everyone in the Data Assimilation Research Centre for shining light on data assimilation and being truly welcoming. The support of the community of the Department of Mathematics and Statistics is highly appreciated.

Considering the non-academic part of the world, my dear friends made sure to give me some perspective and that I spend enough book-free time. For that I am ever grateful, especially to the groups that grew to have the names of Šaka and GIS, and my housemate Paula.

I am incredibly lucky to have my family, which provided me with unbounded love and support in the endeavours that led me here. For that and your interest in what I am working on - *Ačiū jums*!

[1] I am NOT sorry for brewing it this way.

Contents

List of Tables

List of Figures

Abbreviations

4D-Var	Four-dimensional variational data assimilation
3D-Var	Three-dimensional variational data assimilation
CG	Conjugate gradient method
CVT	Control variable transform
ECMWF	European Centre for Medium-Range Weather Forecasts
EVD	Eigenvalue decomposition
LMP	Limited memory preconditioner
MINRES	Minimal residual method
NWP	Numerical weather prediction
REVD	Randomised eigenvalue decomposition
RR	Rayleigh-Ritz procedure
RSVD	Randomised singular value decomposition
SPD	Symmetric positive definite
SVD	Singular value decomposition

Chapter 1

Introduction

Data assimilation is used to obtain an estimate of a state of a dynamical system and is essential in numerous applications, including atmospheric reanalysis, environmental forecasting for hazards like flooding, and numerical weather prediction (NWP), where it provides initial conditions for the weather model [Dee et al., 2011, García-Pintado et al., 2015, Kalnay, 2002]. The forecasting requires integrating dynamical models and hence is an initial value problem. In NWP, the importance of accurate initial conditions is emphasised by the fact that the atmosphere is chaotic, that is, a small change in its state at one time can result in a very different state at a later time (e.g., [Kalnay, 2002]). Accurate forecasting and hence data assimilation is vital for governments and businesses to prepare for extreme events and a longer lead time makes the process easier [Kovats and Ebi, 2006, Doong et al., 2012].

Data assimilation combines observations of the dynamical system with a prior guess of the state (background) taking into account their error statistics. In NWP, there may be 10^8 observations, whereas the state may consist of $10^9 - 10^{10}$ variables [Bauer et al., 2021]. This is because it includes values for meteorological variables like temperature, wind direction, moisture, pressure and others at every grid point at multiple vertical levels [Coiffier, 2011]. There may be 10^7 grid points and 10^2 vertical levels of the atmosphere [Bauer et al., 2021]. The size of the system makes the problem extremely challenging. There is also a time constraint on the computations because of the commitment to issue forecasts at specific times. Efficient data assimilation is hence of vast importance and it requires suitable linear algebra techniques [Freitag, 2020].

Variational data assimilation methods are used in NWP centres like the European Centre for Medium-Range Weather Forecasts (ECMWF) and the UK Met Office [Bonavita and Lean, 2021, Clayton et al., 2013]. The weak constraint four dimensional variational (4D-Var) data assimilation method has received interest in the data assimilation community (e.g., [Fisher and Gürol, 2017, Shaw and Daescu, 2017, Bowler, 2017, Freitag and Green, 2018, Laloyaux et al., 2020a, Laloyaux et al., 2020b]), because it accounts for the error of the dynamical model and thus a more accurate estimate of the state of the system can be obtained. The state is estimated by minimising a series of quadratic cost functions (inner loop minimisations) that require the solution of very large sparse systems of linear

equations. One can choose to write these systems with symmetric positive definite (SPD) or symmetric saddle point coefficient matrices depending on factors such as the need for parallel computations [Fisher and Gürol, 2017].

These systems are solved using iterative methods of which the Krylov subspace methods are the most popular, but their convergence can be slow [Saad, 2003]. It is accepted that preconditioning is needed to improve their performance (for example, [Benzi, 2002, Wathen, 2015]). In this technique, a modified (preconditioned) system is solved. The preconditioning has to be designed in a way that the preconditioned system is solved faster than the original one and the solution of the original system is easily recovered. If the original system has potential for parallel computations, then preconditioning needs to preserve it. The design process is thus highly problem dependent. Well-known results show a relationship between the convergence of Krylov subspace solvers and the eigenvalues of the coefficient matrices (e.g., Lecture 38 of [Trefethen and Bau, III, 1997]). This suggests that effective preconditioning can be obtained when the eigenvalue distribution of the preconditioned coefficient matrix is better suited for the solver than the original one. A way to achieve this is to use an approximation of the inverse of the coefficient matrix for preconditioning. These often make use of low-rank matrix approximations.

Randomised methods for low-rank approximations of a matrix $\mathbf{A}$ constitute a flourishing research theme in numerical linear algebra (see, e.g., [Martinsson and Tropp, 2020] and references therein). They have been also used to design solvers for strong constraint 4D-Var [Bousserez et al., 2020]. The randomised methods consist of two stages: first, a random matrix is used to generate a subspace that approximates the range of $\mathbf{A}$, and then this subspace is used to construct a small matrix whose cheap to obtain decomposition is employed to approximate $\mathbf{A}$. It has been shown that a high quality approximation of $\mathbf{A}$ can be constructed with high probability when $\mathbf{A}$ has rapidly decreasing singular values [Halko et al., 2011]. The randomised methods are designed as block methods, that is they require matrix-matrix products, which can be parallelised and hence are suitable for efficient computations on current computers. We use randomised methods to construct preconditioners for weak constraint 4D-Var.

1.1 Thesis aims

In this thesis, we consider the following research questions related to preconditioning the weak constraint 4D-Var.

1. How can we precondition the linear systems of equations arising in the so-called forcing formulation of the weak constraint 4D-Var independently of the previously solved systems? Current preconditioning practices depend on the previous inner loops [Tshimanga et al., 2008], and may not be successful when the linear systems change significantly from one inner loop to the next.

2. How do the extreme eigenvalues of the coefficient matrices change when new observations are introduced? The number of observations that are used in data assimilation

for NWP is ever increasing and this can affect the convergence of the Krylov subspace solvers.

3. How can we precondition the linear systems in the so-called state formulation so that the potential for time-parallel computations is preserved? In the state formulation, the model linearised at different times can be integrated in parallel [Fisher and Gürol, 2017] and hence effective preconditioning should not impair this.

4. How can we include more information about the observations when preconditioning the saddle point systems? This may improve the performance of the preconditioning when more observations of the dynamical system are used in data assimilation.

1.2 Outline

In **Chapter 2**, we introduce the theory of data assimilation. We start with the required components and introduce the strong constraint 4D-Var method. We then discuss ways to precondition the linear systems that arise in the incremental formulation of the method, namely using the control variable transform (CVT), also known as first level preconditioning, and limited memory preconditioners (LMPs). These ideas are then used in designing preconditioning for the weak constraint method. We formulate three linear systems that can be solved in the incremental formulation, specifically two SPD systems and a 3×3 block saddle point system. A discussion of their features as well as previous attempts to precondition these systems finalises the chapter.

In **Chapter 3**, mathematical ideas and results that form the basis for our work are presented. We state results for specific matrices, their decompositions and relationships concerning their eigenvalues and singular values. Low-rank matrix approximations for large matrices are presented, focusing on the eigenvalue and singular value decompositions. We discuss traditional Lanczos and subspace iteration methods used for these decompositions and introduce randomised methods. The conjugate gradient and minimal residual methods for solving linear systems are presented; the results connecting their convergence with the distribution of the eigenvalues are shown. We also describe mathematical properties of the LMPs and block diagonal Schur complement preconditioners.

In **Chapter 4**, research question 1 is addressed. LMPs have been used in strong constraint 4D-Var, where they have been constructed using estimates of eigenvalues and eigenvectors obtained cheaply in the previous inner loop [Tshimanga et al., 2008]. Such preconditioners can be expected to perform well if the coefficient matrices do not change significantly from one loop to another. We propose a way to construct LMPs in every inner loop independently of the previous ones. This is done using a randomised eigenvalue decomposition and we explore three variants. Numerical experiments with the advection and Lorenz 96 models show that these preconditioners improve the performance of the solvers.

In **Chapter 5**, we consider research question 2 for the linear systems in the state formulation. A 3×3 block saddle point system has been introduced and studied by [Fisher

and Gürol, 2017]. We introduce a reduced 2×2 block saddle point system that can also be used in this formulation. We describe how the extreme eigenvalues of the 3×3 block and 2×2 block saddle point and the SPD coefficient matrices change when new observations are introduced. These theorems hold for the general observation error covariance matrix in the 3×3 block system, and for a diagonal observation error covariance matrix in the other systems. We also provide bounds for the eigenvalues of the coefficient matrices. These results can be used to better understand how to design effective preconditioners that remain useful when the number of observations is increased.

In **Chapter 6**, research question 3 is examined for the SPD system in the state formulation. [Fisher and Gürol, 2017] suggested extending the CVT technique by using an approximation of the linearised model, but did not recommend a suitable approximation. We propose using a randomised singular value decomposition to construct an approximation which preserves the time-parallelism. A way to incorporate the background and model error in this approximation is also explored. It is shown that the exact CVT technique is not always useful but that the randomised preconditioner can give good results when the CVT performs well.

In **Chapter 7**, we tackle research question 4, and research question 2 for the SPD system with a diagonal error covariance matrix in the forcing formulation. We focus on the block diagonal Schur preconditioners. These are constructed with an approximation of the inverse of the Schur complement. The Schur complement in our application coincides with the SPD coefficient matrix in the state formulation. We suggest approximating the inverse of it using the cheap to construct and apply randomised LMPs, which were also used in Chapter 4. They include information about the observations of the dynamical system, which was omitted in previous attempts to precondition [Gratton et al., 2018a, Freitag and Green, 2018, Tabeart and Pearson, 2021]. The importance of this information is expected to grow when the number of observations increases. We further provide results on the eigenvalues of the preconditioned 3×3 block and 2×2 block matrices, and show how a specific choice of preconditioner relates the eigenvalues of the preconditioned systems to the eigenvalues of the SPD systems in the state and forcing formulations.

In **Chapter 8**, we conclude the thesis and provide an outlook for future endeavours.

Chapter 2

Data assimilation

In this chapter, we present the strong constraint 4D-Var data assimilation method and its incremental formulation. Preconditioning approaches for the latter are discussed. The weak constraint 4D-Var method is presented in relation to the strong constraint formulation. We discuss two formulations of incremental weak constraint 4D-Var and present large sparse linear systems of equations, whose solution provides the analysis update. The chapter is concluded with an overview of previously used preconditioning techniques for the three systems of equations.

2.1 Strong constraint 4D-Var

Consider the evolution of a dynamical system over a time window, which is described by taking snapshots of the state of the system at discrete times, i.e., using vectors $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$, $\mathbf{x}_i \in \mathbb{R}^n$ that define the state at times $t_0 < t_1 < \cdots < t_N$. Data assimilation provides an estimate of this trajectory, called the analysis, by combining a prior estimate and observations of the system [Kalnay, 2002].

A dynamical model $\mathcal{M}_i$ describes the evolution of the system from time t_i to t_{i+1}. When complicated systems are simulated, the model is imperfect and does not describe all the physical processes that influence the state, includes approximations, and the represented processes can be poorly resolved [Warner, 2010]. Model discretisation introduces more errors, because processes that occur on smaller spatial scales than the grid are not resolved [Coiffier, 2011]. Hence, the trajectory obtained by integrating the model for a long time may be far from the real state of the system (see, for example, [Kalnay, 2002, Allen et al., 2006, Bauer et al., 2015]).

Observations of the system at time t_i are denoted $\mathbf{y}_i \in \mathbb{R}^{q_i}$. They can come from various sources, for example, in NWP observations come from weather stations, satellites, ocean buoys and so on (e.g., [Rabier, 2005]). The nonlinear observation operator $\mathcal{H}_i : \mathbb{R}^n \to \mathbb{R}^{q_i}$ maps the model state $\mathbf{x}_i$ to the observation space, that is, the following holds

$$\mathbf{y}_i = \mathcal{H}_i(\mathbf{x}_i^t) + \boldsymbol{\epsilon}_i, \tag{2.1}$$

where $\mathbf{x}_i^t$ is the true state of the dynamical system at time t_i and $\boldsymbol{\epsilon}_i$ is the observation error. $\mathcal{H}_i$ can be complicated, for example, when the observations of the cloud related

reflectances are compared to the state described by meteorological variables like temperature, wind speed an so on (e.g., [Kostka et al., 2014]). The observation error $\boldsymbol{\epsilon}_i$ arises due to the instruments that are used to obtain the measurements and how the observations are represented in data assimilation system, for example, different scales of the observations and the model, and error in the observation operator [Andersson and Thépaut, 2010, Janjić et al., 2018]. In NWP, not all of the state is observed.

The prior estimate $\mathbf{x}^b \in \mathbb{R}^n$ called the background usually comes from a short range forecast, i.e., running the numerical model for a short period of time. Hence, errors in $\mathbf{x}^b$ contain the model errors. If the short range forecast was initialised by a previous analysis, then $\mathbf{x}^b$ includes information from the previously used observations (see, e.g., [Bannister, 2008a] for a general discussion).

The information on the errors of the data is incorporated into the data assimilation process. The exact errors are unknown and only the statistics of the errors are available. A common assumption, which we use in this thesis, is that the errors are Gaussian, hence they can be described using the mean and error covariances. It is assumed that the errors have zero mean. The error covariance matrices are denoted as follows: $\mathbf{B}$ for the background and $\mathbf{R}_i$ for the observation error at time t_i. It is assumed that the observation errors are not correlated in time.

A variational approach to data assimilation requires minimising a cost function to find the analysis. We present the strong constraint four dimensional variational (4D-Var) cost function:

$$J(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}(\mathbf{x}_i))^T\mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)), \tag{2.2}$$

where $\mathbf{x}_i$ satisfies the strong model constraint

$$\mathbf{x}_i = \mathcal{M}_{i-1}(\mathbf{x}_{i-1}), \tag{2.3}$$

i.e., it is assumed that the model errors can be ignored. In this case, the analysis trajectory is defined by the state at the initial time $\mathbf{x}_0$.

Evaluating (2.2) or its gradient requires integrating the nonlinear model $\mathcal{M}_i$, which can be expensive for large systems, and the minimisation of a nonlinear function is a hard problem [Nocedal and Wright, 2006]. We discuss a way to approximate the solution in the following section.

2.1.1 Incremental 4D-Var

[Courtier et al., 1994] proposed approximating the analysis using the incremental approach. Consider a perturbation $\delta\mathbf{x}_i$ to $\mathbf{x}_i$. Then the first-order approximations of the nonlinear model and observation operator are

$$\mathcal{M}_i(\mathbf{x}_i + \delta\mathbf{x}_i) \approx \mathcal{M}_i(\mathbf{x}_i) + \mathbf{M}_i\delta\mathbf{x}_i, \tag{2.4}$$

$$\mathcal{H}_i(\mathbf{x}_i + \delta\mathbf{x}_i) \approx \mathcal{H}_i(\mathbf{x}_i) + \mathbf{H}_i\delta\mathbf{x}_i, \tag{2.5}$$

where

$$\delta \mathbf{x}_{i+1} = \mathbf{M}_i \delta \mathbf{x}_i. \tag{2.6}$$

$\mathbf{M}_i$ and $\mathbf{H}_i$ are the Jacobian matrices of $\mathcal{M}_i$ and $\mathcal{H}_i$, respectively, that is they are the model and observation operators linearised at $\mathbf{x}_i$, and they are known as the tangent linear model and tangent linear observation operator in the data assimilation literature.

The analysis is approximated sequentially with $(j+1)$th approximation

$$\mathbf{x}_0^{(j+1)} = \mathbf{x}_0^{(j)} + \delta \mathbf{x}_0^{(j)}, \tag{2.7}$$

where $\delta \mathbf{x}_0^{(j)}$ minimises the quadratic cost function

$$\begin{aligned} J^{\delta}(\delta \mathbf{x}_0^{(j)}) = &(\delta \mathbf{x}_0^{(j)} - (\mathbf{x}^b - \mathbf{x}_0^{(j)}))^T \mathbf{B}^{-1} (\delta \mathbf{x}_0^{(j)} - (\mathbf{x}^b - \mathbf{x}_0^{(j)})) \\ &+ \frac{1}{2} \sum_{i=0}^{N} (\mathbf{H}_i^{(j)} \delta \mathbf{x}_i^{(j)} - (\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i^{(j)})))^T \mathbf{R}_i^{-1} (\mathbf{H}_i^{(j)} \delta \mathbf{x}_i^{(j)} - (\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i^{(j)}))), \end{aligned} \tag{2.8}$$

subject to the linear model constraint (2.6). The incremental approach is organised into inner and outer loops. The inner loop consists of minimising (2.8). The outer loop includes linearising the model and observation operator at $\mathbf{x}_i^{(j)}$, evaluating $\mathbf{x}^b - \mathbf{x}_0^{(j)}$ and $\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i^{(j)})$ subject to the model constraint (2.3), and updating (2.7). The first outer loop is usually started by setting $\mathbf{x}_0^{(0)} = \mathbf{x}^b$. This is a Gauss-Newton method [Gratton et al., 2007] and $\mathbf{x}_0^{(j)}$ converges to the minimiser of (2.2) under certain conditions (see, for example, Section 10.3 of [Nocedal and Wright, 2006]).

To gain computational savings the linearised model in the inner loop can be run at a coarser resolution and with simpler representation of some physical processes than the nonlinear model in the outer loop [Trémolet, 2004]. The resolution of the linearised model can be increased in the subsequent inner loops when $\mathbf{x}_0^{(j)}$ is expected to be closer to the solution [Veerse and Thépaut, 1998], as is done at the ECMWF [ECMWF, 2020]. The inner loop solver is usually stopped after a fixed number of iterations before convergence is reached, for example, the China Meteorological Administration allows 50 iterations [Zhang et al., 2019]. Sufficient conditions for convergence of $\mathbf{x}_0^{(j)}$ as j increases are presented by [Gratton et al., 2007]. In operational settings, only a small number of outer loops is performed, e.g., three outer loops at the ECMWF [ECMWF, 2020].

Because (2.8) is a quadratic cost function, its minimum can be found by solving a large sparse linear system (e.g., Chapter 5 of [Nocedal and Wright, 2006]). We introduce the

following notation to define the system

$$\hat{\mathbf{H}}^{(j)} = \begin{pmatrix} \mathbf{H}_0^{(j)} \\ \mathbf{H}_1^{(j)}\mathbf{M}_{0,0}^{(j)} \\ \mathbf{H}_2^{(j)}\mathbf{M}_{0,1}^{(j)} \\ \vdots \\ \mathbf{H}_N^{(j)}\mathbf{M}_{0,N-1}^{(j)} \end{pmatrix} \in \mathbb{R}^{q\times n}, \tag{2.9}$$

$$\mathbf{R} = diag(\mathbf{R}_0, \ldots, \mathbf{R}_N) \in \mathbb{R}^{q\times q}, \tag{2.10}$$

$$\mathbf{d}^{(j)} = \begin{pmatrix} \mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0^{(j)}) \\ \mathbf{y}_1 - \mathcal{H}_1(\mathbf{x}_1^{(j)}) \\ \vdots \\ \mathbf{y}_N - \mathcal{H}_N(\mathbf{x}_N^{(j)}) \end{pmatrix} \in \mathbb{R}^{q}, \tag{2.11}$$

where $q = \Sigma_{i=0}^{N} q_i$, $\mathbf{R}$ is block diagonal and

$$\mathbf{M}_{i,l}^{(j)} = \mathbf{M}_l^{(j)} \ldots \mathbf{M}_i^{(j)} \tag{2.12}$$

denotes the linearised model integration from time t_i to t_{l+1}.

The update $\delta\mathbf{x}_0^{(j)}$ is the solution of

$$\mathcal{A}\delta\mathbf{x}_0^{(j)} = \mathbf{B}^{-1}(\mathbf{x}^b - \mathbf{x}_0^{(j)}) + (\hat{\mathbf{H}}^{(j)})^T\mathbf{R}^{-1}\mathbf{d}^{(j)}, \tag{2.13}$$

$$\text{where } \mathcal{A} = \mathbf{B}^{-1} + (\hat{\mathbf{H}}^{(j)})^T\mathbf{R}^{-1}\hat{\mathbf{H}}^{(j)}. \tag{2.14}$$

$\mathcal{A} \in \mathbb{R}^{n\times n}$ is the Hessian of (2.8) and it is symmetric positive definite. In operational settings, matrices $\mathbf{B}$, $\mathbf{R}$ and $\hat{\mathbf{H}}$ are too large to be formed explicitly and only operators that return the matrix-vector products with these matrices are available (e.g., [Bannister, 2008b, Fisher et al., 2009]). Hence, iterative methods are used to solve (2.13). The conjugate gradient (CG) method is often used for such systems [Fisher, 1998]. We further discuss a technique used to improve the CG performance.

2.1.2 Preconditioning

The idea of preconditioning is to map the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.15}$$

where $\mathbf{A} \in \mathbb{R}^{m\times m}$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^m$, to another system which can be solved to a desired accuracy faster. We denote the preconditioner $\mathbf{P} \in \mathbb{R}^{m\times m}$. It can be applied in the following ways (e.g., Chapter 9 of [Saad, 2003]):

- left preconditioning

$$\mathbf{P}\mathbf{A}\mathbf{x} = \mathbf{P}\mathbf{b}, \tag{2.16}$$

- right preconditioning

$$\mathbf{A}\mathbf{P}\tilde{\mathbf{x}} = \mathbf{b}, \tag{2.17}$$

$$\text{where } \mathbf{P}\tilde{\mathbf{x}} = \mathbf{x}, \tag{2.18}$$

- split preconditioning

$$\mathbf{C}^T\mathbf{A}\mathbf{C}\tilde{\mathbf{x}} = \mathbf{C}^T\mathbf{b}, \tag{2.19}$$

$$\text{where } \mathbf{P} = \mathbf{C}\mathbf{C}^T, \tag{2.20}$$

$$\mathbf{C}\tilde{\mathbf{x}} = \mathbf{x}. \tag{2.21}$$

$\mathbf{P}$ and $\mathbf{C} \in \mathbb{R}^{m\times m}$ are nonsingular. The choice of left, right, or split preconditioning is problem dependent, as is the construction of $\mathbf{P}$. We detail the desired mathematical properties of $\mathbf{P}$ in Section 3.6.

When solving (2.13), CG convergence depends on the distribution of the eigenvalues of $\mathcal{A}$ with eigenvalues clustered away from zero expected to give fast convergence (see Section 3.4 for a more in depth discussion). We discuss preconditioning strategies that are used in data assimilation and aim to improve the CG convergence by changing the distribution of the eigenvalues in the following sections.

2.1.3 First level preconditioning

The background covariance matrix $\mathbf{B}$ is badly conditioned and difficult to estimate [Bannister, 2008a]. A control variable transform (CVT) technique has been used in operational centres to tackle this [Rabier et al., 2000, Lorenc et al., 2000, Gauthier et al., 2007]. A change of variable is introduced

$$\delta\mathbf{x}_0^{(j)} = \mathbf{B}^{1/2}\delta\mathbf{w}_0^{(j)}, \tag{2.22}$$

where $\mathbf{B}^{1/2}$ is the unique symmetric square root of $\mathbf{B}$. $\mathbf{B}^{1/2}$ is usually obtained by using knowledge of the physical system and choosing new variables $\delta\mathbf{w}_0$ that are uncorrelated with each other [Lawless, 2013]. Then the error covariance matrix is the identity. This is equivalent to first level (split) preconditioning for (2.13) with $\mathbf{B} = \mathbf{B}^{1/2}\mathbf{B}^{1/2}$ [Haben et al., 2011b]. The preconditioned system is then

$$\mathcal{A}^{pr}\delta\mathbf{w}_0^{(j)} = \mathbf{B}^{-1/2}(\mathbf{x}^b - \mathbf{x}_0^{(j)}) + \mathbf{B}^{1/2}(\hat{\mathbf{H}}^{(j)})^T\mathbf{R}^{-1}\mathbf{d}^{(j)}, \tag{2.23}$$

$$\text{where } \mathcal{A}^{pr} = \mathbf{I} + \mathbf{B}^{1/2}(\hat{\mathbf{H}}^{(j)})^T\mathbf{R}^{-1}\hat{\mathbf{H}}^{(j)}\mathbf{B}^{1/2} \tag{2.24}$$

If the dynamical system is not fully observed, the term $\mathbf{B}^{1/2}(\hat{\mathbf{H}}^{(j)})^T\mathbf{R}^{-1}\hat{\mathbf{H}}^{(j)}\mathbf{B}^{1/2}$ is symmetric positive semi-definite and has zero and positive eigenvalues. Hence, the preconditioned Hessian $\mathcal{A}^{pr}$ is symmetric positive definite with smallest eigenvalues equal to one. It has been shown that first level preconditioning with $\mathbf{B} = \mathbf{B}^{1/2}\mathbf{B}^{1/2}$ improves the conditioning (e.g., [Haben et al., 2011a, Haben et al., 2011b, Haben, 2011]).

To increase the potential for parallel calculations when solving (2.23), [Bousserez and Henze, 2018, Bousserez et al., 2020] introduced a randomised solution algorithm Randomized Incremental Optimal Technique (RIOT). RIOT uses a randomised eigenvalue decomposition to estimate a few leading eigenvectors of $\mathcal{A}^{pr}$ and these are employed to construct an approximation to $\delta\mathbf{x}_0^{(j)}$. We discuss the randomised methods for matrix decompositions in more detail in Section 3.3.3. Due to the nature of the randomised algorithms, the quality of the approximation to $\delta\mathbf{x}_0^{(j)}$ strongly depends on how rapidly the largest eigenvalues of $\mathcal{A}^{pr}$ decrease.

2.1.4 Second level preconditioning

Second level preconditioning can be used to further improve the convergence rate when solving (2.23).

Because the smallest eigenvalues of $\mathcal{A}^{pr}$ are equal to one, it is natural to aim to reduce the largest eigenvalues. Limited memory preconditioners (LMPs) described by [Tshimanga et al., 2008, Gratton et al., 2011, Tshimanga, 2007] have been used in operational centres [Moore et al., 2011, Mogensen et al., 2012, Laloyaux et al., 2018]. These preconditioners are constructed with estimates of the largest eigenvalues and eigenvectors (eigenpairs) of $\mathcal{A}^{pr}$ and aim to reduce the largest eigenvalues of the preconditioned system (we discuss LMPs in Section 3.6.1).

[Tshimanga et al., 2008] detailed a computationally cheap way to obtain the eigenpair estimates. A Lanczos and CG connection can be used to obtain estimates of the extreme eigenvalues and eigenvectors of the coefficient matrix after the system is solved using CG (see Section 4.4.4 for a detailed description). Because in data assimilation we solve a sequence of systems with coefficient matrices $(\mathcal{A}^{pr})^{(1)}, (\mathcal{A}^{pr})^{(2)}, (\mathcal{A}^{pr})^{(3)}, \ldots$, the inner loops can be performed in the following way:

1. Solve unpreconditioned system with the coefficient matrix $(\mathcal{A}^{pr})^{(1)}$;
2. Use Lanczos and CG connection to obtain estimates of a few eigenpairs of $(\mathcal{A}^{pr})^{(1)}$;
3. Use these eigenpairs to construct an LMP in a factored form $\mathcal{P}^{(1)} = \mathcal{T}^{(1)}\mathcal{T}^{(1)\,T}$;
4. Solve the preconditioned system with the preconditioned coefficient matrix $\mathcal{T}^{(1)\,T}(\mathcal{A}^{pr})^{(2)}\mathcal{T}^{(1)}$;
5. Use Lanczos and CG connection to obtain estimates of a few eigenpairs of $\mathcal{T}^{(1)\,T}(\mathcal{A}^{pr})^{(2)}\mathcal{T}^{(1)}$;
6. Use these eigenpairs to construct an LMP $\mathcal{P}^{(2)} = \mathcal{T}^{(2)}\mathcal{T}^{(2)\,T}$;
7. Solve the preconditioned system with the preconditioned coefficient matrix $\mathcal{T}^{(2)\,T}\mathcal{T}^{(1)\,T}(\mathcal{A}^{pr})^{(2)}\mathcal{T}^{(1)}\mathcal{T}^{(2)}$;
8. Continue for subsequent inner loops.

Because the Lanczos and CG connection returns eigenpairs of the preconditioned coefficient matrix, e.g., $\mathcal{T}^{(1)}(\mathcal{A}^{pr})^{(2)}\mathcal{T}^{(1)}$, the preconditioners are used in all subsequent inner loops. Hence, the preconditioner becomes more expensive to apply. Preconditioning for the first system with $(\mathcal{A}^{pr})^{(1)}$ is not defined.

2.2 Weak constraint 4D-Var

In the weak constraint 4D-Var data assimilation method the perfect model assumption is rejected and hence model errors are taken into account (see, for example, [Trémolet, 2006, Trémolet, 2007]). Various formulations of the weak constraint 4D-Var cost function

have been described by [Trémolet, 2006]. We consider two of them in the thesis. The model is a weak constraint, i.e.,

$$\mathbf{x}_i = \mathcal{M}_{i-1}(\mathbf{x}_{i-1}) + \boldsymbol{\eta}_i, \tag{2.25}$$

where $\boldsymbol{\eta}_i \in \mathbb{R}^n$ is random Gaussian model error with zero mean and covariance matrix $\mathbf{Q}_i \in \mathbb{R}^{n\times n}$. In this thesis, we assume that the model error is not correlated in time.

The forcing formulation minimises a cost function that estimates $\mathbf{x}_0$ and $\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \ldots, \boldsymbol{\eta}_N$:

$$J_f(\mathbf{x}_0, \boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_N) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i))^T\mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)) \tag{2.26}$$
$$+ \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{\eta}_i^T\mathbf{Q}_i^{-1}\boldsymbol{\eta}_i,$$

where $\mathbf{x}_i$ satisfies the weak model constraint (2.25) (e.g., [Trémolet, 2006]). Obtaining the full trajectory $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$ requires integrating the model over the time window sequentially.

Alternatively, the cost function can operate on the full trajectory $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$. This gives the cost function known as a state formulation (e.g., [Trémolet, 2006]):

$$J_s(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i))^T\mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)) \tag{2.27}$$
$$+ \frac{1}{2}\sum_{i=0}^{N-1}(\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i))^T\mathbf{Q}_{i+1}^{-1}(\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i)).$$

In both forcing and state formulations, the solution is now $N+1$ times larger than in the strong constraint case. This requires a lot of computational resources and makes the efficiency of the solution methods even more important and we focus on this in the thesis. The size of the problem is reduced in a variant of (2.25), where the model error is assumed to be constant; this is used to correct the model bias in a part of stratosphere at ECMWF [Leutbecher et al., 2017, Laloyaux et al., 2020b], but it does not account for the random error. Estimating $\mathbf{Q}_i$ is another challenging area of research, because it is problematic to separate the observation and model error [Laloyaux et al., 2020a].

2.2.1 Incremental weak constraint 4D-Var

The incremental formulation in Section 2.1.1 can be extended for the weak constraint case. We introduce the following vectors (following [Gratton et al., 2018b]).

$$\mathbf{p}^{(j)} = \begin{pmatrix} \mathbf{x}_0^{(j)} \\ \boldsymbol{\eta}_1^{(j)} \\ \vdots \\ \boldsymbol{\eta}_N^{(j)} \end{pmatrix}, \delta\mathbf{p}^{(j)} = \begin{pmatrix} \delta\mathbf{x}_0^{(j)} \\ \delta\boldsymbol{\eta}_1^{(j)} \\ \vdots \\ \delta\boldsymbol{\eta}_N^{(j)} \end{pmatrix}, \mathbf{b}^{(j)} = \begin{pmatrix} \mathbf{x}^b - \mathbf{x}_0^{(j)} \\ -\boldsymbol{\eta}_1^{(j)} \\ \vdots \\ -\boldsymbol{\eta}_N^{(j)} \end{pmatrix},$$

$$\mathbf{x}^{(j)} = \begin{pmatrix} \mathbf{x}_0^{(j)} \\ \mathbf{x}_1^{(j)} \\ \vdots \\ \mathbf{x}_N^{(j)} \end{pmatrix}, \ \delta\mathbf{x}^{(j)} = \begin{pmatrix} \delta\mathbf{x}_0^{(j)} \\ \delta\mathbf{x}_1^{(j)} \\ \vdots \\ \delta\mathbf{x}_N^{(j)} \end{pmatrix},$$

where $\mathbf{p}^{(j)}, \delta\mathbf{p}^{(j)}, \mathbf{b}^{(j)}, \mathbf{x}^{(j)}, \delta\mathbf{x}^{(j)} \in \mathbb{R}^{n(N+1)}$. We define matrices

$$\mathbf{L}^{(j)} = \begin{pmatrix} \mathbf{I} & & & & \\ -\mathbf{M}_0^{(j)} & \mathbf{I} & & & \\ & -\mathbf{M}_1^{(j)} & \mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{M}_{N-1}^{(j)} & \mathbf{I} \end{pmatrix} \in \mathbb{R}^{n(N+1)\times n(N+1)}, \tag{2.28}$$

$$(\mathbf{L}^{-1})^{(j)} = \begin{pmatrix} \mathbf{I} & & & & \\ \mathbf{M}_{0,0}^{(j)} & \mathbf{I} & & & \\ \mathbf{M}_{0,1}^{(j)} & \mathbf{M}_{1,1}^{(j)} & \underline{\mathbf{I}} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{M}_{0,N-1}^{(j)} & \mathbf{M}_{1,N-1}^{(j)} & \cdots & \mathbf{M}_{N-1,N-1}^{(j)} & \mathbf{I} \end{pmatrix} \in \mathbb{R}^{n(N+1)\times n(N+1)}, \tag{2.29}$$

$$\mathbf{H}^{(j)} = diag(\mathbf{H}_0^{(j)}, \ldots, \mathbf{H}_N^{(j)}) \in \mathbb{R}^{q\times n(N+1)}, \tag{2.30}$$

$$\mathbf{D} = diag(\mathbf{B}, \mathbf{Q}_1, \ldots, \mathbf{Q}_N) \in \mathbb{R}^{n(N+1)\times n(N+1)}, \tag{2.31}$$

where $\mathbf{M}_{i,l}^{(j)}$ is as in (2.12) and matrices $\mathbf{H}^{(j)}$ and $\mathbf{D}$ are block diagonal.

For the forcing formulation, we update

$$\mathbf{p}^{(j+1)} = \mathbf{p}^{(j)} + \delta\mathbf{p}^{(j)}, \tag{2.32}$$

where $\delta\mathbf{p}^{(j)}$ is found as the minimiser of

$$J_f^{\delta}(\delta\mathbf{p}^{(j)}) = \frac{1}{2}||\delta\mathbf{p}^{(j)} - \mathbf{b}^{(j)}||^2_{\mathbf{D}^{-1}} + \frac{1}{2}||\mathbf{H}^{(j)}(\mathbf{L}^{-1})^{(j)}\delta\mathbf{p}^{(j)} - \mathbf{d}^{(j)}||^2_{\mathbf{R}^{-1}}, \tag{2.33}$$

where $\|\mathbf{a}\|^2_{\mathbf{A}} = \mathbf{a}^T\mathbf{A}\mathbf{a}$ for an SPD $\mathbf{A}$, $\mathbf{d}^{(j)}$ and $\mathbf{R}$ are as in (2.11) and (2.10), respectively.

For the state formulation, the update is

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} + \delta\mathbf{x}^{(j)}, \tag{2.34}$$

where $\delta\mathbf{x}^{(j)}$ minimises

$$J_s^{\delta}(\delta\mathbf{x}^{(j)}) = \frac{1}{2}||\mathbf{L}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{b}^{(j)}||^2_{\mathbf{D}^{-1}} + \frac{1}{2}||\mathbf{H}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{d}^{(j)}||^2_{\mathbf{R}^{-1}}. \tag{2.35}$$

We omit the superscript (j) in the following sections.

The differences in the sensitivities of both forcing and state formulations and their conditioning has been studied in a thesis by [El-Said, 2015]. As in the strong constraint formulation, (2.33) and (2.35) can be minimised by solving linear systems of equations. We introduce three formulations and discuss their different potential for parallel computations in the following section.

2.2.2 Linear systems

The update $\delta\mathbf{p}$ in the forcing formulation is also a solution of the following system

$$\mathcal{A}_f\delta\mathbf{p} = \mathbf{D}^{-1}\mathbf{b}^{(j)} + \mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}, \tag{2.36}$$

$$\text{where } \mathcal{A}_f = (\mathbf{D}^{-1} + \mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}) \in \mathbb{R}^{n(N+1)\times n(N+1)}. \tag{2.37}$$

In the state formulation, the following system arises

$$\mathcal{A}_s\delta\mathbf{x} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{b} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}, \tag{2.38}$$

$$\text{where } \mathcal{A}_s = (\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}) \in \mathbb{R}^{n(N+1)\times n(N+1)}, \tag{2.39}$$

$\mathcal{A}_f$ and $\mathcal{A}_s$ are the Hessians of (2.33) and (2.35), respectively. These coefficient matrices are symmetric positive definite and CG is the method of choice as in the strong constraint case. The most computationally expensive part of CG are matrix-vector products because of the tangent linear model $\mathbf{M}_i$ and its adjoint $\mathbf{M}_i^T$ in $\mathbf{L}$ and $\mathbf{L}^T$, respectively.

Notice that $\mathcal{A}_f$ includes $\mathbf{L}^{-1}$ and $\mathcal{A}_s$ includes $\mathbf{L}$. This defines the different possibilities for parallel in time computations as described by [Fisher and Gürol, 2017], i.e., running the tangent linear model linearised at different times in parallel. Matrix-vector products with $\mathbf{L}^{-1}$ are essentially sequential. Let $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \ldots, \mathbf{z}_N^T)^T$, $\mathbf{z}_i \in \mathbb{R}^n$, then

$$\mathbf{L}^{-1}\mathbf{z} = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{M}_0\mathbf{z}_0 + \mathbf{z}_1 \\ \mathbf{M}_1(\mathbf{M}_0\mathbf{z}_0 + \mathbf{z}_1) + \mathbf{z}_2 \\ \vdots \\ \mathbf{M}_{N-1}(\mathbf{M}_{N-2}\ldots\mathbf{M}_0\mathbf{z}_0 + \mathbf{M}_{N-2}\ldots\mathbf{M}_1\mathbf{z}_1 + \cdots + \mathbf{z}_{N-1}) + \mathbf{z}_N \end{pmatrix}, \tag{2.40}$$

thus the tangent linear model has to be integrated sequentially. Contrarily, matrix-vector products with $\mathbf{L}$ can be parallelised in the time dimension, because

$$\mathbf{L}\mathbf{z} = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 - \mathbf{M}_0\mathbf{z}_0 \\ \mathbf{z}_2 - \mathbf{M}_1\mathbf{z}_1 \\ \vdots \\ \mathbf{z}_N - \mathbf{M}_{N-1}\mathbf{z}_{N-1} \end{pmatrix}. \tag{2.41}$$

Hence, the state formulation (2.38) has more potential for parallel in time computations compared to the forcing formulation (2.36). Even though the tangent linear model can be run in parallel in (2.38), the adjoint model in $\mathbf{L}^T$ can only be applied after $\mathbf{D}^{-1}\mathbf{L}\mathbf{z}$ is computed.

Motivated by a lack of suitable preconditioners for (2.38), [Fisher and Gürol, 2017] introduced a saddle point formulation to obtain $\delta\mathbf{x}$. In this formulation the tangent linear and adjoint models in $\mathbf{L}$ and $\mathbf{L}^T$, respectively, can be run at the same time. Let $\boldsymbol{\lambda} \in \mathbb{R}^{(N+1)n}$ and $\boldsymbol{\mu} \in \mathbb{R}^q$ be Lagrange multipliers that satisfy equations

$$\mathbf{D}\boldsymbol{\lambda} = (\mathbf{b} - \mathbf{L}\delta\mathbf{x}) \in \mathbb{R}^{(N+1)n}, \tag{2.42}$$

$$\mathbf{R}\boldsymbol{\mu} = (\mathbf{d} - \mathbf{H}\delta\mathbf{x}) \in \mathbb{R}^q. \tag{2.43}$$

Then the optimality constraint, where the gradient of (2.35) with respect to $\delta\mathbf{x}$ is equal to the zero vector, gives

$$\mathbf{0} = \mathbf{L}^T\mathbf{D}^{-1}(\mathbf{L}\delta\mathbf{x} - \mathbf{b}) + \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\delta\mathbf{x} - \mathbf{d}) \tag{2.44}$$

$$= -(\mathbf{L}^T\boldsymbol{\lambda} + \mathbf{H}^T\boldsymbol{\mu}). \tag{2.45}$$

Taking (2.42), (2.43) and (2.45) we obtain the following system

$$\mathcal{A}_3 \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}, \tag{2.46}$$

$$\text{where } \mathcal{A}_3 = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & \mathbf{0} \end{pmatrix}. \tag{2.47}$$

$\mathcal{A}_3 \in \mathbb{R}^{(2(N+1)n+q)\times(2(N+1)n+q)}$ is a sparse symmetric indefinite saddle point matrix of 3×3 block form. The minimal residual method (MINRES) of [Paige and Saunders, 1975] is the iterative solver of choice. System (2.46) is more than twice as large as the positive definite systems (2.36) and (2.38). However, matrix-vector products with every block in (2.46) can be parallelised, hence this formulation may be preferred if enough computational resources are available and parallel computations are essential.

2.2.3 Preconditioning

Effective preconditioning is essential in order to use the weak constraint 4D-Var in operational settings; otherwise a sufficient quality solution may not be obtained in the given time because of the slow convergence. The unique features of each of the three systems (2.36), (2.38) and (2.46) have to be taken into account when designing suitable preconditioners. We now examine the previous efforts and point to our work.

Forcing formulation

The first level preconditioning idea in Section 2.1.3 can be extended for the forcing formulation, so that the preconditioned coefficient matrix is equal to the sum of an identity and a low-rank matrix. Let $\mathbf{D} = \mathbf{D}^{1/2}\mathbf{D}^{1/2}$, where $\mathbf{D}^{1/2}$ is the unique symmetric square root, and $\mathbf{D}^{1/2}\delta\widetilde{\mathbf{p}} = \delta\mathbf{p}$. The increment $\delta\widetilde{\mathbf{p}}$ is obtained by solving

$$\mathcal{A}_f^{pr}\delta\widetilde{\mathbf{p}} = \mathbf{D}^{-1/2}\mathbf{b} + \mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}, \tag{2.48}$$

$$\text{where } \mathcal{A}_f^{pr} = \mathbf{I} + \mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}. \tag{2.49}$$

$\mathcal{A}_f^{pr}$ is symmetric positive definite and its smallest eigenvalues are equal to one. It can be preconditioned using the same second level preconditioners as in the strong constraint case, discussed in Section 2.1.4. In this thesis, we explore a new way to construct LMPs for the forcing formulation (Chapter 4).

State formulation

Extending the first level preconditioning idea for the state formulation requires split preconditioning (2.38) with $\mathbf{P} = \mathbf{L}^{-1}\mathbf{D}^{1/2}(\mathbf{L}^{-1}\mathbf{D}^{1/2})^T$. Then the state formulation with first level preconditioning is the same as (2.48), i.e., we obtain the forcing formulation with first level preconditioning. This is not desirable if potential for time-parallel computations is important.

[Fisher and Gürol, 2017] suggested using approximation $\widetilde{\mathbf{L}}^{-1}$ of $\mathbf{L}^{-1}$ in the first level preconditioning for the state formulation, that is, solving the system

$$\mathcal{A}_s^{pr}\delta\widetilde{\mathbf{x}} = \mathbf{D}^{1/2}\widetilde{\mathbf{L}}^{-T}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{b} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}), \tag{2.50}$$

$$\text{where } \mathcal{A}_s^{pr} = \mathbf{D}^{1/2}\widetilde{\mathbf{L}}^{-T}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}, \tag{2.51}$$

$$\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}\delta\widetilde{\mathbf{x}} = \delta\mathbf{x}. \tag{2.52}$$

A suitable $\widetilde{\mathbf{L}}^{-1}$ was not found. [Gratton et al., 2018b] obtained bounds for the eigenvalues of $\mathcal{A}_s^{pr}$ when the linearised model $\mathbf{M}_i$ in $\widetilde{\mathbf{L}}^{-1}$ is approximated by zero and identity matrices. These bounds indicate that a good approximation of the model may be needed for the preconditioner to be effective. Note that this preconditioner and analysis ignores the observation term $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$.

We propose a new way to approximate $\mathbf{L}^{-1}$ in the first level preconditioner and retain the potential for parallel in time computations in Chapter 6. We also observe that the first level preconditioning is not always useful even if the exact $\mathbf{L}^{-1}$ is used.

Saddle point formulation

The computationally most expensive blocks of the saddle point system (2.46) are the off-diagonal blocks $\mathbf{L}$ and $\mathbf{L}^T$. This is unusual; in many other applications, in which saddle point systems arise, matrix-vector products with the off-diagonal blocks are cheap (see, e.g., [Benzi et al., 2005]). Hence, many preconditioners described in numerical linear algebra literature are not suitable for (2.46) because they include the exact off-diagonal blocks.

Variants of inexact constraint [Bergamaschi et al., 2007, Bergamaschi et al., 2011], block diagonal and block triangular [Benzi and Wathen, 2008] preconditioners for (2.46) have been considered by [Fisher and Gürol, 2017, Gratton et al., 2018a, Fisher et al.,

2018, Freitag and Green, 2018, Tabeart and Pearson, 2021]:

$$\text{(Inexact constraint)} \quad \mathcal{P}_M = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \widetilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \widetilde{\mathbf{L}}^T & \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{2.53}$$

$$\text{(Block diagonal)} \quad \mathcal{P}_B = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{S}} \end{pmatrix}, \tag{2.54}$$

$$\text{(Block triangular)} \quad \mathcal{P}_T = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \widetilde{\mathbf{L}} \\ \mathbf{0} & \mathbf{R} & \widetilde{\mathbf{H}} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{S}} \end{pmatrix}. \tag{2.55}$$

Inverses of these preconditioners are used when solving the system (2.46):

$$\mathcal{P}_M^{-1} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{L}}^{-T} \\ \mathbf{0} & \mathbf{R}^{-1} & \mathbf{0} \\ \widetilde{\mathbf{L}}^{-1} & \mathbf{0} & -\widetilde{\mathbf{L}}^{-1}\mathbf{D}\widetilde{\mathbf{L}}^{-T} \end{pmatrix}, \tag{2.56}$$

$$\mathcal{P}_B^{-1} = \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{S}}^{-1} \end{pmatrix}, \tag{2.57}$$

$$\mathcal{P}_T^{-1} = \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} & -\mathbf{D}\widetilde{\mathbf{L}}\widetilde{\mathbf{S}}^{-1} \\ \mathbf{0} & \mathbf{R}^{-1} & -\mathbf{R}\widetilde{\mathbf{H}}\widetilde{\mathbf{S}}^{-1} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{S}}^{-1} \end{pmatrix}. \tag{2.58}$$

$\widetilde{\mathbf{S}}^{-1}$ is an approximation to the inverse of negative Schur complement of the (1,1) block $\begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix}$. The exact $\mathbf{S}^{-1}$ is equal to the inverse Hessian of the state formulation $\mathcal{A}_s^{-1}$. With a proper choice of $\widetilde{\mathbf{S}}^{-1}$, MINRES can be used to solve (2.46) preconditioned with the block diagonal $\mathcal{P}_B^{-1}$. The inexact constraint and block triangular preconditioners are not symmetric and thus they cannot be used with MINRES. The generalized minimal residual method (GMRES) of [Saad and Schultz, 1986] is the method of choice then.

When generating the block diagonal preconditioner, the observation term in $\mathbf{S}^{-1}$ was ignored by setting $\widetilde{\mathbf{S}}^{-1} = \widetilde{\mathbf{L}}^{-1}\mathbf{D}\widetilde{\mathbf{L}}^{-T}$, where $\widetilde{\mathbf{L}}^{-1}$ is an approximation to $\mathbf{L}^{-1}$ [Fisher and Gürol, 2017, Gratton et al., 2018a, Freitag and Green, 2018, Tabeart and Pearson, 2021]. $\widetilde{\mathbf{L}}^{-1}$ was obtained by setting the tangent linear model $\mathbf{M}_i$ to zero or identity matrices [Fisher and Gürol, 2017, Gratton et al., 2018a, Freitag and Green, 2018]. This may be unrealistic when the state of the dynamical system changes fast. [Tabeart and Pearson, 2021] proposed approximating $\mathbf{L}^{-1}$ by setting $\mathbf{M}_i$ to zero at every kth time and using exact $\mathbf{M}_i$ at other times; larger k values increases the amount of information the preconditioner uses but requires more time-sequential computations. They also approximated non-diagonal $\mathbf{R}^{-1}$ in the preconditioners. This preconditioner accelerates the convergence more than when the $\mathbf{M}_i$ is set to a zero matrix.

[Freitag and Green, 2018] showed that the preconditioners made the problem harder to solve compared to the unpreconditioned case when using a low-rank GMRES solver, where

a low-rank matrix is obtained to approximate $\delta\mathbf{x}$. [Gratton et al., 2018a] concluded that $\mathcal{P}_M$ outperforms other preconditioners over multiple inner loops, but it was not compared to solving the unpreconditioned system.

When solving the positive definite systems (2.36) and (2.38), the decrease of the quadratic cost function at every iteration of CG is monotonic (at least in exact arithmetic; see Section 3.4). This is not the case when solving the saddle point problem with GMRES or MINRES [Gratton et al., 2018a], i.e., the quadratic cost function value can increase during the solution process. This is not desirable, because in operational applications it is not possible to run the iterative process until convergence and the solver is terminated after a fixed number of iterations. [Gratton et al., 2018a] suggested a Safeguarded SADDLE solution algorithm, where GMRES termination criterion depends on reducing the quadratic cost function value and is checked at every jth iteration. Choice of j depends on finding a balance between frequent costly evaluations of the quadratic cost function and running more than necessary iterations of GMRES. For both Safeguarded SADDLE and regular GMRES it is desirable that the preconditioner helps to reduce the quadratic cost function value in the beginning of the iterative process. We explore a new way to approximate the full $\mathbf{S}^{-1}$ in the block diagonal preconditioner in Chapter 7.

2.3 Summary

In this chapter, we defined the strong and weak constraint 4D-Var data assimilation methods and their incremental formulations. The state and forcing formulations of the weak constraint problem were presented. We discussed linear systems of equations that arise in the incremental formulations and how the first level preconditioning (control variable transform), which is used in the strong constraint formulation, can be extended to precondition the weak constraint positive definite problems. The second level preconditioning using the limited memory preconditioners (LMPs) in the strong constraint case was reviewed. We considered the inexact constrain, block diagonal, and block triangular preconditioners for the saddle point system and how they have been constructed in the previous work. We now present mathematical ideas and results used in this thesis.

Chapter 3

Mathematical background

In this chapter, we present mathematical theory and ideas that are used in the following chapters. Specific matrices, their features and decompositions, and vector and matrix norms are described in Section 3.1. Section 3.2 contains results on eigenvalues and singular values. Methods for low-rank eigenvalue and singular value approximations are discussed in Section 3.3. We examine the conjugate gradient and the minimal residual methods and their convergence in Sections 3.4 and 3.5, respectively. The theory of preconditioning including limited memory, and block diagonal Schur complement preconditioners are considered in Section 3.6. Throughout the chapter we provide references to standard texts where the theorems can be found.

3.1 Matrix theory

In this thesis, we consider linear systems with symmetric coefficient matrices. We start the section by defining a symmetric matrix and consider its special cases.

Definition 3.1. *A matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{A} = [a_{ij}]$ *is symmetric if* $a_{ij} = a_{ji}$ *for* $i, j \in \{1, 2, \ldots, n\}$, *that is, if* $\mathbf{A} = \mathbf{A}^T$.

Definition 3.2. *A symmetric matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *is positive definite (SPD) if* $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ *for all nonzero vectors* $\mathbf{x} \in \mathbb{R}^n$.

Definition 3.3. *A symmetric matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *is positive semi-definite (SPSD) if* $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$ *for all vectors* $\mathbf{x} \in \mathbb{R}^n$.

Definition 3.4. *A symmetric matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *is indefinite if there exist vectors* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{y} \in \mathbb{R}^n$ *such that* $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ *and* $\mathbf{y}^T\mathbf{A}\mathbf{y} < 0$.

Theorem 3.5 (Section 9.12.1 of [Lütkepohl, 1996])**.** *If* $\mathbf{A}$ *and* $\mathbf{B}$ *are* $n \times n$ *symmetric positive definite and symmetric positive semi-definite, respectively, then* $\mathbf{A}+\mathbf{B}$ *is symmetric positive definite.*

Theorem 3.5 explains why the coefficient matrices $\mathcal{A}_f$ and $\mathcal{A}_s$ in (2.37) and (2.39) are SPD. The singular value and eigenvalue decompositions are presented next. These decompositions are explored to understand the behaviour of iterative methods and accelerate their convergence.

Theorem 3.6 (Theorem 2.4.1 of [Golub and Van Loan, 2013]). *Let* $\mathbf{A}$ *be a real* $m \times n$ *matrix, then there exist orthogonal matrices* $\mathbf{U}$ *(*$m \times m$*) and* $\mathbf{V}$ *(*$n \times n$*) such that*

$$\mathbf{U}^T\mathbf{A}\mathbf{V} = \mathbf{\Sigma} \in \mathbb{R}^{m\times n},$$

where $\mathbf{\Sigma}$ *has nonzero entries on the diagonal only, that is,* $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \ldots, \sigma_p)$*,* $p = min\{m, n\}$*, and* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$*.*

σ_i is the singular value of $\mathbf{A}$ and the columns $\mathbf{u}_i$ and $\mathbf{v}_i$ of $\mathbf{U}$ and $\mathbf{V}$ are called the ith left and right singular vectors, respectively.

Theorem 3.7 (Theorem 8.1.1 of [Golub and Van Loan, 2013]). *Let* $\mathbf{A}$ *be a symmetric* $n \times n$ *matrix, then there exists a real orthogonal* $n \times n$ *matrix* $\mathbf{V}$ *such that*

$$\mathbf{V}^T\mathbf{A}\mathbf{V} = \mathbf{\Lambda} = diag(\lambda_1(\mathbf{A}), \lambda_2(\mathbf{A}), \ldots, \lambda_n(\mathbf{A})).$$

$\lambda_i(\mathbf{A})$ is the eigenvalue of $\mathbf{A}$ and the column $\mathbf{v}_i$ of $\mathbf{V}$ is the corresponding eigenvector. We order eigenvalues of $\mathbf{A}$ in the following way

$$\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \cdots \geq \lambda_n(\mathbf{A}). \tag{3.1}$$

When we want to emphasise the largest and smallest eigenvalues, we write $\lambda_{max}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A})$. Eigenvalues $\lambda_i(\mathbf{A})$ are the roots of a characteristic polynomial $p_{\mathbf{A}}(\lambda) = det(\lambda\mathbf{I} - \mathbf{A})$. There is the following relationship between the eigenvalues of $\mathbf{A}$ and $\mathbf{A}^{-1}$.

Theorem 3.8 (Section 5.2.1 of [Lütkepohl, 1996]). *Let* λ *be an eigenvalue of a nonsingular* $\mathbf{A}$ *with associated eigenvector* $\mathbf{v}$*. Then* λ^{-1} *is an eigenvalue of* $\mathbf{A}^{-1}$ *with associated eigenvector* $\mathbf{v}$*.*

Hence, if $\lambda_{max}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A})$ are the largest and smallest eigenvalues of $\mathbf{A}$, respectively, then $\lambda_{max}(\mathbf{A}^{-1}) = 1/\lambda_{min}(\mathbf{A})$ and $\lambda_{min}(\mathbf{A}^{-1}) = 1/\lambda_{max}(\mathbf{A})$. The same relationship holds for the singular values of a nonsingular $\mathbf{A}$, because $\mathbf{A}^{-1} = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$. Vector and matrix norms are defined in the following.

Definition 3.9 (Section 2.2.1 of [Golub and Van Loan, 2013]). *A vector norm on* $\mathbb{R}^n$ *is a function* $f : \mathbb{R}^n \to \mathbb{R}$ *that satisfies the following properties:*

$$f(\mathbf{x}) \geq 0, \quad \mathbf{x} \in \mathbb{R}^n \quad (f(\mathbf{x}) = 0 \iff \mathbf{x} = 0) \tag{3.2}$$

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}), \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^n \tag{3.3}$$

$$f(\alpha\mathbf{x}) = |\alpha| f(\mathbf{x}), \quad \alpha \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^n. \tag{3.4}$$

An often used class of vector norms is the p-norm.

Definition 3.10. *Let* $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ *and* $p \geq 1$*. Then the* p*-norm of* $\mathbf{x}$ *is*

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{1/p}.$$

In this thesis, we use the 2-norm. Matrix p-norms are derived using the vector p-norms.

Definition 3.11. *The p-norm of matrix* $\mathbf{A}$ *is*

$$\|\mathbf{A}\|_p = \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p}.$$

As for the vectors, we consider the 2-norm for the matrices. It is known that $\|\mathbf{A}\|_2 = \sigma_{max}$, where σ_{max} is the largest singular value of $\mathbf{A}$. For a nonsingular $\mathbf{A}$, $\|\mathbf{A}^{-1}\|_2 = 1/\sigma_{min}$, where σ_{min} is the smallest singular value of $\mathbf{A}$. If $\mathbf{A}$ is SPD, then σ_{max} coincides with the largest eigenvalue of $\mathbf{A}$ (we show this in Theorem 3.36). This fact is useful when considering the condition number of a matrix.

Definition 3.12. *Let* $\mathbf{A}$ *be a nonsingular* $n \times n$ *matrix. Then its condition number relative to the norm* $\|\cdot\|$ *is defined as*

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|. \tag{3.5}$$

The condition number of $\mathbf{A}$ depicts how sensitive the solution to $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{b} \in \mathbb{R}^n$ is to small perturbations to $\mathbf{A}$ and $\mathbf{b}$. If $\kappa(\mathbf{A})$ is large, then $\mathbf{A}$ is said to be ill-conditioned and small changes in $\mathbf{A}$ and $\mathbf{b}$ may lead to very different solutions (Lecture 12 of [Trefethen and Bau, III, 1997]).

A 2-norm condition number is also used to describe the worst case convergence behaviour for the iterative solution methods, such as conjugate gradient method (see Section 3.4). In this case, $\kappa(\mathbf{A}) = \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 = \sigma_{max}/\sigma_{min}$. If $\mathbf{A}$ is SPD, then

$$\kappa(\mathbf{A}) = \lambda_{max}(\mathbf{A})/\lambda_{min}(\mathbf{A}). \tag{3.6}$$

We present ideas of range and rank of a matrix. These are used in describing the matrix approximation algorithms.

Definition 3.13. *The range of a matrix* $\mathbf{A} \in \mathbb{R}^{n\times m}$ *is defined as a subspace containing all possible combinations of its columns, that is,*

$$range(\mathbf{A}) = \{\mathbf{Ax} \mid \mathbf{x} \in \mathbb{C}^m\} = span\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}, \tag{3.7}$$

where $\mathbf{a}_i$ *is the* i*th column of* $\mathbf{A}$*.*

Definition 3.14. *The rank* $rank(\mathbf{A})$ *of a matrix* $\mathbf{A}$ *is the maximum number of linearly independent rows or columns of* $\mathbf{A}$*.*

Theorem 3.15 (Section 4.3.3 of [Lütkepohl, 1996])**.** *Let* $\mathbf{A}$ *be an* $m \times n$ *and* $\mathbf{B}$ *an* $n \times k$ *matrices. Then*

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

Theorem 3.16 (Theorem 1.3 of [Trefethen and Bau, III, 1997])**.** *Let* $\mathbf{A}$ *be an* $n \times n$ *matrix.* $\mathbf{A}$ *is nonsingular if and only if* $rank(\mathbf{A}) = n$*.*

In this work we consider some low-rank matrices, where an $n \times m$ matrix $\mathbf{A}$ is low-rank if $rank(\mathbf{A}) \ll min\{m, n\}$. Notice that by Theorem 3.15 a product of a nonsingular square matrix and a low-rank matrix is low-rank and hence singular by Theorem 3.16. This is

taken into account when a low-rank matrix approximation is used to approximate $\mathbf{L}^{-1}$ defined in (2.28) (see Chapter 6).

We now introduce block matrices and theorems regarding their inverses and determinants. A Schur complement in a block matrix is presented as well. It defines the relationship between the saddle point matrices in the incremental weak constraint 4D-Var and the SPD formulation $\mathcal{A}_s$ in (2.39) (Chapter 5), and arises in preconditioning the saddle point systems (Chapter 7).

Definition 3.17. *An $n \times m$ matrix $\mathbf{A}$ is called a block matrix if it is written in terms of submatrices, that is*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \dots & \mathbf{A}_{1k} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \dots & \mathbf{A}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{l1} & \mathbf{A}_{l2} & \dots & \mathbf{A}_{lk} \end{pmatrix},$$

where $\mathbf{A}_{ij}$ is an $n_i \times m_i$ submatrix and $\sum_{i=1}^{l} n_i = n$ and $\sum_{j=1}^{k} m_i = m$.

Theorem 3.18 (Section 3.5.3 of [Lütkepohl, 1996])**.** *If $n_i \times n_i$ matrices $\mathbf{A}_i$ are nonsingular for $i \in \{1, 2, \dots, k\}$, then*

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_k \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}_1^{-1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2^{-1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_k^{-1} \end{pmatrix}.$$

Theorem 3.19 (Theorem 3 of [Silvester, 2000])**.** *If $\mathbf{F} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$, $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are $n \times n$ matrices, and $\mathbf{CD} = \mathbf{DC}$, then*

$$det(\mathbf{F}) = det(\mathbf{AD} - \mathbf{BC}).$$

Theorem 3.20 (Section 9.11.2 of [Lütkepohl, 1996])**.** *Let $\mathbf{A}$ and $\mathbf{D}$ be square matrices and $\mathbf{F} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$. If $\mathbf{A}$ is nonsingular, then*

$$det(\mathbf{F}) = det(\mathbf{A}) det(\mathbf{D} - \mathbf{CA}^{-1}\mathbf{B}). \tag{3.8}$$

Definition 3.21. *Let $\mathbf{A}$ be a nonsingular $n \times n$ matrix, $\mathbf{B}$ an $n \times m$, $\mathbf{C}$ an $m \times n$, $\mathbf{D}$ an $m \times m$ matrix and $\mathbf{F} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$. Then the Schur complement of $\mathbf{A}$ in $\mathbf{F}$ is the $m \times m$ matrix $\mathbf{S} = \mathbf{D} - \mathbf{CA}^{-1}\mathbf{B}$.*

In data assimilation, SPD covariance matrices that describe the relationship between the errors in different variables are used. A covariance matrix $\mathbf{F} \in \mathbb{R}^{n \times n}$ can be written as

$$\mathbf{F} = \mathbf{\Sigma C \Sigma}, \tag{3.9}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with the error standard deviation on the diagonal, and $\mathbf{C} \in \mathbb{R}^{n\times n}$ is a correlation matrix. In our numerical experiments, we assume that the standard deviation is the same for all the variables and denote it σ. Then

$$\mathbf{F} = \sigma^2 \mathbf{C}. \tag{3.10}$$

Two types of correlation matrices $\mathbf{C}$, which can be used for equally spaced variables on a circle, are employed in this thesis. The second-order auto-regressive (SOAR) correlation matrix is based on a SOAR correlation function [Daley, 1993], and it has been used in the UK Met Office [Lorenc, 1992, Lorenc et al., 2000]. [Haben, 2011] describes how to obtain the covariance matrix from the correlation function.

Definition 3.22. *Let $s_1, s_2, \ldots, s_n$ be equally spaced grid points on a circle with $\theta_{i,j}$ denoting the angle between s_i and s_j, and a be the radius of the circle. Then the (i,j)th element of the second-order auto-regressive (SOAR) correlation matrix $\mathbf{C}_S \in \mathbb{R}^{n\times n}$ is*

$$\mathbf{C}_S(i,j) = \left(1 + \frac{|2a\sin(\theta_{i,j}/2)|}{L}\right) \exp\left(-\frac{|2a\sin(\theta_{i,j}/2)|}{L}\right), \tag{3.11}$$

where $L > 0$ is the correlation length scale.

A Laplacian correlation matrix $\mathbf{C}_L \in \mathbb{R}^{n\times n}$ is defined using its inverse. It has been shown that $\mathbf{C}_L$ is SPD if $n > 5$ (Theorem 5.2.1 of [Haben, 2011]).

Definition 3.23. *Let $s_1, s_2, \ldots, s_n$ be equally spaced grid points on a circle with the great circle distance Δs between adjacent grid points. Then the inverse of the Laplacian correlation matrix $\mathbf{C}_L \in \mathbb{R}^{n\times n}$ is*

$$\mathbf{C}_L^{-1} = \gamma^{-1}\left(\mathbf{I} + \frac{L^4}{2\Delta s^4}(\mathbf{L}_{ss})^2\right), \tag{3.12}$$

where $L > 0$ is the correlation length scale, $\gamma > 0$ is a normalisation constant that ensures that the largest entry of $\mathbf{C}_L$ is equal to one, and $\mathbf{L}_{ss}$ is a second derivative matrix given by

$$\mathbf{L}_{ss} = \begin{pmatrix} -2 & 1 & 0 & 0 & \ldots & 0 & 1 \\ 1 & -2 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ 0 & & \ddots & \ddots & \ddots & & 1 \\ 1 & 0 & \ldots & & & 1 & -2 \end{pmatrix}. \tag{3.13}$$

In the following chapters, we consider randomised methods that use random Gaussian matrices, which we define now.

Definition 3.24. *Matrix $\mathbf{G} \in \mathbb{R}^{m\times n}$ is Gaussian if its entries are independent standard normal random variables with zero mean and variance equal to one.*

3.2 Results on eigenvalues and singular values

Chapters 5 and 7 contain results bounding the eigenvalues of symmetric matrices, including $\mathcal{A}_f, \mathcal{A}_s$ and $\mathcal{A}_3$. They are obtained using well-known theorems, which we discuss in this section. We refer to an eigenvalue λ of $\mathbf{A}$ and the corresponding eigenvector $\mathbf{v}$ as an eigenpair $(\lambda, \mathbf{v})$. Theorems that consider the similarity $\mathbf{A} = \mathbf{F}\mathbf{B}\mathbf{F}^{-1}$ and congruence $\mathbf{A} = \mathbf{F}\mathbf{B}\mathbf{F}^T$ transformations are presented next.

Theorem 3.25 (Fact 1.1 of [Parlett, 1998])**.** *Let* $\mathbf{A}$*,* $\mathbf{B}$*, and* $\mathbf{F}$ *be* $n \times n$ *matrices, such that*

$$\mathbf{A} = \mathbf{F}\mathbf{B}\mathbf{F}^{-1}. \tag{3.14}$$

If $(\lambda, \mathbf{v})$ *is an eigenpair of* $\mathbf{B}$*, then* $(\lambda, \mathbf{F}\mathbf{v})$ *is an eigenpair of* $\mathbf{A}$*.* $\mathbf{A}$ *and* $\mathbf{B}$ *are said to be similar.*

Theorem 3.26 (Sylvester's inertia theorem, Fact 1.6 of [Parlett, 1998])**.** *Each* $n \times n$ *matrix* $\mathbf{A}$ *is congruent to a diagonal matrix* $\mathbf{B} = diag(\mathbf{I}_\pi, -\mathbf{I}_\mu, \mathbf{0}_\zeta)$*, that is, there exists an* $n \times n$ *matrix* $\mathbf{F}$ *such that*

$$\mathbf{A} = \mathbf{F}\mathbf{B}\mathbf{F}^T, \tag{3.15}$$

where $\mathbf{I}_\pi$ *and* $\mathbf{I}_\mu$ *are the* $\pi \times \pi$ *and* $\mu \times \mu$ *identity matrices, respectively, and* $\mathbf{0}_\zeta$ *is a* $\zeta \times \zeta$ *zero matrix. The number triple* (π, μ, ζ) *is called inertia of* $\mathbf{A}$ *and depends only on* $\mathbf{A}$*;* π, μ, ζ *are the number of positive, negative, and zero eigenvalues of* $\mathbf{A}$*.*

The SPD, SPSD, and indefinite matrices defined in the previous section can be characterised using their eigenvalues.

Theorem 3.27 (Section 5.2 of [Lütkepohl, 1996])**.** *A symmetric matrix* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *is SPD if and only if all its eigenvalues are real and positive.*

Theorem 3.28 (Section 5.2 of [Lütkepohl, 1996])**.** *A symmetric matrix* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *is SPSD if and only if all its eigenvalues are real and nonnegative.*

Theorem 3.29. *A symmetric matrix* $\mathbf{A} \in \mathbb{R}^{n \times n}$ *is indefinite if and only if it has both positive and negative eigenvalues.*

In the following chapters, we consider block diagonal error covariance matrices $\mathbf{D}$ and $\mathbf{R}$ defined in (2.31) and (2.10). Their eigenvalues can be obtained by finding the eigenvalues of the diagonal blocks as shown in the following theorem.

Theorem 3.30. *Let* $\mathbf{A}$ *be an* $n \times n$ *block diagonal matrix,* $\mathbf{A}_i$ *be nonsingular* $n_i \times n_i$ *matrices for* $i \in \{1, 2, \ldots, k\}$*, and*

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{A}_k \end{pmatrix}.$$

Then the eigenvalues of $\mathbf{A}$ *are the eigenvalues of* $\mathbf{A}_i$ *for* $i \in \{1, 2, \ldots, k\}$*.*

Proof. The result for $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_1 \end{pmatrix}$ can be found in Section 9.11.1 of [Lütkepohl, 1996]. The result for the general case follows by considering $\mathbf{B}_1 = diag(\mathbf{A}_2, \mathbf{B}_2)$ with $\mathbf{B}_2 = diag(\mathbf{A}_3, \mathbf{B}_3)$, ..., $\mathbf{B}_{k-2} = diag(\mathbf{A}_{k-1}, \mathbf{A}_k)$. □

We further present theorems that consider eigenvalues of a sum of matrices, and show that eigenvalues of a matrix and its principal submatrix exhibit interlacing. These are useful when considering the change of the eigenvalues of $\mathcal{A}_f$, $\mathcal{A}_s$, and $\mathcal{A}_3$ when new observations of the dynamical system are introduced (Chapters 5 and 7).

Theorem 3.31 (Section 8.1.2 of [Golub and Van Loan, 2013])**.** *If* $\mathbf{A}$ *and* $\mathbf{C}$ *are* $n \times n$ *symmetric matrices, then*

$$\lambda_k(\mathbf{A}) + \lambda_{min}(\mathbf{C}) \leq \lambda_k(\mathbf{A} + \mathbf{C}) \leq \lambda_k(\mathbf{A}) + \lambda_{max}(\mathbf{C}), \quad k \in \{1, 2, \ldots, n\}.$$

Theorem 3.32 (Cauchy's Interlace Theorem, Theorem 4.2 in Chapter 4 of [Stewart and Sun, 1990])**.** *If* $\mathbf{A}$ *is an* $n \times n$ *symmetric matrix and* $\mathbf{C}$ *is a* $(n-1) \times (n-1)$ *principal submatrix of* $\mathbf{A}$ *(a matrix obtained by eliminating an* i*th row and an* i*th column of* $\mathbf{A}$*), then*

$$\lambda_n(\mathbf{A}) \leq \lambda_{n-1}(\mathbf{C}) \leq \lambda_{n-1}(\mathbf{A}) \leq \cdots \leq \lambda_2(\mathbf{A}) \leq \lambda_1(\mathbf{C}) \leq \lambda_1(\mathbf{A}).$$

The Rayleigh quotient is used in approximating eigenvalues and we employ it in deriving bounds for the eigenvalues of symmetric matrices. The definition of the Rayleigh quotient follows.

Definition 3.33. *Let* $\mathbf{A}$ *be a symmetric* $n \times n$ *matrix. The Rayleigh quotient for* $\mathbf{A}$ *is defined as*

$$\rho(\mathbf{u}; \mathbf{A}) = \frac{\mathbf{u}^* \mathbf{A} \mathbf{u}}{\mathbf{u}^* \mathbf{u}}, \tag{3.16}$$

where $\mathbf{u} \in \mathbb{C}^n$ *is nonzero and* $\mathbf{u}^*$ *is its conjugate transpose.*

The Rayleigh quotient is bounded by the smallest and largest eigenvalues of $\mathbf{A}$.

Theorem 3.34 (Fact 1.8 of [Parlett, 1998])**.** *Let* $\mathbf{A}$ *be a symmetric* $n \times n$ *matrix. Then the Rayleigh quotient* $\rho(\mathbf{u}; \mathbf{A})$ *ranges over the interval* $[\lambda_{min}(\mathbf{A}), \lambda_{max}(\mathbf{A})]$.

If $\mathbf{A}$ is not a square matrix, a similar bound can be obtained using $\mathbf{A}^*\mathbf{A}$ and the smallest singular value of $\mathbf{A}$.

Theorem 3.35 (Section 5.5 of [Lütkepohl, 1996])**.** *Let* $\mathbf{A}$ *be an* $m \times n$ *matrix with* $m \geq n$ *and* $\sigma_{min}(\mathbf{A})$ *be its smallest singular value. Then*

$$\sigma_{min}(\mathbf{A}) = \min_{\mathbf{u} \neq 0} \left(\frac{\mathbf{u}^* \mathbf{A}^* \mathbf{A} \mathbf{u}}{\mathbf{u}^* \mathbf{u}} \right)^{1/2}, \tag{3.17}$$

where $\mathbf{u} \in \mathbb{C}^n$.

Hence, the following holds

$$\sigma_{min}(\mathbf{A}) \leq \frac{\|\mathbf{Au}\|_2}{\|\mathbf{u}\|_2}. \tag{3.18}$$

We now consider the relationship between the singular values of a matrix and eigenvalues of some related matrices. These are used to bound eigenvalues of the coefficient matrices in Chapter 5.

Theorem 3.36 (Corollary 4.4.4 of [Horn and Johnson, 2012]). *If* $\mathbf{A}$ *is a symmetric* $n \times n$ *matrix, then there is an orthogonal* $n \times n$ *matrix* $\mathbf{U}$*such that*

$$\mathbf{A} = \mathbf{U\Sigma U}^T, \tag{3.19}$$

where $\mathbf{\Sigma}$ *is a nonnegative diagonal* $n \times n$ *matrix with singular values of* $\mathbf{A}$ *on the diagonal.*

It thus follows from Theorems 3.7 and 3.36 that for a symmetric matrix the singular value and the eigenvalue decompositions coincide up to the sign of the eigenvalues. The following two theorems consider the relationships between the two decompositions when $\mathbf{A}$ may not be symmetric.

Theorem 3.37 (Jordan-Wielandt Theorem, Theorem 4.2 in Chapter 1 of [Stewart and Sun, 1990]). *Let* $\mathbf{U}^H$ *be the conjugate transpose of* $\mathbf{U}$ *and*

$$\mathbf{U}^H\mathbf{AV} = \begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \mathbf{\Sigma} = diag(\sigma_1, \cdots, \sigma_n)$$

be the singular value decomposition of $\mathbf{A} \in \mathbb{C}^{m\times n}$, $m \geq n$. *Then the eigenvalues of the matrix*

$$\mathbf{C} = \begin{pmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^H & \mathbf{0} \end{pmatrix}$$

are $\pm\sigma_1, \cdots, \pm\sigma_n$, *corresponding to the eigenvectors* $\begin{pmatrix} \mathbf{u}_i \\ \pm\mathbf{v}_i \end{pmatrix}$, $i = 1, \cdots, n$, *where* $\mathbf{u}_i$ *and* $\mathbf{v}_i$ *are the* i*th columns of* $\mathbf{U}$ *and* $\mathbf{V}$, *respectively.* $\mathbf{C}$ *also has* $m - n$ *zero eigenvalues with eigenvectors* $\begin{pmatrix} \mathbf{u}_i \\ \mathbf{0} \end{pmatrix}$, $i = n + 1, \cdots, m$.

Theorem 3.38 (Theorem 5.4 of [Trefethen and Bau, III, 1997]). *The nonzero singular values of an* $m \times n$ *matrix* $\mathbf{A}$ *are the square roots of the nonzero eigenvalues of* $\mathbf{A}^T\mathbf{A}$ *or* $\mathbf{AA}^T$; $\mathbf{A}^T\mathbf{A}$ *and* $\mathbf{AA}^T$ *have the same nonzero eigenvalues.*

The following result is used to bound the eigenvalues of the saddle point matrix in the 3×3 block formulation (2.46) (Chapter 5).

Theorem 3.39 (Lemma 2.1 of [Rusten and Winther, 1992]). *Consider a saddle point matrix*

$$\mathbf{A} = \begin{pmatrix} \mathbf{M} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix}, \tag{3.20}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ *is SPD,* $\mathbf{B} \in \mathbb{R}^{n\times m}$ *with* $m \leq n$ *and* $rank(\mathbf{B}) = m$*. Let* μ_{min} *and* μ_{max} *be the smallest and largest eigenvalues of* $\mathbf{M}$ *and* σ_{min} *and* σ_{max} *be the smallest nonzero and largest singular values of* $\mathbf{B}$*. Then the negative eigenvalues of* $\mathbf{A}$ *lie in the interval*

$$I_- = \left[\frac{1}{2}\left(\mu_{min} - \sqrt{\mu_{min}^2 + 4\sigma_{max}^2}\right), \frac{1}{2}\left(\mu_{max} - \sqrt{\mu_{max}^2 + 4\sigma_{min}^2}\right)\right] \tag{3.21}$$

and the positive eigenvalues of $\mathbf{A}$ *lie in the interval*

$$I_+ = \left[\mu_{min}, \frac{1}{2}\left(\mu_{max} + \sqrt{\mu_{max}^2 + 4\sigma_{max}^2}\right)\right]. \tag{3.22}$$

3.3 Low-rank matrix approximations

The singular value and eigenvalue decompositions can be used to accelerate the solution of linear systems of equations. For large problems, however, computing the full decomposition may take too much time and memory issues can arise, because the eigenvectors are in general not sparse even if the matrix is sparse (Chapter 4 of [Stewart, 2001]). Hence, only a subset of the singular vectors and values or eigenpairs are found and low-rank matrix approximations are used. Most algorithms for this consist of two stages:

1. Construct a subspace that contains approximations to the singular vectors or eigenvectors.

2. Compute the approximations to the singular vectors or eigenvectors from the subspace in stage 1.

The two stages can be repeated until the approximation reaches the desired accuracy, and the approximations in the second stage can be used to improve the subspace in the first stage.

3.3.1 Low-rank singular value decomposition

The appeal of the low-rank singular value decomposition (SVD) is explained by the following theorem showing that the SVD of $\mathbf{A}$ contains the most energy of $\mathbf{A}$ among all low-rank approximations, where the energy is measured in 2-norm.

Theorem 3.40 (Eckart-Young Theorem)**.** *Let* $\mathbf{A}$ *be an* $m \times n$ *matrix with* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ *nonzero singular values, and* $\mathbf{u}_i$ *and* $\mathbf{v}_i$*,* $i \in \{1, 2, \ldots, r\}$ *as its left and right singular vectors, respectively. For any* k *with* $0 \leq k \leq r$*, define*

$$\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T. \tag{3.23}$$

Then

$$\|\mathbf{A} - \mathbf{A}_k\|_2 = \inf_{\substack{\mathbf{X}\in\mathbb{C}^{m\times n} \\ rank(\mathbf{X})\leq k}} \|\mathbf{A} - \mathbf{X}\|_2 = \sigma_{k+1}, \tag{3.24}$$

where $\sigma_{k+1} = 0$ *if* $k = min\{m, n\}$*.*

Algorithms for computing the SVD use results in Theorems 3.37 or 3.38 and the problem is reduced to computing an eigenvalue decomposition [Berry et al., 2005]. In this thesis, we use the subspace iteration method for computing the SVD (Chapter 6). This method for the eigenvalue decomposition (EVD) is presented in Algorithm 3 in Section 3.3.2.

For a symmetric matrix, the SVD coincides with EVD up to the signs of the singular values. It is thus natural to consider the low-rank EVD of a symmetric matrix.

3.3.2 Low-rank eigenvalue decomposition

We now focus on the low-rank EVD. Stages 1 and 2 of the approximation problem are addressed separately by discussing the Krylov susbspaces and the Rayleigh-Ritz procedure. The methods for two stages are then combined together in the Lanczos and subspace iteration methods.

Krylov subspaces

The stage 1 of the approximation problem can be performed by employing *Krylov subspaces*, which arise in the solution of linear systems when the conjugate gradient or minimal residual methods are used (sections 3.4 and 3.5). We now define the Krylov subspace.

Definition 3.41. *Let* $\mathbf{A}$ *be an* $n \times n$ *matrix and* $\mathbf{v} \neq \mathbf{0}$ *an* $n \times 1$ *vector. The* m*th Krylov subspace based on* $\mathbf{A}$ *and* $\mathbf{v}$ *is*

$$\mathcal{K}_m(\mathbf{A}, \mathbf{v}) = span\{\mathbf{v}, \mathbf{A}\mathbf{v}, \ldots, \mathbf{A}^{m-1}\mathbf{v}\}. \tag{3.25}$$

That is, any $\mathbf{w} \in \mathcal{K}_m(\mathbf{A}, \mathbf{v})$, where $\mathbf{w}$ is an $n \times 1$ vector, can be written as

$$\mathbf{w} = \gamma_1\mathbf{v} + \gamma_1\mathbf{A}\mathbf{v} + \cdots + \gamma_m\mathbf{A}^{m-1}\mathbf{v}. \tag{3.26}$$

The sequence of Krylov subspaces is nested, i.e., (e.g., Theorem 3.3, Chapter 4 of [Stewart, 2001])

$$\mathcal{K}_m(\mathbf{A}, \mathbf{v}) \subset \mathcal{K}_{m+1}(\mathbf{A}, \mathbf{v}). \tag{3.27}$$

This sequence can contain approximations to eigenvectors of $\mathbf{A}$ [Stewart, 2001]. To show that, we assume that $\mathbf{A}$ is symmetric and has a set of orthonormal eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. Then we can write $\mathbf{v}$ in the following form

$$\mathbf{v} = \alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2 + \cdots + \alpha_n\mathbf{u}_n, \tag{3.28}$$

where $\alpha_i = \mathbf{u}_i^T\mathbf{v}$. Consider the matrix polynomial

$$p(\mathbf{A}) = \gamma_1\mathbf{I} + \gamma_1\mathbf{A} + \cdots + \gamma_m\mathbf{A}^{m-1}. \tag{3.29}$$

Then (3.26) can be written as $\mathbf{w} = p(\mathbf{A})\mathbf{v}$ and using (3.28), we have

$$\mathbf{w} = p(\mathbf{A})\mathbf{v} = \alpha_1 p(\mathbf{A})\mathbf{u}_1 + \alpha_2 p(\mathbf{A})\mathbf{u}_2 + \cdots + \alpha_n p(\mathbf{A})\mathbf{u}_n. \tag{3.30}$$

Because $p(\mathbf{A})\mathbf{u}_1 = p(\lambda_i\mathbf{I})\mathbf{u}_1$,

$$\mathbf{w} = \alpha_1 p(\lambda_1\mathbf{I})\mathbf{u}_1 + \alpha_2 p(\lambda_2\mathbf{I})\mathbf{u}_2 + \cdots + \alpha_n p(\lambda_n\mathbf{I})\mathbf{u}_n. \tag{3.31}$$

Hence, $\mathbf{w}$ is a good approximation to the eigenvector $\mathbf{u}_i$ if there is a polynomial p such that $p(\lambda_i\mathbf{I})$ is large compared to $p(\lambda_j\mathbf{I})$, $j \neq i$. In practice, Krylov subspaces generate good approximations to the eigenvectors associated with the extreme eigenvalues. If the two largest eigenvalues are close to each other, then for small m the values of the polynomial p will be large for both of these and more iterations may be needed to generate satisfactory approximations. Notice that if $\mathbf{v}$ is orthogonal to any eigenvector $\mathbf{u}_i$, then $\alpha_i = 0$ and from (3.31) we see that $\mathbf{u}_i$ does not belong to $\mathcal{K}_m(\mathbf{A}, \mathbf{v})$. However, in practice the rounding errors can cause the loss of orthogonality between $\mathbf{A}^m\mathbf{v}$ and $\mathbf{u}_i$ when m increases (Chapter 13 of [Parlett, 1998]).

Rayleigh-Ritz procedure

Rayleigh-Ritz (RR) procedure is a popular way to perform the stage 2, that is, extract the approximations to the eigenvectors contained in a subspace. It is motivated by the following theorem, that shows that eigenvectors of a large matrix can be obtained by finding eigenvectors of a smaller matrix. We first define an invariant subspace.

Definition 3.42. *Subspace $\mathcal{X} \subset \mathbb{R}^n$ is called an invariant subspace of $\mathbf{A} \in \mathbb{R}^{n\times n}$ if $\mathbf{Ax} \in \mathcal{X}$ for every $\mathbf{x} \in \mathcal{X}$.*

Theorem 3.43 (Theorem 1.2, Chapter 4 of [Stewart, 2001])**.** *Let $\mathcal{X} \subset \mathbb{R}^n$ be an invariant subspace of $\mathbf{A} \in \mathbb{R}^{n\times n}$ and let the columns of $\mathbf{X} \in \mathbb{R}^{n\times m}$ be the basis for $\mathcal{X}$. Then there exists a unique $\mathbf{K} \in \mathbb{R}^{m\times m}$ such that*

$$\mathbf{AX} = \mathbf{XK}. \tag{3.32}$$

$\mathbf{K}$ is given by

$$\mathbf{K} = \mathbf{X}^I\mathbf{AX}, \tag{3.33}$$

where $\mathbf{X}^I \in \mathbb{R}^{m\times n}$ satisfies $\mathbf{X}^I\mathbf{X} = \mathbf{I}$. If $(\lambda, \mathbf{u})$ is an eigenpair of $\mathbf{A}$, then $(\lambda, \mathbf{X}^I\mathbf{u})$ is an eigenpair of $\mathbf{K}$. Conversely, if $(\lambda, \mathbf{u})$ is an eigenpair of $\mathbf{K}$, then $(\lambda, \mathbf{Xu})$ is an eigenpair of $\mathbf{A}$.

Note that if $\mathbf{X}$ is orthogonal, then $\mathbf{X}^I = \mathbf{X}^T$.

Usually the subspace $\widetilde{\mathcal{X}}$ obtained in stage 1 is not invariant and contains only approximations $\hat{\mathbf{u}}$ to the eigenvectors $\mathbf{u}$ of $\mathbf{A}$. It is then expected that eigenvectors $\mathbf{w}$ of $\widetilde{\mathbf{K}} = \widetilde{\mathbf{X}}^T\mathbf{A}\widetilde{\mathbf{X}}$, where the columns of $\widetilde{\mathbf{X}}$ are the orthonormal basis for $\widetilde{\mathcal{X}}$, give good approximations $\tilde{\mathbf{u}} = \mathbf{Xw}$ to $\hat{\mathbf{u}}$. We present the RR procedure in Algorithm 1. It requires finding an EVD of an $m \times m$ matrix. In practice, m is small and EVD can be computed using, e.g., QR algorithm (Chapter 8 of [Parlett, 1998]). Vectors $\tilde{\mathbf{u}}$ are called Ritz vectors, eigenvalues θ and eigenvectors $\mathbf{w}$ of $\widetilde{\mathbf{K}}$ are called Ritz values and primitive Ritz vectors, respectively.

Algorithm 1 Rayleigh-Ritz procedure for computing approximations of eigenpairs of symmetric $\mathbf{A}$

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, orthogonal matrix $\widetilde{\mathbf{X}} \in \mathbb{R}^{n\times m}$, $m < n$

Output: orthogonal $\widetilde{\mathbf{U}} \in \mathbb{R}^{n\times m}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{m\times m}$ with approximations to eigenvalues of $\mathbf{A}$ on the diagonal

1: Form $\widetilde{\mathbf{K}} = \widetilde{\mathbf{X}}^T\mathbf{A}\widetilde{\mathbf{X}} \in \mathbb{R}^{m\times m}$
2: Form EVD of $\widetilde{\mathbf{K}}$: $\widetilde{\mathbf{K}} = \mathbf{W}\mathbf{\Theta}\mathbf{W}^T$, where $\mathbf{W},\ \mathbf{\Theta} \in \mathbb{R}^{m\times m}$
3: Form Ritz vectors $\widetilde{\mathbf{U}} = \widetilde{\mathbf{X}}\mathbf{W} \in \mathbb{R}^{n\times m}$

The RR procedure generates an optimal collection of approximations $(\theta_i, \tilde{\mathbf{u}}_i)$ in the sense that for any orthonormal basis $\widetilde{\mathbf{X}}$ of $\widetilde{\mathcal{X}}$ and diagonal matrix $\mathbf{\Delta}$ the norm

$$\|\mathbf{A}\widetilde{\mathbf{X}} - \widetilde{\mathbf{X}}\mathbf{\Delta}\|_2 \tag{3.34}$$

is minimised when the columns of $\widetilde{\mathbf{X}}$ are the Ritz vectors $\tilde{\mathbf{u}}$ and $\mathbf{\Delta}$ entries are the Ritz values θ (Chapter 11 of [Parlett, 1998]). That is, Ritz vectors and values minimise the 2-norm of the residual $\mathbf{R} = \mathbf{A}\widetilde{\mathbf{X}} - \widetilde{\mathbf{X}}\Delta$. However, in general, no Ritz vector is expected to be the closest unit vector in $\widetilde{\mathcal{X}}$ to any eigenvector of $\mathbf{A}$, that is, there is no guarantee that the norm $\|\tilde{\mathbf{x}} - \mathbf{u}\|_2$, where $\tilde{\mathbf{x}} \in \widetilde{\mathcal{X}}$ is a unit vector and $\mathbf{u}$ is an eigenvector of $\mathbf{A}$, is minimised by setting $\tilde{\mathbf{x}} = \tilde{\mathbf{u}}$.

Lanczos method

The Lanczos method combines the Krylov subspaces and Rayleigh-Ritz procedure to approximate eigenpairs of symmetric matrices (Chapter 13 of [Parlett, 1998]). The idea is based on the fact that if the columns of

$$\mathbf{Q}_m = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m) \in \mathbb{R}^{n\times m} \tag{3.35}$$

form an orthonormal basis for $\mathcal{K}_m(\mathbf{A}, \mathbf{v})$ and $\mathbf{A}$ is symmetric, then (Section 12.7 of [Parlett, 1998])

$$\mathbf{T}_m = \mathbf{Q}_m^T\mathbf{A}\mathbf{Q}_m \tag{3.36}$$

$$= \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{pmatrix}, \tag{3.37}$$

where $\alpha_i = \mathbf{q}_i^T\mathbf{A}\mathbf{q}_i$, (3.38)

$\beta_i = \mathbf{q}_{i+1}^T\mathbf{A}\mathbf{q}_i$. (3.39)

Eigenvalues and eigenvectors of $\mathbf{T}_m$ are the Ritz values and primitive Ritz vectors of $\mathbf{A}$. $\mathbf{T}_m$ can be obtained from $\mathbf{T}_{m-1}$ by appending α_m and β_{m-1}.

At every Lanczos iteration, a new orthonormal basis vector $\mathbf{q}_m$ (Lanczos vector) for the Krylov subspace is generated and values α_m and β_{m-1} are computed. We present the Lanczos method in Algorithm 2. It often returns a good approximation to the extreme eigenpairs of $\mathbf{A}$ in a small number of iterations compared to n [Golub and Van Loan, 2013].

Algorithm 2 Lanczos method for computing Ritz values and vectors of a symmetric $\mathbf{A}$

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, starting vector $\mathbf{q}_1 \in \mathbb{R}^n$ such that $\|\mathbf{q}_1\|_2 = 1$

Output: orthogonal $\widetilde{\mathbf{U}} \in \mathbb{R}^{n\times m}$ with Ritz vectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{m\times m}$ with Ritz values of $\mathbf{A}$ on the diagonal

1: Set $\mathbf{q}_0 = \mathbf{0}$, $\beta_0 = 0$
2: **for** $m = 1, 2, 3...$ until convergence criteria is satisfied **do**
3: $\quad \mathbf{z}_m = \mathbf{A}\mathbf{q}_m - \beta_{m-1}\mathbf{q}_{m-1}$
4: $\quad \alpha_m = \mathbf{z}_m^T\mathbf{q}_m$
5: $\quad \mathbf{z}_m = \mathbf{z}_m - \alpha_m\mathbf{q}_m$
6: $\quad \beta_m = \mathbf{z}_m^T\mathbf{z}_m$
7: $\quad \mathbf{q}_m = \frac{\mathbf{z}_m}{\beta_m}$
8: Form EVD of $\mathbf{T}_m$ in (3.37): $\mathbf{T}_m = \mathbf{W}\mathbf{\Theta}\mathbf{W}^T$, where $\mathbf{W}$, $\mathbf{\Theta} \in \mathbb{R}^{m\times m}$
9: Form Ritz vectors $\widetilde{\mathbf{U}} = \mathbf{Q}_m\mathbf{W} \in \mathbb{R}^{n\times m}$, where $\mathbf{Q}_m$ is as in (3.35)

Notice that the algorithm orthogonalises $\mathbf{q}_{m+1}$ against $\mathbf{q}_m$ and $\mathbf{q}_{m-1}$ only. In exact arithmetic, $\mathbf{q}_{m+1}$ is orthogonal to all $\mathbf{q}_j$, $j < m+1$. However, in floating point arithmetic the orthogonality between the Lanczos vectors $\mathbf{q}_i$ can be lost quickly, because of the roundoff error. This usually happens when one of the Ritz vectors is close to convergence (Section 13.6 of [Parlett, 1998]). The loss of orthogonality results in the algorithm finding approximations to the same eigenpairs repeatedly [Parlett, 1998]. The duplicate Ritz values are called 'ghost' values and they can be avoided by complete orthogonalisation of the Lanczos vectors at every iteration, that is orthogonalising $\mathbf{q}_i$ against $\mathbf{q}_j$ for all $j < i$ (Section 10.3.5 of [Golub and Van Loan, 2013]).

Subspace iteration method

The subspace iteration method is a classic method for approximating a few leading eigenpairs, that is largest eigenvalues and associated eigenvectors (e.g., Chapter 5 of [Saad, 2011]). It can be considered as an extension of a power method which is used to find a leading eigenpair (Chapter 4 of [Parlett, 1998]). The power method is based on the fact that if $\mathbf{A}$ is a diagonalisable $n \times n$ matrix with eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$ and associated eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$, then $\mathbf{A}^k\mathbf{v}$ for an $n \times 1$ vector $\mathbf{v}$ can be written as

$$\mathbf{A}^k\mathbf{v} = \alpha_1\lambda_1^k\mathbf{u}_1 + \alpha_2\lambda_2^k\mathbf{u}_2 + \cdots + \alpha_n^k\lambda_n\mathbf{u}_n \tag{3.40}$$

and if $\alpha_1 \neq 0$, that is, if $\mathbf{v}$ is not orthogonal to $\mathbf{u}_1$, then as k increases $\mathbf{A}^k\mathbf{v}$ converges to a good approximation of $\mathbf{u}_1$. The power method considers subspaces generated by products $\mathbf{A}\mathbf{v}$, $\mathbf{A}^2\mathbf{v}$, ... in a sequence. The Krylov subspace can be contemplated as an extension

of such subspace generation method when the previous subspaces are retained (Chapter 14 of [Parlett, 1998]).

The subspace iteration considers a subspace $\mathcal{S}^k$ spanned by the columns of $\mathbf{A}^k\mathbf{V}$, where $\mathbf{V}$ is an $n \times m$ matrix, instead of the subspace generated by one vector $\mathbf{A}^k\mathbf{v}$ in the power method. In general, $\mathcal{S}^k$ includes a better approximation to $\mathbf{u}_1$ than the power method and contains approximations to the other leading eigenpairs (Chapter 6 of [Stewart, 2001]). To prevent $\mathcal{S}^k$ from converging to the subspace generated by $\mathbf{u}_1$, orthogonal bases for $\mathcal{S}^k$ are considered throughout the subspace iteration method.

The kth iteration of the subspace iteration method transforms the orthogonal basis for $\mathcal{S}^{k-1}$ to orthogonal basis for $\mathcal{S}^k$. The RR procedure is then used to extract the eigenpairs from $\mathcal{S}^k$. We detail the subspace iteration method in Algorithm 3.

Algorithm 3 Subspace iteration method with RR projection for computing Ritz values and vectors of a symmetric $\mathbf{A}$

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, orthogonal matrix $\mathbf{V}_0 \in \mathbb{R}^{n\times m}$

Output: orthogonal $\mathbf{V} \in \mathbb{R}^{n\times m}$ with Ritz vectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{m\times m}$ with Ritz values of $\mathbf{A}$ on the diagonal

1: **for** $k = 1, 2, 3...$ until convergence criteria is satisfied **do**
2: $\quad\mathbf{Z}_k = \mathbf{A}\mathbf{V}_{k-1}$
3: $\quad$Orthonormalise $\mathbf{Z}_k$ into $\mathbf{Q}_k$
4: $\quad$Form $\mathbf{G}_k = \mathbf{Q}_k^T\mathbf{A}\mathbf{Q}_k$
5: $\quad$Form EVD of $\mathbf{G}_k$: $\mathbf{G}_k = \mathbf{W}_k\mathbf{\Theta}_k\mathbf{W}_k^T$, where $\mathbf{W}_k,\ \mathbf{\Theta}_k \in \mathbb{R}^{m\times m}$
6: $\quad$Form Ritz vectors $\mathbf{V}_k = \mathbf{Q}_k\mathbf{W}_k \in \mathbb{R}^{n\times m}$

The low-rank SVD can be also found using Algorithm 3 by changing the step 5: EVD is replaced with SVD $\mathbf{G}_k = \mathbf{W}_k\mathbf{\Sigma}_k\mathbf{U}^T$, where $\mathbf{\Sigma}_k$ is diagonal matrix with singular values of $\mathbf{G}_k$ on its diagonal, and the columns of $\mathbf{W}_k$ and $\mathbf{U}^T$ are the left and right singular vectors of $\mathbf{G}_k$, respectively.

The subspace iteration method is simple to implement. The convergence of the ith leading Ritz vector depends on the ratio $|\lambda_{m+1}/\lambda_i|$ (Chapter 5 of [Saad, 2011]). It can be slow compared to the Lanczos method, but if there is a large gap between the eigenvalues we are looking for and the rest of the spectrum, that is if $|\lambda_{m+1} - \lambda_m|$ is large, then the convergence can be achieved in one or a few iterations (Chapter 14 of [Parlett, 1998]). A larger than needed subspace can be considered to accelerate the convergence. That is, if we require approximating the m leading eigenpairs, the convergence can be accelerated by using $\mathbf{V}_0 \in \mathbb{R}^{n\times(m+l)}$ in Algorithm 3, because $|\lambda_{m+l+1}/\lambda_i|$ is smaller than $|\lambda_{m+1}/\lambda_i|$.

3.3.3 Randomised methods

A thriving area of research in linear algebra focuses on randomised methods, where randomness is used to address the first stage of the low-rank approximation problem, i.e., finding a subspace that contains eigenvectors or singular vectors of $\mathbf{A}$ [Halko et al., 2011, Martinsson and Tropp, 2020]. A deterministic method is used to extract the approximation in

the second stage. The randomised methods are attractive for large problems when using parallel computing, because they minimise the communication between the processors and they are block methods, that is, they require matrix-matrix products which are easy to parallelise on current computers.

Randomised methods generate orthonormal basis $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k$ for a subspace that approximates the range of an $m \times n$ matrix $\mathbf{A}$, that is, if $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k)$, then

$$\mathbf{A} \approx \mathbf{Q}\mathbf{Q}^T\mathbf{A}. \tag{3.41}$$

The subspace spanned by $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k$ contains the leading left singular vectors of $\mathbf{A}$ or the leading eigenvectors if $m = n$. The general randomised algorithm for range approximation is presented in Algorithm 4. It is called a proto-algorithm by [Halko et al., 2011].

Algorithm 4 Randomised algorithm for approximating the range of $\mathbf{A}$

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, target rank k, oversampling parameter l

Output: orthonormal $\mathbf{Q} \in \mathbb{R}^{m\times(k+l)}$ whose range approximates the range of $\mathbf{A}$

1: Draw a random matrix $\mathbf{\Omega} \in \mathbb{R}^{n\times(k+l)}$

2: Form $\mathbf{Y} = \mathbf{A}\mathbf{\Omega} \in \mathbb{R}^{m\times(k+l)}$

3: Orthonormalise $\mathbf{Y}$ into $\mathbf{Q}$

Notice that Algorithm 4 coincides with the subspace iteration method in Algorithm 3 if it is started with a random matrix and the RR projection (steps 4-6) is omitted. The randomisation removes the risk that the subspace iteration can be detrimentally affected by a bad choice of a start matrix. We mentioned that a larger than required subspace can be used in the subspace iteration method to improve the performance; this practice is called using the 'guard vectors' by [Duff and Scott, 1993]. Such practice is referred to as oversampling for the randomised methods and we denote the oversampling parameter l. Large matrices may need larger values of l, whereas good approximations of matrices with rapidly decreasing singular values or eigenvalues may be obtained with small l values [Halko et al., 2011]. In general, [Halko et al., 2011] claim that setting l to five or ten results in a good performance of the randomised methods. If a very high quality approximation is required, larger values of l can be considered.

The expected quality of the approximation using randomised methods can be described when a Gaussian start matrix $\mathbf{\Omega}$ is used. The following theorem bounds the expected error.

Theorem 3.44 (Theorem 10.6 of [Halko et al., 2011])**.** *Let $\sigma_1 \geq \sigma_2 \geq \ldots$ be singular values of $\mathbf{A} \in \mathbb{R}^{m\times n}$, $k \geq 2$ be the target rank, and $l \geq 2$ the oversampling parameter, where $k + l \leq \min\{m, n\}$. Then the expected approximation error using Algorithm 4 with a standard normal $\mathbf{\Omega}$ is bounded by the following:*

$$\mathbb{E}\,\|\mathbf{A} - \mathbf{Q}\mathbf{Q}^T\mathbf{A}\|_2 \leq \left(1 + \sqrt{\frac{k}{l-1}}\right)\sigma_{k+1} + \frac{e\sqrt{k+l}}{l}\left(\sum_{j>k}\sigma_j^2\right)^{1/2}, \tag{3.42}$$

where e is the exponential constant.

Note that σ_{k+1} is the smallest possible error of the approximation by Theorem 3.40. If the singular values decay fast, then $\sum_{j>k} \sigma_j^2$ is small and a good approximation can be expected. If the decay of the singular values is slow, then the power method can be used to improve the performance of the randomised method. In this case, matrix $\mathbf{Y} = (\mathbf{A}\mathbf{A}^T)^q\mathbf{A}\mathbf{\Omega}$ is used in step 2 of Algorithm 4 (Section 9.3 of [Halko et al., 2011]). $(\mathbf{A}\mathbf{A}^T)^q\mathbf{A}$ has the same singular vectors as $\mathbf{A}$, but its singular values are equal to σ_i^{2q+1} where σ_i is a singular value of $\mathbf{A}$. Setting q to two or three may give a satisfactory approximation (Section 11.6 of [Martinsson and Tropp, 2020]).

3.4 The conjugate gradient method

The conjugate gradient (CG) method is a Krylov subspace iterative method used to solve systems of linear equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.43}$$

where $\mathbf{A}$ is a symmetric positive definite $n \times n$ matrix, and $\mathbf{b}$ is an $n \times 1$ vector.

If $\mathbf{x}_0$ is the initial guess for the solution $\mathbf{x}$ of (3.43) and the initial residual is $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, then the Krylov subspace is defined as $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) = span\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{m-1}\mathbf{r}_0\}$. CG generates a nested sequence of subspaces $\mathcal{K}_1(\mathbf{A}, \mathbf{r}_0) \subset \mathcal{K}_1(\mathbf{A}, \mathbf{r}_0) \subset \cdots \subseteq \mathcal{K}_n(\mathbf{A}, \mathbf{r}_0)$ by constructing iterates $\mathbf{x}_m$ such that $\mathbf{r}_m$ is orthogonal to $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$, that is $\mathbf{r}_m^T\mathbf{v} = 0$ for every $\mathbf{v} \in \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$. The algorithm is presented in Algorithm 5 (Algorithm 38.1 of [Trefethen and Bau, III, 1997]).

Algorithm 5 Conjugate gradient (CG) method for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$

Input: symmetric positive definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, right hand side $\mathbf{b} \in \mathbb{R}^n$

Output: approximate solution $\mathbf{x}_m \in \mathbb{R}^n$

1: Set $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{p}_0 = \mathbf{r}_0$
2: **for** $m = 1, 2, 3...$ until convergence criteria is satisfied **do**
3: $\quad \alpha_m = \frac{\mathbf{r}_{m-1}^T\mathbf{r}_{m-1}}{\mathbf{p}_{m-1}^T\mathbf{A}\mathbf{p}_{m-1}}$
4: $\quad \mathbf{x}_m = \mathbf{x}_{m-1} + \alpha\mathbf{p}_{m-1}$
5: $\quad \mathbf{r}_m = \mathbf{r}_{m-1} - \alpha_m\mathbf{A}\mathbf{p}_{m-1}$
6: $\quad \beta_m = \frac{\mathbf{r}_m^T\mathbf{r}_m}{\mathbf{r}_{m-1}^T\mathbf{r}_{m-1}}$
7: $\quad \mathbf{p}_m = \mathbf{r}_m + \beta_m\mathbf{p}_{m-1}$

The convergence criteria is set depending on the linear system that is solved and the application in which it arises. In data assimilation, research on the stopping criteria suggest that it can be based on the relative change in the norm of the gradient of the quadratic cost function [Lawless and Nichols, 2006], and the tolerance should not be too small so that the solution is not fitted to the observation error [Laroche and Gauthier, 1998]. Examples of the stopping criteria used in operational setting include change in the value of the quadratic cost function in the Met Office [Rawlins et al., 2007], and a fixed number of iterations in ECMWF [Fisher et al., 2009].

The error of every CG iterate is $\mathbf{e}_m = \mathbf{x} - \mathbf{x}_m$. CG monotonically minimises the *error* $\mathbf{A}$*-norm* (Lecture 38 of [Trefethen and Bau, III, 1997]) defined as

$$\|\mathbf{e}_m\|_{\mathbf{A}} = \sqrt{\mathbf{e}_m^T \mathbf{A} \mathbf{e}_m}. \tag{3.44}$$

This norm can be expressed in terms of a quadratic cost function $\phi(\mathbf{A}) = 0.5\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}$ (Lecture 38 of [Trefethen and Bau, III, 1997]), that is

$$\|\mathbf{e}_m\|_{\mathbf{A}} = 2\phi(\mathbf{A}) + c, \tag{3.45}$$

where c is a constant. Thus CG guarantees the monotonic minimisation of $\phi(\mathbf{A})$. Solving the SPD systems (2.36) and (2.38) in data assimilation using CG gives a monotonic minimisation of the quadratic cost functions (2.33) and (2.35), respectively.

The convergence of CG can be described using eigenvalues. A well known bound on the error $\mathbf{A}$-norm considers the 2-norm condition number $\kappa = \lambda_{max}(\mathbf{A})/\lambda_{min}(\mathbf{A})$ of $\mathbf{A}$.

Theorem 3.45 (Theorem 38.5 of [Trefethen and Bau, III, 1997])**.** *Let* $\mathbf{A}$ *be an SPD matrix and* κ *its 2-norm condition number. If* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *is solved using CG, then the* $\mathbf{A}$*-norms of the error* $\mathbf{e}_m$ *satisfy*

$$\frac{\|\mathbf{e}_m\|_{\mathbf{A}}}{\|\mathbf{e}_0\|_{\mathbf{A}}} \leq 2 \Big/ \left(\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^m + \left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^{-m} \right) \leq 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^m. \tag{3.46}$$

The bound in (3.46) decreases when κ, that is the distance between the largest and smallest eigenvalues, decreases. This is an upper bound that describes the worst case convergence, and does not take into account information on clustering of the eigenvalues. This is considered in the following theorem.

Theorem 3.46 (Theorem 38.3 of [Trefethen and Bau, III, 1997])**.** *Assume that CG has not converged before iteration* m*,* P_m *is a set of polynomials* p *of degree* m *with* $p(0) = 1$*, then the problem*

$$\min_{p_m \in P_m} \|p_m(\mathbf{A})\mathbf{e}_0\|_{\mathbf{A}}, \tag{3.47}$$

has a unique solution p_m *and* $\mathbf{e}_m = p_m(\mathbf{A})\mathbf{e}_0$ *is the error of iterate* $\mathbf{x}_m$*. Consequently we have*

$$\frac{\|\mathbf{e}_m\|_{\mathbf{A}}}{\|\mathbf{e}_0\|_{\mathbf{A}}} = \inf_{p_m \in P_m} \frac{\|p_m(\mathbf{A})\mathbf{e}_0\|_{\mathbf{A}}}{\|\mathbf{e}_0\|_{\mathbf{A}}} \leq \inf_{p_m \in P_m} \max_{\lambda \in \Lambda(\mathbf{A})} |p_m(\lambda)|, \tag{3.48}$$

where $\Lambda(\mathbf{A})$ *is the spectrum of* $\mathbf{A}$*.*

Hence, CG will converge in a few iterations if polynomials $p_m(\Lambda(\mathbf{A}))$ are small and rapidly decrease as n increases. This may be the case when the eigenvalues lie in small clusters or they are well separated from zero. The following corollary of Theorem 3.46 considers the clustered eigenvalues.

Theorem 3.47 (Theorem 38.4 of [Trefethen and Bau, III, 1997])**.** *If* $\mathbf{A}$ *has* m *distinct eigenvalues, then CG converges in at most* m *iterations.*

The results that we discussed are valid in the case when exact arithmetic is used. In practice, floating point arithmetic means that the orthogonality is lost and the theoretical guarantees do not hold. However, if CG is used for matrices that have suitable eigenvalue distribution, then the convergence can be achieved quickly (Lecture 38 of [Trefethen and Bau, III, 1997]). The linear systems solved with CG are often large and running n iterations is too expensive. We discuss preconditioning that can be used to transform the spectrum to a one that is better suited for CG in Section 3.6.

3.5 The minimal residual method

The minimal residual method (MINRES) solves (3.43), where $\mathbf{A}$ is symmetric positive definite or symmetric indefinite. It is often used in the latter case and we employ it for such systems in our work.

MINRES generates approximations $\mathbf{x}_m$ by creating a sequence of Krylov subspaces $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$, where $\mathbf{r}_m$ is orthogonal to $\mathbf{A}\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) = span\{\mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^m\mathbf{r}_0\}$. At every iteration $\mathbf{x}_m$ is chosen such that $\|\mathbf{r}_m\|_2$ is minimised over $\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$. Because the sequence of Krylov subspaces is nested, $\|\mathbf{r}_m\|_2$ is minimised monotonically. A possible implementation is presented in Algorithm 6 (Chapter 2 of [Greenbaum, 1997]).

Algorithm 6 Minimal residual (MINRES) method for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$

Input: symmetric positive definite or indefinite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, right hand side $\mathbf{b} \in \mathbb{R}^n$

Output: approximate solution $\mathbf{x}_m \in \mathbb{R}^n$

Set $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{r}_0 = \mathbf{b}$, $\mathbf{p}_0 = \mathbf{r}_0$
Compute $\mathbf{s}_0 = \mathbf{A}\mathbf{p}_0$
for $m = 1, 2, 3...$ until convergence criteria is satisfied **do**
    $\alpha_{m-1} = \frac{\mathbf{r}_{m-1}^T \mathbf{s}_{m-1}}{\mathbf{s}_{m-1}^T \mathbf{s}_{m-1}}$
    $\mathbf{x}_m = \mathbf{x}_{m-1} + \alpha_{m-1}\mathbf{p}_{m-1}$
    $\mathbf{r}_m = \mathbf{r}_{m-1} - \alpha_{m-1}\mathbf{s}_{m-1}$
    $\mathbf{p}_m = \mathbf{s}_{m-1}$
    $\mathbf{s}_m = \mathbf{A}\mathbf{s}_{m-1}$
    for $l = 1, 2$ **do**
        $b_{m-l}^{(m)} = \frac{\mathbf{s}_m^T \mathbf{s}_{m-1}}{\mathbf{s}_{m-l}^T \mathbf{s}_{m-l}}$
        $\mathbf{p}_m \leftarrow \mathbf{p}_m - b_{m-l}^{(m)}\mathbf{p}_{m-l}$
        $\mathbf{s}_m \leftarrow \mathbf{s}_m - b_{m-l}^{(m)}\mathbf{s}_{m-l}$

The worst case convergence of MINRES can be described with the following bound on the residual norm

$$\frac{\|\mathbf{r}_m\|}{\|\mathbf{r}_0\|} \leq \min_{p_m \in P_m} \max_{\lambda \in \Lambda(\mathbf{A})} |p_m(\lambda)|, \tag{3.49}$$

where p_m, P_m, and $\Lambda(\mathbf{A})$ are as in Theorem 3.46 [Simoncini and Szyld, 2013]. Hence, in exact arithmetic MINRES will convergence in at most m iterations if $\mathbf{A}$ has m dis-

tinct eigenvalues, and eigenvalues clustered away from zero may result in fast convergence [Wathen, 2015].

MINRES does not have a clear link to minimisation of a quadratic cost function like CG. Hence, the nonmonotonic decrease of the quadratic cost function when solving the 3×3 block system in data assimilation (see the discussion in Section 2.2.3 Saddle point formulation).

3.6 Preconditioning

Preconditioning is essential when Krylov subspace methods like CG or MINRES are used to solve large linear systems of equations [Benzi, 2002, Ferronato, 2012, Wathen, 2015, Pearson and Pestana, 2020]. Consider the left preconditioning, that is, when we solve

$$\mathbf{PAx} = \mathbf{Pb}, \tag{3.50}$$

instead of (3.43). The preconditioner $\mathbf{P}$ is constructed so that the preconditioned coefficient matrix $\mathbf{PA}$ has more favourable features than $\mathbf{A}$. These features depend on the convergence theory for the solver, e.g., if the system is solved using CG then the aim can be to reduce the condition number of the coefficient matrix, or improve the clustering of the eigenvalues when CG or MINRES is used. There is no universally good preconditioner though and they are problem dependent. In general, the requirements for the preconditioner are:

- be cheap to construct;
- be cheap to apply;
- accelerate the convergence or ensure that a higher accuracy solution is reached in a given computational time.

The preconditioner has to be cheap to apply, because for large problems $\mathbf{PA}$ is not formed and matrix vector products with $\mathbf{P}$ are performed in every iteration. If the solver has to be terminated after a fixed number of iterations, then we are interested in the performance of the preconditioner in the beginning of the iterative process.

When CG or MINRES is used, SPD preconditioners are required. Many preconditioning techniques for CG are based on approximating $\mathbf{A}^{-1}$; $\mathbf{PA}$ then approximates an identity matrix. Such a strategy is not suitable for MINRES, because $\mathbf{A}^{-1}$ is indefinite [Wathen, 2015]. We discuss specific preconditioners used in this thesis for CG and MINRES in the following sections.

3.6.1 Limited memory preconditioners

Preconditioner $\mathbf{P}$ which approximates an SPD $\mathbf{A}^{-1}$ can be constructed using ideas from quasi-Newton optimisation methods. These methods are used for unconstrained minimisation of smooth functions $f(\mathbf{x})$ by updating the estimate $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \in \mathbb{R}^n$, where

the search direction is $\mathbf{p}_k = -\mathbf{H}_k \nabla f_k$, ∇f_k is the gradient of $f(\mathbf{x})$ evaluated at $\mathbf{x}_k$, and $\mathbf{H}_k \in \mathbb{R}^{n\times n}$ is an SPD approximation of the inverse of the Hessian $\nabla^2 f(\mathbf{x})$ (e.g., chapter 6 of [Nocedal and Wright, 2006]). $\mathbf{H}_k$ is updated with the newest information about $f(\mathbf{x})$ at every iteration of the minimisation and imposing different conditions on $\mathbf{H}_k$ gives rise to different quasi-Newton methods.

One popular method is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update (e.g., [Vlček and Lukšan, 2019]) with the block version derived by [Schnabel, 1983]

$$\mathbf{H}_k = \mathbf{S}(\mathbf{S}^T\mathbf{Y})^{-1}\mathbf{S}^T + (\mathbf{I} - \mathbf{Y}(\mathbf{S}^T\mathbf{Y})^{-1}\mathbf{S}^T)^T\mathbf{H}_0(\mathbf{I} - \mathbf{Y}(\mathbf{S}^T\mathbf{Y})^{-1}\mathbf{S}^T), \tag{3.51}$$

$$\text{where } \mathbf{H}_k\mathbf{Y} = \mathbf{S}, \tag{3.52}$$

$\mathbf{Y}, \mathbf{S} \in \mathbb{R}^{n\times k}$, $k < n$, $\mathbf{S}^T\mathbf{Y} \in \mathbb{R}^{k\times k}$ is nonsingular and $\mathbf{H}_0$ is an initial approximation. The columns of $\mathbf{Y}$ and $\mathbf{S}$ are $\mathbf{y}_i = \nabla f_{i+1} - \nabla f_i$ and $\mathbf{s}_i = \mathbf{x}_{i+1} - \mathbf{x}_i$, $i \in \{1, 2, \ldots, k\}$, respectively. Storing all the vectors $\mathbf{y}_i$ and $\mathbf{s}_i$ can be too expensive for large problems, hence a limited memory version can be used where only m newest vectors are used for the update and the older ones are discarded, so that $\mathbf{S} = (\mathbf{s}_{k-m+1}, \ldots, \mathbf{s}_k)$ and $\mathbf{Y} = (\mathbf{y}_{k-m+1}, \ldots, \mathbf{y}_k)$.

Such an approach is used in constructing a class of limited memory preconditioners (LMPs) discussed by [Tshimanga et al., 2008, Gratton et al., 2011, Tshimanga, 2007]. Let $\mathbf{H}_0 = \mathbf{M}$, where $\mathbf{M}$ is SPD, and $\mathbf{Y} = \mathbf{AS}$, then LMP for $\mathbf{A}$ is

$$\mathbf{P}_m = \mathbf{S}(\mathbf{S}^T\mathbf{AS})^{-1}\mathbf{S}^T + (\mathbf{I} - \mathbf{AS}(\mathbf{S}^T\mathbf{AS})^{-1}\mathbf{S}^T)^T\mathbf{M}(\mathbf{I} - \mathbf{AS}(\mathbf{S}^T\mathbf{AS})^{-1}\mathbf{S}^T). \tag{3.53}$$

(3.53) is also called balancing preconditioner and is used in domain decomposition methods [Tang et al., 2009]. In this setup, it can be obtained by combining three $n \times n$ preconditioners $\hat{\mathbf{P}}$, $\mathbf{Q}$, and $\mathbf{M}$, where

$$\hat{\mathbf{P}} = \mathbf{I} - \mathbf{AQ} \tag{3.54}$$

$$\mathbf{Q} = \mathbf{Z}(\mathbf{Z}^T\mathbf{AZ})^{-1}\mathbf{Z}^T, \tag{3.55}$$

$\mathbf{Z} \in \mathbb{R}^{n\times m}$ is a projection subspace matrix with rank k, and $\hat{\mathbf{P}}$ is a projection matrix. If we set $\mathbf{S} = \mathbf{Z}$, then

$$\mathbf{P}_m = \mathbf{Q} + \hat{\mathbf{P}}^T\mathbf{M}\hat{\mathbf{P}}. \tag{3.56}$$

$\mathbf{M}$ is considered to be a first level preconditioner, which removes the smallest eigenvalues. In data assimilation, we set $\mathbf{M} = \mathbf{I}$, because the smallest eigenvalues of the preconditioned matrix are equal to one after the the control variable transform is applied (see Section 2.1.3).

The limited memory preconditioners have properties that are advantageous when considering the eigenvalue distribution of the preconditioned coefficient matrix. The following theorem shows that $\mathbf{P}_m$ does not expand the spectrum of the original matrix if it includes eigenvalues at one, that is, some eigenvalues of the preconditioned matrix are equal to one and the rest are bounded by the smallest and largest eigenvalues of $\mathbf{A}$.

Theorem 3.48 (Theorem 3.4 of [Gratton et al., 2011])**.** *Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be eigenvalues of $\mathbf{MA}$, $\mathbf{P}_m$ be defined as in* (3.53)*, and let $\bar{\mathbf{W}}$ be an $n \times n - m$ matrix such*

that $\bar{\mathbf{W}}^T\mathbf{AS}(\mathbf{S}^T\mathbf{AS})^{-1/2} = \mathbf{0}$ *and* $\bar{\mathbf{W}}^T\mathbf{A}\bar{\mathbf{W}} = \mathbf{I}$. *Then the set of eigenvalues* $\mu_1, \mu_2, \ldots, \mu_n$ *of* $\mathbf{P}_m\mathbf{A}$ *can be split into two subsets*

$$\lambda_j \leq \mu_j \leq \lambda_{j+m}, \quad j \in \{1, 2, \ldots, n-m\}, \tag{3.57}$$

where $\mu_1, \mu_2, \ldots, \mu_{n-m}$ *are also the eigenvalues of* $\bar{\mathbf{W}}^T\mathbf{AMAS}(\mathbf{S}^T\mathbf{AS})^{-1/2}$, *and*

$$\mu_j = 1, \quad j \in \{n-m+1, \ldots, n\}. \tag{3.58}$$

In addition, the condition number κ *of* $\mathbf{P}_m\mathbf{A}$ *can be bounded*

$$\kappa \leq \frac{\max\{1, \sigma_n\}}{\min\{1, \sigma_1\}}. \tag{3.59}$$

Recall that in data assimilation the first level preconditioning creates a cluster of eigenvalues at one. As shown by the following theorem, this cluster is preserved by the LMP and may be increased given an appropriate choice of $\mathbf{S}$.

Theorem 3.49 (Theorem 3.7 of [Gratton et al., 2011])**.** *Let* $\mathbf{P}_m$ *be defined as in* (3.53). *If* r *and* p *denote the multiplicity of one as an eigenvalue of* $\mathbf{MA}$ *and* $\mathbf{P}_m\mathbf{A}$, *respectively, then*

$$\max\{m, r\} \leq p \leq \min\{r + 2m, n\}. \tag{3.60}$$

Moreover, if the r *independent vectors of* $\mathbf{MA}$ *associated with eigenvalue one are* $\mathbf{A}$*-conjugate to the columns of* $\mathbf{S}$, *then*

$$r + m \leq p \leq \min\{r + 2m, n\}, \tag{3.61}$$

while if the columns of $\mathbf{S}$ *are* m *independent eigenvectors of* $\mathbf{MA}$ *associated with eigenvalue one, then* $p = r$.

For large problems the matrix $\mathbf{AS}$ in $\mathbf{P}_m$ is not formed and every application of $\mathbf{P}_m$ requires computing matrix-vector products with $\mathbf{A}$, which may dominate the cost of the CG iteration. This may result in $\mathbf{P}_m$ being too computationally expensive to use. Expression (3.53) can be simplified if $\mathbf{S}$ is constructed using specific vectors. We discuss two formulations, which have been used in the data assimilation setting, namely the spectral- and Ritz-LMPs [Moore et al., 2011, Mogensen et al., 2012, Laloyaux et al., 2018]. In the following, we assume that the first level preconditioning has been performed separately and set $\mathbf{M} = \mathbf{I}$.

A spectral-LMP can be obtained if m eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ of $\mathbf{A}$ and the corresponding orthonormal eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ are available. Set $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m)$, $\mathbf{\Lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, then using the low-rank eigendecomposition $\mathbf{AV} = \mathbf{V\Lambda}$ and orthogonality $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, and setting $\mathbf{S} = \mathbf{V}$ we obtain

$$\mathbf{S}(\mathbf{S}^T\mathbf{AS})^{-1}\mathbf{S}^T = \mathbf{V\Lambda}^{-1}\mathbf{V}, \tag{3.62}$$

$$\mathbf{AS}(\mathbf{S}^T\mathbf{AS})^{-1}\mathbf{S}^T = \mathbf{VV}^T. \tag{3.63}$$

Then the LMP in (3.53) can be simplified to what is known as a spectral-LMP

$$\mathbf{P}_m^{sp} = \mathbf{I} - \sum_{i=1}^{m}(1 - \lambda_i^{-1})\mathbf{v}_i\mathbf{v}_i^T. \tag{3.64}$$

The split version is $\mathbf{P}_m^{sp} = \mathbf{C}_m^{sp}(\mathbf{C}_m^{sp})^T$ with (Section 2.3.1 of [Tshimanga, 2007])

$$\mathbf{C}_m^{sp} = \prod_{i=1}^{m}\left(\mathbf{I} - \left(1 - (\sqrt{\lambda_i})^{-1}\right)\mathbf{v}_i\mathbf{v}_i^T\right). \tag{3.65}$$

A Ritz-LMP is obtained using Ritz values $\theta_1, \ldots, \theta_m$ and corresponding orthogonal Ritz vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$. Set $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$, $\mathbf{\Theta} = diag(\theta_1, \ldots, \theta_m)$ and $\mathbf{S} = \mathbf{U}$, then the Ritz-LMP is

$$\mathbf{P}_m^{Rt} = (\mathbf{I} - \mathbf{U}\mathbf{\Theta}^{-1}\mathbf{U}^T\mathbf{A})(\mathbf{I} - \mathbf{A}\mathbf{U}\mathbf{\Theta}^{-1}\mathbf{U}^T) + \mathbf{U}\mathbf{\Theta}^{-1}\mathbf{U}^T. \tag{3.66}$$

Notice that this formulation still requires computing matrix-vector products with $\mathbf{A}$. If θ_i and $\mathbf{u}_i$ are obtained using the Lanczos method, then (3.66) can be simplified to $\mathbf{P}_m^{Rt} = \mathbf{C}_m^{Rt}(\mathbf{C}_m^{Rt})^T$ with

$$\mathbf{C}_m^{Rt} = \prod_{i=1}^{m}\left(\mathbf{I} - \left(1 - (\sqrt{\theta_i})^{-1}\right)\mathbf{u}_i\mathbf{u}_i^T - \frac{\mathbf{e}_m^T\mathbf{w}_i}{\sqrt{\theta_i}}\beta_{m-1}\mathbf{u}_i\mathbf{q}_{m+1}^T\right), \tag{3.67}$$

where $\mathbf{w}_i$ is a primitive Ritz vector, $\mathbf{e}_m$ is a zero vector with 1 as its mth entry, $\mathbf{q}_{m+1}$ is a Lanczos vector and β_{m-1} is an off-diagonal entry of matrix $\mathbf{T}_m$ used in the Lanczos process.

From (3.65) and (3.67), we see that the Ritz-LMP can be considered to be the spectral-LMP with a correction term to account for the error in using approximations of the eigenpairs. In practice, the exact eigenpairs are usually unavailable and the spectral-LMP is constructed using Ritz approximations. It has been shown that this may have a detrimental effect to its performance and the Ritz-LMP can outperform the spectral-LMP in the data assimilation setting [Tshimanga et al., 2008]. However, the following theorem shows that if the Ritz approximations are well converged, then the spectral-LMP performs like the Ritz-LMP.

Theorem 3.50 (Theorem 4.5 of [Gratton et al., 2011])**.** *Suppose that m Ritz pairs $(\theta_i, \mathbf{u}_i)$ of $\mathbf{A}$ are obtained using the Lanczos method. Let $\mathbf{P}_m^{Rt}$ be the corresponding Ritz-LMP and $\widetilde{\mathbf{P}}_m^{sp}$ the spectral-LMP in* (3.64) *constructed using θ_i and $\mathbf{u}_i$ instead of λ_i and $\mathbf{v}_i$. Then*

$$\|\mathbf{P}_m^{Rt} - \widetilde{\mathbf{P}}_m^{sp}\| \leq (2 + \|\boldsymbol{\omega}\|_2)\|\boldsymbol{\omega}\|_2, \tag{3.68}$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_m)^T$ with $\omega_i = \frac{\mathbf{e}_m^T\mathbf{w}_i}{\theta_i}\beta_m$, and $\mathbf{e}_m^T$, $\mathbf{w}_i$, and β_m defined as in (3.67).

Because $|(\mathbf{e}_m^T\mathbf{w}_i)\beta_m| = \|\mathbf{A}\mathbf{u}_i - \theta_i\mathbf{u}_i\|_2$ is the residual norm for the Ritz pairs (Section 13.2 of [Parlett, 1998]), well converged Ritz pairs give small ω_i and the spectral-LMP acts like the Ritz-LMP.

Obtaining Ritz pairs using the Lanczos method is computationally expensive, and such preconditioner may not be useful. We explore a cheap way to obtain the Ritz pairs in Chapter 4.

3.6.2 Block diagonal Schur complement preconditioners

Consider system (3.43), where $\mathbf{A}$ is an indefinite matrix of the form

$$\mathbf{A} = \begin{pmatrix} \mathbf{B} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{pmatrix}, \tag{3.69}$$

with an SPD $\mathbf{B} \in \mathbb{R}^{k\times k}$ and full rank $\mathbf{C} \in \mathbb{R}^{n-k\times k}$. A way to precondition $\mathbf{A}$ comes from the following idea. Consider the symmetric positive definite matrix

$$\mathbf{P} = \begin{pmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{S} \end{pmatrix}, \tag{3.70}$$

where $\mathbf{S} = \mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T$ is the negative Schur complement of $\mathbf{B}$ in $\mathbf{A}$. The preconditioned matrix $\mathbf{P}_m^{-1}\mathbf{A}$ has three distinct eigenvalues equal to 1 and $0.5(1 \pm \sqrt{5})$, and MINRES converges in three iterations (see, e.g., section 10.1.1. of [Benzi et al., 2005]). In practice, preconditioning with (3.70) is too expensive, but it is expected that a good preconditioner $\widetilde{\mathbf{P}}$ can be constructed by using suitable SPD approximations $\widetilde{\mathbf{B}}$ to $\mathbf{B}$ and $\widetilde{\mathbf{S}}$ to $\mathbf{S}$ (section 5.2 of [Wathen, 2015]). Then the preconditioner is

$$\widetilde{\mathbf{P}} = \begin{pmatrix} \widetilde{\mathbf{B}} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{S}} \end{pmatrix}. \tag{3.71}$$

In practice we use $\widetilde{\mathbf{P}}^{-1}$, that is,

$$\widetilde{\mathbf{P}}^{-1} = \begin{pmatrix} \widetilde{\mathbf{B}}^{-1} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{S}}^{-1} \end{pmatrix}, \tag{3.72}$$

and we need approximations of the inverses of $\mathbf{B}$ and $\mathbf{S}$.

The following well-known result bounds the eigenvalues of $\widetilde{\mathbf{P}}^{-1}\mathbf{A}$ and shows that when $\widetilde{\mathbf{B}}$ and $\widetilde{\mathbf{S}}$ approximate $\mathbf{B}$ and $\mathbf{S}$ in a way that the largest and smallest eigenvalues of $\widetilde{\mathbf{B}}^{-1}\mathbf{B}$ and $\widetilde{\mathbf{S}}^{-1}\mathbf{S}$ are close to one, then eigenvalues are contained in three clusters away from zero and fast convergence of MINRES can be expected.

Theorem 3.51 ([Rees and Wathen, 2009])**.** *Let* $\mathbf{B}$*,* $\widetilde{\mathbf{B}}$*,* $\mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}$ *and* $\widetilde{\mathbf{S}}$ *be positive definite matrices,* $\mathbf{A}$ *as in* (3.69) *and* $\widetilde{\mathbf{P}}$ *as in* (3.71)*. If we denote* $\lambda_{min}(\widetilde{\mathbf{B}}^{-1}\mathbf{B}) = \delta$*,* $\lambda_{max}(\widetilde{\mathbf{B}}^{-1}\mathbf{B}) = \Delta$*,* $\lambda_{min}(\widetilde{\mathbf{S}}^{-1}\mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}) = \phi$ *and* $\lambda_{max}(\widetilde{\mathbf{S}}^{-1}\mathbf{C}^T\mathbf{B}^{-1}\mathbf{C}) = \Phi$*, where* $\lambda_{min}(\mathbf{C})$ *and* $\lambda_{max}(\mathbf{C})$ *are the smallest and largest eigenvalues of* $\mathbf{C}$*, respectively, then the eigenvalues* λ *of* $\widetilde{\mathbf{P}}^{-1}\mathbf{A}$ *are real and are bounded by*

$$\frac{1}{2}\left(\delta - \sqrt{\delta^2 + 4\Delta\Phi}\right) \leq \lambda \leq \frac{1}{2}\left(\Delta - \sqrt{\Delta^2 + 4\delta\phi}\right), \tag{3.73}$$

$$\delta \leq \lambda \leq \Delta, \tag{3.74}$$

$$\frac{1}{2}\left(\delta + \sqrt{\delta^2 + 4\delta\phi}\right) \leq \lambda \leq \frac{1}{2}\left(\Delta + \sqrt{\Delta^2 + 4\Delta\Phi}\right). \tag{3.75}$$

3.7 Summary

In this chapter, we presented results on specific matrices and their eigenvalues and singular values, which are used to better understand and bound the eigenvalues of the coefficient

matrices in the incremental weak constraint 4D-Var (Chapters 5 and 7). CG and MINRES methods, that are used to solve the linear systems, and the connection between their convergence and the distribution of the eigenvalues of the coefficient matrices were discussed. We examined using preconditioning to improve the convergence of CG and MINRES, which is explored in Chapters 4, 6 and 7. We specifically concentrated on the limited memory and block diagonal Schur complement preconditioners. In our work (Chapters 4 and 7), these are constructed using a low-rank eigenvalue decomposition, and a low-rank singular value decomposition is exploited in Chapter 6. The Lanczos and subspace iteration as well as the randomised methods that can be used to obtain the low-rank approximations were discussed. In the following chapter, we concentrate on preconditioning the linear systems of equations arising from the forcing formulation.

Chapter 4

Second level preconditioning for the forcing formulation

In this chapter, we consider the research question 1 by using LMPs to precondition the linear systems of equations in the forcing formulation independently of the previous inner loops. Three methods for randomised eigenvalue decomposition are used to construct the preconditioner. We explore if such preconditioning is useful compared to using no preconditioning and using LMPs constructed with the information from the previous inner loops, and consider the following questions. Do the results get better when the preconditioner is constructed with more approximations to the eigenvalues and eigenvectors? Should we use a large oversampling to obtain better results?

This chapter, except the summary in Section 4.8, is based on the paper: Daužickaitė, I., Lawless, A.S., Scott, J.A. and van Leeuwen, P.J. (2021) Randomised preconditioning for the forcing formulation of weak-constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 147(740), 3719 - 3734.

4.1 Abstract

There is growing awareness that errors in the model equations cannot be ignored in data assimilation methods such as four-dimensional variational assimilation (4D-Var). If allowed for, more information can be extracted from observations, longer time windows are possible, and the minimisation process is easier, at least in principle. Weak constraint 4D-Var estimates the model error and minimises a series of quadratic cost functions, which can be achieved using the conjugate gradient (CG) method; minimising each cost function is called an inner loop. CG needs preconditioning to improve its performance. In previous work, limited memory preconditioners (LMPs) have been constructed using approximations of the eigenvalues and eigenvectors of the Hessian in the previous inner loop. If the Hessian changes significantly in consecutive inner loops, the LMP may be of limited usefulness. To circumvent this, we propose using randomised methods for low-rank eigenvalue decomposition and use these approximations to cheaply construct LMPs using information from the current inner loop. Three randomised methods are compared. Numerical

experiments in idealized systems show that the resulting LMPs perform better than the existing LMPs. Using these methods may allow more efficient and robust implementations of incremental weak constraint 4D-Var.

4.2 Introduction

In numerical weather prediction, data assimilation provides the initial conditions for the weather model and hence influences the accuracy of the forecast [Kalnay, 2002]. Data assimilation uses observations of a dynamical system to correct a previous estimate (background) of the system's state. Statistical knowledge of the errors in the observations and the background is incorporated in the process. A variational data assimilation method called weak constraint 4D-Var provides a way to also take the model error into account [Trémolet, 2006], which can lead to a better analysis (e.g. [Trémolet, 2007]).

We explore the weak constraint 4D-Var cost function. In its incremental version, the state is updated by a minimiser of the linearised version of the cost function. The minimiser can be found by solving a large sparse linear system. The process of solving each system is called an inner loop. Because the second derivative of the cost function, the Hessian, is symmetric positive definite, the systems may be solved with the conjugate gradient (CG) method [Hestenes and Stiefel, 1952], whose convergence rate depends on the eigenvalue distribution of the Hessian. Limited memory preconditioners (LMPs) have been shown to improve the convergence of CG when minimising the strong constraint 4D-Var cost function [Fisher, 1998, Tshimanga et al., 2008]. Strong constraint 4D-Var differs from weak constraint 4D-Var by making the assumption that the dynamical model has no error.

LMPs can be constructed using approximations to the eigenvalues and eigenvectors (eigenpairs) of the Hessian. The Lanczos and CG connection (Section 6.7 of [Saad, 2003]) can be exploited to obtain approximations to the eigenpairs of the Hessian in one inner loop, and these approximations may then be employed to construct the preconditioner for the next inner loop [Tshimanga et al., 2008]. This approach does not describe how to precondition the first inner loop, and the number of CG iterations used on the ith inner loop limits the number of vectors available to construct the preconditioner on the $(i+1)$th inner loop. Furthermore, the success of preconditioning relies on the assumption that the Hessians do not change significantly from one inner loop to the next.

In this paper, we propose addressing these drawbacks by using easy to implement subspace iteration methods (see Chapter 5 of [Saad, 2011]) to obtain approximations of the largest eigenvalues and corresponding eigenvectors of the Hessian in the current inner loop. The subspace iteration method first approximates the range of the Hessian by multiplying it with a start matrix (for approaches to choosing it see, e.g., [Duff and Scott, 1993]) and the speed of convergence depends on the choice of this matrix (e.g., [Gu, 2015]). A variant of subspace iteration, which uses a Gaussian random start matrix, is called Randomised Eigenvalue Decomposition (REVD). REVD has been popularised

by probabilistic analysis [Halko et al., 2011, Martinsson and Tropp, 2020]. It has been shown that REVD, which is equivalent to one iteration of the subspace iteration method, can often generate a satisfactory approximation of the largest eigenpairs of a matrix that has rapidly decreasing eigenvalues. Because the Hessian is symmetric positive definite, a randomised Nyström method for computing a low-rank eigenvalue decomposition can also be used. It is expected to give a higher quality approximation than REVD (e.g. [Halko et al., 2011]). We explore these two methods and another implementation of REVD, which is based on the *ritzit* implementation of the subspace method [Rutishauser, 1971]. The methods differ in the number of matrix-matrix products with the Hessian. Even though more computations are required to generate the preconditioner in the current inner loop compared with using information from the previous inner loop, the randomised methods are block methods and hence easily parallelisable.

In Section 4.3, we discuss the weak constraint 4D-Var method and, in Section 4.4, we consider LMPs and ways to obtain spectral approximations. The three randomised methods are examined in Section 4.5. Numerical experiments with linear advection and Lorenz 96 models are presented in Section 4.6, followed by a concluding discussion in Section 4.7.

4.3 Weak constraint 4D-Var

We are interested in estimating the state evolution of a dynamical system $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$, with $\mathbf{x}_i \in \mathbb{R}^n$, at times $t_0, t_1, \ldots, t_N$. Prior information about the state at t_0 is called the background and is denoted by $\mathbf{x}^b \in \mathbb{R}^n$. It is assumed that $\mathbf{x}^b$ has Gaussian errors with zero mean and covariance matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$. Observations of the system at time t_i are denoted by $\mathbf{y}_i \in \mathbb{R}^{q_i}$ and their errors are assumed to be Gaussian with zero mean and covariance matrix $\mathbf{R}_i \in \mathbb{R}^{q_i \times q_i}$ ($q_i \ll n$). An observation operator $\mathcal{H}_i$ maps the model variables into the observed quantities at the correct location, i.e. $\mathbf{y}_i = \mathcal{H}_i(\mathbf{x}_i) + \boldsymbol{\zeta}_i$, where $\boldsymbol{\zeta}_i$ is the observation error. We assume that the observation errors are uncorrelated in time.

The dynamics of the system are described using a nonlinear model $\mathcal{M}_i$ such that

$$\mathbf{x}_{i+1} = \mathcal{M}_i(\mathbf{x}_i) + \boldsymbol{\eta}_{i+1}, \tag{4.1}$$

where $\boldsymbol{\eta}_{i+1}$ is the model error at time t_{i+1}. The model errors are assumed to be uncorrelated in time and to be Gaussian with zero mean and covariance matrix $\mathbf{Q}_i \in \mathbb{R}^{n\times n}$.

The forcing formulation of the nonlinear weak constraint 4D-Var cost function, in which we solve for the initial state and the model error realizations, is

$$J(\mathbf{x}_0, \boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_N) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)) \tag{4.2}$$

$$+ \frac{1}{2}\sum_{i=1}^{N} \boldsymbol{\eta}_i^T \mathbf{Q}_i^{-1} \boldsymbol{\eta}_i,$$

where $\mathbf{x}_i$ satisfies the model constraint in equation (4.1) [Trémolet, 2006]. The analysis (approximation of the state evolution over the time window) $\mathbf{x}_0^a, \mathbf{x}_1^a, \ldots, \mathbf{x}_N^a$ can be obtained from the minimiser of (4.2) using constraints in equation (4.1).

4.3.1 Incremental 4D-Var

One way to compute the analysis is to approximate the minimum of (4.2) with an inexact Gauss-Newton algorithm [Gratton et al., 2007], where a sequence of quadratic cost functions is minimised. In this approach, we update $\mathbf{x}_0$ and the model error

$$\mathbf{p}^{(j+1)} = \mathbf{p}^{(j)} + \delta\mathbf{p}^{(j)}, \tag{4.3}$$

where $\mathbf{p}^{(j)} = (\mathbf{x}_0^{(j)T}, \boldsymbol{\eta}_1^{(j)T}, \ldots, \boldsymbol{\eta}_N^{(j)T})^T$ is the jth approximation and the update is $\delta\mathbf{p}^{(j)} = (\delta\mathbf{x}_0^{(j)T}, \delta\boldsymbol{\eta}_1^{(j)T}, \ldots, \delta\boldsymbol{\eta}_N^{(j)T})^T$. The jth approximation of the state $\mathbf{x}^{(j)} = (\mathbf{x}_0^{(j)\ T}, \ldots, \mathbf{x}_N^{(j)\ T})^T$ is calculated with (4.1) using $\mathbf{p}^{(j)}$. The update $\delta\mathbf{p}^{(j)}$ is obtained by minimising the following cost function

$$J^{\delta}(\delta\mathbf{p}^{(j)}) = \frac{1}{2}||\delta\mathbf{p}^{(j)} - \mathbf{b}^{(j)}||^2_{\mathbf{D}^{-1}} + \frac{1}{2}||\mathbf{H}^{(j)}(\mathbf{L}^{-1})^{(j)}\delta\mathbf{p}^{(j)} - \mathbf{d}^{(j)}||^2_{\mathbf{R}^{-1}}, \tag{4.4}$$

where $||\mathbf{a}||^2_{\mathbf{A}^{-1}} = \mathbf{a}^T\mathbf{A}^{-1}\mathbf{a}$ and the covariance matrices are block diagonal, i.e. $\mathbf{D} = diag(\mathbf{B}, \mathbf{Q}_1, \ldots, \mathbf{Q}_N) \in \mathbb{R}^{n(N+1)\times n(N+1)}$ and $\mathbf{R} = diag(\mathbf{R}_0, \ldots, \mathbf{R}_N) \in \mathbb{R}^{q\times q}$, where $q = \Sigma_{i=0}^N q_i$. We use the notation (following [Gratton et al., 2018b]) $\mathbf{H}^{(j)} = diag(\mathbf{H}_0^{(j)}, \ldots, \mathbf{H}_N^{(j)}) \in \mathbb{R}^{q\times n(N+1)}$, where $\mathbf{H}_i^{(j)}$ is the linearised observation operator, and

$$(\mathbf{L}^{-1})^{(j)} = \begin{pmatrix} \mathbf{I} & & & & \\ \mathbf{M}_{0,0}^{(j)} & \mathbf{I} & & & \\ \mathbf{M}_{0,1}^{(j)} & \mathbf{M}_{1,1}^{(j)} & \mathbf{I} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{M}_{0,N-1}^{(j)} & \mathbf{M}_{1,N-1}^{(j)} & \cdots & \mathbf{M}_{N-1,N-1}^{(j)} & \mathbf{I} \end{pmatrix}, \tag{4.5}$$

$$\mathbf{b}^{(j)} = \begin{pmatrix} \mathbf{x}^b - \mathbf{x}_0^{(j)} \\ -\boldsymbol{\eta}_1^{(j)} \\ \vdots \\ -\boldsymbol{\eta}_N^{(j)} \end{pmatrix}, \tag{4.6}$$

$$\mathbf{d}^{(j)} = \begin{pmatrix} \mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0^{(j)}) \\ \mathbf{y}_1 - \mathcal{H}_1(\mathbf{x}_1^{(j)}) \\ \vdots \\ \mathbf{y}_N - \mathcal{H}_N(\mathbf{x}_N^{(j)}) \end{pmatrix}, \tag{4.7}$$

where $\mathbf{M}_{i,l}^{(j)} = \mathbf{M}_l^{(j)} \ldots \mathbf{M}_i^{(j)}$ and $\mathbf{M}_i^{(j)}$ is the linearised model, i.e. $\mathbf{M}_{i,l}^{(j)}$ denotes the linearised model integration from time t_i to t_{l+1}, $(\mathbf{L}^{-1})^{(j)} \in \mathbb{R}^{n(N+1)\times n(N+1)}$, $\mathbf{x}^{(j)}, \delta\mathbf{x}^{(j)}, \mathbf{b}^{(j)} \in \mathbb{R}^{n(N+1)}$ and $\mathbf{d}^{(j)} \in \mathbb{R}^q$. The outer loop consists of updating (4.3), calculating $\mathbf{x}^{(j)}, \mathbf{b}^{(j)}, \mathbf{d}^{(j)}$, and linearising $\mathcal{H}_i$ and $\mathcal{M}_i$ for the next inner loop.

The minimum of the quadratic cost function (4.4) can be found by solving a linear system

$$\boldsymbol{\mathcal{A}}^{(j)}\delta\mathbf{p}^{(j)} = \mathbf{D}^{-1}\mathbf{b}^{(j)} + (\mathbf{L}^{-T})^{(j)}(\mathbf{H}^{T})^{(j)}\mathbf{R}^{-1}\mathbf{d}^{(j)}, \tag{4.8}$$

$$\boldsymbol{\mathcal{A}}^{(j)} = (\mathbf{D}^{-1} + (\mathbf{L}^{-T})^{(j)}(\mathbf{H}^{T})^{(j)}\mathbf{R}^{-1}(\mathbf{H})^{(j)}(\mathbf{L}^{-1})^{(j)}) \in \mathbb{R}^{n(N+1)\times n(N+1)}, \tag{4.9}$$

where $\boldsymbol{\mathcal{A}}^{(j)}$ is the Hessian of (4.4), which is symmetric positive definite. These large sparse systems are usually solved with the conjugate gradient (CG) method, whose convergence properties depend on the spectrum of $\boldsymbol{\mathcal{A}}^{(j)}$ (see Section 4.4.1 for a discussion). In general, clustered eigenvalues result in fast convergence. We consider a technique to cluster eigenvalues of $\boldsymbol{\mathcal{A}}^{(j)}$ in the following section. From now on we omit the superscript (j).

4.3.2 Control Variable Transform

A control variable transform, which is also called first level preconditioning, maps the variables $\delta\mathbf{p}$ to $\delta\tilde{\mathbf{p}}$, whose errors are uncorrelated (see, e.g. Section 3.2 of [Lawless, 2013]). This can be denoted by the transformation $\mathbf{D}^{1/2}\delta\tilde{\mathbf{p}} = \delta\mathbf{p}$, where $\mathbf{D} = \mathbf{D}^{1/2}\mathbf{D}^{1/2}$ and $\mathbf{D}^{1/2}$ is the symmetric square root. The update $\delta\tilde{\mathbf{p}}$ is then the solution of

$$\boldsymbol{\mathcal{A}}^{pr}\delta\tilde{\mathbf{p}} = \mathbf{D}^{-1/2}\mathbf{b} + \mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{d}, \tag{4.10}$$

$$\text{where } \boldsymbol{\mathcal{A}}^{pr} = \mathbf{I} + \mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}. \tag{4.11}$$

Here, $\boldsymbol{\mathcal{A}}^{pr}$ is the sum of the identity matrix and a rank q positive semi-definite matrix. Hence, it has a cluster of $n(N+1) - q$ eigenvalues at one and q eigenvalues that are greater than one. Thus, the spectral condition number $\kappa = \lambda_{max}/\lambda_{min}$ (here λ_{max} and λ_{min} are the largest and smallest eigenvalues of $\boldsymbol{\mathcal{A}}^{pr}$, respectively) is equal to λ_{max}. We discuss employing second level preconditioning to reduce the condition number while also preserving the cluster of the eigenvalues at one. In the subsequent sections, we use notation that is common in numerical linear algebra. Namely, we use $\mathbf{A}$ for the Hessian with first level preconditioning, $\mathbf{x}$ for the unknown and $\mathbf{b}$ for the right hand side of the system of linear equations. Thus, we denote (4.10) by

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4.12}$$

where the right hand side $\mathbf{b}$ is known and $\mathbf{x}$ is the required solution. We assume throughout that $\mathbf{A}$ is symmetric positive definite.

4.4 Preconditioning weak constraint 4D-Var

4.4.1 Preconditioned conjugate gradient

The CG method (see, e.g. [Saad, 2003]) is a popular Krylov subspace method for solving systems of the form (4.12). A well known bound for the error at the ith CG iteration $\boldsymbol{\epsilon}_i = \mathbf{x} - \mathbf{x}_i$ is

$$\frac{||\boldsymbol{\epsilon}_i||_{\mathbf{A}}}{||\boldsymbol{\epsilon}_0||_{\mathbf{A}}} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{i}, \tag{4.13}$$

where κ is the spectral condition number and $||\boldsymbol{\epsilon}_i||^2_{\mathbf{A}} = \boldsymbol{\epsilon}_i^T \mathbf{A} \boldsymbol{\epsilon}_i$ (see, e.g., Section 5.1. of [Nocedal and Wright, 2006]). Note that this bound describes the worst-case convergence and only takes into account the largest and smallest eigenvalues. The convergence of CG also depends on the distribution of the eigenvalues of $\mathbf{A}$ (as well as the right hand side $\mathbf{b}$ and the initial guess $\mathbf{x}_0$); eigenvalues clustered away from zero suggest rapid convergence (Lecture 38 of [Trefethen and Bau, III, 1997]). Otherwise, CG can display slow convergence and preconditioning is used to try to tackle this problem (Chapter 9 of [Saad, 2003]). Preconditioning aims to map the system (4.12) to another system that has the same solution, but different properties that imply faster convergence. Ideally, the preconditioner $\mathbf{P}$ should be cheap both to construct and to apply, and the preconditioned system should be easy to solve.

If $\mathbf{P}$ is a symmetric positive definite matrix that approximates $\mathbf{A}^{-1}$ and is available in factored form $\mathbf{P} = \mathbf{C}\mathbf{C}^T$, the following system is solved

$$\mathbf{C}^T \mathbf{A} \mathbf{C} \hat{\mathbf{x}} = \mathbf{C}^T \mathbf{b}, \tag{4.14}$$

where $\hat{\mathbf{x}} = \mathbf{C}^{-1}\mathbf{x}$. Split preconditioned CG (PCG) for solving (4.14) is described in Algorithm 7 (see, for example, Algorithm 9.2 of [Saad, 2003]). Note that every CG iteration involves one matrix-vector product with $\mathbf{A}$ (the product $\mathbf{A}\mathbf{p}_{j-1}$ is stored in step 3 and reused in step 5) and this is expensive in weak constraint 4D-Var, because it involves running the linearised model throughout the assimilation window through the factor $\mathbf{L}^{-1}$.

Algorithm 7 Split preconditioned CG for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$

Input: $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, $\mathbf{b} \in \mathbb{R}^{n_A}$, preconditioner $\mathbf{P} = \mathbf{C}\mathbf{C}^T \in \mathbb{R}^{n_A \times n_A}$, initial solution $\mathbf{x}_0 \in \mathbb{R}^{n_A}$

Output: solution $\mathbf{x}_j \in \mathbb{R}^{n_A}$

1: Compute $\mathbf{r}_0 = \mathbf{C}^T(\mathbf{b} - \mathbf{A}\mathbf{x}_0)$, and $\mathbf{p}_0 = \mathbf{C}\mathbf{r}_0$
2: **for** $j = 1, 2, \ldots,$ until convergence **do**
3: $\quad \alpha_j = (\mathbf{r}_{j-1}^T \mathbf{r}_{j-1})/(\mathbf{p}_{j-1}^T \mathbf{A} \mathbf{p}_{j-1})$
4: $\quad \mathbf{x}_j = \mathbf{x}_{j-1} + \alpha_j \mathbf{p}_{j-1}$
5: $\quad \mathbf{r}_j = \mathbf{r}_{j-1} - \alpha_j \mathbf{C}^T \mathbf{A} \mathbf{p}_{j-1}$
6: $\quad \beta_j = (\mathbf{r}_j^T \mathbf{r}_j)/(\mathbf{r}_{j-1}^T \mathbf{r}_{j-1})$
7: $\quad \mathbf{p}_j = \mathbf{C}\mathbf{r}_j + \beta_j \mathbf{p}_{j-1}$

4.4.2 Limited memory preconditioners

In weak constraint 4D-Var, the preconditioner $\mathbf{P}$ approximates the inverse Hessian. Hence, $\mathbf{P}$ can be obtained using Quasi-Newton methods for unconstrained optimization that construct an approximation of the Hessian matrix, which is updated regularly (see, for example, Chapter 6 of [Nocedal and Wright, 2006]). A popular method to approximate the Hessian is BFGS (named after Broyden, Fletcher, Goldfarb, and Shanno, who proposed it), but it is too expensive in terms of storage and updating the approximation. Instead, the so-called block BFGS method (derived by [Schnabel, 1983]) uses only a limited number

of vectors to build the Hessian, and when new vectors are added older ones are dropped. This is an example of a limited memory preconditioner (LMP), and the one considered by Tshimanga et al. (see [Tshimanga et al., 2008, Gratton et al., 2011] and [Tshimanga, 2007]) in the context of strong constraint 4D-Var. An LMP for an $n_A \times n_A$ symmetric positive definite matrix $\mathbf{A}$ is defined as follows

$$\mathbf{P}_k = (\mathbf{I}_{n_A} - \mathbf{S}(\mathbf{S}^T\mathbf{A}\mathbf{S})^{-1}\mathbf{S}^T\mathbf{A})(\mathbf{I}_{n_A} - \mathbf{A}\mathbf{S}(\mathbf{S}^T\mathbf{A}\mathbf{S})^{-1}\mathbf{S}^T) + \mathbf{S}(\mathbf{S}^T\mathbf{A}\mathbf{S})^{-1}\mathbf{S}^T, \tag{4.15}$$

where $\mathbf{S}$ is an $n_A \times k$ ($k \leq n_A$) matrix with linearly independent columns $\mathbf{s}_1, \ldots, \mathbf{s}_k$, and $\mathbf{I}_{n_A}$ is the $n_A \times n_A$ identity matrix [Gratton et al., 2011]. $\mathbf{P}_k$ is symmetric positive definite and if $k = n_A$ then $(\mathbf{S}^T\mathbf{A}\mathbf{S})^{-1} = \mathbf{S}^{-1}\mathbf{A}^{-1}\mathbf{S}^{-T}$ and $\mathbf{P}_k = \mathbf{A}^{-1}$. In data assimilation, we have $k \ll n_A$, hence the name LMPs. $\mathbf{P}_k$ is called a balancing preconditioner in [Tang et al., 2009].

A potential problem for practical applications of (4.15) is the need for expensive matrix-matrix products with $\mathbf{A}$. Simpler formulations of (4.15) are obtained by imposing more conditions on the vectors $\mathbf{s}_1, \ldots, \mathbf{s}_k$. Two formulations that [Tshimanga et al., 2008] call spectral-LMP and Ritz-LMP have been used, for example, in ocean data assimilation in the Regional Ocean Modeling System (ROMS) [Moore et al., 2011] and the variational data assimilation software with the Nucleus for European Modelling of the Ocean (NEMO) ocean model (NEMOVAR) [Mogensen et al., 2012], and coupled climate reanalysis in Coupled ECMWF ReAnalysis (CERA) [Laloyaux et al., 2018].

To obtain the spectral-LMP, let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be orthonormal eigenvectors of $\mathbf{A}$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_k$. Set $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_k)$ and $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_k)$ so that $\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$. If $\mathbf{s}_i = \mathbf{v}_i$, $i = 1, \ldots, k$, then the LMP in (4.15) is the spectral-LMP $\mathbf{P}_k^{sp}$ (it is called a deflation preconditioner in [Giraud and Gratton, 2006]), which can be simplified to

$$\mathbf{P}_k^{sp} = \mathbf{I}_{n_A} - \sum_{i=1}^{k}(1 - \lambda_i^{-1})\mathbf{v}_i\mathbf{v}_i^T. \tag{4.16}$$

Then $\mathbf{P}_k^{sp} = \mathbf{C}_k^{sp}(\mathbf{C}_k^{sp})^T$ with (presented in Section 2.3.1 of [Tshimanga, 2007])

$$\mathbf{C}_k^{sp} = \prod_{i=1}^{k}\left(\mathbf{I}_{n_A} - \left(1 - \left(\sqrt{\lambda_i}\right)^{-1}\right)\mathbf{v}_i\mathbf{v}_i^T\right). \tag{4.17}$$

In many applications, including data assimilation, exact eigenpairs are not known, and their approximations, called Ritz values and vectors, are used (we discuss these in the following section). If $\mathbf{u}_1, \ldots, \mathbf{u}_k$ are orthogonal Ritz vectors, then the following relation holds: $\mathbf{U}^T\mathbf{A}\mathbf{U} = \mathbf{\Theta}$, where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$, $\mathbf{\Theta} = diag(\theta_1, \ldots, \theta_k)$ and θ_i is a Ritz value. Setting $\mathbf{s}_i = \mathbf{u}_i$, $i = 1, \ldots, k$, the Ritz-LMP $\mathbf{P}_k^{Rt}$ is

$$\mathbf{P}_k^{Rt} = (\mathbf{I}_{n_A} - \mathbf{U}\mathbf{\Theta}^{-1}\mathbf{U}^T\mathbf{A})(\mathbf{I}_{n_A} - \mathbf{A}\mathbf{U}\mathbf{\Theta}^{-1}\mathbf{U}^T) + \mathbf{U}\mathbf{\Theta}^{-1}\mathbf{U}^T. \tag{4.18}$$

Each application of $\mathbf{P}_k^{Rt}$ requires a matrix-matrix product with $\mathbf{A}$. If the Ritz vectors are obtained by the Lanczos process (described in Section 4.4.4 below), then (4.18) can

be further simplified, so that no matrix-matrix products with $\mathbf{A}$ are needed (see Section 4.2.2. of [Gratton et al., 2011] for details).

An important property is that if an LMP is constructed using k vectors then at least k eigenvalues of the preconditioned matrix $\mathbf{C}^T\mathbf{A}\mathbf{C}$ will be equal to 1, and the remaining eigenvalues will lie between the smallest and largest eigenvalues of $\mathbf{A}$ (see Theorem 3.4 of [Gratton et al., 2011]). Moreover, if $\mathbf{A}$ has a cluster of eigenvalues at 1, then LMPs preserve this cluster. This is crucial when preconditioning (4.10): because the LMPs preserve the $n(N+1)-q$ smallest eigenvalues of $\boldsymbol{\mathcal{A}}^{pr}$ that are equal to 1, the CG convergence can be improved by decreasing the largest eigenvalues. Hence, it is preferable to use the largest eigenpairs or their approximations.

In practice, both spectral-LMP and Ritz-LMP use Ritz vectors and values to construct the LMPs. It has been shown that the Ritz-LMP can perform better than the spectral-LMP in a strong constraint 4D-Var setting by correcting for the inaccuracies in the estimates of eigenpairs [Tshimanga et al., 2008]. However, [Gratton et al., 2011] (their Theorem 4.5) have shown that if the preconditioners are constructed with Ritz vectors and values that have converged, then the spectral-LMP acts like the Ritz-LMP.

4.4.3 Ritz information

Calculating or approximating all the eigenpairs of a large sparse matrix is impractical. Hence, only a subset is approximated to construct the LMPs. This is often done by extracting approximations from a subspace, and the Rayleigh-Ritz (RR) procedure is a popular method for doing this.

Assume that $\mathcal{Z} \subset \mathbb{R}^{n_A}$ is an invariant subspace of $\mathbf{A}$, i.e. $\mathbf{A}\mathbf{z} \in \mathcal{Z}$ for every $\mathbf{z} \in \mathcal{Z}$, and the columns of $\mathbf{Z} \in \mathbb{R}^{n_A \times m}$, $m < n_A$, form an orthonormal basis for $\mathcal{Z}$. If $(\lambda, \hat{\mathbf{y}})$ is an eigenpair of $\mathbf{K} = \mathbf{Z}^T\mathbf{A}\mathbf{Z} \in \mathbb{R}^{m \times m}$, then $(\lambda, \mathbf{v})$, where $\mathbf{v} = \mathbf{Z}\hat{\mathbf{y}}$, is an eigenpair of $\mathbf{A}$ (see, e.g. Theorem 1.2 in Chapter 4 of [Stewart, 2001]). Hence, eigenvalues of $\mathbf{A}$ that lie in the subspace $\mathcal{Z}$ can be extracted by solving a small eigenvalue problem.

However, generally the computed subspace $\widetilde{\mathcal{Z}}$ with orthonormal basis as columns of $\widetilde{\mathbf{Z}}$ is not invariant. Hence, only approximations $\tilde{\mathbf{v}}$ to the eigenvectors $\mathbf{v}$ belong to $\widetilde{\mathcal{Z}}$. The RR procedure computes approximations $\mathbf{u}$ to $\tilde{\mathbf{v}}$. We give the RR procedure in Algorithm 8, where the eigenvalue decomposition is abbreviated as EVD. Approximations to eigenvalues λ are called Ritz values θ, and $\mathbf{u}$ are the Ritz vectors. Eigenvectors of $\widetilde{\mathbf{K}} = \widetilde{\mathbf{Z}}^T\mathbf{A}\widetilde{\mathbf{Z}}$, which is the projection of $\mathbf{A}$ onto $\widetilde{\mathcal{Z}}$, are denoted by $\mathbf{w}$ and are called primitive Ritz vectors.

4.4.4 Spectral information from CG

[Tshimanga et al., 2008] use Ritz pairs of the Hessian in one inner loop to construct LMPs for the following inner loop, i.e. information on $\mathbf{A}^{(0)}$ is used to precondition $\mathbf{A}^{(1)}$, and so on. Success relies on the Hessians not changing significantly from one inner loop to the next. Ritz information can be obtained from the Lanczos process that is connected to CG, hence information for the preconditioner can be gathered at a negligible cost.

Algorithm 8 Rayleigh-Ritz procedure for computing approximations of eigenpairs of symmetric $\mathbf{A}$

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, orthogonal matrix $\widetilde{\mathbf{Z}} \in \mathbb{R}^{n_A \times m}$, $m < n_A$

Output: orthogonal $\mathbf{U} \in \mathbb{R}^{n_A \times m}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{m \times m}$ with approximations to eigenvalues of $\mathbf{A}$ on the diagonal

1: Form $\widetilde{\mathbf{K}} = \widetilde{\mathbf{Z}}^T \mathbf{A} \widetilde{\mathbf{Z}} \in \mathbb{R}^{m \times m}$

2: Form EVD of $\widetilde{\mathbf{K}}$: $\widetilde{\mathbf{K}} = \mathbf{W} \mathbf{\Theta} \mathbf{W}^T$, where $\mathbf{W}$, $\mathbf{\Theta} \in \mathbb{R}^{m \times m}$

3: Form Ritz vectors $\mathbf{U} = \widetilde{\mathbf{Z}} \mathbf{W} \in \mathbb{R}^{n_A \times m}$

The Lanczos process is used to obtain estimates of a few extremal eigenvalues and corresponding eigenvectors of a symmetric matrix $\mathbf{A}$ (Section 10.1 of [Golub and Van Loan, 2013]). It produces a sequence of tridiagonal matrices $\mathbf{T}_j \in \mathbb{R}^{j \times j}$, whose largest and smallest eigenvalues converge to the largest and smallest eigenvalues of $\mathbf{A}$. Given a starting vector $\mathbf{f}_0$, it also computes an orthonormal basis $\mathbf{f}_0, \ldots, \mathbf{f}_{j-1}$ for the Krylov subspace $\mathcal{K}_j = span\{\mathbf{f}_0, \mathbf{A}\mathbf{f}_0, \ldots, \mathbf{A}^{j-1}\mathbf{f}_0\}$. Ritz values θ_i are obtained as eigenvalues of a tridiagonal matrix, which has the following structure:

$$\mathbf{T}_j = \begin{pmatrix} \gamma_1 & \tau_1 & & \\ \tau_1 & \gamma_2 & \tau_2 & \\ & \ddots & \ddots & \ddots \\ & & \tau_{j-1} & \gamma_j \end{pmatrix}. \tag{4.19}$$

The Ritz vectors of $\mathbf{A}$ are $\mathbf{u}_i = \mathbf{F}_j \mathbf{w}_i$, where $\mathbf{F}_j = (\mathbf{f}_0, \ldots, \mathbf{f}_{j-1})$ and an eigenvector $\mathbf{w}_i$ of $\mathbf{T}_j$ is a primitive Ritz vector. Eigenpairs of $\mathbf{T}_j$ can be obtained using a symmetric tridiagonal QR algorithm or Jacobi procedures (see, e.g. Section 8.5 of [Golub and Van Loan, 2013]).

Saad (see Section 6.7.3 of [Saad, 2003]) discusses obtaining entries of $\mathbf{T}_j$ when solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ with CG. At the jth iteration of CG, new entries of $\mathbf{T}_j$ are calculated as follows

$$\gamma_j = \begin{cases} \frac{1}{\alpha_j} & \text{for } j = 1 \\ \frac{1}{\alpha_j} + \frac{\beta_{j-1}}{\alpha_{j-1}} & \text{for } j > 1 \end{cases} \tag{4.20}$$

$$\tau_j = \frac{\sqrt{\beta_j}}{\alpha_j}, \tag{4.21}$$

and the vector $\mathbf{f}_j = \mathbf{r}_j / ||\mathbf{r}_j||$, where $||\mathbf{r}_j||^2 = \mathbf{r}_j^T \mathbf{r}_j$ and α_j, β_j and $\mathbf{r}_j$ are as in Algorithm 7. Hence, obtaining eigenvalue information requires normalizing the residual vectors and finding eigenpairs of the tridiagonal matrix $\mathbf{T}_j$. In data assimilation, the dimension of $\mathbf{T}_j$ is small, because the cost of matrix-vector products restricts the number of CG iterations in the previous inner loop. Hence its eigenpairs can be calculated cheaply. However, caution has to be taken to avoid 'ghost' values, i.e. repeated Ritz values, due to the loss of orthogonality in CG (Section 10.3.5 of [Golub and Van Loan, 2013]). This can be addressed using a complete reorthogonalization in every CG iteration, which is done in the CONGRAD routine used at the European Centre for Medium Range Weather Forecasts

[ECMWF, 2020]. This makes every CG iteration more expensive, but CG may converge in fewer iterations [Fisher, 1998].

4.5 Randomised eigenvalue decomposition

If the Hessian in one inner loop differs significantly from the Hessian in the previous inner loop, then it may not be useful to precondition the former with an LMP that is constructed with information from the latter. Employing the Lanczos process to obtain eigenpair estimates and use them to construct an LMP in the same inner loop is too computationally expensive, because each iteration of the Lanczos process requires a matrix-vector product with the Hessian, thus the cost is similar to the cost of CG. Hence, we explore a different approach.

Subspace iteration is a simple procedure to obtain approximations to the largest eigenpairs (see, e.g., Chapter 5 of [Saad, 2011]). It is easily understandable and can be implemented in a straightforward manner, although its convergence can be very slow if the largest eigenvalues are not well separated from the rest of the spectrum. The accuracy of subspace iteration may be enhanced by using an RR projection.

Such an approach is used in the Randomised Eigenvalue Decomposition (REVD: see, e.g., [Halko et al., 2011]). This takes a Gaussian random matrix, i.e. a matrix with independent standard normal random variables with zero mean and variance equal to one as its entries, and applies one iteration of the subspace iteration method with RR projection, hence obtaining a rank m approximation $\mathbf{A} \approx \mathbf{Z}_1(\mathbf{Z}_1^T\mathbf{A}\mathbf{Z}_1)\mathbf{Z}_1^T$, where $\mathbf{Z}_1 \in \mathbb{R}^{n_A \times m}$ is orthogonal. We present REVD in Algorithm 9. An important feature of REVD is the observation that the accuracy of the approximation is enhanced with oversampling (which is also called 'using guard vectors' in [Duff and Scott, 1993]), i.e. working on a larger space than the required number of Ritz vectors. [Halko et al., 2011] claim that setting the oversampling parameter to 5 or 10 is often sufficient.

Algorithm 9 Randomised eigenvalue decomposition, REVD

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, target rank k, an oversampling parameter l

Output: orthogonal $\mathbf{U}_1 \in \mathbb{R}^{n_A \times k}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta}_1 \in \mathbb{R}^{k \times k}$ with approximations to the largest eigenvalues of $\mathbf{A}$ on the diagonal

1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n_A \times (k+l)}$
2: Form a sample matrix $\mathbf{Y} = \mathbf{A}\mathbf{G} \in \mathbb{R}^{n_A \times (k+l)}$
3: Orthonormalize the columns of $\mathbf{Y}$ to obtain orthonormal $\mathbf{Z}_1 \in \mathbb{R}^{n_A \times (k+l)}$
4: Form $\mathbf{K}_1 = \mathbf{Z}_1^T\mathbf{A}\mathbf{Z}_1 \in \mathbb{R}^{(k+l)\times(k+l)}$
5: Form EVD of $\mathbf{K}$: $\mathbf{K} = \mathbf{W}_1\mathbf{\Theta}_1\mathbf{W}_1^T$, where $\mathbf{W}_1,\ \mathbf{\Theta}_1 \in \mathbb{R}^{(k+l)\times(k+l)}$, elements of $\mathbf{\Theta}_1$ are sorted in decreasing order
6: Remove last l columns and rows of $\mathbf{\Theta}_1$, so that $\mathbf{\Theta}_1 \in \mathbb{R}^{k \times k}$
7: Remove last l columns of $\mathbf{W}_1$, so that $\mathbf{W}_1 \in \mathbb{R}^{(k+l)\times k}$
8: Form $\mathbf{U}_1 = \mathbf{Z}_1\mathbf{W}_1 \in \mathbb{R}^{n_A \times k}$.

Randomised algorithms are designed to minimise the communication instead of the flop count. The expensive parts of Algorithm 9 are the two matrix-matrix products $\mathbf{AG}$ and $\mathbf{AZ}_1$ in steps 2 and 4, that is, in each of these steps, matrix $\mathbf{A}$ has to be multiplied with $(k+l)$ vectors, which in serial computations would be essentially the cost of $2(k+l)$ iterations of unpreconditioned CG. However, note that these matrix-matrix products can be parallelised.

In weak constraint 4D-Var, $\mathbf{A}$ is the Hessian, hence it is symmetric positive definite and its eigenpairs can also be approximated using a randomised Nyström method (Algorithm 5.5. of [Halko et al., 2011]), which is expected to give much more accurate results than REVD [Halko et al., 2011]. We present the Nyström method in Algorithm 10, where singular value decomposition is abbreviated as SVD. It considers a more elaborate rank m approximation than in REVD: $\mathbf{A} \approx (\mathbf{AZ}_1)(\mathbf{Z}_1^T\mathbf{AZ}_1)^{-1}(\mathbf{AZ}_1)^T = \mathbf{FF}^T$, where $\mathbf{Z}_1 \in \mathbb{R}^{n_A \times m}$ is orthogonal (obtained in the same way as in REVD, e.g. using a tall skinny QR (TSQR) decomposition [Demmel et al., 2012]). The eigenvalues of $\mathbf{FF}^T$ are the squares of the singular values of $\mathbf{F}$ (see section 2.4.2 of [Golub and Van Loan, 2013]). In numerical computations we store matrices $\mathbf{E}^{(1)} = \mathbf{AZ}_1$ and $\mathbf{E}^{(2)} = \mathbf{Z}_1^T\mathbf{E}^{(1)} = \mathbf{Z}_1^T\mathbf{AZ}_1$ (step 4), perform the Cholesky factorization of $\mathbf{E}^{(2)} = \mathbf{C}^T\mathbf{C}$ (step 5) and obtain $\mathbf{F}$ by solving the triangular system $\mathbf{FC} = \mathbf{E}^{(1)}$.

Algorithm 10 Randomised eigenvalue decomposition for symmetric positive semi-definite $\mathbf{A}$, Nyström

Input: symmetric positive semi-definite matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, target rank k, an oversampling parameter l

Output: orthogonal $\mathbf{U}_2 \in \mathbb{R}^{n_A \times k}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta}_2 \in \mathbb{R}^{k \times k}$ with approximations to the largest eigenvalues of $\mathbf{A}$ on the diagonal

1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n_A \times (k+l)}$
2: Form a sample matrix $\mathbf{Y} = \mathbf{AG} \in \mathbb{R}^{n_A \times (k+l)}$
3: Orthonormalize the columns of $\mathbf{Y}$ to obtain orthonormal $\mathbf{Z}_1 \in \mathbb{R}^{n_A \times (k+l)}$
4: Form matrices $\mathbf{E}^{(1)} = \mathbf{AZ}_1 \in \mathbb{R}^{n_A \times (k+l)}$ and $\mathbf{E}^{(2)} = \mathbf{Z}_1^T\mathbf{E}^{(1)} \in \mathbb{R}^{(k+l) \times (k+l)}$
5: Form a Cholesky factorization $\mathbf{E}^{(2)} = \mathbf{C}^T\mathbf{C}$
6: Solve $\mathbf{FC} = \mathbf{E}^{(1)}$ for $\mathbf{F} \in \mathbb{R}^{n_A \times (k+l)}$
7: Form SVD of $\mathbf{F}$: $\mathbf{F} = \mathbf{U}_2\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}_2$, $\mathbf{V} \in \mathbb{R}^{n_A \times (k+l)}$, $\mathbf{\Sigma} \in \mathbb{R}^{(k+l) \times (k+l)}$, elements of $\mathbf{\Sigma}$ are sorted in decreasing order
8: Remove last l columns of $\mathbf{U}_2$, so that $\mathbf{U}_2 \in \mathbb{R}^{n_A \times k}$
9: Remove last l columns and rows of $\mathbf{\Sigma}$, so that $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$, and set $\mathbf{\Theta}_2 = \mathbf{\Sigma}^2$

The matrix-matrix product with $\mathbf{A}$ at step 4 of Algorithms 9 and 10 is removed in Rutishauser's implementation of subspace iteration with RR projection called *ritzit* [Rutishauser, 1971]. It can be derived in the following manner (see Chapter 14 of [Parlett, 1998]). Assume that $\mathbf{G}_3 \in \mathbb{R}^{n_A \times m}$ is an orthogonal matrix and the sample matrix is $\mathbf{Y}_3 = \mathbf{AG}_3 = \mathbf{Z}_3\mathbf{R}_3$, where $\mathbf{Z}_3 \in \mathbb{R}^{n_A \times m}$ is orthogonal and $\mathbf{R}_3 \in \mathbb{R}^{m \times m}$ is upper triangu-

lar. Then a projection of $\mathbf{A}^2$ onto the column space of $\mathbf{G}_3$ is $\hat{\mathbf{K}} = \mathbf{Y}_3^T\mathbf{Y}_3 = \mathbf{R}_3^T\mathbf{Z}_3^T\mathbf{Z}_3\mathbf{R}_3 = \mathbf{R}_3^T\mathbf{R}_3$. Then $\mathbf{K}_3 = \mathbf{R}_3\mathbf{R}_3^T = \mathbf{R}_3\mathbf{R}_3^T\mathbf{R}_3\mathbf{R}_3^{-1} = \mathbf{R}_3\hat{\mathbf{K}}\mathbf{R}_3^{-1}$, which is similar to $\hat{\mathbf{K}}$ and hence has the same eigenvalues. This leads to another implementation of REVD presented in Algorithm 11. This is a single pass algorithm, meaning that $\mathbf{A}$ has to be accessed just once, and to the best of our knowledge this method has not been considered in the context of randomised eigenvalue approximations.

Algorithm 11 Randomised eigenvalue decomposition based on *ritzit*, REVD_*ritzit*

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, target rank k, an oversampling parameter l

Output: orthogonal $\mathbf{U}_3 \in \mathbb{R}^{n_A \times k}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta}_3 \in \mathbb{R}^{k \times k}$ with approximations to the largest eigenvalues of $\mathbf{A}$ on the diagonal

1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n_A \times (k+l)}$
2: Orthonormalize the columns of $\mathbf{G}$ to obtain orthonormal $\mathbf{G}_3$
3: Form a sample matrix $\mathbf{Y}_3 = \mathbf{A}\mathbf{G}_3 \in \mathbb{R}^{n_A \times (k+l)}$
4: Compute QR decomposition $\mathbf{Y}_3 = \mathbf{Z}_3\mathbf{R}_3$ to obtain orthogonal $\mathbf{Z}_3 \in \mathbb{R}^{n_A \times (k+l)}$ and upper triangular $\mathbf{R}_3 \in \mathbb{R}^{(k+l)\times(k+l)}$
5: Form $\mathbf{K}_3 = \mathbf{R}_3\mathbf{R}_3^T \in \mathbb{R}^{(k+l)\times(k+l)}$
6: Form EVD of $\mathbf{K}_3$: $\mathbf{K}_3 = \mathbf{W}_3\mathbf{\Theta}_3^2\mathbf{W}_3^T$, where $\mathbf{W}_3$, $\mathbf{\Theta}_3^2 \in \mathbb{R}^{(k+l)\times(k+l)}$, elements of $\mathbf{\Theta}_3$ are sorted in decreasing order
7: Remove last l columns and rows of $\mathbf{\Theta}_3^2$, so that $\mathbf{\Theta}_3^2 \in \mathbb{R}^{k\times k}$
8: Remove last l columns of $\mathbf{W}_3$, so that $\mathbf{W}_3 \in \mathbb{R}^{(k+l)\times k}$
9: Form $\mathbf{U}_3 = \mathbf{Z}_3\mathbf{W}_3 \in \mathbb{R}^{n_A \times k}$.

Note that the Ritz vectors given by Algorithms 9, 10 and 11 are different. Although Algorithm 11 accesses the matrix $\mathbf{A}$ only once, it requires an additional orthogonalisation of a matrix of size $n_A \times (k + l)$.

In Table 4.1, we summarise some properties of the Lanczos, REVD, Nyström and REVD_*ritzit* methods when they are used to compute Ritz values and vectors to generate a preconditioner for $\mathbf{A}$ in incremental data assimilation. Note that the cost of applying the spectral-LMP depends on the number of vectors k used in its construction and is independent of which method is used to obtain them. The additional cost of using randomised algorithms arises only once per inner loop when the preconditioner is generated. We recall that in these algorithms the required EVD or SVD of the small matrix can be obtained cheaply and the most expensive parts of the algorithms are the matrix-matrix products of $\mathbf{A}$ and $n_A \times (k+l)$ matrices. If enough computational resources are available, these can be parallelised. In the best case scenario, all $k + l$ matrix-vector products can be performed at the same time, making the cost of the matrix-matrix product equivalent to the cost of one iteration of CG plus communication between the processors.

When a randomised method is used to generate the preconditioner, an inner loop is performed as follows. Estimates of the Ritz values of the Hessian and the corresponding Ritz vectors are obtained with a randomised method (Algorithm 9, 10 or 11) and used

	Lanczos	REVD	Nyström	REVD_*ritzit*
Information source	Previous inner loop	Current inner loop	Current inner loop	Current inner loop
Preconditioner for the first inner loop	No	Yes	Yes	Yes
k dependence on the previous inner loop	Bounded by the number of CG iterations	Independent	Independent	Independent
Matrix-matrix products with $\mathbf{A}$	None	2 products with $n_A \times (k+l)$ matrices	2 products with $n_A \times (k+l)$ matrices	1 product with $n_A \times (k+l)$ matrix
QR decomposition	None	None	None	$\mathbf{Y}_3 \in \mathbb{R}^{n_A \times (k+l)}$
Orthogonalisation	None	$\mathbf{Y} \in \mathbb{R}^{n_A \times (k+l)}$	$\mathbf{Y} \in \mathbb{R}^{n_A \times (k+l)}$	$\mathbf{G} \in \mathbb{R}^{n_A \times (k+l)}$
Cholesky factorization	None	None	$\mathbf{E}^{(2)} \in \mathbb{R}^{(k+l)\times(k+l)}$	None
Triangular solve	None	None	$\mathbf{FC} = \mathbf{E}^{(1)}$ for $\mathbf{F} \in \mathbb{R}^{n_A \times (k+l)}$	None
Deterministic EVD	$\mathbf{T}_k \in \mathbb{R}^{k\times k}$ *	$\mathbf{K} \in \mathbb{R}^{(k+l)\times(k+l)}$	None	$\mathbf{K}_3 \in \mathbb{R}^{(k+l)\times(k+l)}$
Deterministic SVD	None	None	$\mathbf{F} \in \mathbb{R}^{n_A \times (k+l)}$	None

Table 4.1: A summary of the properties of the different methods of obtaining k Ritz vectors and values to generate the preconditioner for a $n_A \times n_A$ matrix $\mathbf{A}$ in the ith inner loop. Here l is the oversampling parameter. * applies for CG with reorthogonalization.

to construct an LMP. Then the system (4.12) with the exact Hessian $\mathbf{A}$ is solved with PCG (Algorithm 7) using the LMP. The state is updated in the outer loop using the PCG solution.

4.6 Numerical experiments

We demonstrate our proposed preconditioning strategies using two models: a simple linear advection model to explore the spectra of the preconditioned Hessian and the nonlinear Lorenz 96 model [Lorenz, 1996] to explore the convergence of split preconditioned CG (PCG) . We perform identical twin experiments, where $\mathbf{x}^t = ((\mathbf{x}_0^t)^T, \ldots, (\mathbf{x}_N^t)^T)^T$ denotes the reference trajectory. The observations and background state are generated by adding noise with covariance matrices $\mathbf{R}$ and $\mathbf{B}$ to $\mathcal{H}_i(\mathbf{x}_i^t)$ and $\mathbf{x}_0$, respectively. We use direct observations, thus the observation operator $\mathcal{H}_i$ is linear.

We use covariance matrices $\mathbf{R}_i = \sigma_o^2 \mathbf{I}_{q_i}$, where q_i is the number of observations at time t_i, $\mathbf{Q}_i = \sigma_q^2 \mathbf{C}_q$, where $\mathbf{C}_q$ is a Laplacian correlation matrix [Johnson et al., 2005], and $\mathbf{B} = \sigma_b^2 \mathbf{C}_b$, where $\mathbf{C}_b$ is a second-order auto-regressive correlation matrix [Daley, 1993].

We assume that first level preconditioning has already been applied (recall (4.10)). In data assimilation, using Ritz-LMP as formulated in (4.18) is impractical because of the matrix products with $\mathbf{A}$ and we cannot use a simple formulation of Ritz-LMP when the Ritz values and vectors are obtained with the randomised methods. Hence, we use the spectral-LMP. However, as we mentioned in Section 4.4.2, the spectral-LMP that is constructed with well converged Ritz values and vectors acts like Ritz-LMP. When we

consider the second inner loop, we compare the spectral-LMPs with information from the randomised methods with the spectral-LMP constructed with information obtained with the Matlab function *eigs* in the previous inner loop. *eigs* returns a highly accurate estimate of a few largest or smallest eigenvalues and corresponding eigenvectors. We will use the term randomised LMP to refer to the spectral-LMPs that are constructed with information from the randomised methods, and deterministic LMP to refer to the spectral-LMP that is constructed with information from *eigs*.

The computations are performed with Matlab R2017b. Linear systems are solved using the Matlab implementation of PCG (function *pcg*), which was modified to allow split preconditioning to maintain the symmetric coefficient matrix at every loop.

4.6.1 Advection model

First, we consider the linear advection model:

$$\frac{\partial u(z,t)}{\partial t} + \frac{\partial u(z,t)}{\partial z} = 0, \tag{4.22}$$

where $z \in [0,1]$ and $t \in (0,T)$. An upwind numerical scheme is used to discretise (4.22) (see, e.g. Chapter 4 of [Morton and Mayers, 1994]). To allow us to compute all the eigenvalues (described in the following section), we consider a small system with the linear advection model. The domain is divided into $n = 40$ equally spaced grid points, with grid spacing $\Delta z = 1/n$. We run the model for 51 time steps ($N = 50$) with the time step size $\Delta t = 1/N$, hence $\mathbf{A}$ is a 2040×2040 matrix. The Courant number is $C = 0.8$ (the upwind scheme is stable with $C \in [0,1]$). The initial conditions are Gaussian,

$$u(z,0) = 6 \exp\left(-\frac{(z-0.5)^2}{2 \times 0.1^2}\right),$$

and the boundary conditions are periodic $u(1,t) = u(0,t)$.

We set $\sigma_o = 0.05$, $\sigma_q = 0.05$ and $\sigma_b = 0.1$. $\mathbf{C}_q$ and $\mathbf{C}_b$ have length scales equal to $10\Delta z$. Every fourth model variable is observed at every fifth time step, ensuring that there is an observation at the final time step (100 observations in total). Because the model and the observational operator $\mathcal{H}_i$ are linear, the cost function (4.2) is quadratic and its minimiser is found in the first loop of the incremental method.

Eigenvalues of the preconditioned matrix

We apply the randomised LMPs in the first inner loop. Note that if the deterministic LMP is used, it is unclear how to precondition the first inner loop. We explore what effect the randomised LMPs have on the eigenvalues of $\mathbf{A}$. The oversampling parameter l is set to 5 and the randomised LMPs are constructed with $k = 25$ vectors.

The Ritz values of $\mathbf{A}$ given by the randomised methods are compared to those computed using *eigs* (Figure 4.1a). The Nyström method produces a good approximation of the largest eigenvalues, while REVD gives a slightly worse approximation, except for the largest five eigenvalues. The REVD_*ritzit* method underestimates the largest eigenvalues significantly. The largest eigenvalues of the preconditioned matrices are smaller

than the largest eigenvalue of $\mathbf{A}$ (Figure 4.1b). However, the smallest eigenvalues of the preconditioned matrices are less than one and hence applying the preconditioner expands the spectrum of $\mathbf{A}$ at the lower boundary (Figure 4.1c), so that Theorem 3.4 of [Gratton et al., 2011], which considers the non-expansiveness of the spectrum of the Hessian after preconditioning with an LMP, does not hold. This happens because the formulation of the spectral-LMP is derived assuming that the eigenvalues and eigenvectors are exact, while the randomized methods provide only approximations. Note that even though REVD_*ritzit* gives the worst approximations of the largest eigenvalues of the Hessian, using the randomised LMP with information from REVD_*ritzit* reduces the largest eigenvalues of the preconditioned matrix the most and the smallest eigenvalues are close to one. Using the randomised LMP with estimates from Nyström gives similar results. Hence, the condition number of the preconditioned matrix is lower when the preconditioners are constructed with REVD_*ritzit* or Nyström compared to REVD.

The values of the quadratic cost function at the first ten iterations of PCG are shown in Figure 4.1d. Using the randomised LMP that is constructed with information from REVD is detrimental to the PCG convergence compared with using no preconditioning. Using information from the Nyström and REVD_*ritzit* methods results in similar PCG convergence and low values of the quadratic cost function are reached in fewer iterations than without preconditioning. The PCG convergence may be explained by the favourable distribution of the eigenvalues after preconditioning using Nyström and REVD_*ritzit*, and the smaller than one eigenvalues when using REVD. These results, however, do not necessarily generalize to an operational setting as this system is well conditioned while operational settings are not. This will be investigated further in the next section.

4.6.2 Lorenz 96 model

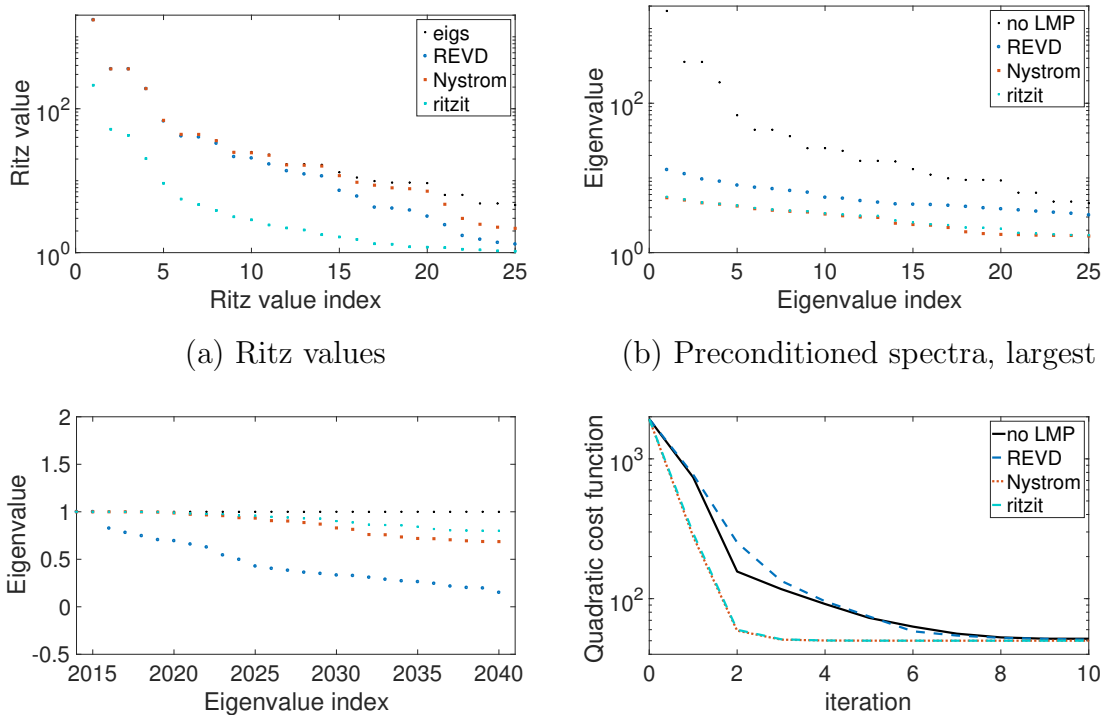
We next use the Lorenz 96 model to examine what effect the randomised LMPs have on PCG performance. In the Lorenz 96 model, the evolution of the n components $X^j,\ j \in \{1, 2, \ldots, n\}$ of $\mathbf{x}_i$ is governed by a set of n coupled ODEs:

$$\frac{dX^j}{dt} = -X^{j-2}X^{j-1} + X^{j-1}X^{j+1} - X^j + F, \tag{4.23}$$

where $X^{-1} = X^{n-1}, X^0 = X^n$ and $X^{n+1} = X^1$, and $F = 8$. The equations are integrated using a fourth order Runge-Kutta scheme [Butcher, 1987]. We set $n = 80$ and $N = 150$ (the size of $\mathbf{A}$ is 12080×12080) and observe every tenth model variable at every tenth time step (120 observations in total), ensuring that there are observations at the final time step. The grid point distance is $\Delta X = 1/n$ and the time step is set to $\Delta t = 2.5 \times 10^{-2}$.

For the covariance matrices we use $\sigma_o = 0.15$ and $\sigma_b = 0.2$. $\mathbf{C}_b$ has length scale equal to $2\Delta X$. Two setups are used for the model error covariance matrix:

- $\sigma_q = 0.1$ and $\mathbf{C}_q$ has length scale $L_q = 2\Delta X$ (the same as $\mathbf{C}_b$);
- $\sigma_q = 0.05$ and $\mathbf{C}_q$ has length scale $L_q = 0.25\Delta X$.



(a) Ritz values

(b) Preconditioned spectra, largest

(c) Preconditioned spectra, smallest

(d) Quadratic cost function

Figure 4.1: Advection problem. (a) The 25 largest eigenvalues of $\mathbf{A}$ (*eigs*) and their estimates given by randomised methods; the largest eigenvalues and their estimates given by REVD and Nyström coincide. (b) The largest eigenvalues of $\mathbf{A}$ (no LMP, the same as *eigs* in (a)) and $(\mathbf{C}_{25}^{sp})^T\mathbf{A}\mathbf{C}_{25}^{sp}$, where $\mathbf{C}_{25}^{sp}$ is constructed with Ritz values in (a) and corresponding Ritz vectors. (c) The smallest eigenvalues of $\mathbf{A}$ and $(\mathbf{C}_{25}^{sp})^T\mathbf{A}\mathbf{C}_{25}^{sp}$. (d) Quadratic cost function value versus PCG iteration when solving systems with $\mathbf{A}$ and $(\mathbf{C}_{25}^{sp})^T\mathbf{A}\mathbf{C}_{25}^{sp}$.

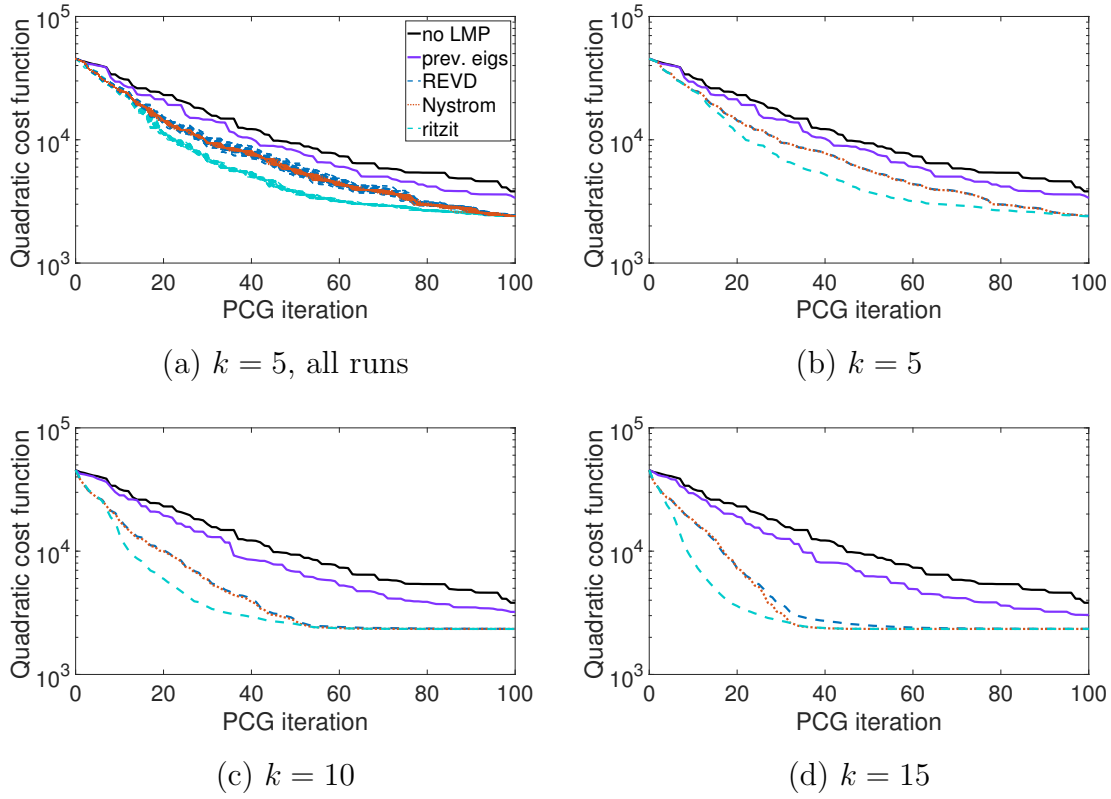
In our numerical experiments, the preconditioners have very similar effect using both setups. Hence, we present results for the case $\sigma_q = 0.1$ and $L_q = 2\Delta X$ in the following sections, except Figure 4.3.

The first outer loop is performed and no second level preconditioning is used in the first inner loop, where PCG is run for 100 iterations or until the relative residual norm reaches 10^{-6}. In the following sections, we use randomised and deterministic LMPs in the second inner loop. PCG has the same stopping criteria as in the first inner loop.

Minimising the inner loop cost function

In Figure 4.2, we compare the performance of the randomised LMPs with the deterministic LMP. We also consider the effect of varying k, the number of vectors used to construct the preconditioner. We set the oversampling parameter to $l = 5$. Because results from the randomized methods depend on the random matrix used, we perform 50 experiments with different realizations for the random matrix. We find that the different realizations lead to very similar results (see Figure 4.2a).

Independently of the k value, there is an advantage in using the second level preconditioning. The reduction in the value of the quadratic cost function is faster using randomised LMPs compared with deterministic LMPs, with REVD_*ritzit* performing the best after the first few iterations. The more information we use in the preconditioner (i.e. the higher k value), the faster REVD_*ritzit* shows better results than the other methods. The performances of the REVD and Nyström methods are similar. Note that as k increases, the storage (see Table 4.1) and work per PCG iteration increase. Examination of the Ritz values given by the randomised methods shows that REVD_*ritzit* gives the worse estimate of the largest eigenvalues, as was the case when using the advection model. We calculated the smallest eigenvalue of the preconditioned matrix $(\mathbf{C}_5^{sp})^T\mathbf{A}\mathbf{C}_5^{sp}$ using *eigs*. When $\mathbf{C}_5^{sp}$ is constructed using REVD_*ritzit* or Nyström, the smallest eigenvalue of $(\mathbf{C}_5^{sp})^T\mathbf{A}\mathbf{C}_5^{sp}$ is equal to one, whereas using REVD it is approximately 0.94. This may explain why the preconditioner constructed using REVD may not perform as well as other randomised preconditioners, but it is not entirely clear why the preconditioner that uses REVD_*ritzit* shows the best performance.



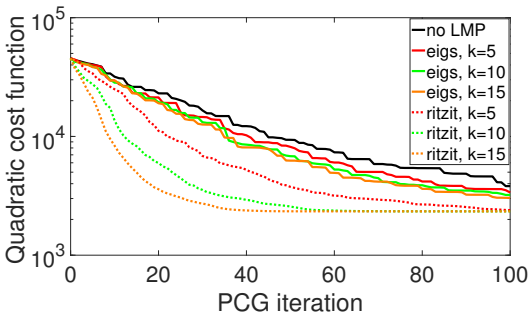
(a) $k = 5$, all runs (b) $k = 5$

(c) $k = 10$ (d) $k = 15$

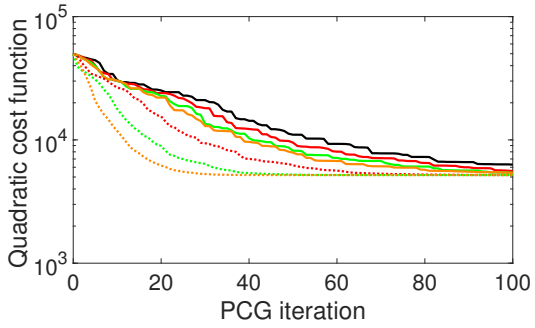
Figure 4.2: A comparison of the value of the quadratic cost function at every PCG iteration when spectral-LMP is constructed with $k \in \{5, 10, 15\}$ Ritz values and vectors obtained with the randomised methods in the current inner loop, and function *eigs* in the previous inner loop. We also show no second level preconditioning (no LMP), which is the same in all four panels. For the randomised methods, (a) shows 50 experiments for $k = 5$ and the rest display means over 50 experiments. Here $\sigma_q = 0.1$ and $L_q = 2\Delta X$.

The PCG convergence when using the deterministic LMP and the randomised LMP

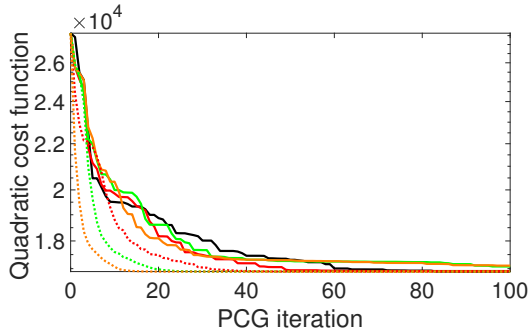
with information from REVD_*ritzit* with different k values is compared in Figure 4.3 for both setups of the model error covariance matrix. We also show an additional case where the model error covariance matrix is constructed setting $\sigma_q = \sigma_b/100 = 0.002$ and $L_q = 0.25\Delta X$. In this case, the performances of the REVD and Nyström methods are very similar, outperforming no preconditioning after the first 10-15 iterations, with better performance for higher k values (results not shown). Moreover, REVD_*ritzit* again outperforms the deterministic LMP from the first PCG iterations. For the deterministic LMP in Figure 4.3, varying k has little effect, especially in the initial iterations. However, for REVD_*ritzit* in general, increasing k results in a greater decrease of the cost function. Setting $k = 5$ gives better initial results compared with $k = 10$ in the $\sigma_q = 0.002$ case, but the larger k value performs better after that. Also, at any iteration of PCG we obtain a lower value of the quadratic cost function using the randomised LMP with $k = 5$ compared to the deterministic LMP with $k = 15$, which uses exact eigenpair information from the Hessian of the previous loop.



(a) $\sigma_q = 0.1$ and $L_q = 2\Delta X$



(b) $\sigma_q = 0.05$ and $L_q = 0.25\Delta X$



(c) $\sigma_q = 0.002$ and $L_q = 0.25\Delta X$

Figure 4.3: A comparison of the values of the quadratic cost function at every PCG iteration when using deterministic LMP with information from the previous loop (eigs) and the randomised LMP with information from REVD_*ritzit* for different k values (5, 10 and 15). No second level preconditioning is also shown (case (a) is the same as in Figure 4.2). In cases (a), (b) and (c) the model error covariance matrices are constructed using parameters σ_q and L_q.

Effect of the observation network

To understand the sensitivities of the results from the different LMPs to the observation network, we consider a system with the same parameters as in the previous section, where we had 120 observations, but we now observe

- every fifth model variable at every fifth time step (480 observations in total);
- every second variable at every second time step (3000 observations in total).

The oversampling parameter is again set to $l = 5$ and we set $k = 5$ and $k = 15$ for both observation networks. Since the number of observations is equal to the number of eigenvalues that are larger than one and there are more observations than in the previous section, there are more eigenvalues that are larger than one after the first level preconditioning. Because all 50 experiments with different Gaussian matrices in the previous section were close to the mean, we perform 10 experiments for each randomised method, solve the systems, and report the means of the quadratic cost function.

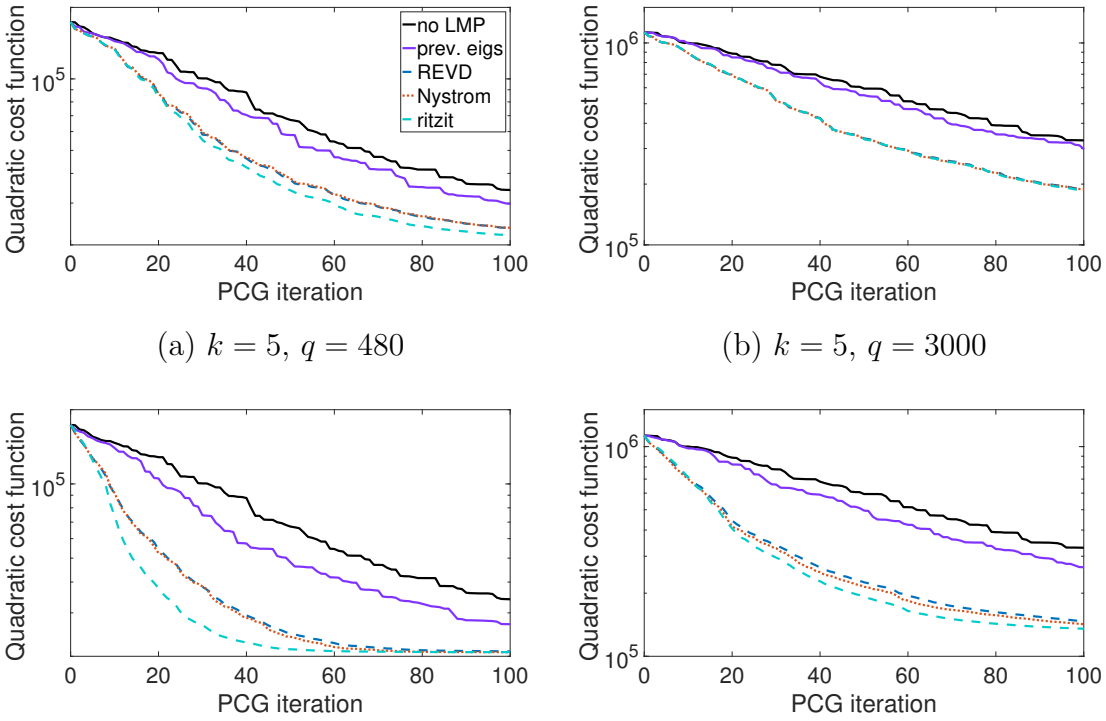
The results are presented in Figure 4.4. Again, the randomised LMPs perform better than the deterministic LMP. However, if the preconditioner is constructed with a small amount of information about the system ($k = 5$ for both systems and $k = 15$ for the system with 3000 observations), then there is little difference in the performance of different randomised LMPs. Also, when the number of observations is increased, more iterations of PCG are needed to get any improvement in the minimisation of the quadratic cost function when using the deterministic LMP over using no second level preconditioning.

When comparing the randomised and deterministic LMPs with different values of k for these systems, we obtain similar results to those in Figure 4.3a, i.e. it is more advantageous to use the randomised LMP constructed with $k = 5$ than using the deterministic LMP constructed with $k = 15$.

Effect of oversampling

We next consider the effect of increasing the value of the oversampling parameter l. The observation network is as in Section 4.6.2: Minimising the inner loop cost function (120 observations in total). We set $k = 15$ and perform the second inner loop 50 times for every value of $l \in \{5, 10, 15\}$ with all three randomised methods. The standard deviation of the value of the quadratic cost function at every iteration is presented in Figure 4.5.

For all the methods, the standard deviation is greatest in the first iterations of PCG. It is reduced when the value of l is increased and the largest reduction happens in the first iterations. However, REVD_*ritzit* is the least sensitive to the increase of the oversampling. With all values of l, REVD_*ritzit* has the largest standard deviation in the first few iterations, but it stills gives the largest reduction of the quadratic cost function. Hence, large oversampling is not necessary if REVD_*ritzit* is used.



(a) $k = 5$, $q = 480$

(b) $k = 5$, $q = 3000$

(c) $k = 15$, $q = 480$

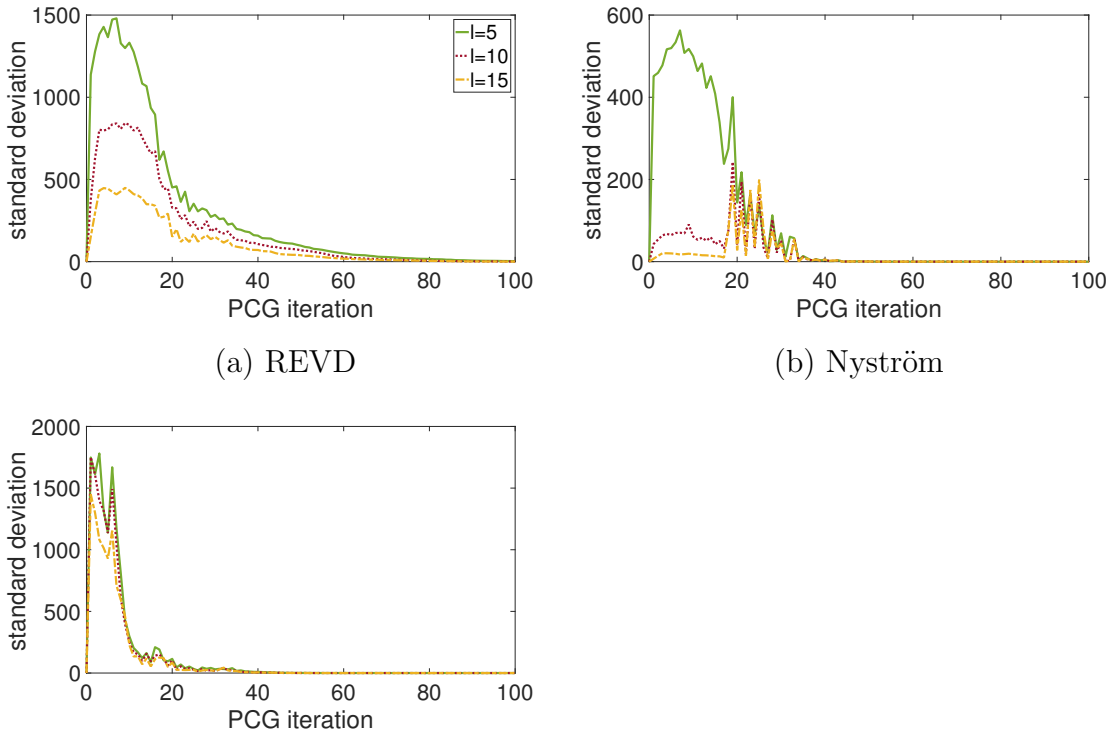
(d) $k = 15$, $q = 3000$

Figure 4.4: As in Figure 4.2, but for two systems with q observations; 10 experiments are done for each randomised method and the mean values plotted.

4.7 Conclusions and future work

We have proposed a new randomised approach to second level preconditioning of the incremental weak constraint 4D-Var forcing formulation. It can be preconditioned with an LMP that is constructed using approximations of eigenpairs of the Hessian. Previously, by using the Lanzcos and CG connection these approximations were obtained at a very low cost in one inner loop and then used to construct the LMP in the following inner loop. We have considered three methods (REVD, Nyström and REVD_*ritzit*) that employ randomisation to compute the approximations. These methods can be used to cheaply construct the preconditioner in the current inner loop, with no dependence on the previous inner loop, and are parallelisable.

Numerical experiments with the linear advection and Lorenz 96 models have shown that the randomised LMPs constructed with approximate eigenpairs improve the convergence of PCG more than deterministic LMPs with information from the previous loop, especially after the initial PCG iterations. The quadratic cost function reduces more rapidly when using a randomised LMP rather than a deterministic LMP, even if the randomised LMP is constructed with fewer vectors than the deterministic LMP. Also, for the randomised LMPs, the more information about the system we use (i.e. more approximations of eigenpairs are used to construct the preconditioner), the greater the reduction in the quadratic cost function, with a possible exception in the first PCG iterations for low k values and very small model error. Using more information to construct a deterministic



(a) REVD

(b) Nyström

(c) REVD_*ritzit*

Figure 4.5: Standard deviation of the quadratic cost function at every iteration of PCG when spectral-LMP is constructed with different randomised methods. For every randomised method we perform 50 experiments. Here $\sigma_q = 0.1$ and $L_q = 2\Delta X$.

LMP may not result in larger reduction of the quadratic cost function, especially in the first iterations of PCG, which is in line with results in [Tshimanga et al., 2008]. However, if not enough information is included in the randomised LMP, then preconditioning may have no effect on the first few iterations of PCG.

Of the randomised methods considered, the best overall performance was for the REVD_*ritzit*. However, if we run a small number of PCG iterations, the preconditioners obtained with different randomised methods give similar results. In the case of a very small model error, using REVD and Nyström is useful after the initial iterations of PCG whereas REVD_*ritzit* improves the reduction of quadratic cost function from the start. The performance was independent of the choice of the random Gaussian start matrix and it may be improved with oversampling.

In this work we apply randomised methods to generate a preconditioner, which is then used to accelerate the solution of the exact inner loop problem (4.12) with the PCG method (as discussed in Section 4.5). A different approach has been explored by [Bousserez and Henze, 2018] and [Bousserez et al., 2020], who presented and tested a randomised solution algorithm called the Randomized Incremental Optimal Technique (RIOT) in data assimilation. RIOT is designed to be used instead of PCG and employs a randomised eigenvalue decomposition of the Hessian (using a different method than the ones presented in this paper) to directly construct the solution $\mathbf{x}$ in (4.12), which approximates the

solution given by PCG.

The randomised preconditioning approach can also be employed to minimise other quadratic cost functions, including the strong constraint 4D-Var formulation. Further exploration of other single-pass versions of the randomised methods for the eigenvalue decomposition, that are discussed in [Halko et al., 2011], may be useful. In particular, the single-pass version of the Nyström method is potentially attractive. If a large number of Ritz vectors are used to construct the preconditioner, more attention can be paid to choosing the value of the oversampling parameter l in the randomised methods. In some cases a better approximation may be obtained if l linearly depends on the target rank of the approximation [Nakatsukasa, 2020].

Acknowledgements

We are grateful to Dr. Adam El-Said for his code for the weak constraint 4D-Var assimilation system. We would like to thank two anonymous reviewers, whose comments helped us to improve the manuscript.

4.8 Summary

We explored preconditioning the linear systems in the forcing formulation independently of previous inner loops by using randomised eigenvalue decompositions. We showed that in this way LMPs can be constructed cheaply if computational resources for parallelising matrix-matrix products are available. All three randomised methods considered outperformed no preconditioning and LMPs constructed using eigenpairs from the previous inner loop. Increasing the number of eigenpairs used to construct the LMPs gave better results, although the effect may appear only after a few PCG iterations. The REVD_*ritzit* implementation of the randomised method showed the best results of the three randomised methods. There may be no or a very small difference between the results with different methods if the number of eigenpairs in LMP is very small compared to the size of the system. We saw that large oversampling is not required.

Because REVD_*ritzit* implementation showed best results, we use this randomised method to construct LMPs in the block diagonal Schur complement preconditioners for the saddle point systems in Chapter 7. The results on oversampling motivate us to use the recommended value of $l = 5$ in Chapters 6 and 7.

The randomised LMPs remained effective when the number of observations was increased, whereas LMPs constructed using eigenpairs from the previous inner loop were less effective in the first PCG iterations. We explore how the extreme eigenvalues of the coefficient matrix in the forcing formulation change when new observations are added in Chapter 7. In the following chapter, we study the effect of adding observations on the extreme eigenvalues of the systems in the state formulation.

Chapter 5

Spectral estimates for coefficient matrices in state formulation

In this chapter, we look at the research question 2 for the systems in state formulation. We wish to know how the extreme eigenvalues of the coefficient matrices change when new observations are introduced. This can be used to assess the worst case convergence of CG and MINRES. We also provide bounds for the eigenvalues to estimate how changes in eigenvalues of the error covariance matrices and singular values of the blocks including the linearised model and linearised observation operator affect the eigenvalues of the coefficient matrices. A numerical example with the Lorenz 96 model is used to understand how tight the bounds are and how the convergence of CG and MINRES is affected by the new observations.

This chapter, except the summary in Section 5.8, is based on the paper: Daužickaitė, I., Lawless, A.S., Scott, J.A. and van Leeuwen, P.J. (2020) Spectral estimates for saddle point matrices arising in weak constraint four-dimensional variational data assimilation. *Numerical Linear Algebra with Applications*, 27(5): e2313.

5.1 Abstract

We consider the large sparse symmetric linear systems of equations that arise in the solution of weak constraint four-dimensional variational data assimilation, a method of high interest for numerical weather prediction. These systems can be written as saddle point systems with a 3×3 block structure but block eliminations can be performed to reduce them to saddle point systems with a 2×2 block structure, or further to symmetric positive definite systems. In this paper, we analyse how sensitive the spectra of these matrices are to the number of observations of the underlying dynamical system. We also obtain bounds on the eigenvalues of the matrices. Numerical experiments are used to confirm the theoretical analysis and bounds.

5.2 Introduction

Data assimilation estimates the state of a dynamical system by combining observations of the system with a prior estimate. The latter is called a background state and it is usually an output of a numerical model that simulates the dynamics of the system. The impact that the observations and the background state have on the state estimate depends on their errors whose statistical properties we assume are known. Data assimilation is used to produce initial conditions in numerical weather prediction (NWP) [Kalnay, 2002, Swinbank, 2010], as well as other areas, e.g. flood forecasting [Chen et al., 2013], research into atmospheric composition [Elbern et al., 1997], and neuroscience [Moye and Diekman, 2018]. In operational applications, the process is made more challenging by the size of the system, e.g. the numerical model may be operating on 10^8 state variables and $10^5 - 10^6$ observations may be incorporated [Nichols, 2010, Lawless, 2013]. Moreover, there is usually a constraint on the time that can be spent on calculations.

The solution, called the analysis, is obtained by combining the observations and the background state in an optimal way. One approach is to solve a weighted least-squares problem, which requires minimising a cost function. An active research topic in this area is the weak constraint four-dimensional variational (4D-Var) data assimilation method [Trémolet, 2006, Trémolet, 2007, El-Said, 2015, Bonavita et al., 2017, Fisher and Gürol, 2017, Gratton et al., 2018a, Freitag and Green, 2018]. It is employed in the search for states of the system over a time period, called the assimilation window. This method uses a cost function that is formulated under the assumption that the numerical model is not perfect and penalises the weighted discrepancy between the analysis and the observations, the analysis and the background state, and the difference between the analysis and the trajectory given by integrating the dynamical model.

Effective minimisation techniques need evaluations of the cost function and its gradient that involve expensive operations with the dynamical model and its linearised variant. Such approaches are impractical in operational applications. One way to approximate the minimum of the weak constraint 4D-Var is to use an inexact Gauss-Newton method [Gratton et al., 2007], in which a series of linearised quadratic cost functions with a low resolution model are minimised [Courtier et al., 1994], and the minima are used to update the high resolution state estimate. The state estimate update is found by solving sparse symmetric linear systems using an iterative method [Saad, 2003].

To increase the potential of using parallel computations when computing the state update with weak constraint 4D-Var, [Fisher and Gürol, 2017] introduced a symmetric 3×3 block saddle point formulation. The resulting large symmetric linear systems are solved using Krylov subspace solvers [Freitag and Green, 2018, Saad, 2003, Benzi et al., 2005]. One criteria that affects their convergence is the spectra of the coefficient matrices [Benzi et al., 2005]. We derive bounds for the eigenvalues of the 3×3 block matrix using the work of [Rusten and Winther, 1992]. Also, inspired by the practice in solving saddle point systems that arise from interior point methods [Greif et al., 2014, Morini et al., 2017], we reduce the 3×3 block system to a 2×2 block saddle point formulation and

derive eigenvalue bounds for this system. We also consider a 1×1 block formulation with a positive definite coefficient matrix, which corresponds to the standard method [Trémolet, 2006, Trémolet, 2007]. Some of the blocks in the 3×3 and 2×2 block saddle point coefficient matrices, and the 1×1 block positive definite coefficient matrix depend on the available observations of the dynamical system. We present a novel examination of how adding new observations influences the spectrum of these coefficient matrices.

In Section 5.3, we formulate the data assimilation problem and introduce weak constraint 4D-Var with the 3×3 block and 2×2 block saddle point formulations and the 1×1 block symmetric positive definite formulation. Eigenvalue bounds for the saddle point and positive definite matrices and results on how the extreme eigenvalues and the bounds depend on the number of observations are presented in Section 5.4. Section 5.5 illustrates the theoretical considerations using numerical examples, and concluding remarks and future directions are presented in Section 5.6.

5.3 Variational Data Assimilation

The state of the dynamical system of interest at times $t_0 < t_1 < ... < t_N$ is represented by the state vectors $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$ with $\mathbf{x}_i \in \mathbb{R}^n$. A nonlinear model $\mathcal{M}_i$ that is assumed to have errors describes the transition from the state at time t_i to the state at time t_{i+1}, i.e.

$$\mathbf{x}_{i+1} = \mathcal{M}_i(\mathbf{x}_i) + \boldsymbol{\eta}_{i+1}, \tag{5.1}$$

where $\boldsymbol{\eta}_i$ represents the model error at time t_i. It is further assumed that the model errors are Gaussian with zero mean and covariance matrix $\mathbf{Q}_i \in \mathbb{R}^{n \times n}$, and that they are uncorrelated in time, i.e. there is no relationship between the model errors at different times. In NWP, the model comes from the discretization of the partial differential equations that describe the flow and thermodynamics of a stratified multiphase fluid in interaction with radiation [Kalnay, 2002]. It also involves parameters that characterize processes arising at spatial scales that are smaller than the distance between the grid points [Rood, 2010]. Errors due to the discretization of the equations, errors in the boundary conditions, inaccurate parameters, and so on are components of the model error [Griffith and Nichols, 2000].

The background information about the state at time t_0 is denoted by $\mathbf{x}^b \in \mathbb{R}^n$. $\mathbf{x}^b$ usually comes from a previous short range forecast and is chosen to be the first guess of the state. It is assumed that the background term has errors that are Gaussian with zero mean and covariance matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$.

Observations of the dynamical system at time t_i are given by $\mathbf{y}_i \in \mathbb{R}^{q_i}$. In NWP, there are considerably fewer observations than state variables, i.e. $q_i << n$. Also, there may be indirect observations of the variables in the state vector and a comparison is obtained by mapping the state variables to the observation space using a nonlinear operator $\mathcal{H}_i$. For example, satellite observations used in NWP provide top of the atmosphere radiance data, whereas the model operates on different meteorological variables, e.g. temperature,

pressure, wind and so on [Andersson and Thépaut, 2010]. Hence, values of meteorological variables are transformed into top of the atmosphere radiances in order to compare the model output with the observations. In this case, the operator $\mathcal{H}_i$ is nonlinear and complicated. Approximations made when mapping the state variables to the observation space, different spatial and temporal scales between the model and some observations (observations may give information at a finer resolution than the model), and preprocessing, or quality control, of the observations (see, e.g., Section 5.8 of [Kalnay, 2002]) comprise the representativity error [Janjić et al., 2018]. The observation error is made up of the representativity error combined with the error arising due to the limited precision of the measurements. It is assumed to be Gaussian with zero mean and covariance matrix $\mathbf{R}_i \in \mathbb{R}^{q_i \times q_i}$. The observation errors are assumed to be uncorrelated in time [Lawless, 2013].

5.3.1 Weak constraint 4D-Var

In weak constraint 4D-Var, the analysis $\mathbf{x}_0^a, \mathbf{x}_1^a, \ldots, \mathbf{x}_N^a$ is obtained by minimising the following nonlinear cost function

$$J(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)) \tag{5.2}$$
$$+ \frac{1}{2}\sum_{i=0}^{N-1}(\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i))^T \mathbf{Q}_{i+1}^{-1}(\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i)).$$

This cost function is referred to as the state control variable formulation. Here, the control variables are defined as the variables with respect to which the cost function is minimised, i.e. $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$ in (5.2). Choosing different control variables and obtaining different formulations of the cost function is possible [Trémolet, 2006]. If the model is assumed to have no errors (i.e. $\mathbf{x}_{i+1} = \mathcal{M}_i(\mathbf{x}_i)$), the cost function simplifies as the last term in (5.2) is removed; this is called strong constraint 4D-Var. Rejecting this perfect model assumption and using weak constraint 4D-Var may lead to a better analysis [Trémolet, 2007].

Iterative gradient-based optimisation methods are used in practical data assimilation [Talagrand, 2010, Lawless, 2013]. These require the cost function and its gradient to be evaluated at every iteration. In operational applications, integrating the model over the assimilation window to evaluate the cost function is computationally expensive. The gradient is obtained by the adjoint method (see, e.g., Section 2 of [Lawless, 2013] and Section 3.2 of [Talagrand, 2010] for an introduction), which is a few times more computationally expensive than evaluating the cost function. This makes the minimisation of the nonlinear weak constraint 4D-Var cost function impractical. Hence, approximations have to be made. We introduce such an approach in the following section.

5.3.2 Incremental formulation

Minimisation of the 4D-Var cost function in an operational setting is made feasible by employing an iterative Gauss-Newton method, as first proposed by [Courtier et al., 1994] for the strong constraint 4D-Var. In this approach, the solution of the weak constraint 4D-Var is approximated by solving a sequence of linearised problems, i.e. the $(l+1)$th approximation of the state is

$$\mathbf{x}_i^{(l+1)} = \mathbf{x}_i^{(l)} + \delta\mathbf{x}_i^{(l)}, \quad i \in \{0, 1, \dots, N\}, \tag{5.3}$$

where $\delta\mathbf{x}_i^{(l)}$ is obtained as the minimiser of the linearised cost function

$$\begin{aligned} J^{\delta}(\delta\mathbf{x}_0^{(l)}, \delta\mathbf{x}_1^{(l)}, \dots, \delta\mathbf{x}_N^{(l)}) &= (\delta\mathbf{x}_0^{(l)} - \hat{\mathbf{b}}^{(l)})^T \mathbf{B}^{-1} (\delta\mathbf{x}_0^{(l)} - \hat{\mathbf{b}}^{(l)}) \\ &+ \frac{1}{2}\sum_{i=0}^{N} (\mathbf{H}_i^{(l)}\delta\mathbf{x}_i^{(l)} - \mathbf{d}_i^{(l)})^T \mathbf{R}_i^{-1} (\mathbf{H}_i^{(l)}\delta\mathbf{x}_i^{(l)} - \mathbf{d}_i^{(l)}) \\ &+ \frac{1}{2}\sum_{i=0}^{N-1} (\mathbf{M}_i^{(l)}\delta\mathbf{x}_i^{(l)} - \delta\mathbf{x}_{i+1}^{(l)} - \boldsymbol{\eta}_{i+1}^{(l)})^T \mathbf{Q}_{i+1}^{-1} (\mathbf{M}_i^{(l)}\delta\mathbf{x}_i^{(l)} - \delta\mathbf{x}_{i+1}^{(l)} - \boldsymbol{\eta}_{i+1}^{(l)}), \end{aligned} \tag{5.4}$$

where $\hat{\mathbf{b}}^{(l)} = \mathbf{x}^b - \mathbf{x}_0^{(l)}$, $\mathbf{d}_i^{(l)} = \mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i^{(l)})$, $\boldsymbol{\eta}_i^{(l)} = \mathbf{x}_i^{(l)} - \mathcal{M}_{i-1}(\mathbf{x}_{i-1}^{(l)})$ (as in (5.1)) and $\mathbf{M}_i^{(l)}$ and $\mathbf{H}_i^{(l)}$ are the model $\mathcal{M}_i$ and the observation operator $\mathcal{H}_i$, respectively, linearised at $\mathbf{x}_i^{(l)}$. Minimisation of (5.4) is called the inner loop. The lth outer loop consists of updating the approximation of the state (5.3), linearising the model $\mathcal{M}_i$ and the observation operator $\mathcal{H}_i$, and computing the values $\hat{\mathbf{b}}^{(l)}$, $\mathbf{d}_i^{(l)}$ and $\boldsymbol{\eta}_i^{(l)}$. This cost function is quadratic, which allows the use of effective minimisation techniques, such as conjugate gradient (see Chapter 5 of [Nocedal and Wright, 2006]). In NWP, the computational cost of minimising the 4D-Var cost function is reduced by using a version of the inner loop cost function that is defined for a model with lower spatial resolution, i.e., on a coarser grid [Fisher, 1998]. We do not consider such an approach in the subsequent work, because our results on the change of the spectra of the coefficient matrices and the bounds (that are introduced in the following section) hold for models with any spatial resolution.

For ease of notation, we introduce the following four-dimensional (in the sense that they contain information in space and time) vectors and matrices.

$$\mathbf{x}^{(l)} = \begin{pmatrix} \mathbf{x}_0^{(l)} \\ \mathbf{x}_1^{(l)} \\ \vdots \\ \mathbf{x}_N^{(l)} \end{pmatrix}, \ \delta\mathbf{x}^{(l)} = \begin{pmatrix} \delta\mathbf{x}_0^{(l)} \\ \delta\mathbf{x}_1^{(l)} \\ \vdots \\ \delta\mathbf{x}_N^{(l)} \end{pmatrix}, \mathbf{b}^{(l)} = \begin{pmatrix} \hat{\mathbf{b}}^{(l)} \\ -\boldsymbol{\eta}_1^{(l)} \\ \vdots \\ -\boldsymbol{\eta}_N^{(l)} \end{pmatrix}, \ \mathbf{d}^{(l)} = \begin{pmatrix} \mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0^{(l)}) \\ \mathbf{y}_1 - \mathcal{H}_1(\mathbf{x}_1^{(l)}) \\ \vdots \\ \mathbf{y}_N - \mathcal{H}_N(\mathbf{x}_N^{(l)}) \end{pmatrix},$$

where $\mathbf{x}^{(l)}, \delta\mathbf{x}^{(l)}, \mathbf{b}^{(l)} \in \mathbb{R}^{(N+1)n}$ and $\mathbf{d}^{(l)} \in \mathbb{R}^q, q = \Sigma_{i=0}^N q_i$. We also define the matrices

$$\mathbf{L}^{(l)} = \begin{pmatrix} \mathbf{I} & & & & \\ -\mathbf{M}_0^{(l)} & \mathbf{I} & & & \\ & -\mathbf{M}_1^{(l)} & \mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{M}_{N-1}^{(l)} & \mathbf{I} \end{pmatrix}, \qquad \mathbf{H}^{(l)} = \begin{pmatrix} \mathbf{H}_0^{(l)} & & & \\ & \mathbf{H}_1^{(l)} & & \\ & & \ddots & \\ & & & \mathbf{H}_N^{(l)} \end{pmatrix},$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, $\mathbf{L}^{(l)} \in \mathbb{R}^{(N+1)n\times(N+1)n}$ and $\mathbf{H}^{(l)} \in \mathbb{R}^{q\times(N+1)n}$. We define the block diagonal covariance matrices

$$\mathbf{D} = \begin{pmatrix} \mathbf{B} & & & \\ & \mathbf{Q}_1 & & \\ & & \ddots & \\ & & & \mathbf{Q}_N \end{pmatrix} \quad \text{and} \quad \mathbf{R} = \begin{pmatrix} \mathbf{R}_0 & & & \\ & \mathbf{R}_1 & & \\ & & \ddots & \\ & & & \mathbf{R}_N \end{pmatrix},$$

$\mathbf{D} \in \mathbb{R}^{(N+1)n\times(N+1)n}$ and $\mathbf{R} \in \mathbb{R}^{q\times q}$. The state update (5.3) may then be written as

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} + \delta\mathbf{x}^{(l)}, \tag{5.5}$$

and the quadratic cost function (5.4) becomes

$$J^{\delta}(\delta\mathbf{x}^{(l)}) = \frac{1}{2}||\mathbf{L}^{(l)}\delta\mathbf{x}^{(l)} - \mathbf{b}^{(l)}||^2_{\mathbf{D}^{-1}} + \frac{1}{2}||\mathbf{H}^{(l)}\delta\mathbf{x}^{(l)} - \mathbf{d}^{(l)}||^2_{\mathbf{R}^{-1}}, \tag{5.6}$$

where $||\mathbf{a}||^2_{\mathbf{A}^{-1}} = \mathbf{a}^T\mathbf{A}^{-1}\mathbf{a}$. We omit the superscript (l) for the outer iteration in the subsequent discussions. The minimum of (5.6) can be found by solving linear systems. We discuss different formulations of these in the next three subsections.

3×3 block saddle point formulation

In pursuance of exploiting parallel computations in data assimilation, [Fisher and Gürol, 2017] proposed obtaining the state increment $\delta\mathbf{x}$ by solving a saddle point system (see also [Freitag and Green, 2018]). New variables are introduced

$$\boldsymbol{\lambda} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{L}\delta\mathbf{x}) \ \in \mathbb{R}^{(N+1)n}, \tag{5.7}$$

$$\boldsymbol{\mu} = \mathbf{R}^{-1}(\mathbf{d} - \mathbf{H}\delta\mathbf{x}) \ \in \mathbb{R}^{q}. \tag{5.8}$$

The gradient of the cost function (5.6) with respect to $\delta\mathbf{x}$ provides the optimality constraint

$$\mathbf{0} = \mathbf{L}^T\mathbf{D}^{-1}(\mathbf{L}\delta\mathbf{x} - \mathbf{b}) + \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\delta\mathbf{x} - \mathbf{d}) \tag{5.9}$$

$$= -(\mathbf{L}^T\boldsymbol{\lambda} + \mathbf{H}^T\boldsymbol{\mu}). \tag{5.10}$$

Multiplying (5.7) by $\mathbf{D}$ and (5.8) by $\mathbf{R}$ together with (5.10), yields a coupled linear system of equations:

$$\mathcal{A}_3 \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}, \tag{5.11}$$

where the coefficient matrix is given by

$$\mathcal{A}_3 = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(2(N+1)n+q)\times(2(N+1)n+q)}. \tag{5.12}$$

$\mathcal{A}_3$ is a sparse symmetric indefinite saddle point matrix that has a 3×3 block form. Such systems are explored in the optimization literature [Greif et al., 2014, Morini et al.,

2016, Morini et al., 2017]. When solving these systems iteratively, it is usually assumed that calculations involving the blocks on the diagonal are computationally expensive, while the off-diagonal blocks are cheap to apply and easily approximated. However, in our application, operations with the diagonal blocks are relatively cheap and the off-diagonal blocks are computationally expensive, particularly because of the integrations of the model and its adjoint in $\mathbf{L}$ and $\mathbf{L}^T$.

Recall that the sizes of the blocks $\mathbf{R}$, $\mathbf{H}$ and $\mathbf{H}^T$ depend on the number of observations q. Thus, the size of $\mathcal{A}_3$ and possibly some of its characteristics are also affected by q. The saddle point systems that arise in different outer loops vary in the right hand sides and the linearisation states of $\mathbf{L}$ and $\mathbf{H}$.

Because of the memory requirements of sparse direct solvers, they cannot be used to solve the 3×3 block saddle point systems that arise in an operational setting. Iterative solvers (such as MINRES, SYMMLQ [Paige and Saunders, 1975], GMRES [Saad and Schultz, 1986]) need to be used. These Krylov subspace methods require matrix-vector products with $\mathcal{A}_3$ at each iteration. Note that the matrix-vector product $\mathcal{A}_3\mathbf{q}$, $\mathbf{q}^T = (\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T)$, $\mathbf{q}_1, \mathbf{q}_3 \in \mathbb{R}^{(N+1)n}, \mathbf{q}_2 \in \mathbb{R}^q$, involves multiplying $\mathbf{D}$ and $\mathbf{L}^T$ by $\mathbf{q}_1$, $\mathbf{R}$ and $\mathbf{H}^T$ by $\mathbf{q}_2$, and $\mathbf{L}$ and $\mathbf{H}$ by $\mathbf{q}_3$. These matrix-vector products may be performed in parallel. Furthermore, multiplication of each component of each block matrix with the respective part of the vector $\mathbf{q}_i$ can be performed in parallel. The possibility of multiplying a vector with the blocks in $\mathbf{L}$ and $\mathbf{L}^T$ in parallel is particularly attractive, because the expensive operations involving the linearised model $\mathbf{M}_i$ and its adjoint $\mathbf{M}_i^T$ can be done at the same time for every $i \in \{0, 1, \ldots, N-1\}$.

2×2 block saddle point formulation

The saddle point systems with 3×3 block coefficient matrices that arise from interior point methods are often reduced to 2×2 block systems [Greif et al., 2014, Morini et al., 2017]. The 2×2 block formulation has not been used in data assimilation before. Because of its smaller size, it may be advantageous. Therefore, we now explore using this approach in the weak constraint 4D-Var setting.

Multiplying equation (5.7) by $\mathbf{D}$ and eliminating $\boldsymbol{\mu}$ in (5.10) gives the following equivalent system of equations

$$\mathcal{A}_2 \begin{pmatrix} \boldsymbol{\lambda} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ -\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d} \end{pmatrix}, \tag{5.13}$$

where

$$\mathcal{A}_2 = \begin{pmatrix} \mathbf{D} & \mathbf{L} \\ \mathbf{L}^T & -\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \end{pmatrix} \in \mathbb{R}^{2(N+1)n \times 2(N+1)n}. \tag{5.14}$$

The reduced matrix $\mathcal{A}_2$ is a sparse symmetric indefinite saddle point matrix with a 2×2 block form. Unlike the 3×3 block matrix $\mathcal{A}_3$, its size is independent of the number of observations. $\mathcal{A}_2$ involves the matrix $\mathbf{R}^{-1}$, which is usually available in data assimilation applications. The computationally most expensive blocks $\mathbf{L}$ and $\mathbf{L}^T$ are again the off-diagonal blocks.

Solving (5.13) in parallel might be less appealing compared to solving (5.11) in parallel: for a Krylov subspace method, the $(2,2)$ block $-\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ need not be formed separately, that is, only operators to perform the matrix-vector products with $\mathbf{H}^T$, $\mathbf{R}^{-1}$ and $\mathbf{H}$ need to be stored. Hence, a matrix-vector product $\mathcal{A}_2\mathbf{q}$, $\mathbf{q}^T = (\mathbf{q}_1^T, \mathbf{q}_3^T)$, $\mathbf{q}_1, \mathbf{q}_3 \in \mathbb{R}^{(N+1)n}$, requires multiplying $\mathbf{D}$ and $\mathbf{L}^T$ by $\mathbf{q}_1$, $\mathbf{L}$ and $\mathbf{H}$ by $\mathbf{q}_3$ (which may be done in parallel) and subsequently $\mathbf{R}^{-1}$ by $\mathbf{H}\mathbf{q}_3$, followed by $-\mathbf{H}^T$ by $\mathbf{R}^{-1}\mathbf{H}\mathbf{q}_3$. Hence, the cost of matrix-vector products for the 3×3 and 2×2 block formulations differs in that the former needs matrix-vector products with $\mathbf{R}$ while the latter requires products with $\mathbf{R}^{-1}$, and the 2×2 block formulation requires some sequential calculations. However, notice that the expensive calculations that involve applying the operators $\mathbf{L}$ and $\mathbf{L}^T$ (the linearised model and its adjoint) can still be performed in parallel.

1×1 block formulation

The 2×2 block system can be further reduced to a 1×1 block system, that is, to the standard formulation (see, e.g., [Trémolet, 2006] and [El-Said, 2015] for a more detailed consideration):

$$(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\delta\mathbf{x} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{b} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}. \tag{5.15}$$

Observe that the coefficient matrix

$$\begin{aligned} \mathcal{A}_1 &= \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \\ &= \begin{pmatrix} \mathbf{L}^T & \mathbf{H}^T \end{pmatrix} \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{L} \\ \mathbf{H} \end{pmatrix} \end{aligned} \tag{5.16}$$

is the negative Schur complement of $\begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix}$ in $\mathcal{A}_3$. The matrix $\mathcal{A}_1$ is block tridiagonal and symmetric positive definite, hence the conjugate gradient method by [Hestenes and Stiefel, 1952] can be used. The computations with the linearised model in $\mathbf{L}$ at every time step can again be performed in parallel. However, the adjoint of the linearised model in $\mathbf{L}^T$ can only be applied after the computations with the model are finished, thus limiting the potential for parallelism.

5.4 Eigenvalues of the saddle point formulations

One factor that influences the rate of convergence of Krylov subspace iterative solvers for symmetric systems is the spectrum of the coefficient matrix (see, for example, Section 9 in the survey paper of [Benzi et al., 2005] and Lectures 35 and 38 in the textbook by [Trefethen and Bau, III, 1997] for a discussion). [Simoncini and Szyld, 2013] have shown that any eigenvalues of the saddle point systems that lie close to zero can cause the iterative solver MINRES to stagnate for a number of iterations while the rate of convergence can improve if the eigenvalues are clustered.

In the following, we examine how the eigenvalues of the block matrices $\mathcal{A}_3$, $\mathcal{A}_2$, and $\mathcal{A}_1$ change when new observations are added. This is done by considering the shift in

the extreme eigenvalues of these matrices, that is the smallest and largest positive and negative eigenvalues. We then present bounds for the eigenvalues of these matrices.

5.4.1 Preliminaries

In order to determine how changing the number of observations influences the spectra of $\mathcal{A}_3$, $\mathcal{A}_2$, and $\mathcal{A}_1$, we explore the extreme singular values and eigenvalues of some blocks in $\mathcal{A}_3$, $\mathcal{A}_2$ and $\mathcal{A}_1$. We state two theorems that we will require. Here, we employ the notation $\lambda_k(\mathbf{A})$ to denote the kth largest eigenvalue of a matrix $\mathbf{A}$ and subscripts *min* and *max* are used to denote the smallest and largest eigenvalues, respectively.

Theorem 5.1 (See Section 8.1.2 of [Golub and Van Loan, 2013])**.** *If* $\mathbf{A}$ *and* $\mathbf{C}$ *are* $n \times n$ *Hermitian matrices, then*

$$\lambda_k(\mathbf{A}) + \lambda_{min}(\mathbf{C}) \leq \lambda_k(\mathbf{A}+\mathbf{C}) \leq \lambda_k(\mathbf{A}) + \lambda_{max}(\mathbf{C}), \quad k \in \{1, 2, \ldots, n\}.$$

Theorem 5.2 (Cauchy's Interlace Theorem, see Theorem 4.2 in Chapter 4 of [Stewart and Sun, 1990])**.** *If* $\mathbf{A}$ *is an* $n \times n$ *Hermitian matrix and* $\mathbf{C}$ *is a* $(n-1) \times (n-1)$ *principal submatrix of* $\mathbf{A}$ *(a matrix obtained by eliminating a row and a corresponding column of* $\mathbf{A}$*), then*

$$\lambda_n(\mathbf{A}) \leq \lambda_{n-1}(\mathbf{C}) \leq \lambda_{n-1}(\mathbf{A}) \leq \cdots \leq \lambda_2(\mathbf{A}) \leq \lambda_1(\mathbf{C}) \leq \lambda_1(\mathbf{A}).$$

In the following lemmas we describe how the smallest and largest singular values of $(\mathbf{L}^T \ \mathbf{H}^T)$ (here $\mathbf{L}$ and $\mathbf{H}$ are as defined in Section 5.3.2) and the extreme eigenvalues of the observation error covariance matrix $\mathbf{R}$ change when new observations are introduced. The same is done for the largest eigenvalues of $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ assuming that $\mathbf{R}$ is diagonal. In these lemmas the subscript $k \in \{0, 1, \ldots, (N+1)n - 1\}$ denotes the number of available observations and the subscript $k+1$ indicates that a new observation is added to the system with k observations, i.e. matrices $\mathbf{R}_k \in \mathbb{R}^{k \times k}$ and $\mathbf{H}_k \in \mathbb{R}^{k \times (N+1)n}$ correspond to a system with k observations and $\mathbf{R}_{k+1}$ and $\mathbf{H}_{k+1}$ correspond to the system with an additional observation. We write $\mathbf{R}_{k+1} = \begin{pmatrix} \mathbf{R}_k & \mathbf{r} \\ \mathbf{r}^T & \alpha \end{pmatrix}$ and $\mathbf{H}_{k+1} = \begin{pmatrix} \mathbf{H}_k \\ \mathbf{h}_{k+1}^T \end{pmatrix}$, where $\mathbf{r} \in \mathbb{R}^k$, $\alpha \in \mathbb{R}^1$, $\alpha > 0$ and $\mathbf{h}_{k+1} \in \mathbb{R}^{(N+1)n}$ correspond to the new observation.

Lemma 5.3. *Let* ω_{min} *and* ω_{max} *be the smallest and largest singular values of* $(\mathbf{L}^T \ \mathbf{H}_k^T)$*, and* ϕ_{min} *and* ϕ_{max} *be the smallest and largest singular values of* $(\mathbf{L}^T \ \mathbf{H}_{k+1}^T)$*. Then*

$$\omega_{min}^2 \leq \phi_{min}^2 \quad \textit{and} \quad \omega_{max}^2 \leq \phi_{max}^2$$

i.e. the smallest and largest singular values of $(\mathbf{L}^T \ \mathbf{H}^T)$ *increase or are unchanged when new observations are added.*

Proof. We consider the eigenvalues of $\mathbf{L}^T\mathbf{L} + \mathbf{H}_k^T\mathbf{H}_k$ and $\mathbf{L}^T\mathbf{L} + \mathbf{H}_{k+1}^T\mathbf{H}_{k+1}$, which are the squares of the singular values of $(\mathbf{L}^T \ \mathbf{H}_k^T)$ and $(\mathbf{L}^T \ \mathbf{H}_{k+1}^T)$, respectively (see Section 2.4.2 of [Golub and Van Loan, 2013]). We can write

$$\mathbf{H}_{k+1}^T\mathbf{H}_{k+1} = \begin{pmatrix} \mathbf{H}_k^T & \mathbf{h}_{k+1} \end{pmatrix} \begin{pmatrix} \mathbf{H}_k \\ \mathbf{h}_{k+1}^T \end{pmatrix} = \mathbf{H}_k^T\mathbf{H}_k + \mathbf{h}_{k+1}\mathbf{h}_{k+1}^T.$$

Then by Theorem 5.1,

$$\omega_{min}^2 + \lambda_{min}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) \leq \phi_{min}^2, \quad k \in \{0, 1, \ldots, (N+1)n-1\},$$

and since $\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T$ is a rank 1 symmetric positive semi-definite matrix, $\lambda_{min}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) = 0$.

The proof for the largest singular values is analogous. □

Lemma 5.4. *Consider the observation error covariance matrices* $\mathbf{R}_k \in \mathbb{R}^{k\times k}$ *and* $\mathbf{R}_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$. *Then*

$$\lambda_{min}(\mathbf{R}_{k+1}) \leq \lambda_{min}(\mathbf{R}_k) \quad \textit{and} \quad \lambda_{max}(\mathbf{R}_k) \leq \lambda_{max}(\mathbf{R}_{k+1}), \quad k \in \{0, 1, \ldots, (N+1)n-1\},$$

i.e. the largest (respectively, smallest) eigenvalue of $\mathbf{R}$ *increases (respectively, decreases), or is unchanged when new observations are introduced.*

Proof. When adding an observation, a row and a corresponding column are appended to $\mathbf{R}_k$ while the other entries of $\mathbf{R}_k$ are unchanged. The result is immediate by applying Theorem 5.2. □

Lemma 5.5. *If the observation errors are uncorrelated, i.e.* $\mathbf{R}$ *is diagonal, then*

$$\lambda_{max}(\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k) \leq \lambda_{max}(\mathbf{H}_{k+1}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{k+1}), \quad k \in \{0, 1, \ldots, (N+1)n-1\},$$

i.e. for diagonal $\mathbf{R}$, *the largest eigenvalue of* $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ *increases or is unchanged when new observations are introduced.*

Proof. The proof is similar to that of Lemma 5.3. For diagonal $\mathbf{R}_{k+1}$:

$$\mathbf{R}_{k+1}^{-1} = \begin{pmatrix} \mathbf{R}_k^{-1} & \\ & \alpha^{-1} \end{pmatrix}, \ \alpha > 0.$$

Then

$$\mathbf{H}_{k+1}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{k+1} = \begin{pmatrix} \mathbf{H}_k^T & \mathbf{h}_{k+1} \end{pmatrix} \begin{pmatrix} \mathbf{R}_k^{-1} & \\ & \alpha^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{H}_k \\ \mathbf{h}_{k+1}^T \end{pmatrix} = \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k + \alpha^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T.$$

Hence, by Theorem 5.1,

$$\lambda_{max}(\mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k) + \alpha^{-1}\lambda_{min}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) \leq \lambda_{max}(\mathbf{H}_{k+1}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{k+1}),$$
$$k \in \{0, 1, \ldots, (N+1)n-1\}, \quad (5.17)$$

and since $\lambda_{min}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) = 0$ the result is proved. □

Notation

In the following, we use the notation given in Table 5.1 for the eigenvalues of $\mathcal{A}_3$, $\mathcal{A}_2$ and $\mathcal{A}_1$, and the eigenvalues and singular values of the blocks within them. We use subscripts *min* and *max* to denote the smallest and largest eigenvalues or singular values, respectively,

Matrix	$\mathcal{A}_3$	$\mathcal{A}_2$	$\mathcal{A}_1$	$\mathbf{D}$	$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$	$\mathbf{R}$
Eigenvalue	γ_i	ζ_i	χ_i	ψ_i	ν_i	ρ_i

Matrix	$(\mathbf{L}^T \quad \mathbf{H}^T)$	$\mathbf{L}$
Singular value	θ_i	σ_i

Table 5.1: Notation for the eigenvalues and singular values.

and θ_{min} denote the smallest non-zero singular value of $(\mathbf{L}^T \quad \mathbf{H}^T)$. In addition, $||\cdot||$ denotes the L_2 norm.

We also use

$$\tau_{min} = min\{\psi_{min}, \rho_{min}\}, \tag{5.18}$$

$$\tau_{max} = max\{\psi_{max}, \rho_{max}\}. \tag{5.19}$$

5.4.2 Bounds for the 3×3 block formulation

To determine the numbers of positive and negative eigenvalues of $\mathcal{A}_3$ given in (5.12), we write $\mathcal{A}_3$ as a congruence transformation

$$\begin{aligned}
\mathcal{A}_3 =& \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & \mathbf{0} \end{pmatrix} \\
=& \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \mathbf{L}^T & \mathbf{H}^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \\
=& \hat{\mathbf{L}}\hat{\mathbf{B}}\hat{\mathbf{L}}^T,
\end{aligned}$$

where $\mathbf{I} \in \mathbb{R}^{(N+1)n \times (N+1)n}$ is the identity matrix. Thus, by Sylvester's law of inertia (see Section 8.1.5 of [Golub and Van Loan, 2013]), $\mathcal{A}_3$ and $\hat{\mathbf{B}}$ have the same inertia, that is, the same number of positive, negative, and zero eigenvalues. Since the blocks $\mathbf{D}^{-1}$, $\mathbf{R}^{-1}$ and $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} = \mathcal{A}_1$ are symmetric positive definite matrices, $\mathcal{A}_3$ has $(N+1)n+q$ positive and $(N+1)n$ negative eigenvalues. In the following theorem, we explore how the extreme eigenvalues of $\mathcal{A}_3$ change when new observations are introduced.

Theorem 5.6. *The smallest and largest negative eigenvalues of $\mathcal{A}_3$ either move away from zero or are unchanged when new observations are introduced. The same holds for the largest positive eigenvalue, while the smallest positive eigenvalue approaches zero or is unchanged.*

Proof. Let $\mathcal{A}_{3,k}$ denote $\mathcal{A}_3$ where $q = k$. To account for an additional observation, a row and a corresponding column is added to $\mathcal{A}_3$, hence $\mathcal{A}_{3,k}$ is a principal submatrix of $\mathcal{A}_{3,k+1}$. Let

$$\lambda_{-(N+1)n}(\mathcal{A}_{3,k}) \leq \lambda_{-((N+1)n-1)}(\mathcal{A}_{3,k}) \leq \cdots \leq \lambda_{-1}(\mathcal{A}_{3,k}) < 0, \tag{5.20}$$

$$0 < \lambda_1(\mathcal{A}_{3,k}) \leq \cdots \leq \lambda_{(N+1)n+k}(\mathcal{A}_{3,k}) \tag{5.21}$$

be the eigenvalues of $\mathcal{A}_{3,k}$, and

$$\lambda_{-(N+1)n}(\mathcal{A}_{3,k+1}) \leq \lambda_{-((N+1)n-1)}(\mathcal{A}_{3,k+1}) \leq \cdots \leq \lambda_{-1}(\mathcal{A}_{3,k+1}) < 0, \tag{5.22}$$

$$0 < \lambda_1(\mathcal{A}_{3,k+1}) \leq \cdots \leq \lambda_{(N+1)n+k+1}(\mathcal{A}_{3,k+1}) \tag{5.23}$$

be the eigenvalues of $\mathcal{A}_{3,k+1}$. Then by Theorem 5.2:

$$\begin{aligned}
\text{smallest negative eigenvalues :} \quad & \lambda_{-(N+1)n}(\mathcal{A}_{3,k+1}) \leq \lambda_{-(N+1)n}(\mathcal{A}_{3,k}), \\
\text{largest negative eigenvalues :} \quad & \lambda_{-1}(\mathcal{A}_{3,k+1}) \leq \lambda_{-1}(\mathcal{A}_{3,k}), \\
\text{smallest positive eigenvalues :} \quad & \lambda_1(\mathcal{A}_{3,k+1}) \leq \lambda_1(\mathcal{A}_{3,k}), \\
\text{largest positive eigenvalues :} \quad & \lambda_{(N+1)n+k}(\mathcal{A}_{3,k}) \leq \lambda_{(N+1)n+k+1}(\mathcal{A}_{3,k+1}).
\end{aligned}$$

□

To obtain information on not only how the eigenvalues of $\mathcal{A}_3$ change because of new observations, but also on where the eigenvalues lie when the number of observations is fixed, we formulate intervals for the negative and positive eigenvalues of $\mathcal{A}_3$.

Theorem 5.7. *The negative eigenvalues of $\mathcal{A}_3$ lie in the interval*

$$I_- = \Big[\frac{1}{2}\Big(\tau_{min} - \sqrt{\tau_{min}^2 + 4\theta_{max}^2}\Big), \frac{1}{2}\Big(\tau_{max} - \sqrt{\tau_{max}^2 + 4\theta_{min}^2}\Big)\Big] \tag{5.24}$$

and the positive eigenvalues lie in the interval

$$I_+ = \Big[\tau_{min}, \frac{1}{2}\Big(\tau_{max} + \sqrt{\tau_{max}^2 + 4\theta_{max}^2}\Big)\Big], \tag{5.25}$$

where τ_{min}, τ_{max}, and θ_i are defined in (5.18), (5.19), *and Table 5.1.*

Proof. Lemma 2.1 of [Rusten and Winther, 1992] gives eigenvalue intervals for matrices of the form $\mathbf{A} = \begin{pmatrix} \mathbf{C} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{0} \end{pmatrix}$. Applying these intervals in the case $\mathbf{C} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix}$ and $\mathbf{E}^T = \begin{pmatrix} \mathbf{L}^T & \mathbf{H}^T \end{pmatrix}$ yields the required results. □

We present two corollaries that describe how the bounds in Theorem 5.7 change if additional observations are introduced and conclude that the change of the bounds is consistent with the results in Theorem 5.6.

Corollary 5.8. *The interval for the positive eigenvalues of $\mathcal{A}_3$ in* (5.25) *either expands or is unchanged when new observations are added.*

Proof. First, consider the positive upper bound $\frac{1}{2}\Big(\tau_{max} + \sqrt{\tau_{max}^2 + 4\theta_{max}^2}\Big)$. By Lemma 5.3, θ_{max}^2 increases or is unchanged when additional observations are included. If $\tau_{max} = \rho_{max}$, the same holds for τ_{max} (by Lemma 5.4). If $\tau_{max} = \psi_{max}$, changing the number of observations does not affect τ_{max}. Hence, the positive upper bound increases or is unchanged.

The positive lower bound τ_{min} is unaltered if $\tau_{min} = \psi_{min}$. If $\tau_{min} = \rho_{min}$, the bound decreases or is unchanged by Lemma 5.4. □

Corollary 5.9. *If $\tau_{max} = \psi_{max}$, the upper bound for the negative eigenvalues of $\mathcal{A}_3$ in* (5.24) *is either unchanged or moves away from zero when new observations are added. If $\tau_{min} = \psi_{min}$, the same holds for the lower bound for negative eigenvalues in* (5.24)*.*

Proof. The results follow from the facts that ψ_{max} and ψ_{min} do not change if observations are added, whereas θ_{min} and θ_{max} increase or are unchanged by Lemma 5.3. □

If $\tau_{max} = \rho_{max}$ or $\tau_{min} = \rho_{min}$, it is unclear how the interval for the negative eigenvalues in (5.24) changes, because $\sqrt{\tau_{min}^2 + 4\theta_{max}^2}$ can increase, decrease or be unchanged, and both τ_{max} and $\sqrt{\tau_{max}^2 + 4\theta_{min}^2}$ can increase or be unchanged.

5.4.3 Bounds for the 2×2 block formulation

$\mathcal{A}_2$ given in (5.14) is equal to the following congruence transformation

$$\mathcal{A}_2 = \begin{pmatrix} \mathbf{D} & \mathbf{L} \\ \mathbf{L}^T & -\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \end{pmatrix} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{L}^T & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{D} & \mathbf{L} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \tag{5.26}$$

where $\mathbf{I} \in \mathbb{R}^{(N+1)n \times (N+1)n}$ is the identity matrix. Then by Sylvester's law, $\mathcal{A}_2$ has $(N+1)n$ positive and $(N+1)n$ negative eigenvalues. The change of the extreme negative and positive eigenvalues of $\mathcal{A}_2$ due to the additional observations is analysed in the subsequent theorem. However, the result holds only in the case of uncorrelated observation errors, unlike the general analysis for $\mathcal{A}_3$ in Theorem 5.6.

Theorem 5.10. *If the observation errors are uncorrelated, i.e., $\mathbf{R}$ is diagonal, then the smallest and largest negative eigenvalues of $\mathcal{A}_2$ either move away from zero or are unchanged when new observations are added. Contrarily, the smallest and largest positive eigenvalues of $\mathcal{A}_2$ approach zero or are unchanged.*

Proof. Matrices $\mathbf{D}$ and $\mathbf{L}$ do not depend on the number of observations. In Lemma 5.5, we have shown that $\mathbf{H}_{k+1}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{k+1} = \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k + \alpha^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T$, $(\alpha > 0)$ for diagonal $\mathbf{R}$. Hence, when $\mathcal{A}_{2,k}$ denotes $\mathcal{A}_2$ with $q = k$, we can write

$$\mathcal{A}_{2,k+1} = \mathcal{A}_{2,k} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\alpha^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T \end{pmatrix} = \mathcal{A}_{2,k} + \mathcal{E}_2,$$

where $\mathcal{E}_2$ has negative and zero eigenvalues. Let

$$\lambda_{-(N+1)n}(\mathcal{A}_{2,k}) \leq \cdots \leq \lambda_{-1}(\mathcal{A}_{2,k}) < 0 < \lambda_1(\mathcal{A}_{2,k}) \leq \cdots \leq \lambda_{(N+1)n}(\mathcal{A}_{2,k})$$

be the eigenvalues of $\mathcal{A}_{2,k}$, and

$$\lambda_{-(N+1)n}(\mathcal{A}_{2,k+1}) \leq \cdots \leq \lambda_{-1}(\mathcal{A}_{2,k+1}) < 0 < \lambda_1(\mathcal{A}_{2,k+1}) \leq \cdots \leq \lambda_{(N+1)n}(\mathcal{A}_{2,k+1})$$

be the eigenvalues of $\mathcal{A}_{2,k+1}$. By Theorem 5.1,

smallest negative eigenvalues :

$$\lambda_{-(N+1)n}(\mathcal{A}_{2,k}) - \alpha^{-1}\lambda_{max}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) \leq \lambda_{-(N+1)n}(\mathcal{A}_{2,k+1}) \leq \lambda_{-(N+1)n}(\mathcal{A}_{2,k}),$$

largest negative eigenvalues :

$$\lambda_{-1}(\mathcal{A}_{2,k}) - \alpha^{-1}\lambda_{max}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) \leq \lambda_{-1}(\mathcal{A}_{2,k+1}) \leq \lambda_{-1}(\mathcal{A}_{2,k}),$$

smallest positive eigenvalues :

$$\lambda_{1}(\mathcal{A}_{2,k}) - \alpha^{-1}\lambda_{max}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) \leq \lambda_{1}(\mathcal{A}_{2,k+1}) \leq \lambda_{1}(\mathcal{A}_{2,k}),$$

largest positive eigenvalues :

$$\lambda_{(N+1)n}(\mathcal{A}_{2,k}) - \alpha^{-1}\lambda_{max}(\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T) \leq \lambda_{(N+1)n}(\mathcal{A}_{2,k+1}) \leq \lambda_{(N+1)n}(\mathcal{A}_{2,k}).$$

□

We further search for the intervals in which the negative and positive eigenvalues of $\mathcal{A}_2$ lie. We follow a similar line of thought as in [Silvester and Wathen, 1994], with the energy arguments for any non-zero vector $\mathbf{w} \in \mathbb{R}^{(N+1)n}$

$$\psi_{min}||\mathbf{w}||^2 \leq \mathbf{w}^T\mathbf{D}\mathbf{w} \leq \psi_{max}||\mathbf{w}||^2, \tag{5.27}$$

$$-\nu_{max}||\mathbf{w}||^2 \leq -\mathbf{w}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{w} \leq -\nu_{min}||\mathbf{w}||^2, \tag{5.28}$$

$$\sigma_{min}||\mathbf{w}|| \leq ||\mathbf{L}^T\mathbf{w}|| \leq \sigma_{max}||\mathbf{w}||, \tag{5.29}$$

$$\theta_{min}||\mathbf{w}|| \leq ||(\mathbf{L}^T\ \mathbf{H}^T)^T\mathbf{w}|| \leq \theta_{max}||\mathbf{w}||. \tag{5.30}$$

Theorem 5.11. *The negative eigenvalues of $\mathcal{A}_2$ lie in the interval*

$$I_- = \left[\frac{1}{2}\left(\psi_{min} - \nu_{max} - \sqrt{(\psi_{min}+\nu_{max})^2 + 4\sigma_{max}^2}\right), min\left\{\beta_1, max\left\{\beta_2, \beta_3\right\}\right\}\right], \tag{5.31}$$

where

$$\beta_1 = \frac{1}{2}\left(\psi_{max} - \nu_{min} - \sqrt{(\psi_{max}+\nu_{min})^2 + 4\sigma_{min}^2}\right), \tag{5.32}$$

$$\beta_2 = -\rho_{max}^{-1}\theta_{min}^2, \tag{5.33}$$

$$\beta_3 = \frac{1}{2}\left(\psi_{max} - \sqrt{\psi_{max}^2 + 4\theta_{min}^2}\right), \tag{5.34}$$

and the positive ones lie in the interval

$$I_+ = \left[I_+^{(1)}, I_+^{(2)}\right], \tag{5.35}$$

where

$$I_+^{(1)} = \frac{1}{2}\left(\psi_{min} - \nu_{max} + \sqrt{(\psi_{min}+\nu_{max})^2 + 4\sigma_{min}^2}\right), \tag{5.36}$$

$$I_+^{(2)} = \frac{1}{2}\left(\psi_{max} - \nu_{min} + \sqrt{(\psi_{max}+\nu_{min})^2 + 4\sigma_{max}^2}\right). \tag{5.37}$$

Proof. Assume that $(\mathbf{u}^T, \mathbf{v}^T)^T,\ \mathbf{u}, \mathbf{v} \in \mathbb{R}^{(N+1)n}$ is an eigenvector of $\mathcal{A}_2$ with an eigenvalue ζ. Then the eigenvalue equations are

$$\mathbf{D}\mathbf{u} + \mathbf{L}\mathbf{v} = \zeta\mathbf{u}, \tag{5.38}$$

$$\mathbf{L}^T\mathbf{u} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{v} = \zeta\mathbf{v}. \tag{5.39}$$

We note that if $\mathbf{u} = \mathbf{0}$ then $\mathbf{v} = \mathbf{0}$ by (5.38) and if $\mathbf{v} = \mathbf{0}$ then $\mathbf{u} = \mathbf{0}$ by (5.39). Hence, $\mathbf{u}, \mathbf{v} \neq \mathbf{0}$.

First, we consider $\zeta > 0$. Equation (5.39) gives $\mathbf{v} = (\mathbf{I}\zeta + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{L}^T\mathbf{u}$, where $\mathbf{I} \in \mathbb{R}^{(N+1)n\times(N+1)n}$. The matrix $\mathbf{I}\zeta + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is positive definite, hence nonsingular. We multiply (5.38) by $\mathbf{u}^T$ and use the previous expression for $\mathbf{v}$ to get

$$\mathbf{u}^T\mathbf{D}\mathbf{u} + \mathbf{u}^T\mathbf{L}(\mathbf{I}\zeta + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{L}^T\mathbf{u} = \zeta||\mathbf{u}||^2. \tag{5.40}$$

The eigenvalues of $(\mathbf{I}\zeta+\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$ in increasing order are $(\zeta+\nu_{max})^{-1}, \ldots, (\zeta+\nu_{min})^{-1}$. Then

$$\begin{aligned}\mathbf{u}^T\mathbf{L}(\mathbf{I}\zeta + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{L}^T\mathbf{u} \geq& \frac{1}{\zeta + \nu_{max}}||\mathbf{L}^T\mathbf{u}||^2 &(5.41)\\ \geq& \frac{1}{\zeta + \nu_{max}}\sigma^2_{min}||\mathbf{u}||^2 \quad \text{[by (5.29)]}. &(5.42)\end{aligned}$$

Hence, this inequality together with (5.27) and (5.40) gives

$$\zeta||\mathbf{u}||^2 \geq \psi_{min}||\mathbf{u}||^2 + \frac{1}{\zeta + \nu_{max}}\sigma^2_{min}||\mathbf{u}||^2 \tag{5.43}$$

and solving

$$\zeta^2 + (\nu_{max} - \psi_{min})\zeta - \psi_{min}\nu_{max} - \sigma^2_{min} \geq 0 \tag{5.44}$$

results in

$$\zeta \geq \frac{1}{2}\left(\psi_{min} - \nu_{max} + \sqrt{(\psi_{min} + \nu_{max})^2 + 4\sigma^2_{min}}\right). \tag{5.45}$$

Similarly, using the upper bound from (5.27) and employing (5.40) yields the upper bound

$$\zeta \leq \frac{1}{2}\left(\psi_{max} - \nu_{min} + \sqrt{(\psi_{max} + \nu_{min})^2 + 4\sigma^2_{max}}\right). \tag{5.46}$$

Now consider the case $\zeta < 0$. Since $\mathbf{D} - \zeta\mathbf{I}$ is positive definite, from (5.38)

$$\mathbf{u} = -(\mathbf{D} - \zeta\mathbf{I})^{-1}\mathbf{L}\mathbf{v}. \tag{5.47}$$

Using this expression and multiplying (5.39) by $\mathbf{v}^T$ gives

$$-\zeta||\mathbf{v}||^2 = \mathbf{v}^T\mathbf{L}^T(\mathbf{D} - \zeta\mathbf{I}_{(N+1)n})^{-1}\mathbf{L}\mathbf{v} + \mathbf{v}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{v}. \tag{5.48}$$

Then using (5.28), (5.29) and the fact that the smallest eigenvalue of $(\mathbf{D}-\zeta\mathbf{I})^{-1}$ is $(\psi_{max}-\zeta)^{-1}$ results in inequality

$$-\zeta||\mathbf{v}||^2 \geq \sigma^2_{min}||\mathbf{v}||^2\frac{1}{\psi_{max} - \zeta} + \nu_{min}||\mathbf{v}||^2, \tag{5.49}$$

which can be expressed as

$$\zeta^2 - (\psi_{max} - \nu_{min})\zeta - \nu_{min}\psi_{max} - \sigma^2_{min} \geq 0, \tag{5.50}$$

and its solution gives the upper bound

$$\zeta \leq \frac{1}{2}\left(\psi_{max} - \nu_{min} - \sqrt{(\psi_{max} + \nu_{min})^2 + 4\sigma^2_{min}}\right) = \beta_1. \tag{5.51}$$

Notice that the bound (5.51) takes into account information on observations only if the system is fully observed. Otherwise, $q < (N+1)n$ and $\nu_{min} = 0$.

We obtain an alternative upper bound for the negative eigenvalues, that depends on the observational information and might be useful for the fully observed case, too. Equation (5.48) may be written as

$$-\zeta||\mathbf{v}||^2 = \mathbf{v}^T(\mathbf{L}^T\ \mathbf{H}^T)\begin{pmatrix} (\mathbf{D}-\zeta\mathbf{I})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{pmatrix}\begin{pmatrix} \mathbf{L} \\ \mathbf{H} \end{pmatrix}\mathbf{v}. \tag{5.52}$$

Eigenvalues of the 2×2 block matrix in the previous equation are the eigenvalues of $(\mathbf{D}-\zeta\mathbf{I})^{-1}$ and $\mathbf{R}^{-1}$. Thus, by an energy argument (5.27),

$$-\zeta||\mathbf{v}||^2 \geq min\{\rho_{max}^{-1}, (-\zeta+\psi_{max})^{-1}\}||(\mathbf{L}^T\ \mathbf{H}^T)^T\mathbf{v}||^2 \tag{5.53}$$

$$\geq min\{\rho_{max}^{-1}, (-\zeta+\psi_{max})^{-1}\}\theta_{min}^2||\mathbf{v}||^2 \quad \text{[by (5.30)]}. \tag{5.54}$$

Hence,

$$\zeta \leq -\theta_{min}^2\iota, \tag{5.55}$$

where $\iota = min\{\rho_{max}^{-1}, (-\zeta+\psi_{max})^{-1}\}$. If $\iota = \rho_{max}^{-1}$, the upper bound is

$$\zeta \leq -\rho_{max}^{-1}\theta_{min}^2 = \beta_2. \tag{5.56}$$

If $\iota = (-\zeta+\psi_{max})^{-1}$, the following inequality

$$\zeta^2 - \psi_{max}\zeta - \theta_{min}^2 \geq 0 \tag{5.57}$$

gives the bound

$$\zeta \leq \frac{1}{2}\left(\psi_{max} - \sqrt{\psi_{max}^2 + 4\theta_{min}^2}\right) = \beta_3. \tag{5.58}$$

Hence,

$$\zeta \leq max\{\beta_2, \beta_3\}. \tag{5.59}$$

The required upper bound follows from (5.51) and (5.59)

Next, we obtain the lower bound for the negative eigenvalues. Using equation (5.48) with the largest eigenvalue of $(\mathbf{D}-\zeta\mathbf{I})^{-1}$ and other parts of (5.28) and (5.29) yields

$$-\zeta||\mathbf{v}||^2 \leq \sigma_{max}^2||\mathbf{v}||^2\frac{1}{\psi_{min}-\zeta} + \nu_{max}||\mathbf{v}||^2.$$

Solving

$$\zeta^2 - (\psi_{min}-\nu_{max})\zeta - \nu_{max}\psi_{min} - \sigma_{max}^2 \leq 0$$

results in

$$\zeta \geq \frac{1}{2}\Big(\psi_{min} - \nu_{max} - \sqrt{(\psi_{min}+\nu_{max})^2 + 4\sigma_{max}^2}\Big).$$

□

We observe that if the system is not fully observed, then $q < (N+1)n$ and $\nu_{min} = 0$, and the upper bound for the positive eigenvalues and the upper bound for the negative eigenvalues (5.32) in Theorem 5.11 reduces to (2.11) and (2.13) of [Silvester and Wathen, 1994].

We are interested in how the bounds in Theorem 5.11 change if additional observations are introduced. The change to the upper negative bound in (5.31) depends on which of (5.32), (5.33) or (5.34) gives the bound. Hence, in Corollary 5.12 we comment on when (5.34) is larger than (5.33) and Corollary 5.13 describes a setting when the negative upper bound is given by (5.34).

Corollary 5.12.

$$max\{\beta_2, \beta_3\} = \beta_3 \quad \iff \frac{1}{2}(\psi_{max} + \sqrt{\psi_{max}^2 + \theta_{min}^2}) \geq \rho_{max}.$$

Proof. $max\{\beta_2, \beta_3\} = \beta_3$ if and only if

$$\frac{1}{2}\Big(\psi_{max} - \sqrt{\psi_{max}^2 + 4\theta_{min}^2}\Big) \geq -\rho_{max}^{-1}\theta_{min}^2.$$

Rearranging this inequality gives

$$\psi_{max} + 2\rho_{max}^{-1}\theta_{min}^2 \geq \sqrt{\psi_{max}^2 + 4\theta_{min}^2}.$$

Squaring both sides with further rearrangement results in

$$\theta_{min}^2(\rho_{max}^{-1}\psi_{max} + \rho_{max}^{-2}\theta_{min}^2 - 1) \geq 0.$$

Since $\theta_{min}^2 > 0$, this is equivalent to

$$\rho_{max}^2 - \rho_{max}\psi_{max} - \theta_{min}^2 \leq 0,$$

from which it follows that

$$\rho_{max} \leq \frac{1}{2}\Big(\psi_{max} + \sqrt{\psi_{max}^2 + 4\theta_{min}^2}\Big).$$

□

Corollary 5.12 can be used to check if the assumption in the following corollary holds.

Corollary 5.13. *If the system is not fully observed and* $max\{\beta_2, \beta_3\} = \beta_3$*, then the upper bound for the negative eigenvalues of* $\mathcal{A}_2$ *is given by* (5.34)*.*

Proof. The singular values of $\mathbf{L}$ and $(\mathbf{L}^T \quad \mathbf{H}^T)$ are the square roots of the eigenvalues of $\mathbf{L}^T\mathbf{L}$ and $\mathbf{L}^T\mathbf{L} + \mathbf{H}^T\mathbf{H}$, respectively. Hence, by Theorem 5.1,

$$\sigma_{min}^2 + \lambda_{min}(\mathbf{H}^T\mathbf{H}) \leq \theta_{min}^2,$$

where $\lambda_{min}(\mathbf{H}^T\mathbf{H}) \geq 0$, since $\mathbf{H}^T\mathbf{H}$ is symmetric positive semi-definite. Also, if $q < (N+1)n$, then $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is singular, that is, $\nu_{min} = 0$, and from (5.32) and (5.34)

$$\beta_1 = \frac{1}{2}\Big(\psi_{max} - \sqrt{\psi_{max}^2 + 4\sigma_{min}^2}\Big) \geq \frac{1}{2}\Big(\psi_{max} - \sqrt{\psi_{max}^2 + 4\theta_{min}^2}\Big) = \beta_3 = max\{\beta_2, \beta_3\}.$$

□

We further describe how the negative upper bound changes if it is given by (5.32) or (5.34), including the case described in Corollary 5.13.

Corollary 5.14. *If the upper bound for the negative eigenvalues of $\mathcal{A}_2$ in* (5.31) *is given by β_1 or β_3, then the bound moves away from zero or stays the same when new observations are added.*

Proof. β_1 does not change while the system is not fully observed. When the system becomes fully observed, $\nu_{min} > 0$ and β_1 decreases. β_3 decreases or stays the same by Lemma 5.3. □

Note that if the negative upper bound in (5.31) is given by β_2, it is unclear how the bound changes with the number of observations, since both ρ_{max} and θ^2_{min} increase or stay the same. The same is true for the positive bounds in (5.35). Only ν_{max} and ν_{min} depend on the available observations and they are contained in elements with positive and negative signs.

The result in Corollary 5.14 that applies for $\mathcal{A}_2$ with a general $\mathbf{R}$ is consistent with the result in Theorem 5.10 that considers $\mathcal{A}_2$ with a diagonal $\mathbf{R}$. The same holds for the result in the following corollary, that determines how the lower bound for the negative eigenvalues of $\mathcal{A}_2$ changes in the special case of uncorrelated observational errors.

Corollary 5.15. *If the observation error covariance matrix $\mathbf{R}$ is diagonal, the negative lower bound in* (5.31) *moves away from zero or stays the same when additional observations are introduced.*

Proof. The result follows by applying Lemma 5.5 to see how ν_{max} changes. □

In the following corollary, we consider the intervals for the positive eigenvalues of $\mathcal{A}_3$ and $\mathcal{A}_2$ with a fixed number of observations. It suggests that we may expect the positive eigenvalues of $\mathcal{A}_2$ to be more clustered than those of $\mathcal{A}_3$.

Corollary 5.16. *The interval for the positive eigenvalues of $\mathcal{A}_2$ is contained in the interval for the positive eigenvalues of $\mathcal{A}_3$, i.e.*

$$\Big[\frac{1}{2}\Big(\psi_{min} - \nu_{max} + \sqrt{(\psi_{min}+\nu_{max})^2 + 4\sigma^2_{min}}\Big), \frac{1}{2}\Big(\psi_{max} - \nu_{min} + \sqrt{(\psi_{max}+\nu_{min})^2 + 4\sigma^2_{max}}\Big)\Big] \subseteq \Big[\tau_{min}, \frac{1}{2}\Big(\tau_{max} + \sqrt{\tau^2_{max} + 4\theta^2_{max}}\Big)\Big].$$

Proof. As observed in Corollary 5.13,

$$\sigma^2_{max} + \lambda_{min}(\mathbf{H}^T\mathbf{H}) \leq \theta^2_{max}, \tag{5.60}$$

with $\lambda_{min}(\mathbf{H}^T\mathbf{H}) \geq 0$. Also, by definition $\tau_{max} \geq \psi_{max}$ and the following inequality for the upper bound for the positive eigenvalues of $\mathcal{A}_3$ holds

$$\frac{1}{2}\Big(\tau_{max} + \sqrt{\tau^2_{max} + 4\theta^2_{max}}\Big) \geq \frac{1}{2}\Big(\psi_{max} + \sqrt{\psi^2_{max} + 4\theta^2_{max}}\Big). \tag{5.61}$$

Thus, we show that the upper bound for positive eigenvalues of $\mathcal{A}_3$ is larger than the upper bound for positive eigenvalues of $\mathcal{A}_2$:

$$\begin{aligned} \frac{1}{2}\Big(\psi_{max} + \sqrt{\psi^2_{max} + 4\theta^2_{max}}\Big) &\geq \frac{1}{2}\Big(\psi_{max} - \nu_{min} + \sqrt{(\psi_{max}+\nu_{min})^2 + 4\sigma^2_{max}}\Big) \\ \iff \nu_{min} + \sqrt{\psi^2_{max} + 4\theta^2_{max}} &\geq \sqrt{(\psi_{max}+\nu_{min})^2 + 4\sigma^2_{max}} \end{aligned} \tag{5.62}$$

(squaring both sides and simplifying)

$$\iff 2\theta_{max}^2 + \nu_{min}\sqrt{\psi_{max}^2 + 4\theta_{max}^2} \geq \psi_{max}\nu_{min} + 2\sigma_{max}^2 \tag{5.63}$$

(rearranging)

$$\iff 2(\theta_{max}^2 - \sigma_{max}^2) \geq \nu_{min}(\psi_{max} - \sqrt{\psi_{max}^2 + 4\theta_{max}^2}). \tag{5.64}$$

Inequality (5.64) always holds because the left hand side is positive and the right hand side is negative.

We also show that the lower bound for the positive eigenvalues of $\mathcal{A}_3$ is smaller than the lower bound for the positive eigenvalues of $\mathcal{A}_2$:

$$\tau_{min} \leq \frac{1}{2}\Big(\psi_{min} - \nu_{max} + \sqrt{(\psi_{min} + \nu_{max})^2 + 4\sigma_{min}^2}\Big). \tag{5.65}$$

Note that by definition $\tau_{min} \leq \psi_{min}$ and the following inequality always holds

$$\psi_{min} \leq \frac{1}{2}\Big(\psi_{min} - \nu_{max} + \sqrt{(\psi_{min} + \nu_{max})^2 + 4\sigma_{min}^2}\Big), \tag{5.66}$$

because it can be simplified to

$$\psi_{min} + \nu_{max} \leq \sqrt{(\psi_{min} + \nu_{max})^2 + 4\sigma_{min}^2} \tag{5.67}$$

$$\text{(squaring both sides)} \quad \iff (\psi_{min} + \nu_{max})^2 \leq (\psi_{min} + \nu_{max})^2 + 4\sigma_{min}^2 \tag{5.68}$$

$$\iff 0 \leq 4\sigma_{min}^2. \tag{5.69}$$

□

5.4.4 Bounds for the 1×1 block formulation

The system matrix $\mathcal{A}_1$ given by (5.16) is symmetric positive definite and so its eigenvalues are positive. We determine how these change due to additional observations when the observation errors are uncorrelated (as for the extreme eigenvalues of $\mathcal{A}_2$ in Theorem 5.10).

Theorem 5.17. *If the observation errors are uncorrelated, i.e.* $\mathbf{R}$ *is diagonal, then the eigenvalues of* $\mathcal{A}_1$ *move away from zero or are unchanged when new observations are added.*

Proof. Let $\mathcal{A}_{1,k}$ denote $\mathcal{A}_1$ where $q = k$. Then $\mathcal{A}_{1,k+1} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}_{k+1}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{k+1} = \mathcal{A}_{1,k} + \alpha^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T$. The result follows by applying Theorem 5.1. □

We formulate spectral bounds for $\mathcal{A}_1$ that depend on the largest and smallest eigenvalues of $\mathbf{D}$ and $\mathbf{R}$, and the largest and smallest singular values of $(\mathbf{L}^T\ \mathbf{H}^T)$.

Theorem 5.18. *The eigenvalues of* $\mathcal{A}_1$ *lie in the interval*

$$I_+ = \left[\theta_{min}^2/\tau_{max}, \theta_{max}^2/\tau_{min}\right], \tag{5.70}$$

where θ_i *and* τ_i *are defined in Table 5.1, and* (5.18) *and* (5.19).

Proof. Assume that $\mathbf{u} \in \mathbb{R}^{(N+1)n}$ is an eigenvector of $\mathcal{A}_1$. Then the eigenvalue equation premultiplied by $\mathbf{u}^T$ can be written as

$$\chi||\mathbf{u}||^2 = \mathbf{u}^T(\mathbf{L}^T\ \mathbf{H}^T)\begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{pmatrix}\begin{pmatrix} \mathbf{L} \\ \mathbf{H} \end{pmatrix}\mathbf{u}, \tag{5.71}$$

where χ is an eigenvalue of $\mathcal{A}_1$. The smallest and largest eigenvalues of $\begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{pmatrix}$ are τ_{max}^{-1} and τ_{min}^{-1}, respectively. The bounds follow from the following inequalities that are obtained using (5.30):

$$\chi||\mathbf{u}||^2 \geq \tau_{max}^{-1}\mathbf{u}^T(\mathbf{L}^T\ \mathbf{H}^T)\begin{pmatrix} \mathbf{L} \\ \mathbf{H} \end{pmatrix}\mathbf{u} \geq \tau_{max}^{-1}\theta_{min}^2||\mathbf{u}||^2, \tag{5.72}$$

$$\chi||\mathbf{u}||^2 \leq \tau_{min}^{-1}\mathbf{u}^T(\mathbf{L}^T\ \mathbf{H}^T)\begin{pmatrix} \mathbf{L} \\ \mathbf{H} \end{pmatrix}\mathbf{u} \leq \tau_{min}^{-1}\theta_{max}^2||\mathbf{u}||^2. \tag{5.73}$$

□

The following corollary explains how the upper bound for the eigenvalues of $\mathcal{A}_1$ changes with the addition of new observations. This result that applies for $\mathcal{A}_1$ with a general $\mathbf{R}$ is consistent with Theorem 5.17 that considers $\mathcal{A}_1$ with a diagonal $\mathbf{R}$.

Corollary 5.19. *The upper bound in Theorem 5.18 moves away from zero or is unchanged when new observations are added.*

Proof. If $\tau_{min} = \rho_{min}$, τ_{min} decreases by Lemma 5.4. Otherwise τ_{min} does not change. The result follows by applying Lemma 5.3 to determine the change to θ_{max}. □

It is unclear how the lower bound in Theorem 5.18 changes with respect to the number of observations, because both the numerator and denominator grow or stay unchanged by Lemmas 5.3 and 5.4, respectively.

5.4.5 Alternative bounds

Alternative eigenvalue bounds for symmetric saddle point matrices have been formulated by [Axelsson and Neytcheva, 2006]. These depend on the eigenvalues of the matrices $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$, $\mathbf{R}$, $\mathbf{D}$ and $\mathcal{A}_1$, and

$$\xi = max\{|\lambda_i(\mathcal{A}_1^{-1/2}\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}\mathcal{A}_1^{-1/2})|, i = 1, \ldots, (N+1)n\}.$$

Theorem 5.20 (From Theorem 1 (c) of [Axelsson and Neytcheva, 2006])**.** *The negative eigenvalues of $\mathcal{A}_3$ lie in the interval*

$$I_- = \left[\frac{1}{2}\left(\tau_{max} - \sqrt{\tau_{max}^2 + 4\tau_{max}\lambda_{max}(\mathcal{A}_1)}\right), \frac{1}{2}\left(\tau_{min} - \sqrt{\tau_{min}^2 + 4\tau_{min}\lambda_{min}(\mathcal{A}_1)}\right)\right] \tag{5.74}$$

and the positive ones lie in the interval

$$I_+ = \left[\tau_{min}, \frac{1}{2}\left(\tau_{max} + \sqrt{\tau_{max}^2 + 4\tau_{max}\lambda_{max}(\mathcal{A}_1)}\right)\right]. \tag{5.75}$$

Note that the lower bound for the positive eigenvalues in Theorem 5.20 is the same as in Theorem 5.7.

Theorem 5.21 (From Theorem 1 (a) and (b) of [Axelsson and Neytcheva, 2006])**.** *The negative eigenvalues of $\mathcal{A}_2$ lie in the interval*

$$I_- = \left[-\lambda_{max}(\mathcal{A}_1), \frac{-\lambda_{min}(\mathcal{A}_1)}{1 + \frac{\xi \lambda_{min}(\mathcal{A}_1)}{\psi_{min}}} \right], \tag{5.76}$$

and the positive ones lie in the interval

$$I_+ = \left[\psi_{min}, \frac{1}{2} \left(\psi_{max} + \sqrt{\psi_{max}^2 + 4\psi_{max}\lambda_{max}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})} \right) \right]. \tag{5.77}$$

We observe that the bound (5.77) for the positive eigenvalues, unlike our bound in Theorem 5.11, is independent of the number of observations. Also, in practical applications it may not be possible to compute the upper bound for the negative eigenvalues because of the ξ term.

5.5 Numerical Experiments

5.5.1 System setup

We present results of numerical experiments using the Lorenz 96 model [Lorenz, 1996], where the state of the system at time t_i is $\mathbf{x}_i = (X_i^1, X_i^2, \ldots, X_i^n)^T$ and the evolution of $\mathbf{x}_i$ components $X^j,\ j \in \{1, 2, \ldots, n\}$, is governed by a set of n coupled ODEs:

$$\frac{dX^j}{dt} = -X^{j-2}X^{j-1} + X^{j-1}X^{j+1} - X^j + F,$$

where $X^{-1} = X^{n-1}, X^0 = X^n$ and $X^{n+1} = X^1$. This model is continuous in time and discrete in space. We assume that $X^1, X^2 \ldots, X^n$ are equally spaced on a periodic domain of length one and take the space increment to be $\Delta X = 1/n$. We require the linearisation of this model $\mathbf{M}_i^{(l)}$, $i \in \{0, \ldots, N-1\}$ to define $\mathcal{A}_3$, $\mathcal{A}_2$ and $\mathcal{A}_1$. In our experiments, we set $n = 40$ and $F = 8$, since the system shows chaotic behaviour with the latter value. The equations are integrated using a fourth order Runge-Kutta scheme [Butcher, 1987]. The time step is set to $\Delta t = 2.5 \times 10^{-2}$ and the system is run for $N = 15$ time steps.

The assimilation system is set up for so-called identical twin experiments, by which synthetic data are generated using the same model as is used in the assimilation. We generate a reference, or "true", model trajectory $\mathbf{x}^t$ by running the Lorenz 96 model over the time window from prescribed initial conditions and with prescribed Gaussian model errors $\boldsymbol{\eta}_i$. An initial background state $\mathbf{x}^b$ and observations $\mathbf{y}_i$ at each time t_i are then generated by adding Gaussian noise to $\mathbf{x}^t$. Assimilation experiments are run using this background state and observations, assuming that the true state is unknown. The error covariance matrices that are used to generate the model error in $\mathbf{x}^t$ and the observation error in $\mathbf{y}_i$ are also used for the assimilation, i.e. in the 3×3 block, 2×2 block and 1×1 block matrices. These error covariance matrices do not change over time. The

observation error covariance matrix is $\mathbf{R}_i = \sigma_o^2 I_{q_i}$, where q_i is the number of observations at time t_i, (diagonal $\mathbf{R}_i$ is a common choice in data assimilation experiments [Freitag and Green, 2018, Gratton et al., 2018a]) and the model error covariance matrix is equal to the background error covariance matrix $\mathbf{Q}_i = \mathbf{B} = \sigma_b^2 \mathbf{C}_b$, where $\mathbf{C}_b$ is a second-order auto-regressive correlation matrix [Daley, 1993] with correlation length scale 1.5×10^{-2}. We have also performed numerical experiments with $\mathbf{Q}_i = \sigma_q^2 \mathbf{C}_q \neq \mathbf{B}$, where $\mathbf{C}_q$ is a Laplacian correlation matrix [Johnson et al., 2005], and σ_q and σ_b vary by a factor of two. We observed similar results to those presented here. In our experiments, the parameters are chosen so that the observations are close to the real values of the variables, and the background and the model errors are low, in particular, we set $\sigma_o = 10^{-1}$, which is about 5% of the mean of the values in $\mathbf{x}^t$, and $\sigma_b = 5 \times 10^{-2}$. $\mathbf{y}_i$ consists of direct observations of the variables $X^j,\ j \in \{1, 2, \ldots, n\}$ at time t_i, hence the observation operator $\mathcal{H}_i$ is linear.

All computations are performed using Matlab R2016b. In particular, the eigenvalues are computed using the Matlab function *eig*. If only extreme eigenvalues are needed, *eigs* is used, and the extreme singular values are given by *svds*.

5.5.2 Eigenvalue bounds

We present numerically calculated eigenvalue bounds and eigenvalues of $\mathcal{A}_3$, $\mathcal{A}_2$ and $\mathcal{A}_1$ and illustrate their change with the number of observations and the quality of the spectral estimates, presented in Section 5.4. We consider the following observation networks that have different numbers of observations ($q = \sum_{i=0}^{N} q_i$):

a) 1 observation at the final time t_{15},

b) 20 observations, observing every eighth model variable at every fourth time step (at times t_3, t_7, t_{11}, t_{15}),

c) 80 observations, observing every fourth model variable at every second time step (at times $t_1, t_3, t_5, t_7, t_9, t_{11}, t_{13}, t_{15}$),

d) 160 observations, observing every second model variable at every second time step (at the same times as in observation network c)),

e) 320 observations, observing every second model variable at every time step,

f) 640 observations, fully observed system.

In Figure 5.1, we plot the eigenvalues of the matrices $\mathcal{A}_3$, $\mathcal{A}_2$, and $\mathcal{A}_1$ together with the bounds from Theorems 5.7, 5.11, and 5.18, respectively, for each of the observation networks a-f. In these experiments, as expected from Theorem 5.6, as the number of observations increases, the smallest and largest negative and the largest positive eigenvalues of $\mathcal{A}_3$ move away from zero and the smallest positive eigenvalue approaches zero. Also, as determined in Corollary 5.8, the upper bound for the positive eigenvalues of $\mathcal{A}_3$ presented in Figure 5.1a grows and the lower bound stays the same (because the eigenvalues of $\mathbf{R}$ do not change) when more observations are added. The change is too small to observe in the

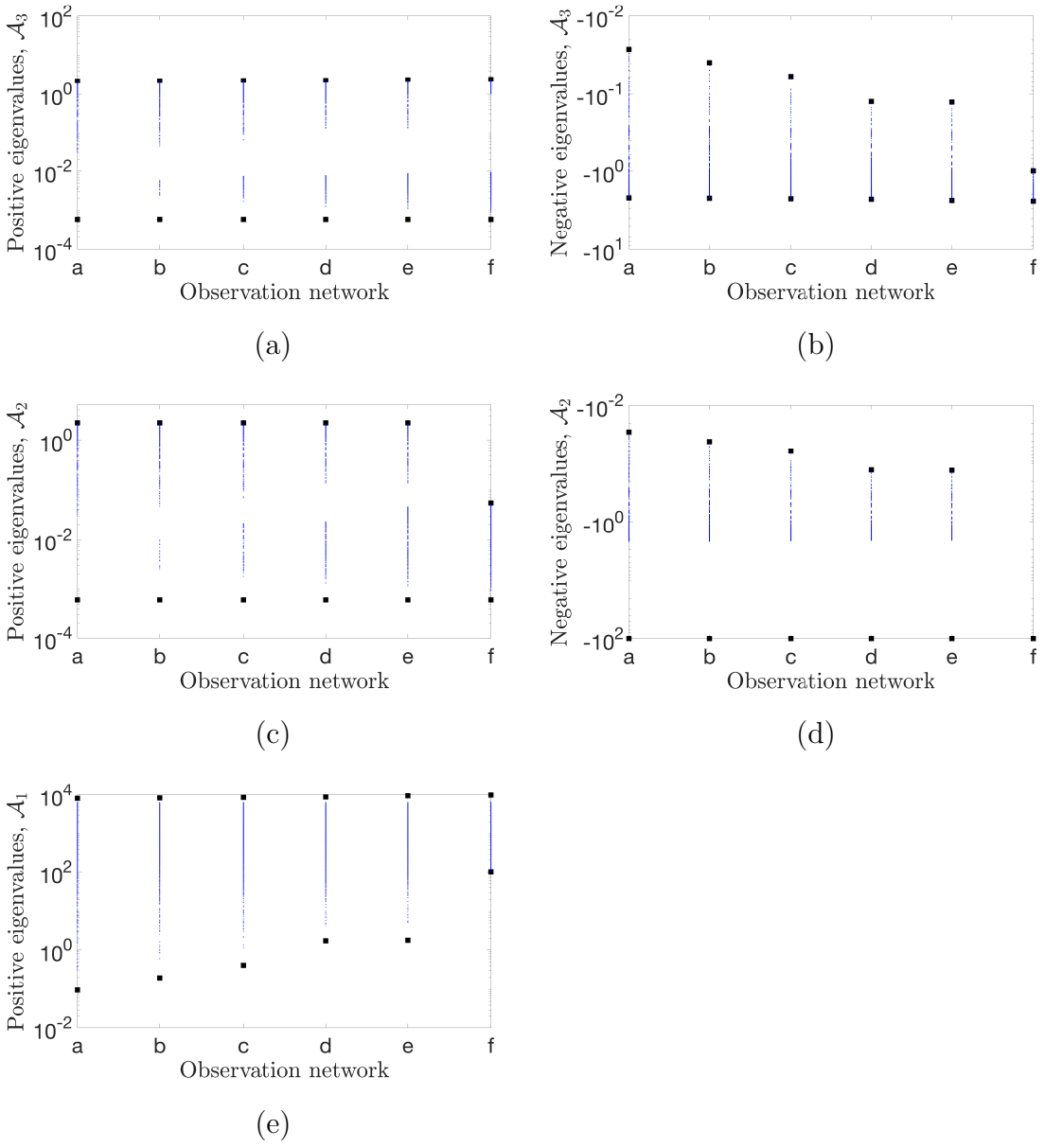


Figure 5.1: Semilogarithmic plots of the positive and negative eigenvalues of the matrices $\mathcal{A}_3$ ((a) and (b)) and $\mathcal{A}_2$ ((c) and (d)), and the positive eigenvalues of $\mathcal{A}_1$ in (e) for the different observation networks (a-f). Eigenvalues are denoted with merged blue dots. The filled black squares mark the bounds for eigenvalues of $\mathcal{A}_3$ in Theorem 5.7, $\mathcal{A}_2$ in Theorem 5.11, and $\mathcal{A}_1$ in Theorem 5.18. Note that the smallest negative eigenvalues of $\mathcal{A}_2$ coincide with the bounds.

plots, hence we report the extreme eigenvalues of $\mathcal{A}_3$ and the intervals from Theorem 5.7 for the networks a), c), e) and f) in Table 5.2. Moreover, the negative bounds for the eigenvalues of $\mathcal{A}_3$ in Figure 5.1b move away from zero. This agrees with Corollary 5.9, because here $\tau_{min} = \psi_{min}$. However, in this setting $\tau_{max} = \rho_{max}$ and the same Corollary cannot be used to explain the change to the upper bound. In general, the outer bounds (the largest positive and the smallest negative) for the eigenvalues of $\mathcal{A}_3$ are tight and the inner bounds (the smallest positive and the largest negative) get tighter as the number of

observations increases.

The positive eigenvalues of $\mathcal{A}_2$ displayed in Figure 5.1c approach zero as observations are added, whereas the negative eigenvalues in Figure 5.1d move away from it. This is consistent with Theorem 5.10, which holds for this experiment because we have chosen diagonal $\mathbf{R}$. The lower bounds for the positive and negative eigenvalues of $\mathcal{A}_2$ stay the same when the observation network is changed. In these bounds only ν_{max} (the largest eigenvalue of $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$) depends on the observations. In our experiments, ν_{max} does not change because of our choice of $\mathbf{H}$ and $\mathbf{R}$. The constant negative lower bound is consistent with Corollary 5.15. The numerical values of the intervals from Theorem 5.11 and of the extreme eigenvalues of $\mathcal{A}_2$ for the networks a), c), e) and f) are presented in Table 5.3. The upper positive bound moves towards zero when the system becomes fully observed and is constant for the other networks, because the smallest eigenvalue ν_{min} of $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is nonzero only for the fully observed system. The negative upper bound for the spectrum of $\mathcal{A}_2$ is given by β_1 in (5.32) for the fully observed system and β_3 in (5.34) otherwise, and moves away from zero, in agreement with Corollary 5.14. Notice that the eigenvalue bounds are tight. Also, the numerical results confirm the statement of Corollary 5.16 that the interval for the positive eigenvalues of $\mathcal{A}_3$ contains the bounds for positive eigenvalues of $\mathcal{A}_2$.

Note that $\mathcal{A}_2$ has q distinct eigenvalues that coincide with the negative lower bound in the plots. The distinct eigenvalues are explained by the bounds for individual eigenvalues in Corollary 5.26 in Appendix 5.7, because in our experiments $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ has eigenvalues that are equal to $\sigma_o^{-2} = 10^2$ and the largest singular value σ_{max} of $\mathbf{L}$ is less than 10. Hence, there are q eigenvalues of $\mathcal{A}_2$ in the interval $[-110, -90]$ and $(N+1)n - q$ eigenvalues no further than 10 from zero.

The eigenvalues of $\mathcal{A}_1$ and their bounds presented in Figure 5.1e move away from zero when more observations are used. This is as expected, because Theorem 5.17 holds for our choice of diagonal $\mathbf{R}$. The variation of the bounds is explained by the fact that with our choice of $\mathbf{R}$ values of τ_{min} and τ_{max} do not change, and θ_{min} and θ_{max} grow. Such behaviour of the upper bound agrees with Corollary 5.19. However, as can be seen in Table 5.4 the upper value of the intervals in Theorem 5.18 are too pessimistic.

Better eigenvalue clustering away from zero when more observations are used can speed up the convergence of iterative solvers when solving the 1×1 block formulation. However, nothing definite can be said about the 3×3 block and 2×2 block formulations: the negative eigenvalues become more clustered, but the smallest positive eigenvalues approach zero when new observations are introduced.

We also calculate the alternative eigenvalue bounds given in Theorems 5.20 and 5.21. With the choice of parameters and observations considered in this section, the bounds given in these theorems are not as sharp as those in Theorems 5.7 and 5.11. However, this is not always the case, as is illustrated in Tables 5.5 and 5.6. Here $\sigma_o = 1.5$, $\sigma_b = 1$ and the observation network d) is used.

O.n.	I_-	Eigenvalues
a)	$[-2.193, -2.66 \times 10^{-2}]$	$[-2.192, -2.99 \times 10^{-2}]$
c)	$[-2.249, -5.88 \times 10^{-2}]$	$[-2.247, -6.18 \times 10^{-2}]$
e)	$[-2.360, -1.28 \times 10^{-1}]$	$[-2.358, -1.31 \times 10^{-1}]$
f)	$[-2.410, -9.96 \times 10^{-1}]$	$[-2.408, -9.96 \times 10^{-1}]$

O.n.	I_+	Eigenvalues
a)	$[5.93 \times 10^{-4}, 2.198]$	$[3.56 \times 10^{-3}, 2.195]$
c)	$[5.93 \times 10^{-4}, 2.254]$	$[1.70 \times 10^{-3}, 2.251]$
e)	$[5.93 \times 10^{-4}, 2.365]$	$[1.13 \times 10^{-3}, 2.362]$
f)	$[5.93 \times 10^{-4}, 2.416]$	$[9.14 \times 10^{-4}, 2.413]$

Table 5.2: Computed spectral intervals and extreme eigenvalues of $\mathcal{A}_3$ from Theorem 5.7 for different observation networks (O.n.).

O.n.	I_-	Eigenvalues
a)	$[-1.0005 \times 10^2, -2.83 \times 10^{-2}]$	$[-1.0001 \times 10^2, -2.99 \times 10^{-2}]$
c)	$[-1.0005 \times 10^2, -6.07 \times 10^{-2}]$	$[-1.0002 \times 10^2, -6.50 \times 10^{-2}]$
e)	$[-1.0005 \times 10^2, -1.29 \times 10^{-1}]$	$[-1.0004 \times 10^2, -1.33 \times 10^{-1}]$
f)	$[-1.0005 \times 10^2, -1.00 \times 10^2]$	$[-1.0005 \times 10^2, -1.00 \times 10^2]$

O.n.	I_+	Eigenvalues
a)	$[6.03 \times 10^{-4}, 2.196]$	$[3.91 \times 10^{-3}, 2.195]$
c)	$[6.03 \times 10^{-4}, 2.196]$	$[1.78 \times 10^{-3}, 2.148]$
e)	$[6.03 \times 10^{-4}, 2.196]$	$[1.15 \times 10^{-3}, 2.101]$
f)	$[6.03 \times 10^{-4}, 5.42 \times 10^{-2}]$	$[9.35 \times 10^{-4}, 5.15 \times 10^{-2}]$

Table 5.3: Computed spectral intervals and extreme eigenvalues of $\mathcal{A}_2$ from Theorem 5.11 for different observation networks (O.n.).

O.n.	I_+	Eigenvalues
a)	$[9.72 \times 10^{-2}, 8.11 \times 10^3]$	$[3.23 \times 10^{-1}, 6.30 \times 10^3]$
c)	$[4.05 \times 10^{-1}, 8.53 \times 10^3]$	$[1.16, 6.32 \times 10^3]$
e)	$[1.75, 9.40 \times 10^3]$	$[5.21, 6.35 \times 10^3]$
f)	$[1.00 \times 10^2, 9.80 \times 10^3]$	$[1.00 \times 10^2, 6.40 \times 10^3]$

Table 5.4: Computed spectral intervals and extreme eigenvalues of $\mathcal{A}_1$ from Theorem 5.18 with different observation networks (O.n.).

Eigenvalues of $\mathcal{A}_3$	Bounds from Th. 5.7	Bounds from Th. 5.20
$[-1.93, -1.38 \times 10^{-2}]$	$[-2.17, -5.83 \times 10^{-3}]$	$[-5.10, -1.33 \times 10^{-2}]$
$[2.98 \times 10^{-1}, 3.59]$	$[2.37 \times 10^{-1}, 3.81]$	$[2.37 \times 10^{-1}, 7.53]$

Table 5.5: Computed spectral intervals and extreme eigenvalues of $\mathcal{A}_3$ from Theorems 5.7 and 5.20 for observation network d) with $\sigma_o = 1.5$ and $\sigma_b = 1$.

Eigenvalues of $\mathcal{A}_2$	Bounds from Th. 5.11	Bounds from Th. 5.21
$[-1.97, -1.39 \times 10^{-2}]$	$[-2.33, -5.83 \times 10^{-3}]$	$[-15.79, -1.33 \times 10^{-2}]$
$[3.00 \times 10^{-1}, 3.51]$	$[2.38 \times 10^{-1}, 3.74]$	$[2.37 \times 10^{-1}, 7.51]$

Table 5.6: Computed spectral intervals and extreme eigenvalues of $\mathcal{A}_2$ from Theorems 5.11 and 5.21 for observation network d) with $\sigma_o = 1.5$ and $\sigma_b = 1$.

5.5.3 Solving the systems

We solve the 3×3 block, 2×2 block, and 1×1 block systems with the coefficient matrices discussed in the previous subsection, and the right hand sides defined in (5.11), (5.13), and (5.15), respectively. The saddle point systems are solved with MINRES and the symmetric positive definite systems are solved with CG. The relative residual at the jth iteration of the iterative method is defined as $||\mathbf{r}_j||/||\mathbf{r}_0||$, where $||\cdot||$ is the L_2 norm and $\mathbf{r}_j$ is the residual on iteration j. The iterative method terminates after 400 iterations or when the relative residual reaches 10^{-4}. The initial guess is taken to be the zero vector.

In Figure 5.2, we plot the relative residuals. Note that the residual reaches 10^{-4} in the fully observed case (observation network f)) when solving each of the systems and convergence is most rapid in this case. This is expected because of the clustering of the eigenvalues. The convergence rates are similar for networks d and e, which is consistent with Figure 5.1. The convergence of MINRES for the observation network a) with a single observation is not explained by the spectra of $\mathcal{A}_3$ and $\mathcal{A}_2$. However, the convergence in other cases agrees with our eigenvalue analysis.

5.6 Conclusions

Weak constraint 4D-Var data assimilation requires the minimisation of a cost function in order to obtain an estimate of the state of a dynamical system. Its solution can be approximated by solving a series of linear systems. We have analysed three different formulations of these systems, namely the standard system with 1×1 block symmetric positive definite coefficient matrix $\mathcal{A}_1$, a new system with a 2×2 block saddle point coefficient matrix $\mathcal{A}_2$, and the version with 3×3 block saddle point coefficient matrix $\mathcal{A}_3$ of [Fisher and Gürol, 2017]. We have focused on the dependency of the coefficient matrices on the number of observations.

We have found that the spectra of $\mathcal{A}_3$, $\mathcal{A}_2$ and $\mathcal{A}_1$ are sensitive to the number of observations and examined how they change when new observations are added. The results

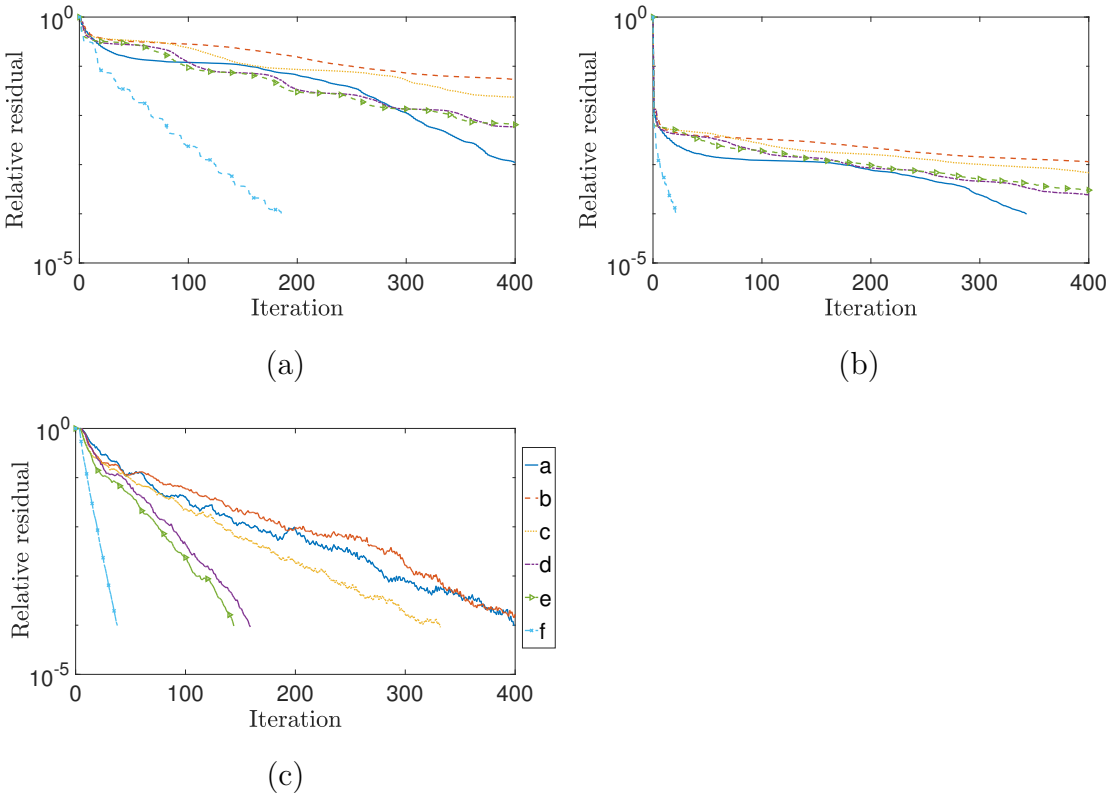


Figure 5.2: Semilogarithmic plots of the relative residual of MINRES when solving the 3×3 block (a) and 2×2 block (b) systems, and the relative residual of CG when solving the 1×1 block (c) system for different observation networks (a-f).

hold with any choice of the blocks in $\mathcal{A}_3$, whereas we can only make inference about the change of the spectra of $\mathcal{A}_2$ and $\mathcal{A}_1$ for uncorrelated observation errors (diagonal $\mathbf{R}$). We have shown that the negative eigenvalues of both $\mathcal{A}_3$ and $\mathcal{A}_2$ move away from zero or are unchanged when observations are added. The smallest and largest positive eigenvalues of $\mathcal{A}_2$, as well as the smallest positive eigenvalue of $\mathcal{A}_3$, approach zero or are unchanged, whereas the largest positive eigenvalue of $\mathcal{A}_3$ moves away from zero or is unchanged. The smallest and largest eigenvalues of $\mathcal{A}_1$ move away from zero or are unchanged. The extreme eigenvalues may cause convergence problems for Krylov subspace solvers, hence we may expect the small positive eigenvalues of $\mathcal{A}_2$ and $\mathcal{A}_3$ to cause these issues when new observations are added. We summarise these results together with the properties of the three systems in Table 5.7.

We have used the work of [Rusten and Winther, 1992] to determine the bounds for the spectrum of $\mathcal{A}_3$ and derived novel bounds for the spectral intervals of the saddle point matrix $\mathcal{A}_2$ and the positive definite matrix $\mathcal{A}_1$. We have observed that the change to the intervals due to new observations is consistent with the change of the extreme eigenvalues of the matrices. Our numerical experiments agree with these findings. In general, the bounds for the saddle point matrices are tight whereas the upper bounds for the positive definite matrix are too pessimistic.

	$\mathcal{A}_3$	$\mathcal{A}_2$	$\mathcal{A}_1$
Type	Symmetric indefinite	Symmetric indefinite	Symmetric positive definite
Iterative solver	MINRES/SYMMLQ	MINRES/SYMMLQ	CG
Order	$2(N+1)n+q$	$2(N+1)n$	$(N+1)n$
$\mathbf{D}^{-1}$ needed	No	No	Yes
$\mathbf{R}^{-1}$ needed	No	Yes	Yes
Sequential matrix products	None	$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$	$\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$, $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$
Eigenvalues that may move towards zero with new observations	Smallest positive	Positive*	None*
Eigenvalues that may move away from zero with new observations	Largest positive, negative	Negative*	All*

Table 5.7: A summary of the properties of the 3×3 block, 2×2 block, and 1×1 block systems. * applies to systems with diagonal $\mathbf{R}$.

Our numerical experiments show slow convergence, particularly with a few observations, and the need for preconditioning is evident. Previous work on the 3×3 block saddle point system considered iteratively solving the problem when inexact constraint preconditioners of [Bergamaschi et al., 2007] are used (see, [Fisher and Gürol, 2017], [Freitag and Green, 2018], [Gratton et al., 2018a]). It was shown that such a preconditioning approach is not optimal and further research into effective preconditioning is still an open question. Preconditioning may transform the coefficient matrix into a non-normal one with GMRES as an iterative solver of choice. Although the spectrum of a non-normal matrix may not be enough to describe the convergence of GMRES [Greenbaum et al., 1996], [Benzi et al., 2005] claim that fast convergence often appears if the spectrum is clustered away from the origin. Hence, a better understanding of the spectrum of $\mathcal{A}_3$, $\mathcal{A}_2$ and $\mathcal{A}_1$ may help in finding a suitable preconditioner for these matrices. We suggest that including the information on observations coming from the observation error covariance matrix $\mathbf{R}$ and the linearised observation operator $\mathbf{H}$ could be beneficial for preconditioning, given that the spectra of all the considered matrices depend on the observations. A design of such preconditioners that are cheap to construct and apply is an interesting area for future research.

Acknowledgments

We would like to kindly thank Dr. Adam El-Said for his code for the weak-constraint 4D-Var assimilation system. We are also grateful to two anonymous reviewers for their

constructive comments that have led to improvements to the paper.

5.7 Appendix: Bounds for individual eigenvalues of $\mathcal{A}_3$ and $\mathcal{A}_2$

We derive bounds for the individual eigenvalues of $\mathcal{A}_3$ and $\mathcal{A}_2$ (Theorems 5.24 and 5.25, respectively). First, we state two theorems that are used in deriving these bounds. The notation of Table 5.1 is used.

Theorem 5.22 (See Theorem 3 in [Silvester, 2000]). *If* $\mathbf{A} = \begin{pmatrix} \mathbf{C} & \mathbf{E} \\ \mathbf{F} & \mathbf{G} \end{pmatrix}$, $\mathbf{C}, \mathbf{E}, \mathbf{F}, \mathbf{G} \in \mathbb{R}^{n \times n}$, *and* $\mathbf{FG} = \mathbf{GF}$, *then*

$$det(\mathbf{A}) = det(\mathbf{CG} - \mathbf{EF}).$$

Theorem 5.23 (Jordan-Wielandt Theorem, see Theorem 4.2 in Chapter 1 of [Stewart and Sun, 1990]). *Let*

$$\mathbf{U}^H \mathbf{A} \mathbf{V} = \begin{pmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \ \mathbf{\Sigma} = diag(\sigma_1, \cdots, \sigma_n)$$

be the singular value decomposition of $\mathbf{A} \in \mathbb{C}^{\mathbb{m} \times \mathbb{n}}$, $m \geq n$. *Then the eigenvalues of the matrix*

$$\mathbf{C} = \begin{pmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^H & \mathbf{0} \end{pmatrix}$$

are $\pm\sigma_1, \cdots, \pm\sigma_n$, *corresponding to the eigenvectors* $\begin{pmatrix} \mathbf{u}_i \\ \pm\mathbf{v}_i \end{pmatrix}$, $i = 1, \cdots, n$, *where* $\mathbf{u}_i$ *and* $\mathbf{v}_i$ *are the* i*-th columns of* $\mathbf{U}$ *and* $\mathbf{V}$, *respectively.* $\mathbf{C}$ *also has* $m - n$ *zero eigenvalues with eigenvectors* $\begin{pmatrix} \mathbf{u}_i \\ \mathbf{0} \end{pmatrix}$, $i = n+1, \cdots, m$.

Theorem 5.24. *Let* $\omega_i,\ i = 1, \ldots, (N+1)n+q$ *be the* i*-th value in* $\{\psi_k, \rho_j | k = 1, \ldots, (N+1)n,\ j = 1, \ldots, q\}$ *(the set of eigenvalues of* $\mathbf{D}$ *and* $\mathbf{R}$*). Then the* k*th eigenvalue of* $\mathcal{A}_3$ *is bounded by*

$$\textit{positive eigenvalues:} \quad \omega_k - \theta_{max} \leq \gamma_k \leq \omega_k + \theta_{max}, \qquad k = 1, \ldots, (N+1)n + q, \tag{5.78}$$

$$\textit{negative eigenvalues:} \quad -\theta_{max} \leq \gamma_{k+(N+1)n+p} < 0, \qquad k = 1, \ldots, (N+1)n. \tag{5.79}$$

Proof. We can write $\mathcal{A}_3$ as a sum of two symmetric matrices:

$$\mathcal{A}_3 = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{0} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & \mathbf{0} \end{pmatrix} = \mathbf{S}_D^{3x3} + \mathbf{S}_L^{3x3}. \tag{5.80}$$

The spectrum of $\mathbf{S}_D^{3x3}$ is the union of the eigenvalues of $\mathbf{D}$, $\mathbf{R}$ and zeros. By Theorem 5.23, the eigenvalues λ of the indefinite matrix $\mathbf{S}_L^{3x3}$ are the singular values of $(\mathbf{L}^T\ \mathbf{H}^T)$ with plus and minus signs, thus $\lambda_{min} = -\theta_{max}$ and $\lambda_{max} = \theta_{max}$.

The result follows from applying Theorem 5.1 to the matrices $\mathbf{S}_D^{3x3}$ and $\mathbf{S}_L^{3x3}$. □

Theorem 5.25. *The eigenvalues of $\mathcal{A}_2$ are bounded by*

$$\begin{aligned} &\textit{positive eigenvalues:} && \psi_k - \sigma_{max} \leq \zeta_k \leq \psi_k + \sigma_{max}, && k = 1, \ldots, (N+1)n. \\ &\textit{negative eigenvalues:} && -\nu_k - \sigma_{max} \leq \zeta_{k+(N+1)n} \leq -\nu_k + \sigma_{max}, && k = 1, \ldots, (N+1)n, \end{aligned} \tag{5.81}$$

Proof. As in Theorem 5.24, we express $\mathcal{A}_2$ as a sum of two symmetric matrices

$$\mathcal{A}_2 = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & -\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{L} \\ \mathbf{L}^T & \mathbf{0} \end{pmatrix} = \mathbf{S}_D^{2x2} + \mathbf{S}_L^{2x2}. \tag{5.82}$$

The rest of the proof is analogous to that of Theorem 5.24. □

Corollary 5.26. *If there are $q < (N+1)n$ observations,* (5.81) *in Theorem 5.25 becomes*

$$-\sigma_{max} \leq \zeta_{k+(N+1)n} \leq 0, \qquad k = 1, \ldots, (N+1)n - q, \tag{5.83}$$

$$-\nu_k - \sigma_{max} \leq \zeta_{k+2(N+1)n-q} < -\nu_k + \sigma_{max}, \qquad k = 1, \ldots, q. \tag{5.84}$$

Proof. The result follows from noticing that $-\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ has $(N+1)n-q$ zero eigenvalues. □

5.8 Summary

We provided eigenvalue bounds for the coefficient matrices in the state formulation. The bounds for the eigenvalues of the saddle point systems are tight, but the bounds for the SPD coefficient matrix are pessimistic. The sensitivity of the extreme eigenvalues of the coefficient matrices to adding new observations was explored. These results hold for general observation error covariance matrices for the 3×3 block saddle point matrix, and for the 2×2 block and SPD matrices with diagonal observation error covariance matrices, that is, when the errors in different observations are uncorrelated. The smallest positive eigenvalue of the 3×3 block and the smallest and largest positive eigenvalues of the 2×2 block matrices move towards zero or are unchanged when new observations are added. The small eigenvalues may cause convergence issues, and they should be addressed when designing the preconditioning (see Chapter 7). The other extreme eigenvalues of the saddle point and SPD matrices move away from zero or stay unchanged.

The SPD system may become easier to solve when the number of observations is increased because of the smallest eigenvalues moving away from zero, and may require different preconditioning strategies than when the number of observations is small. We explore a way to precondition the SPD system in the next chapter.

Chapter 6

First level preconditioning for the SPD state formulation

We tackle the research question 3 in this chapter by using a randomised SVD to approximate the CVT technique. This allows preserving the time-parallelism. We want to know if such preconditioning is useful in the beginning of the iterative process. If yes, does it depend on the number of observations and their errors? Should we incorporate the background and model error information when generating the preconditioner?

This chapter, except the appendix in Section 6.7 and the summary in Section 6.8, is based on the paper: Daužickaitė, I., Lawless, A.S., Scott, J.A. and van Leeuwen, P.J. (2021) On time-parallel preconditioning for the state formulation of incremental weak constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 147(740), 3521 - 3529.

6.1 Abstract

Using a high degree of parallelism is essential for the efficient performance of data assimilation. The state formulation of the incremental weak constraint four-dimensional variational data assimilation method allows parallel calculations in the time dimension. In this approach, the solution is approximated by minimising a series of quadratic cost functions using the conjugate gradient method. To use this method in practice, effective preconditioning strategies that maintain the potential for parallel calculations are needed. We examine approximations to the control variable transform (CVT) technique when the latter is beneficial. The new strategy employs a randomised singular value decomposition and retains the potential for parallelism in the time domain. Numerical results for the Lorenz 96 model show that this approach accelerates the minimisation in the first few iterations, with better results when CVT performs well.

6.2 Introduction

The ever increasing resolution of weather models enhances the importance of parallelisation in data assimilation. Higher potential for parallel computations can be achieved by using suitable data assimilation methods. The state formulation of the weak constraint 4D-Var method, which allows for the model error, is such a method. In its incremental version, a series of quadratic cost functions is minimised via solving a series of linear systems containing the Hessian of the linearised cost function. These are solved with the conjugate gradient (CG) method (e.g., [Saad, 2003]), where the most computationally expensive part is integrating the tangent linear model and its adjoint. It has been shown that these calculations can be parallelised in the time dimension [Fisher and Gürol, 2017].

However, CG needs preconditioning for fast convergence. Efficient preconditioning for the state formulation of incremental weak constraint 4D-Var, which also preserves the potential for parallel in time calculations, is still an open question. By analogy with the standard preconditioning technique (also known as a control variable transform or first level preconditioning) used in strong constraint 4D-Var, [Fisher and Gürol, 2017] suggested using approximations of the tangent linear model. Their search for a suitable approximation was unsuccessful. Our investigation in this paper reveals that preconditioning using the exact tangent linear model can be detrimental to the minimisation in some cases. We focus on approximations in the case when using the exact tangent linear model works well.

In the light of the growing popularity of randomised methods and examples of their use in data assimilation [Bousserez et al., 2020, Daužickaitė et al., 2021b], we propose using a randomised singular value decomposition (RSVD) [Halko et al., 2011] to approximate the tangent linear model. RSVD is a block method that is easy to parallelise in the sense that it requires calculating matrix products with blocks of vectors. Because [Lawless et al., 2008] showed that it is important to take into account the information on the background errors when using model reduction techniques in data assimilation, we also examine an approach where we approximate the tangent linear model in interaction with the background and model error covariance matrices.

We formulate the incremental weak constraint 4D-Var problem and discuss its preconditioning in Section 6.3. Our idea for randomised preconditioning is presented in Section 6.4. Numerical experiments exploring preconditioning using the exact tangent linear model and its low-rank approximation obtained using RSVD are presented in Section 6.5 and we summarize our findings and suggest future directions in Section 6.6.

6.3 Incremental weak constraint 4D-Var

In data assimilation, the prior estimate of a model trajectory is combined with observations over a time window to obtain an improved estimate of the state (analysis) $\mathbf{x}_0^a, \mathbf{x}_1^a, \ldots, \mathbf{x}_N^a$ at times $t_0, t_1, \ldots, t_N$. The prior estimate of the state at t_0 is called the background and is denoted by $\mathbf{x}^b \in \mathbb{R}^n$ and the observations at time t_i are denoted by $\mathbf{y}_i \in \mathbb{R}^{q_i}$. The state variables $\mathbf{x}_i$ are mapped to the observation space using an observation operator $\mathcal{H}_i$. The

nonlinear dynamical model $\mathcal{M}_i$ describes the state evolution from time t_i to t_{i+1}. It is assumed that the background, observations, and model have Gaussian errors with zero mean and covariance matrices $\mathbf{B} \in \mathbb{R}^{n\times n}$, $\mathbf{R}_i \in \mathbb{R}^{q_i\times q_i}$, and $\mathbf{Q}_i \in \mathbb{R}^{n\times n}$, respectively. We assume that the observation and model errors are uncorrelated in time.

In the state formulation of weak constraint 4D-Var, the analysis is the minimiser of the nonlinear cost function

$$J(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{i=0}^{N} \|\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)\|^2_{\mathbf{R}_i^{-1}} + \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i)\|^2_{\mathbf{Q}_{i+1}^{-1}}, \tag{6.1}$$

where $\|\mathbf{a}\|^2_{\mathbf{A}} = \mathbf{a}^T\mathbf{A}\mathbf{a}$ [Trémolet, 2006].

The minimiser of (6.1) can be approximated using an inexact Gauss-Newton algorithm [Gratton et al., 2007]. In this incremental approach, the $(j+1)$th approximation $\mathbf{x}^{(j+1)} = (\mathbf{x}_0^{(j+1)T}, \mathbf{x}_1^{(j+1)T}, \ldots, \mathbf{x}_N^{(j+1)T})^T \in \mathbb{R}^{(N+1)n}$ of the state is

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} + \delta\mathbf{x}^{(j)}, \tag{6.2}$$

where the update is $\delta\mathbf{x}^{(j)} = (\delta\mathbf{x}_0^{(j)T}, \delta\mathbf{x}_1^{(j)T}, \ldots, \delta\mathbf{x}_N^{(j)T})^T \in \mathbb{R}^{(N+1)n}$. $\mathbf{M}_i$ and $\mathbf{H}_i$ are the model and observation operators linearised at $\mathbf{x}_i$; they are known as the tangent linear model and tangent linear observation operator, respectively. We define the following matrices (following [Gratton et al., 2018b])

$$\mathbf{L}^{(j)} = \begin{pmatrix} \mathbf{I} & & & & \\ -\mathbf{M}_0^{(j)} & \mathbf{I} & & & \\ & -\mathbf{M}_1^{(j)} & \mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{M}_{N-1}^{(j)} & \mathbf{I} \end{pmatrix} \in \mathbb{R}^{(N+1)n\times(N+1)n}, \tag{6.3}$$

$$\mathbf{H}^{(j)} = diag(\mathbf{H}_0^{(j)}, \mathbf{H}_1^{(j)}, \ldots, \mathbf{H}_N^{(j)}) \in \mathbb{R}^{q\times(N+1)n}, \tag{6.4}$$

$$\mathbf{D} = diag(\mathbf{B}, \mathbf{Q}_1, \ldots, \mathbf{Q}_N) \in \mathbb{R}^{(N+1)n\times(N+1)n}, \tag{6.5}$$

$$\mathbf{R} = diag(\mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_N) \in \mathbb{R}^{q\times q}, \tag{6.6}$$

where $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, $diag(\cdot)$ denotes a block diagonal matrix and $q = \Sigma_{i=0}^{N} q_i$ is the total number of observations. We use the following notation for vectors

$$\mathbf{b}^{(j)} = \begin{pmatrix} \mathbf{x}^b - \mathbf{x}_0^{(j)} \\ \mathcal{M}_0(\mathbf{x}_0^{(j)}) - \mathbf{x}_1^{(j)} \\ \vdots \\ \mathcal{M}_{N-1}(\mathbf{x}_{N-1}^{(j)}) - \mathbf{x}_N^{(j)} \end{pmatrix} \in \mathbb{R}^{(N+1)n}, \tag{6.7}$$

$$\mathbf{d}^{(j)} = \begin{pmatrix} \mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0^{(j)}) \\ \mathbf{y}_1 - \mathcal{H}_1(\mathbf{x}_1^{(j)}) \\ \vdots \\ \mathbf{y}_N - \mathcal{H}_N(\mathbf{x}_N^{(j)}) \end{pmatrix} \in \mathbb{R}^{q}. \tag{6.8}$$

The update $\delta\mathbf{x}^{(j)}$ is the minimiser of

$$J^{\delta}(\delta\mathbf{x}^{(j)}) = \frac{1}{2}||\mathbf{L}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{b}^{(j)}||^2_{\mathbf{D}^{-1}} + \frac{1}{2}||\mathbf{H}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{d}^{(j)}||^2_{\mathbf{R}^{-1}}. \tag{6.9}$$

Because (6.9) is a quadratic cost function, $\delta\mathbf{x}^{(j)}$ can be found by solving the following large linear systems with the Hessian $\mathbf{A}^{(j)}$ of $J^{\delta}(\delta\mathbf{x}^{(j)})$:

$$\mathbf{A}^{(j)}\delta\mathbf{x}^{(j)} = (\mathbf{L}^T)^{(j)}\mathbf{D}^{-1}\mathbf{b}^{(j)} + (\mathbf{H}^T)^{(j)}\mathbf{R}^{-1}\mathbf{d}^{(j)}, \tag{6.10}$$

$$\text{where } \mathbf{A}^{(j)} = (\mathbf{L}^T)^{(j)}\mathbf{D}^{-1}\mathbf{L}^{(j)} + (\mathbf{H}^T)^{(j)}\mathbf{R}^{-1}\mathbf{H}^{(j)}. \tag{6.11}$$

It is assumed that $q \ll (N+1)n$, thus $(\mathbf{H}^T)^{(j)}\mathbf{R}^{-1}\mathbf{H}^{(j)}$ is symmetric positive semi-definite. Because $(\mathbf{L}^T)^{(j)}\mathbf{D}^{-1}\mathbf{L}^{(j)}$ is symmetric positive definite, $\mathbf{A}^{(j)} \in \mathbb{R}^{(N+1)n\times(N+1)n}$ is symmetric positive definite. Hence the method of choice for solving (6.10) is CG. Each iteration of CG requires one matrix-vector product with $\mathbf{A}^{(j)}$, which is expensive due to the tangent linear model and its adjoint in $\mathbf{L}^{(j)}$ and $(\mathbf{L}^T)^{(j)}$, respectively. [Fisher and Gürol, 2017] noted that the structure of $\mathbf{L}^{(j)}$ allows the matrix-vector products with $\mathbf{A}^{(j)}$ to be parallelised in the time dimension, that is, computation of $\mathbf{L}^{(j)}\mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^{(N+1)n}$, can be parallelised for the model linearised at different times. In the rest of this paper, the superscript (j) is omitted for ease of notation.

In general, CG needs preconditioning for fast convergence. Efficient preconditioning maps the system to another system that can be solved faster and the solution of the original problem can be easily recovered from the solution of the preconditioned problem. Choosing a suitable preconditioner is highly problem dependent. Given the possibility of parallel computations in matrix-vector products with (6.11), the preconditioner should keep this potential.

6.3.1 Preconditioning

We consider an extension of the control variable transform, also known as a first level preconditioning, that is used in 3D-Var, where the model evolution is omitted, and in the strong constraint formulation of 4D-Var, where the model is assumed to have no error (see, e.g., [Lorenc et al., 2000, Rawlins et al., 2007, Lawless, 2013]). The idea is to apply the preconditioner so that the first term of the preconditioned Hessian is equal to identity. Then the preconditioned Hessian is a sum of the identity matrix and a low-rank symmetric positive semi-definite matrix with rank at most q. Its smallest eigenvalue is equal to one and it has at most q eigenvalues that are larger than one. The latter can impair CG convergence if they are not well separated (for a more general discussion see, for example, [Nocedal and Wright, 2006, Liesen and Strakoš, 2013]).

Applying this kind of preconditioning to the state formulation of weak constraint 4D-

Var requires preconditioning with $\mathbf{L}^{-1}\mathbf{D}^{1/2}$, where

$$\mathbf{L}^{-1} = \begin{pmatrix} \mathbf{I} & & & & \\ \mathbf{M}_{0,0} & \mathbf{I} & & & \\ \mathbf{M}_{0,1} & \mathbf{M}_{1,1} & \mathbf{I} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{M}_{0,N-1} & \mathbf{M}_{1,N-1} & \cdots & \mathbf{M}_{N-1,N-1} & \mathbf{I} \end{pmatrix} \tag{6.12}$$

and $\mathbf{M}_{i,j} = \mathbf{M}_j \dots \mathbf{M}_i$ denotes the linearised model integration from time t_i to t_{j+1}. Matrix-vector products with $\mathbf{L}^{-1}$ are sequential in the time dimension, i.e.,

$$\mathbf{L}^{-1}\mathbf{z} = \begin{pmatrix} \mathbf{z}_0 \\ \mathbf{M}_0\mathbf{z}_0 + \mathbf{z}_1 \\ \mathbf{M}_1(\mathbf{M}_0\mathbf{z}_0 + \mathbf{z}_1) + \mathbf{z}_2 \\ \vdots \\ \mathbf{M}_{N-1}(\mathbf{M}_{N-2}\dots\mathbf{M}_0\mathbf{z}_0 + \mathbf{M}_{N-2}\dots\mathbf{M}_1\mathbf{z}_1 + \cdots + \mathbf{z}_{N-1}) + \mathbf{z}_N \end{pmatrix}, \tag{6.13}$$

where $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T, \dots, \mathbf{z}_N^T)^T$. [Fisher and Gürol, 2017] suggested using an approximation $\widetilde{\mathbf{L}}^{-1}$ of $\mathbf{L}^{-1}$ in the preconditioner. Then the preconditioned system to be solved is

$$\mathbf{A}^{pr}\delta\widetilde{\mathbf{x}} = \mathbf{D}^{1/2}\widetilde{\mathbf{L}}^{-T}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{b} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}), \tag{6.14}$$

$$\text{where} \qquad \mathbf{A}^{pr} = \mathbf{D}^{1/2}\widetilde{\mathbf{L}}^{-T}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}, \tag{6.15}$$

$$\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}\delta\widetilde{\mathbf{x}} = \delta\mathbf{x}. \tag{6.16}$$

With an appropriate choice of $\widetilde{\mathbf{L}}^{-1}$, $\mathbf{A}^{pr}$ is symmetric positive definite. $\widetilde{\mathbf{L}}^{-1}$ should be chosen so that it can be applied in parallel. Fisher and Gürol could not find a suitable approximation that would guarantee good convergence. [Gratton et al., 2018a, Gratton et al., 2018b] discussed using $\widetilde{\mathbf{L}}^{-1}$ where $\mathbf{M}_i$ is set to zero or to the identity matrix in (6.12), which may be useful if the model state does not change significantly from one time step to the next. This may be unrealistic. We propose a new approximation strategy that avoids this assumption in the next section.

6.4 Randomised preconditioning

Randomised methods for low-rank matrix approximations have attracted a lot of interest in recent years because they require matrix products with blocks of vectors that can be easily parallelised and it has been shown that good approximations for matrices with rapidly decaying singular values can be obtained with high probability (e.g., [Halko et al., 2011], [Martinsson and Tropp, 2020]). These methods have been explored in data assimilation when designing solvers for strong constraint 4D-Var [Bousserez et al., 2020] and preconditioning for the forcing formulation of the incremental weak constraint 4D-Var [Daužickaitė et al., 2021b].

A low-rank approximation of $\mathbf{L}^{-1}$ cannot be used in (6.14), because it would make (6.15) low-rank and thus singular. Hence, we suggest exploiting the structure of $\mathbf{L}^{-1}$ when generating the preconditioner. We write

$$\mathbf{L}^{-1} = \mathbf{I} + \mathbf{P}, \tag{6.17}$$

where $\mathbf{P}$ is a strictly lower triangular matrix (with 0 on the diagonal). We propose using a rank k approximation $\widetilde{\mathbf{P}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where k is small compared to $(N+1)n$ and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{(N+1)n \times k}$, $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ is a truncated singular value decomposition, that is, $\mathbf{\Sigma}$ is diagonal with approximations to the k largest singular values of $\mathbf{P}$ on the diagonal, and the columns of $\mathbf{U}$ and $\mathbf{V}$ are approximate left and right singular vectors, respectively. Then a nonsingular $\widetilde{\mathbf{L}}^{-1}$ is

$$\widetilde{\mathbf{L}}^{-1} = \mathbf{I} + \widetilde{\mathbf{P}} = \mathbf{I} + \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{6.18}$$

An RSVD can be used to obtain $\widetilde{\mathbf{P}}$. RSVD is essentially one iteration of a classic subspace iteration method [Gu, 2015]. To increase the probability of success, the randomised methods work with larger subspaces than the required rank of the approximation. This is called oversampling. [Halko et al., 2011] indicate that setting the oversampling parameter l to five or ten generally gives good results. We present the RSVD in Algorithm 12. The entries of the Gaussian random matrix are independent standard normal random variables. Note that we remove the smallest l computed singular values and the corresponding singular vectors. In this way the oversampling increases the cost of generating the preconditioner (in particular, the cost of the matrix-matrix products in steps 2 and 4), but not of its application. RSVD needs two expensive matrix-matrix products with $\mathbf{P}$ (steps 2 and 4 in Algorithm 12), where $\mathbf{P}$ is multiplied with a matrix of size $(N+1)n \times (k+l)$. For efficiency, these can be parallelised. These matrix-matrix products consist of products with $\mathbf{M}_{i,j}$. Hence, the cost of generating the preconditioner depends on the cost of integrating the tangent linear model over the assimilation window sequentially.

[Lawless et al., 2008] showed that including the background error covariance matrix when using model reduction methods may lead to better results. Hence, we also explore using an approximation of

$$\mathbf{S} = \mathbf{L}^{-1}\mathbf{D}^{1/2} = \begin{pmatrix} \mathbf{B}^{1/2} & & & & \\ \mathbf{M}_{0,0}\mathbf{B}^{1/2} & \mathbf{Q}_1^{1/2} & & & \\ \mathbf{M}_{0,1}\mathbf{B}^{1/2} & \mathbf{M}_{1,1}\mathbf{Q}_1^{1/2} & \mathbf{Q}_2^{1/2} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{M}_{0,N-1}\mathbf{B}^{1/2} & \mathbf{M}_{1,N-1}\mathbf{Q}_1^{1/2} & \cdots & \mathbf{M}_{N-1,N-1}\mathbf{Q}_{N-1}^{1/2} & \mathbf{Q}_N^{1/2} \end{pmatrix}. \tag{6.19}$$

As when approximating $\mathbf{L}^{-1}$, we write

$$\mathbf{S} = \mathbf{D}^{1/2} + \mathbf{W}, \tag{6.20}$$

where $\mathbf{W}$ is strictly lower triangular. Approximation $\widetilde{\mathbf{S}} = \mathbf{D}^{1/2} + \widetilde{\mathbf{W}}$ can be obtained by using RSVD to generate a low-rank approximation of $\mathbf{W}$. The system to be solved is (6.14) with $\widetilde{\mathbf{L}}^{-1}\mathbf{D}^{1/2}$ replaced by $\widetilde{\mathbf{S}}$.

Algorithm 12 Randomised singular value decomposition (RSVD)

Input: matrix $\mathbf{A} \in \mathbb{R}^{s\times s}$, target rank k, an oversampling parameter l

Output: orthogonal $\mathbf{U} \in \mathbb{R}^{s\times k}$ and $\mathbf{V} \in \mathbb{R}^{s\times k}$ whose columns are approximations to left and right singular vectors of $\mathbf{A}$, respectively, and diagonal $\mathbf{\Sigma} \in \mathbb{R}^{k\times k}$ with approximations to the largest singular values of $\mathbf{A}$

1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{s\times(k+l)}$
2: Form a sample matrix $\mathbf{Y} = \mathbf{AG} \in \mathbb{R}^{s\times(k+l)}$
3: Orthonormalize the columns of $\mathbf{Y}$ to obtain orthonormal $\mathbf{Z} \in \mathbb{R}^{s\times(k+l)}$
4: Form $\mathbf{K} = \mathbf{Z}^T\mathbf{A} \in \mathbb{R}^{(k+l)\times s}$
5: Form SVD of $\mathbf{K}$: $\mathbf{K} = \hat{\mathbf{U}}\mathbf{\Sigma}\mathbf{V}^T$, where $\hat{\mathbf{U}}$, $\mathbf{\Sigma} \in \mathbb{R}^{(k+l)\times(k+l)}$, $\mathbf{V} \in \mathbb{R}^{s\times(k+l)}$
6: Remove last l columns and rows of $\mathbf{\Sigma}$, so that $\mathbf{\Sigma} \in \mathbb{R}^{k\times k}$
7: Remove last l columns of $\hat{\mathbf{U}}$ and $\mathbf{V}$, so that $\hat{\mathbf{U}} \in \mathbb{R}^{(k+l)\times k}$, $\mathbf{V} \in \mathbb{R}^{s\times k}$
8: Form $\mathbf{U} = \mathbf{Z}\hat{\mathbf{U}} \in \mathbb{R}^{s\times k}$.

6.5 Numerical results

We test preconditioning using $\mathbf{L}^{-1}$ and the approximations $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ in (6.14) numerically. Preconditioning using the exact $\mathbf{L}^{-1}$ is considered so that we understand when preconditioning using $\widetilde{\mathbf{L}}^{-1}$ or $\widetilde{\mathbf{S}}$ may be effective but this is not regarded as a practical approach when parallelisation in the time dimension is desired. Identical twin experiments are performed. The background state $\mathbf{x}^b$ is generated by adding random, Gaussian noise with covariance $\mathbf{B}$ to $\mathbf{x}_0^t$, where $\mathbf{x}_i^t$ is the reference state at time t_i. We use direct observations that are obtained by adding random, Gaussian noise with covariance $\mathbf{R}_i$ to $\mathcal{H}_i(\mathbf{x}_i^t)$.

The nonlinear Lorenz 96 model [Lorenz, 1996] is used, where the dynamics of $\mathbf{x}_i = (X^1, \ldots, X^n)^T$ are described by a set of n coupled ODEs:

$$\frac{dX^j}{dt} = -X^{j-2}X^{j-1} + X^{j-1}X^{j+1} - X^j + F \tag{6.21}$$

with conditions $X^{-1} = X^{n-1}, X^0 = X^n$ and $X^{n+1} = X^1$ and $F = 8$. We use a fourth order Runge-Kutta scheme [Butcher, 1987]. We consider the system with $n = 100$ and $N = 149$, so $\mathbf{A}^{pr}$ is a 15000×15000 matrix. The time step is set to $\Delta t = 2.5 \times 10^{-2}$ and the grid point distance is $\Delta X = 1/n$.

The covariance matrices are $\mathbf{B} = 0.2^2\mathbf{C}_b$, $\mathbf{Q}_i = 0.05^2\mathbf{C}_q$, where $\mathbf{C}_b$ is a second-order auto-regressive (SOAR) [Daley, 1993] matrix and $\mathbf{C}_q$ is a Laplacian [Johnson et al., 2005] correlation matrix with length scales $2\Delta X$ and $0.75\Delta X$, respectively. We consider $\mathbf{R}_i = \sigma_o^2\mathbf{I}$ and vary σ_o.

The computations are performed with Matlab R2019b and the linear systems are solved with the Matlab preconditioned conjugate gradient (PCG) implementation *pcg*.

6.5.1 Preconditioning with exact $\mathbf{L}^{-1}$

We have noticed that the effectiveness of the exact preconditioner $\mathbf{L}^{-1}$ depends on how much of the system is observed and the interaction between the model and observation

errors. There are observations at every tenth time step, ensuring that there are observations at the final time. We consider the following cases regarding the observation error variance σ_o and the total number of observations q:

1. $\sigma_o = 1.5 \times 10^{-1}$, $q = 300$ (observing 2% of the system);
2. $\sigma_o = 4.5 \times 10^{-1}$, $q = 300$;
3. $\sigma_o = 1.5 \times 10^{-1}$, $q = 60$ (observing 0.4% of the system).

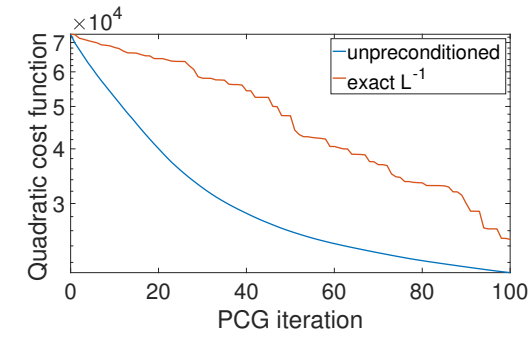
In Figure 6.1, we show that preconditioning using $\mathbf{L}^{-1}$ is not useful in case 1 but can be effective if the observation error variance is increased while keeping the same number of observations (case 2), or if the number of observations is reduced while σ_o is unchanged (case 3).

Note that we compare the value of the quadratic cost function at every PCG iteration without taking into account the cost of the computation, which can be evaluated in terms of runtime or energy consumption and depends on how much parallelism can be achieved (e.g., [Carson and Strakoš, 2020]). If matrix-vector products with $\mathbf{L}$ can be parallelised, then PCG iterations when solving the unpreconditioned system can be performed faster than with preconditioning. Then, in terms of the runtime, preconditioning in case 1 can be even worse than indicated by comparing the quadratic cost function at every PCG iteration. In the same manner, preconditioning using exact $\mathbf{L}^{-1}$ in cases 2 and 3 may not be as effective as displayed. In the following section, we test preconditioning using $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ in cases 2 and 3.

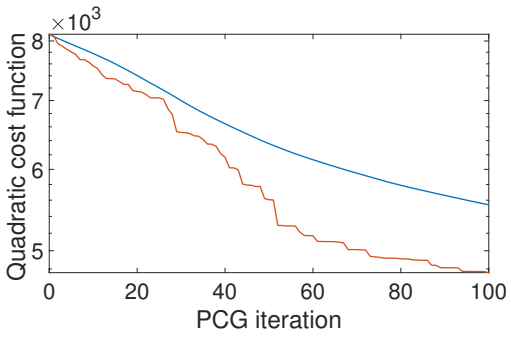
6.5.2 Preconditioning with randomised low-rank approximation

We generate $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ by using rank $k \in \{30, 60, 90\}$ approximations of $\mathbf{P}$ and $\mathbf{W}$ in (6.17) and (6.20), respectively. The oversampling parameter is set to $l = 5$. We found that using $l = 10$ or $l = 15$ does not make a significant difference to the results (not shown). RSVD produces high quality approximations of the singular values of both $\mathbf{P}$ and $\mathbf{W}$. The largest singular values and their approximations are shown in Figure 6.2, where the same random seed is used to generate the random matrix $\mathbf{G}$ for all k values. Matrices $\mathbf{P}$ and $\mathbf{W}$ do not depend on whether case 2 or 3 is considered, because the cases differ in the observation terms. In each case, we run the RSVD algorithm one hundred times with different Gaussian matrices $\mathbf{G}$ and solve the systems with the resulting preconditioners. The spread is illustrated in Figure 6.3 for $\widetilde{\mathbf{S}}$ (see the appendix in Section 6.7 for $\widetilde{\mathbf{P}}$). In both cases, the variation in the values of the cost function is small during the early iterations. This shows that our results are not very sensitive to the choice of $\mathbf{G}$ and, in practice, it is only necessary to run the RSVD algorithm once.

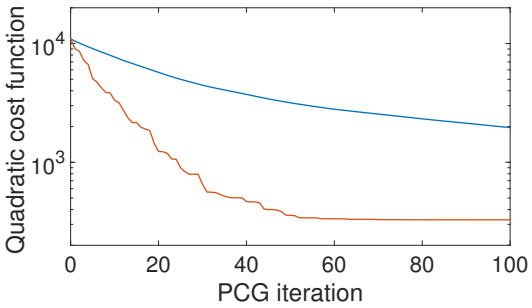
The means of the quadratic cost function in cases 2 and 3 are shown in Figure 6.4. Higher rank approximations in both cases and using $\widetilde{\mathbf{S}}$ in case 3 results in faster minimisation. Notice that in the first few iterations of PCG, preconditioning gives the same improvement regardless of the rank of approximation and whether $\widetilde{\mathbf{L}}^{-1}$ or $\widetilde{\mathbf{S}}$ is used. Preconditioning is more useful in case 3, which has fewer observations. The approximations



(a) Case 1

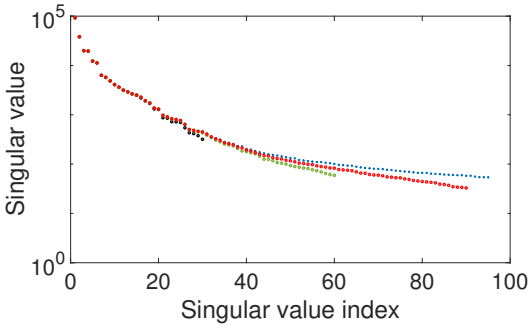


(b) Case 2

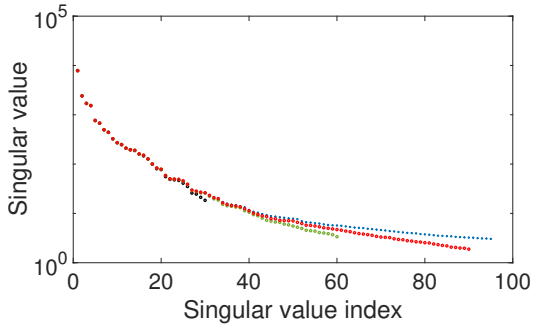


(c) Case 3

Figure 6.1: The values of the quadratic cost functions at every PCG iteration when using no preconditioner and preconditioning using exact $\mathbf{L}^{-1}$. Values of σ_o and the number of observations q for cases 1, 2, and 3 are given in the text.



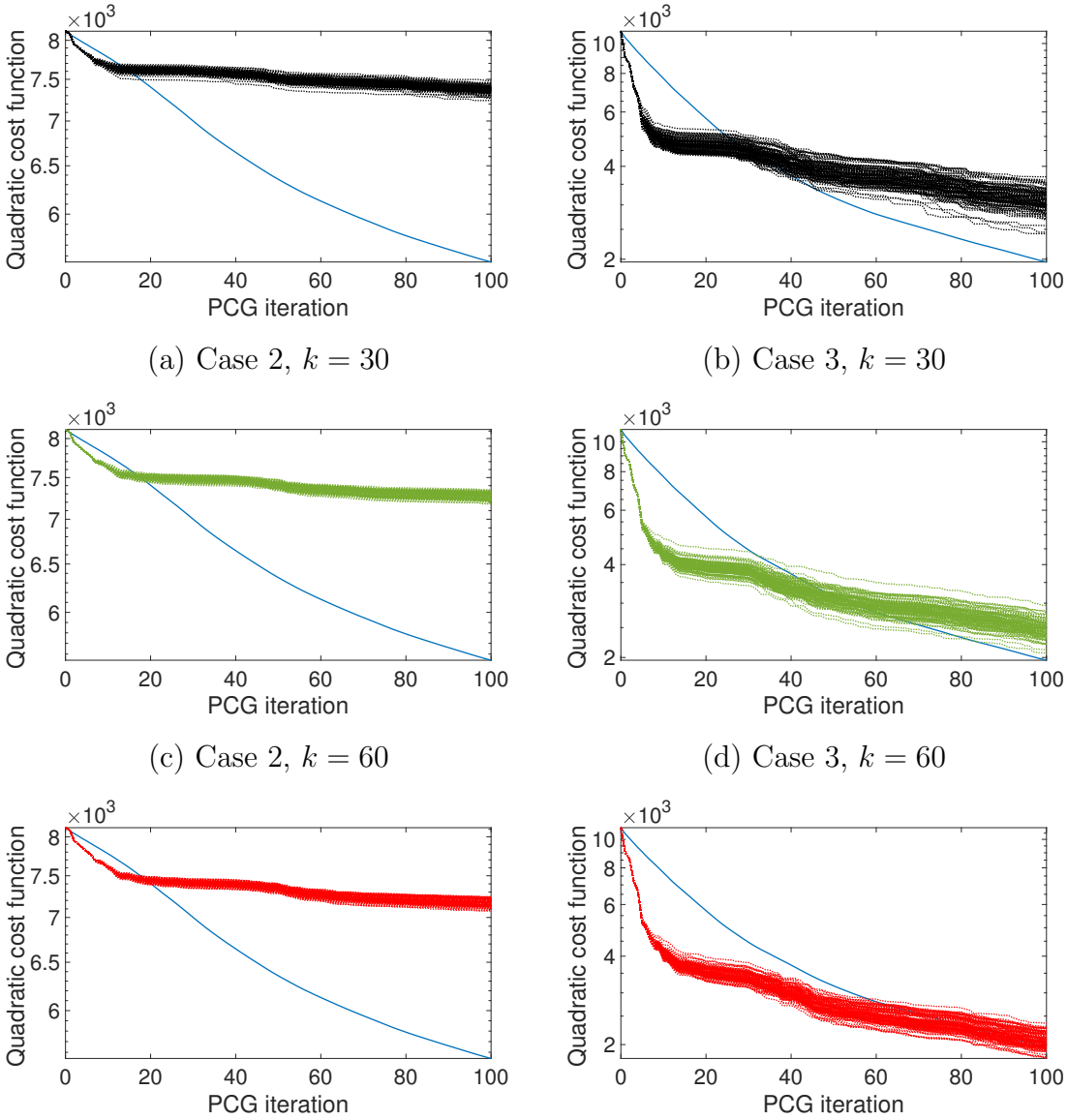
(a)



(b)

Figure 6.2: Largest singular values of (a) $\mathbf{P}$ (blue) and (b) $\mathbf{W}$ (blue) and their approximations given by RSVD when using rank $k = 30$ (black), $k = 60$ (green) and $k = 90$ (red). The largest singular values and their approximations coincide.

used to generate $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ are very low-rank compared to the size of the system and there is a good improvement over the unpreconditioned case when the number of observations is low, especially in the beginning of the iterative process, which is the most relevant in practical settings. In the case with more observations (case 2), the randomised preconditioning is useful if a small number of PCG iterations is run. Since in an operational



(a) Case 2, $k = 30$

(b) Case 3, $k = 30$

(c) Case 2, $k = 60$

(d) Case 3, $k = 60$

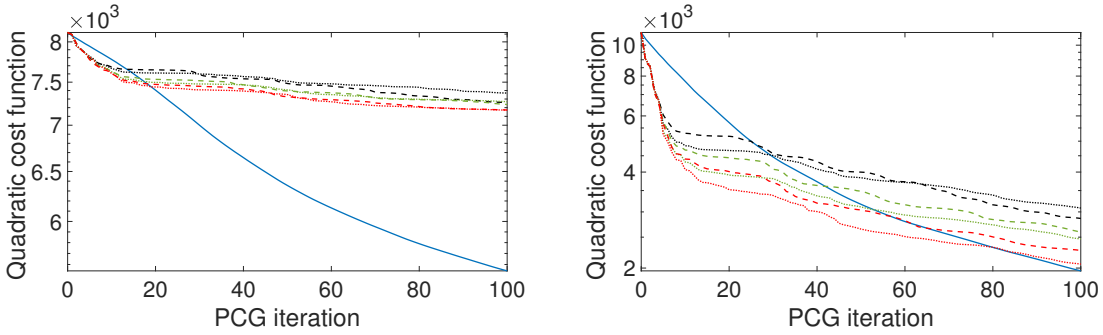
(e) Case 2, $k = 90$

(f) Case 3, $k = 90$

Figure 6.3: Values of the quadratic cost function at every PCG iteration when using no preconditioner (blue solid line) and preconditioning using $\widetilde{\mathbf{S}}$ (dotted lines) that are constructed using rank $k \in \{30, 60, 90\}$ approximation. One hundred realisations of the randomised preconditioner are shown. Values of σ_o and the number of observations q in cases 2 and 3 are given in text.

context we only run a small number of iterations, we are more likely to be in this regime. In cases 2 and 3, using exact $\mathbf{L}^{-1}$ results in a modest (case 2) and a rapid (case 3) decrease of the cost function in the first PCG iterations (Figure 6.1). Our proposed preconditioners replicate such behaviour and if larger k is used then the performance of exact $\mathbf{L}^{-1}$ is followed for more PCG iterations. In case 3, the quadratic cost function value is reduced by a factor of two after five PCG iterations when using exact $\mathbf{L}^{-1}$ in the preconditioner, the same result is obtained after eight ($k = 30$) and six ($k = 60$ and $k = 90$) PCG iterations using $\widetilde{\mathbf{L}}^{-1}$, and six ($k = 30$) and five ($k = 60$ and $k = 90$) PCG iterations using $\widetilde{\mathbf{S}}$. In

case 2, the quadratic cost function is reduced only by a factor of 1.7 in one hundred PCG iterations when preconditioning with the exact $\mathbf{L}^{-1}$. When using our preconditioners the values of the quadratic cost function after one hundred PCG iterations are larger than when using exact $\mathbf{L}^{-1}$ or no preconditioning. This can be addressed by using a larger rank approximation, computational resources permitting.



(a) Case 2 (b) Case 3

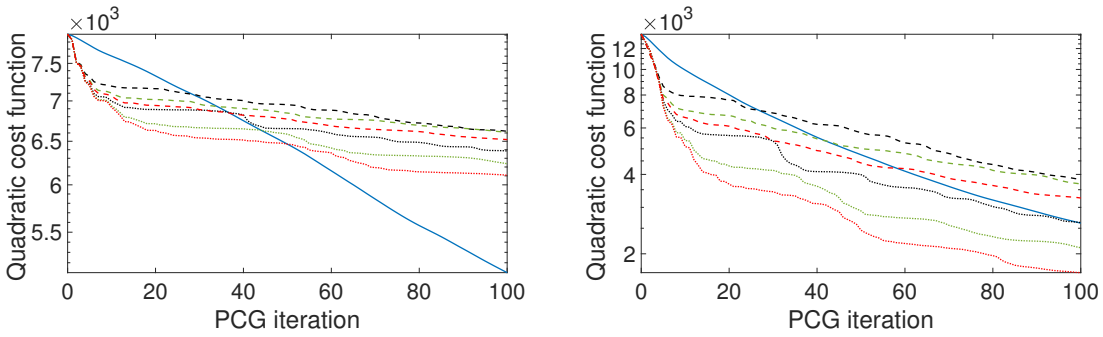
Figure 6.4: Mean values (over one hundred experiments) of the quadratic cost function at every PCG iteration when using no preconditioner (blue solid line) and when preconditioning using $\widetilde{\mathbf{L}}^{-1}$ (dashed) and $\widetilde{\mathbf{S}}$ (dotted) that are constructed using rank $k = 30$ (black), $k = 60$ (green) and $k = 90$ (red) approximation. Values of σ_o and the number of observations q in cases 2 and 3 are given in text.

Large model error

We explore how the preconditioning using approximations of $\mathbf{L}^{-1}$ and $\mathbf{L}^{-1}\mathbf{D}^{1/2}$ compare when the model error is large. The numerical experiments are performed using the same setup as before, but now we set $\mathbf{Q}_i = 0.1^2\mathbf{C}_q$, where $\mathbf{C}_q$ has length scale $2\Delta X$. The means over one hundred runs are presented in Figure 6.5. There is a clear separation between the minimisation using $\widetilde{\mathbf{L}}^{-1}$ and $\widetilde{\mathbf{S}}$ in the preconditioner after the first few PCG iterations, with the latter resulting in faster minimisation. Notice that the preconditioning using both approximations remains useful for more PCG iterations than in the setup with a smaller model error. This can be expected because the increase of length scales of $\mathbf{Q}_i$ has a detrimental effect on the conditioning of the unpreconditioned Hessian (see, e.g., Chapter 6 of [El-Said, 2015]) and hence preconditioning can be more efficient.

6.6 Conclusions

We have considered preconditioning for the state formulation of incremental weak constraint 4D-Var, which closely follows the control variable transform (first level preconditioning) strategy for the strong constraint formulation. We have shown that such preconditioning may not be useful even when using the exact $\mathbf{L}^{-1}$, which also makes the matrix-vector products with the Hessian sequential in the time dimension. In the cases



(a) Case 2 (b) Case 3

Figure 6.5: As in Figure 6.4, but the model error covariance matrix is $\mathbf{Q}_i = 0.1^2\mathbf{C}_q$ and $\mathbf{C}_q$ has length scale $2\Delta X$.

where such preconditioning is useful, a good preconditioner can be obtained by using randomised singular value decompositions to approximate $\mathbf{L}^{-1}$ or $\mathbf{L}^{-1}\mathbf{D}^{1/2}$. These preconditioners are cheap to compute and apply and do allow for parallelization in the time dimension. They can improve the solution of the exact inner loop problem, resulting in a greater reduction of the quadratic cost function in the same number of iterations compared to using no preconditioning or obtaining the same quadratic cost function value in fewer iterations. The effect of the accuracy of the inner loop solution on the analysis has been studied by, for example, [Lawless and Nichols, 2006].

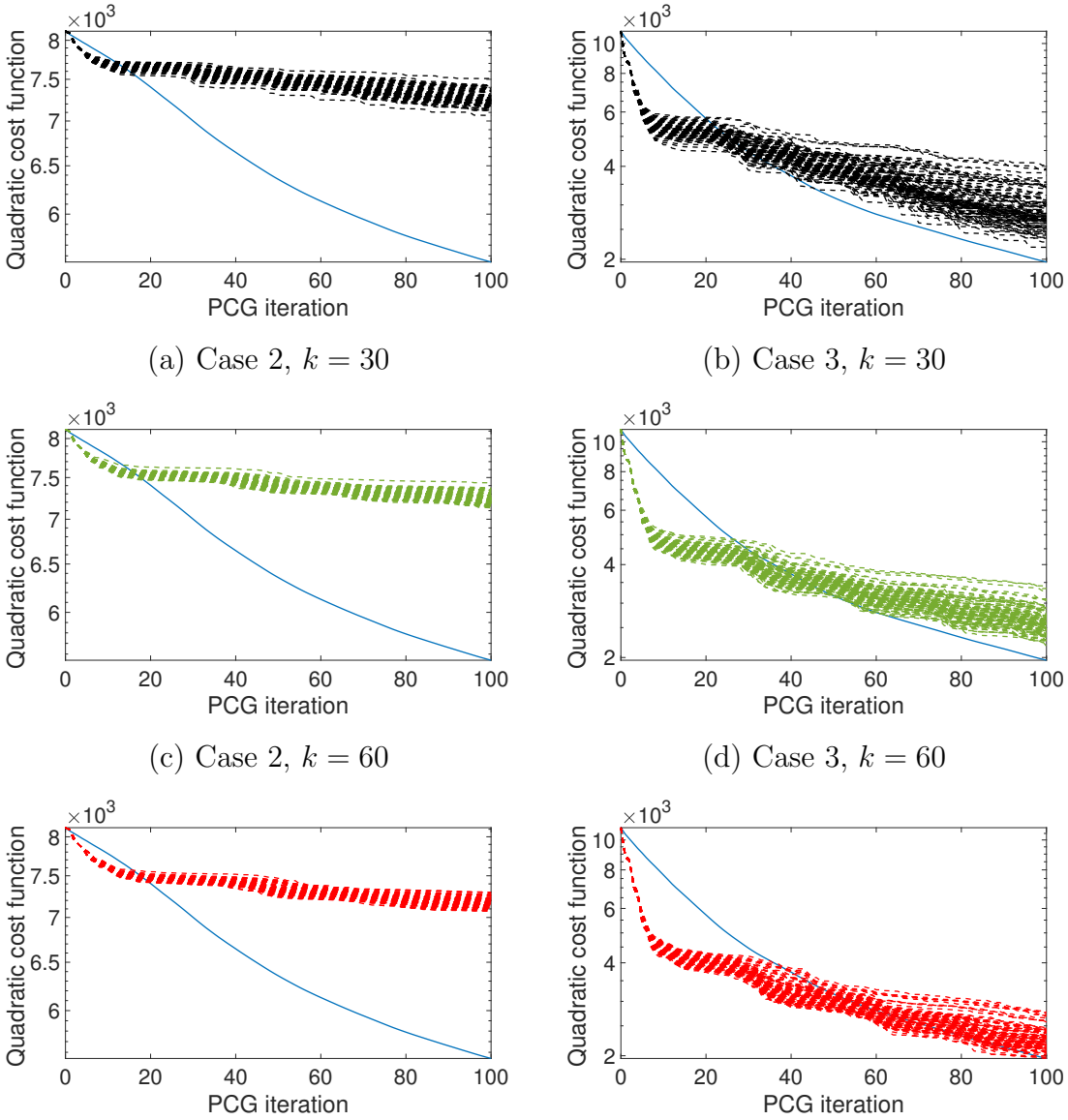
Our results call for caution when designing preconditioning approaches that focus on approximating $\mathbf{L}^{-1}$, especially when the number of observations is high. In practical NWP settings, around 1% of the system is observed, hence approximating $\mathbf{L}^{-1}$ may be useful. Using randomised approximations of $\mathbf{L}^{-1}$ or $\mathbf{L}^{-1}\mathbf{D}^{1/2}$ should be tested using large and more realistic systems, where meaningful evaluations of the runtime and energy consumption can be obtained. A more detailed investigation on when preconditioning with $\mathbf{L}^{-1}$ gives good results would also be useful.

Acknowledgements

We thank Dr. Adam El-Said for his code for the weak constraint 4D-Var assimilation system. We are also grateful for the helpful comments by two anonymous reviewers.

6.7 Appendix: Spread when using $\widetilde{\mathbf{P}}$

We report the spread of one hundred runs when the preconditioner is constructed using $\widetilde{\mathbf{P}}$ (Figure 6.6). The spread in the later iterations is larger than when using $\widetilde{\mathbf{S}}$ (Figure 6.3).



(a) Case 2, $k = 30$

(b) Case 3, $k = 30$

(c) Case 2, $k = 60$

(d) Case 3, $k = 60$

(e) Case 2, $k = 90$

(f) Case 3, $k = 90$

Figure 6.6: As in Figure 6.3, but preconditioning uses $\widetilde{\mathbf{P}}$ instead of $\widetilde{\mathbf{S}}$.

6.8 Summary

We proposed a time-parallel preconditioning strategy for the SPD system arising in the state formulation. It is an approximation of the CVT technique which involves approximating the matrix containing the linearised model. We showed that a preconditioner constructed using the randomised singular value decomposition is useful in the first iterations of CG, which are of the most importance in practical settings. The preconditioner can also be constructed taking into account the model and background errors; this is especially useful if the model error is large. Numerical experiments demonstrated that using the exact CVT technique is not always useful. The effectiveness depends on the number of observations and their error, that is how much influence the observations have; the unpreconditioned state formulation is easier to solve when there are many observations or

when their error is smaller. This may relate to the results in Chapter 5 that show that the eigenvalues of the state SPD matrix can move away from zero when new observations are added and hence the system may become easier to solve. Our preconditioner is thus of interest when a small part of the dynamical system is observed.

In the next chapter we consider how the observation information can be included in the approximation of the inverse Schur complement when preconditioning the saddle point systems.

Chapter 7

Preconditioning for the saddle point systems

In this chapter, we address the research question 4. We precondition the saddle point systems using the block diagonal Schur preconditioner. We investigate the eigenvalues of the preconditioned saddle point matrices, particularly how they relate to the eigenvalues of the SPD matrices in the state and forcing formulations, and how they change when new observations are added, thus touching upon the research question 2. An answer to this research question when considering the SPD system with uncorrelated observation errors in the forcing formulation is also provided.

The required approximation of the inverse Schur complement in the block diagonal preconditioner are constructed using randomised LMPs. We examine if this is useful compared to using no preconditioner and other types of model approximations in the Schur complement where the observation term is discarded. If yes, does the usefulness depend on the number of observations of the system?

7.1 Abstract

Saddle point formulations of linear systems of equations occurring in the incremental weak constraint 4D-var data assimilation method are suitable for time-parallel computations. These large sparse systems can be solved using the MINRES method and preconditioning is needed to do it efficiently. We consider block diagonal preconditioners for the 3×3 block and the 2×2 block formulations. These preconditioners employ approximations of the inverse of a Schur complement, which usually excludes the observation information. We propose a way to incorporate this information in the approximation. This can be achieved by computing the randomised eigenvalue decomposition of the Schur complement and using it to construct limited memory preconditioners. We also analyse how the eigenvalues of the preconditioned coefficient matrices relate to the eigenvalues of the coefficient matrices in the symmetric positive definite formulations of the weak constraint 4D-var, and how sensitive they are to the number of observations. An idealised numerical example illustrates the theory and shows that the new preconditioner improves the minimisation.

7.2 Introduction

Data assimilation uses observations of a dynamical system to improve a prior estimate (background) of the state of this system. It is used in numerical weather prediction to obtain the initial conditions for a weather model [Kalnay, 2002], and in other applications like flood-, air pollution-, and epidemiological-forecasting [García-Pintado et al., 2015, Arcucci et al., 2018, Evensen et al., 2021]. The size of the problem in operational settings and time constraint on computations requires parallelisation. High level of parallelism can be achieved using an incremental weak constraint 4D-Var method, where a series of quadratic cost functions are minimised (e.g., [Trémolet, 2006, Fisher and Gürol, 2017]). Their minima can be found by solving large sparse linear systems of equations, and the 3×3 block and 2×2 block saddle point formulations of these have a lot of potential for time-parallel computations ([Fisher and Gürol, 2017, Daužickaitė et al., 2020]).

The iterative Krylov subspace solvers used to solve these require preconditioning (see, e.g. [Saad, 2003]) and designing an effective preconditioner is a problem-dependent challenge. The computational time constraint in data assimilation often results in solvers being terminated after a fixed number of iterations. Hence, the preconditioner needs to be efficient in the first iterations of the solver. In the early termination case caution has to be exercised, because the change to the value of the quadratic cost function in the incremental weak constraint 4D-Var, whose minimiser is sought by solving the linear systems, is not monotonic when solving the saddle point systems and additional checks may be needed to ensure a reduction in the quadratic cost function value [Gratton et al., 2018a]. If the preconditioner is effective enough to ensure the convergence of the iterative solver in a given time, then the additional checks are not needed, because the minimum of the quadratic cost function is reached.

Previous preconditioning approaches for the 3×3 block formulation used a block diagonal preconditioner that involves approximation of the Schur complement [Gratton et al., 2018a, Freitag and Green, 2018, Tabeart and Pearson, 2021]. The Schur complement is a sum of a symmetric positive definite matrix and a symmetric positive semi-definite matrix; the latter containing the observation information. The previous approximations explicitly excluded the observation term, so that the inverse of the approximation could be computed. Based on our previous work on the forcing formulation of weak constraint 4D-Var [Daužickaitė et al., 2021b], we propose preconditioning using randomised limited memory preconditioners (LMPs) to approximate the inverse of the full Schur complement. LMPs are cheap to apply and can be constructed using approximations of the eigenvalues and eigenvectors (eigenpairs) of the Schur complement, that can be found using the randomised eigenvalue decomposition. This approach can also be used to precondition the 2×2 block formulation.

We analyse the eigenvalues of the preconditioned 3×3 block and 2×2 block saddle point matrices and show their connection to the eigenvalues of the symmetric positive definite (SPD) matrices arising in the standard forcing and state formulations of the incremental weak constraint 4D-Var [Trémolet, 2006, Trémolet, 2007]. We also extend the theory on

how the eigenvalues of the unpreconditioned saddle point matrices depend on the number of observations of the dynamical system [Daužickaitė et al., 2020] to the preconditioned case.

The incremental weak constraint 4D-Var method and previous preconditioning strategies are discussed in Section 7.3. We propose the randomised preconditioning approach in Section 7.4, and Section 7.5 contains the theoretical analysis of the eigenvalues. A numerical example using the Lorenz 96 model is presented in Section 7.6. We conclude the work in Section 7.7.

7.3 Incremental weak constraint 4D-Var

We consider the state evolution of the dynamical system $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N$ with $\mathbf{x}_i \in \mathbb{R}^n$ at times $t_0, t_1, \ldots, t_N$. The background is the previous estimate of the state at time t_0 denoted by $\mathbf{x}^b \in \mathbb{R}^n$. The observations at time t_i are given by $\mathbf{y}_i \in \mathbb{R}^{q_i}$ and the nonlinear observation operator $\mathcal{H}_i$ maps the state variables to the observation space. The nonlinear model $\mathcal{M}_i$ maps the state variables at time t_i to the state at time t_{i+1} and has error $\boldsymbol{\eta}_i \in \mathbb{R}^n$, that is,

$$\mathbf{x}_{i+1} = \mathcal{M}_i(\mathbf{x}_i) + \boldsymbol{\eta}_i. \tag{7.1}$$

The errors in the data are assumed to be Gaussian with zero mean and are described using the background- $\mathbf{B} \in \mathbb{R}^{n\times n}$, the observation- $\mathbf{R}_i \in \mathbb{R}^{q_i \times q_i}$ and the model-error $\mathbf{Q}_i \in \mathbb{R}^{n\times n}$ covariance matrices.

The updated trajectory $\mathbf{x}_0^a, \mathbf{x}_1^a, \ldots, \mathbf{x}_N^a$ is called the analysis. It can be obtained as the minimiser of the following nonlinear cost function, as arises from the so-called state formulation [Trémolet, 2006]

$$\begin{aligned} J(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_N) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)) \\ + \frac{1}{2}\sum_{i=0}^{N-1}(\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i))^T \mathbf{Q}_{i+1}^{-1}(\mathbf{x}_{i+1} - \mathcal{M}_i(\mathbf{x}_i)). \end{aligned} \tag{7.2}$$

For large systems, the direct minimisation of (7.2) is impractical and the analysis is approximated using an approximate Gauss-Newton method [Gratton et al., 2007], also known as the incremental formulation. We use the following notation (following [Gratton et al., 2018a]).

$$\mathbf{x}^{(j)} = \begin{pmatrix} \mathbf{x}_0^{(j)} \\ \mathbf{x}_1^{(j)} \\ \cdots \\ \mathbf{x}_N^{(j)} \end{pmatrix}, \delta\mathbf{x}^{(j)} = \begin{pmatrix} \delta\mathbf{x}_0^{(j)} \\ \delta\mathbf{x}_1^{(j)} \\ \cdots, \\ \delta\mathbf{x}_N^{(j)} \end{pmatrix}, \mathbf{b}^{(j)} = \begin{pmatrix} \mathbf{x}^b - \mathbf{x}_0^{(j)} \\ \mathcal{M}_0(\mathbf{x}_0^{(j)}) - \mathbf{x}_1^{(j)} \\ \vdots \\ \mathcal{M}_{N-1}(\mathbf{x}_{N-1}^{(j)}) - \mathbf{x}_N^{(j)} \end{pmatrix}, \mathbf{d}^{(j)} = \begin{pmatrix} \mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0^{(j)}) \\ \mathbf{y}_1 - \mathcal{H}_1(\mathbf{x}_1^{(j)}) \\ \vdots \\ \mathbf{y}_N - \mathcal{H}_N(\mathbf{x}_N^{(j)}) \end{pmatrix}, \tag{7.3}$$

where $\mathbf{x}^{(j)}$ is the jth approximation of the state, $\mathbf{x}^{(j)}, \delta\mathbf{x}^{(j)}, \mathbf{b}^{(j)} \in \mathbb{R}^{(N+1)n}$, $q = \sum_{i=0}^{N} q_i$

and $\mathbf{d}^{(j)} \in \mathbb{R}^q$. The following matrices are defined

$$\mathbf{L}^{(j)} = \begin{pmatrix} \mathbf{I} & & & & \\ -\mathbf{M}_0^{(j)} & \mathbf{I} & & & \\ & -\mathbf{M}_1^{(j)} & \mathbf{I} & & \\ & & \ddots & \ddots & \\ & & & -\mathbf{M}_{N-1}^{(j)} & \mathbf{I} \end{pmatrix} \in \mathbb{R}^{(N+1)n \times (N+1)n}, \tag{7.4}$$

$$\mathbf{H}^{(j)} = \ diag(\mathbf{H}_0^{(j)}, \mathbf{H}_1^{(j)}, \ldots, \mathbf{H}_N^{(j)}) \in \mathbb{R}^{q \times (N+1)n}, \tag{7.5}$$

$$\mathbf{D} = \ diag(\mathbf{B}, \mathbf{Q}_1, \ldots, \mathbf{Q}_N) \in \mathbb{R}^{(N+1)n \times (N+1)n}, \tag{7.6}$$

$$\mathbf{R} = \ diag(\mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_N) \in \mathbb{R}^{q \times q}, \tag{7.7}$$

where $\mathbf{I}$ is an $n \times n$ identity matrix, $diag(\cdot)$ denotes a block diagonal matrix, and $\mathbf{M}_i$ and $\mathbf{H}_i$ are the model and observation operator, respectively, linearised at time t_i. The analysis is updated incrementally

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} + \delta\mathbf{x}^{(j)}, \tag{7.8}$$

where $\delta\mathbf{x}^{(j)}$ minimises the quadratic cost function

$$J^{\delta}(\delta\mathbf{x}^{(j)}) = \frac{1}{2}||\mathbf{L}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{b}^{(j)}||^2_{\mathbf{D}^{-1}} + \frac{1}{2}||\mathbf{H}^{(j)}\delta\mathbf{x}^{(j)} - \mathbf{d}^{(j)}||^2_{\mathbf{R}^{-1}}, \tag{7.9}$$

where $||\mathbf{a}||^2_{\mathbf{A}^{-1}} = \mathbf{a}^T\mathbf{A}^{-1}\mathbf{a}$ and $\mathbf{A}$ is symmetric positive definite.

The increment $\delta\mathbf{x}^{(j)}$ can be obtained as a solution of the following large sparse linear system of equations with symmetric positive definite coefficient matrix

$$\mathcal{A}_s^{(j)}\delta\mathbf{x}^{(j)} = (\mathbf{L}^T)^{(j)}\mathbf{D}^{-1}\mathbf{b}^{(j)} + (\mathbf{H}^T)^{(j)}\mathbf{R}^{-1}\mathbf{d}^{(j)}, \tag{7.10}$$

$$\text{where } \mathcal{A}_s^{(j)} = (\mathbf{L}^T)^{(j)}\mathbf{D}^{-1}\mathbf{L}^{(j)} + (\mathbf{H}^T)^{(j)}\mathbf{R}^{-1}\mathbf{H}^{(j)}. \tag{7.11}$$

The conjugate gradient (CG) method (e.g., [Saad, 2003]) is the method of choice for such systems. The most expensive part of CG is the matrix-vector product with $\mathcal{A}_s^{(j)}$, especially integrating the linearised model and its adjoint in $\mathbf{L}$ and $\mathbf{L}^T$, respectively. [Fisher and Gürol, 2017] noted that the matrix-vector products with $\mathcal{A}_s^{(j)}$ can be parallelised in the time dimension, that is computations with $\mathbf{M}_i^{(j)}$ can be done independently of $\mathbf{M}_l^{(j)}$, $l \neq i$.

The control variable transform technique, also known as first level preconditioning (e.g., Section 3.2 of [Lawless, 2013]), is used in data assimilation to eliminate the small eigenvalues. In this technique, the coefficient matrix is mapped to a sum of the identity and a low-rank symmetric positive semi-definite matrix, so the smallest eigenvalue is equal to one. This is equivalent to split preconditioning $\mathcal{A}_s^{(j)}$ with $\mathbf{P}^{(j)} = ((\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2})^T((\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2})$. Then the so-called forcing formulation is obtained

$$\mathcal{A}_f^{(j)}\delta\mathbf{p}^{(j)} = \mathbf{D}^{-1/2}\mathbf{b}^{(j)} + \mathbf{D}^{1/2}(\mathbf{L}^{-T})^{(j)}(\mathbf{H}^T)^{(j)}\mathbf{R}^{-1}\mathbf{d}^{(j)}, \tag{7.12}$$

$$\text{where } \mathcal{A}_f^{(j)} = \mathbf{I} + \mathbf{D}^{1/2}(\mathbf{L}^{-T})^{(j)}(\mathbf{H}^T)^{(j)}\mathbf{R}^{-1}\mathbf{H}^{(j)}(\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2} \tag{7.13}$$

$$\text{and } \delta\mathbf{p}^{(j)} = \mathbf{D}^{-1/2}\mathbf{L}^{(j)}\delta\mathbf{x}^{(j)}. \tag{7.14}$$

Here,

$$(\mathbf{L}^{-1})^{(j)} = \begin{pmatrix} \mathbf{I} & & & & \\ \mathbf{M}_{0,0}^{(j)} & \mathbf{I} & & & \\ \mathbf{M}_{0,1}^{(j)} & \mathbf{M}_{1,1}^{(j)} & \mathbf{I} & & \\ \vdots & \vdots & \ddots & \ddots & \\ \mathbf{M}_{0,N-1}^{(j)} & \mathbf{M}_{1,N-1}^{(j)} & \cdots & \mathbf{M}_{N-1,N-1}^{(j)} & \mathbf{I} \end{pmatrix} \tag{7.15}$$

and $\mathbf{M}_{i,l}^{(j)} = \mathbf{M}_l^{(j)} \dots \mathbf{M}_i^{(j)}$ denotes the linearised model integration from time t_i to t_{l+1}. The matrix-vector products with $(\mathbf{L}^{-1})^{(j)}$ and hence with $\mathcal{A}_f^{(j)}$ are essentially sequential in time, that is, computations with $\mathbf{M}_i^{(j)}$ depend on the computations with $\mathbf{M}_l^{(j)}$, $l < i$. Thus, the control variable transform technique is not suitable when time-parallelism is important; for approximation approaches see [Daužickaitė et al., 2021a].

We note the following relationship between the eigenvalues of $\mathcal{A}_s^{(j)}$ and $\mathcal{A}_f^{(j)}$.

Lemma 7.1. *$\mathcal{A}_f^{(j)}$ and $(\mathbf{L}^{-1})^{(j)}\mathbf{D}(\mathbf{L}^{-T})^{(j)}\mathcal{A}_s^{(j)}$ have the same eigenvalues.*

Proof. We write $\mathcal{A}_f^{(j)}$ in the form

$$\mathcal{A}_f^{(j)} = \mathbf{D}^{1/2}(\mathbf{L}^{-T})^{(j)}\mathcal{A}_s^{(j)}(\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2} \tag{7.16}$$

and apply the similarity transformation using $(\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2}$:

$$\begin{aligned} ((\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2})\mathcal{A}_f^{(j)}((\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2})^{-1} = & \\ ((\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2})\mathbf{D}^{1/2}(\mathbf{L}^{-T})^{(j)}\mathcal{A}_s^{(j)}(\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2}((\mathbf{L}^{-1})^{(j)}\mathbf{D}^{1/2})^{-1} & \quad (7.17) \\ = (\mathbf{L}^{-1})^{(j)}\mathbf{D}(\mathbf{L}^{-T})^{(j)}\mathcal{A}_s^{(j)}. & \quad (7.18) \end{aligned}$$

$\mathcal{A}_f^{(j)}$ is hence similar to $(\mathbf{L}^{-1})^{(j)}\mathbf{D}(\mathbf{L}^{-T})^{(j)}\mathcal{A}_s^{(j)}$, and the result follows. □

We omit the superscript (j) in further discussion.

7.3.1 Saddle point formulations

Because of a lack of efficient time-parallel preconditioning for (7.10) and the need to increase the potential for parallelism further, [Fisher and Gürol, 2017] introduced a formulation of the system of linear equations with a 3×3 block saddle point coefficient matrix to obtain $\delta\mathbf{x}$

$$\mathcal{A}_3 \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{0} \end{pmatrix}, \tag{7.19}$$

where

$$\mathcal{A}_3 = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{L} \\ \mathbf{0} & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(2(N+1)n+q)\times(2(N+1)n+q)}. \tag{7.20}$$

$\boldsymbol{\lambda} \in \mathbb{R}^{(N+1)n}$ and $\boldsymbol{\mu} \in \mathbb{R}^q$ are Lagrange multipliers. Notice that (7.20) is more than twice the size of $\mathcal{A}_s$, but now the computations with $\mathbf{L}$ and $\mathbf{L}^T$ can be done independently, i.e., there is more potential for parallel computations.

[Daužickaitė et al., 2020] introduced a reduced 2×2 block saddle point system, which retains the potential for time-parallel model integration, but only solves for $\boldsymbol{\lambda}$ and $\delta\mathbf{x}$:

$$\mathcal{A}_2 \begin{pmatrix} \boldsymbol{\lambda} \\ \delta\mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ -\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d} \end{pmatrix}, \tag{7.21}$$

where

$$\mathcal{A}_2 = \begin{pmatrix} \mathbf{D} & \mathbf{L} \\ \mathbf{L}^T & -\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \end{pmatrix} \in \mathbb{R}^{2(N+1)n \times 2(N+1)n}. \tag{7.22}$$

Both $\mathcal{A}_3$ and $\mathcal{A}_2$ are symmetric and thus systems (7.19) and (7.21) can be solved using MINRES [Paige and Saunders, 1975], which has the advantage of being a three-term recurrence method compared to other methods for saddle point systems, like the generalized minimal residual method (GMRES, [Saad and Schultz, 1986]), that is, in MINRES only two previous iterates are needed to compute the new one. We consider preconditioning for these systems in the following section.

7.3.2 Preconditioning

MINRES requires a symmetric positive definite preconditioner (e.g., [Benzi et al., 2005]). In this work, we consider such block diagonal, or Schur complement, preconditioners $\mathcal{P}_{B,3}$ and $\mathcal{P}_{B,2}$ for the 3×3 block and 2×2 block systems, respectively.

$$\mathcal{P}_{B,3} = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{S}} \end{pmatrix}, \tag{7.23}$$

$$\mathcal{P}_{B,2} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{S}} \end{pmatrix}, \tag{7.24}$$

where $\widetilde{\mathbf{S}}$ is a symmetric positive definite approximation to the negative Schur complement of $\begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix}$ in $\mathcal{A}_3$ and $\mathbf{D}$ in $\mathcal{A}_2$, defined as

$$\mathbf{S} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}. \tag{7.25}$$

Notice that

$$\mathbf{S} = \mathcal{A}_s. \tag{7.26}$$

In practical settings, the inverses of the preconditioners are applied

$$\mathcal{P}_{B,3}^{-1} = \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \widetilde{\mathbf{S}}^{-1} \end{pmatrix}, \tag{7.27}$$

$$\mathcal{P}_{B,2}^{-1} = \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{S}}^{-1} \end{pmatrix}, \tag{7.28}$$

and the preconditioned matrices $\mathcal{P}_B^{-1}\mathcal{A}_3$ and $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ are

$$\mathcal{P}_{B,3}^{-1}\mathcal{A}_3 = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{D}^{-1}\mathbf{L} \\ \mathbf{0} & \mathbf{I} & \mathbf{R}^{-1}\mathbf{H} \\ \widetilde{\mathbf{S}}^{-1}\mathbf{L}^T & \widetilde{\mathbf{S}}^{-1}\mathbf{H}^T & \mathbf{0} \end{pmatrix}, \tag{7.29}$$

$$\mathcal{P}_{B,2}^{-1}\mathcal{A}_2 = \begin{pmatrix} \mathbf{I} & \mathbf{D}^{-1}\mathbf{L} \\ \widetilde{\mathbf{S}}^{-1}\mathbf{L}^T & -\widetilde{\mathbf{S}}^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \end{pmatrix}. \tag{7.30}$$

$\mathcal{P}_{B,3}$ was used for the 3×3 block system in [Gratton et al., 2018a, Freitag and Green, 2018, Tabeart and Pearson, 2021]. $\widetilde{\mathbf{S}}$ did not include observation information, that is, the approximation

$$\widetilde{\mathbf{S}} = \widetilde{\mathbf{L}}^T\mathbf{D}^{-1}\widetilde{\mathbf{L}} \tag{7.31}$$

was considered, because then $\widetilde{\mathbf{S}}^{-1}$ can be computed easily. The approximation $\widetilde{\mathbf{L}}$ to $\mathbf{L}$ in (7.31) needs to be chosen in such a way that the potential for time-parallel computations is preserved. This can be achieved by approximating the model $\mathbf{M}_i$ in $\widetilde{\mathbf{L}}$ with $\widetilde{\mathbf{M}}_i = \mathbf{0}$ or $\widetilde{\mathbf{M}}_i = \mathbf{I}$ [Gratton et al., 2018a, Freitag and Green, 2018]. [Tabeart and Pearson, 2021] generated $\widetilde{\mathbf{L}}$ by setting $\widetilde{\mathbf{M}}_i = \mathbf{0}$ for some times t_i and using the exact model $\widetilde{\mathbf{M}}_i = \mathbf{M}_i$ for others. This accelerates the convergence of MINRES compared to setting $\widetilde{\mathbf{M}}_i$ to zero or identity, but does not include any observation information and reduces some of the potential for time-parallel computations. We propose a way to approximate $\mathbf{S}^{-1}$ without discarding the $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ term, while preserving the time-parallelism in the following section.

7.4 Randomised preconditioning

Spectral limited memory preconditioners (LMPs) use eigenpairs of an SPD matrix to approximate its inverse (e.g., [Gratton et al., 2011]). They have been used to precondition the strong constraint 4D-Var [Tshimanga et al., 2008] and the forcing formulation of the weak constraint 4D-Var (7.12) [Daužickaitė et al., 2021b]. If λ_i, $i \in \{1, 2, \ldots, k\}$, is an eigenvalue of an SPD matrix $\mathbf{A}$ and $\mathbf{v}_i$ is the corresponding eigenvector, then the spectral-LMP defined as

$$\mathbf{P}_k = \mathbf{I} - \sum_{i=1}^{k}(1 - \lambda_i^{-1})\mathbf{v}_i\mathbf{v}_i^T \tag{7.32}$$

is an SPD matrix that approximates $\mathbf{A}^{-1}$. The number of eigenpairs k used to construct $\mathbf{P}_k$ is usually small compared to the size of the system and hence the matrix-vector products with $\mathbf{P}_k$ are cheap to compute.

For large systems, approximations of the eigenpairs are used to construct $\mathbf{P}_k$. The randomised LMP is constructed using approximations of the eigenpairs obtained with a randomised eigenvalue decomposition method. We propose using the randomised LMP to construct an approximation of $\mathbf{S}^{-1} = \mathcal{A}_s^{-1}$, that is we set $\widetilde{\mathbf{S}}^{-1}$ in (7.27) and (7.28) to

$$\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_k. \tag{7.33}$$

Hence, we need to compute the randomised eigenvalue decomposition of $\mathcal{A}_s$ to generate $\widetilde{\mathbf{S}}^{-1}$. These methods can be considered as variants of the classic subspace iteration method started with a random matrix [Gu, 2015]. We present the REVD_*ritzit* method, which is based on the Rutishauser's implementation of the subspace iteration method [Rutishauser, 1971], and has been shown to produce a good randomised LMP for the forcing formulation (7.12) [Daužickaitė et al., 2021b], in Algorithm 13. It is started with a random matrix whose entries are independent standard normal random variables with zero mean and variance equal to one. Oversampling, that is, working on a subspace generated by $k+l$ vectors when looking for k eigenpairs, is used to increase the quality of the approximation; setting l to five or ten is expected to be sufficient in many applications [Halko et al., 2011]. The algorithm requires one matrix-matrix product with $\mathcal{A}_s$ in step 3, which is easy to parallelise on the current computers. The most expensive part of the product is the integration of the linearised model and its adjoint in $\mathbf{L}$ and $\mathbf{L}^T$. Note that this can also be parallelised in time.

Algorithm 13 Randomised eigenvalue decomposition based on *ritzit*, REVD_*ritzit*

Input: symmetric matrix $\mathbf{A} \in \mathbb{R}^{n_A \times n_A}$, target rank k, an oversampling parameter l

Output: orthogonal $\mathbf{U} \in \mathbb{R}^{n_A \times k}$ with approximations to eigenvectors of $\mathbf{A}$ as its columns, and diagonal $\mathbf{\Theta} \in \mathbb{R}^{k \times k}$ with approximations to the largest eigenvalues of $\mathbf{A}$ on the diagonal

1: Form a Gaussian random matrix $\mathbf{G} \in \mathbb{R}^{n_A \times (k+l)}$
2: Orthonormalize the columns of $\mathbf{G}$ to obtain orthonormal $\hat{\mathbf{G}}$
3: Form a sample matrix $\mathbf{Y} = \mathbf{A}\hat{\mathbf{G}} \in \mathbb{R}^{n_A \times (k+l)}$
4: Compute QR decomposition $\mathbf{Y} = \mathbf{Z}\mathbf{R}$ to obtain orthogonal $\mathbf{Z} \in \mathbb{R}^{n_A \times (k+l)}$ and upper triangular $\mathbf{R} \in \mathbb{R}^{(k+l)\times(k+l)}$
5: Form $\mathbf{K} = \mathbf{R}\mathbf{R}^T \in \mathbb{R}^{(k+l)\times(k+l)}$
6: Form EVD of $\mathbf{K}$: $\mathbf{K} = \mathbf{W}\mathbf{\Theta}^2\mathbf{W}^T$, where $\mathbf{W}$, $\mathbf{\Theta}^2 \in \mathbb{R}^{(k+l)\times(k+l)}$, elements of $\mathbf{\Theta}$ are sorted in decreasing order
7: Remove last l columns and rows of $\mathbf{\Theta}^2$, so that $\mathbf{\Theta}^2 \in \mathbb{R}^{k\times k}$
8: Remove last l columns of $\mathbf{W}$, so that $\mathbf{W} \in \mathbb{R}^{(k+l)\times k}$
9: Form $\mathbf{U} = \mathbf{Z}\mathbf{W} \in \mathbb{R}^{n_A \times k}$.

7.5 Eigenvalues of the preconditioned saddle point systems

The convergence of MINRES can be described using the distribution of the eigenvalues of the coefficient matrix, see, for example [Greenbaum, 1997, Simoncini and Szyld, 2013]. Eigenvalues tightly clustered away from zero may be expected to give good convergence. In this section, we present results that connect eigenvalues of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ with eigenvalues of $\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s$ for a general choice of $\widetilde{\mathbf{S}}^{-1}$, and eigenvalues of $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ with eigenvalues of $(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ when $\widetilde{\mathbf{S}} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$, which shows the importance of the interaction between the model term $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$ and the observation term $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ of $\mathcal{A}_s$.

We also explore how the extreme eigenvalues of the preconditioned systems change with introduction of new observations. We use the following theorems.

7.5.1 Preliminaries

Theorem 7.2 (Determinant of a block matrix, see, e.g., Section 9.11.2 of [Lütkepohl, 1996]). *Let* $\mathbf{A}$ *and* $\mathbf{D}$ *be square matrices and* $\mathbf{F} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}$. *If* $\mathbf{A}$ *is invertible, then*

$$det(\mathbf{F}) = det(\mathbf{A})det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}). \tag{7.34}$$

Theorem 7.3 (Eigenvalue bounds for the preconditioned saddle point system, see, e.g., [Rees and Wathen, 2009])). *Let* $\mathbf{A}$, $\widetilde{\mathbf{A}}$, $\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ *and* $\mathbf{Z}$ *be positive definite matrices,* $\mathbf{F} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix}$ *and* $\mathbf{P} = \begin{pmatrix} \widetilde{\mathbf{A}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Z} \end{pmatrix}$. *If we denote* $\lambda_{min}(\widetilde{\mathbf{A}}^{-1}\mathbf{A}) = \delta$, $\lambda_{max}(\widetilde{\mathbf{A}}^{-1}\mathbf{A}) = \Delta$, $\lambda_{min}(\mathbf{Z}^{-1}\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}) = \phi$ *and* $\lambda_{max}(\mathbf{Z}^{-1}\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}) = \Phi$, *where* $\lambda_{min}(\mathbf{C})$ *and* $\lambda_{max}(\mathbf{C})$ *are the smallest and largest eigenvalues of* $\mathbf{C}$, *respectively, then the eigenvalues* λ *of* $\mathbf{P}^{-1}\mathbf{F}$ *are real and are bounded by*

$$\frac{1}{2}\left(\delta - \sqrt{\delta^2 + 4\Delta\Phi}\right) \leq \lambda \leq \frac{1}{2}\left(\Delta - \sqrt{\Delta^2 + 4\delta\phi}\right), \tag{7.35}$$

$$\delta \leq \lambda \leq \Delta, \tag{7.36}$$

$$\frac{1}{2}\left(\delta + \sqrt{\delta^2 + 4\delta\phi}\right) \leq \lambda \leq \frac{1}{2}\left(\Delta + \sqrt{\Delta^2 + 4\Delta\Phi}\right). \tag{7.37}$$

Theorem 7.4 (See Section 8.1.2 of [Golub and Van Loan, 2013]). *If* $\mathbf{A}$ *and* $\mathbf{C}$ *are* $n \times n$ *Hermitian matrices, then*

$$\lambda_k(\mathbf{A}) + \lambda_{min}(\mathbf{C}) \leq \lambda_k(\mathbf{A} + \mathbf{C}) \leq \lambda_k(\mathbf{A}) + \lambda_{max}(\mathbf{C}), \quad k \in \{1, 2, \ldots, n\}.$$

7.5.2 Eigenvalues of the preconditioned 3×3 block formulation

We consider the eigenvalues of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ for an SPD $\widetilde{\mathbf{S}}^{-1}$, and discuss the implications for special cases of $\widetilde{\mathbf{S}}^{-1}$.

Theorem 7.5. *Let* $\mathcal{A}_3$ *and* $\mathcal{P}_{B,3}^{-1}$ *be as defined in* (7.20) *and* (7.27), *respectively, and* γ *be an eigenvalue of* $\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s$. *Then the eigenvalues* λ *of* $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ *are real and*

$$\lambda = \begin{cases} 1, \text{ multiplicity } q, \\ \frac{1}{2}\left(1 \pm \sqrt{1 + 4\gamma}\right), \end{cases} \tag{7.38}$$

where q *is the total number of observations of the dynamical system.*

Proof. We know that λ is real from Theorem 7.3, where $\mathbf{A} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} \mathbf{L} \\ \mathbf{H} \end{pmatrix}$, $\widetilde{\mathbf{A}} = \mathbf{A}$ and $\mathbf{Z} = \widetilde{\mathbf{S}}$.

We show that $\lambda = 1$ using Theorem 7.3. Because $\widetilde{\mathbf{A}} = \mathbf{A}$, we have $\widetilde{\mathbf{A}}^{-1}\mathbf{A} = \mathbf{I}$ and $\delta = \Delta = 1$. Then from (7.36), $\lambda = 1$. In this case, $\mathcal{A}_3\mathbf{v} = \mathcal{P}_{B,3}\mathbf{v}$, where $\mathbf{v} = \left(\mathbf{v}_1^T, \mathbf{v}_2^T, \mathbf{v}_3^T\right)^T$,

$\mathbf{v}_1, \mathbf{v}_3 \in \mathbb{R}^{n(N+1)}$, $\mathbf{v}_2 \in \mathbb{R}^q$, i.e.,

$$\mathbf{D}\mathbf{v}_1 + \mathbf{L}\mathbf{v}_3 = \mathbf{D}\mathbf{v}_1, \tag{7.39}$$

$$\mathbf{R}\mathbf{v}_2 + \mathbf{H}\mathbf{v}_3 = \mathbf{R}\mathbf{v}_2, \tag{7.40}$$

$$\mathbf{L}^T\mathbf{v}_1 + \mathbf{H}^T\mathbf{v}_2 = \widetilde{\mathbf{S}}\mathbf{v}_3. \tag{7.41}$$

From (7.39), $\mathbf{v}_3 = \mathbf{0}$. Then (7.41) gives $\mathbf{v}_1 = -\mathbf{L}^{-T}\mathbf{H}^T\mathbf{v}_2$ and $\mathbf{v} = \left((-\mathbf{L}^{-T}\mathbf{H}^T\mathbf{v}_2)^T, \mathbf{v}_2^T, \mathbf{0}^T\right)^T$ with q choices of linearly independent $\mathbf{v}_2$. Hence, $\lambda = 1$ has multiplicity q.

Now, assume $\lambda \neq 1$ and consider the characteristic polynomial

$$0 = det(\mathcal{P}_{B,3}^{-1}\mathcal{A}_3 - \lambda\mathbf{I}) = det\left(\begin{pmatrix} \mathbf{I}-\lambda\mathbf{I} & \mathbf{0} & \mathbf{D}^{-1}\mathbf{L} \\ \mathbf{0} & \mathbf{I}-\lambda\mathbf{I} & \mathbf{R}^{-1}\mathbf{H} \\ \widetilde{\mathbf{S}}^{-1}\mathbf{L}^T & \widetilde{\mathbf{S}}^{-1}\mathbf{H}^T & -\lambda\mathbf{I} \end{pmatrix}\right). \tag{7.42}$$

Matrix $\begin{pmatrix} \mathbf{I}-\lambda\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}-\lambda\mathbf{I} \end{pmatrix}$ is invertible, thus using Theorem 7.2

$$\begin{aligned} 0 = det\left(\begin{pmatrix} \mathbf{I}-\lambda\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}-\lambda\mathbf{I} \end{pmatrix}\right) \times \\ det\left(-\lambda\mathbf{I} - \begin{pmatrix} \widetilde{\mathbf{S}}^{-1}\mathbf{L}^T & \widetilde{\mathbf{S}}^{-1}\mathbf{H}^T \end{pmatrix}\begin{pmatrix} \mathbf{I}-\lambda\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}-\lambda\mathbf{I} \end{pmatrix}^{-1}\begin{pmatrix} \mathbf{D}^{-1}\mathbf{L} \\ \mathbf{R}^{-1}\mathbf{H} \end{pmatrix}\right) \end{aligned} \tag{7.43}$$

and

$$0 = det\left(-\lambda\mathbf{I} - \begin{pmatrix} \widetilde{\mathbf{S}}^{-1}\mathbf{L}^T & \widetilde{\mathbf{S}}^{-1}\mathbf{H}^T \end{pmatrix}\begin{pmatrix} \mathbf{I}-\lambda\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}-\lambda\mathbf{I} \end{pmatrix}^{-1}\begin{pmatrix} \mathbf{D}^{-1}\mathbf{L} \\ \mathbf{R}^{-1}\mathbf{H} \end{pmatrix}\right) \tag{7.44}$$

$$= det\left(-\lambda\mathbf{I} - (1-\lambda)^{-1}\widetilde{\mathbf{S}}^{-1}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\right) \tag{7.45}$$

$$= det\left(-\lambda\mathbf{I} - (1-\lambda)^{-1}\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s\right) \tag{7.46}$$

$$= det\left((1-\lambda)^{-1}(-\lambda(1-\lambda)\mathbf{I} - \widetilde{\mathbf{S}}^{-1}\mathcal{A}_s)\right). \tag{7.47}$$

Thus, $-\lambda(1-\lambda) = \lambda(\lambda-1)$ is an eigenvalues of $\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s$, that is,

$$\lambda(\lambda-1) = \gamma \tag{7.48}$$

and

$$\lambda = \frac{1}{2}\left(1 \pm \sqrt{1+4\gamma}\right). \tag{7.49}$$

□

Note that $\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s$ is symmetric positive definite and γ are positive. Hence, the smallest positive eigenvalue of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ is equal to one. For large λ the approximation $\lambda(\lambda-1) \approx \lambda^2$ can be used. Then $\lambda \approx \pm\sqrt{\gamma}$ and the modulus of the largest eigenvalue of $\mathcal{P}_B^{-1}\mathcal{A}_3$ is approximately the square root of the largest eigenvalue of $\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s$. We further consider three special cases of $\widetilde{\mathbf{S}}$.

- Assume that $\widetilde{\mathbf{S}}$ does not involve any information on the observations and the exact model is used, that is $\widetilde{\mathbf{S}} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$. Then $\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}\mathcal{A}_s$ and γ are the eigenvalues of $\mathcal{A}_f$ (Lemma 7.1).

- If $\widetilde{\mathbf{S}}$ does not involve any information on the observations and the model approximation is set to zero, then $\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s = \mathbf{D}\mathcal{A}_s$ and γ are the eigenvalues of $\mathcal{A}_s$ preconditioned with $\mathbf{D}$.

- If $\widetilde{\mathbf{S}}$ is a spectral-LMP in (7.32) constructed with the exact eigenpairs of $\mathcal{A}_s$, then some γ are equal to one and the rest are bounded by the smallest and largest eigenvalues of $\mathcal{A}_s$ (see Theorem 3.4 on the spectrum non-expansiveness in [Gratton et al., 2011]).

We can obtain the general bounds in Theorem 7.3 when $\delta = \Delta = 1$ by bounding γ with $\lambda_{min}(\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s) = \phi$ and $\lambda_{max}(\widetilde{\mathbf{S}}^{-1}\mathcal{A}_s) = \Phi$ in Theorem 7.5.

7.5.3 Eigenvalues of the preconditioned 2×2 block formulation

We consider the special case when $\widetilde{\mathbf{S}} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$ and show that in this case the eigenvalues of $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ relate to the eigenvalues of $(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. This shows that the interaction between the terms including the model $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$ and the observation information $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is important when preconditioner uses the exact model in $\widetilde{\mathbf{S}}$.

Theorem 7.6. *Let $\mathcal{A}_2$ and $\mathcal{P}_{B,2}^{-1}$ be as defined in* (7.22) *and* (7.30)*, respectively, with $\widetilde{\mathbf{S}} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$, and let θ be an eigenvalue of $(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. Then the eigenvalues μ of $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ are real and are given by*

$$\mu = \frac{1}{2}\left(1 - \theta \pm \sqrt{\theta^2 + 2\theta + 5}\right). \tag{7.50}$$

Proof. Let μ and $\mathbf{u} = (\mathbf{u}_1^T, \mathbf{u}_2^T)^T \in \mathbb{R}^{2n(N+1)}$, $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^{n(N+1)}$, be an eigenpair of $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$. First, we show that $\mu \neq 1$. Consider the eigenvalue equation $\mathcal{A}_2\mathbf{u} = \mu\mathcal{P}_{B,2}\mathbf{u}$, that is

$$\mathbf{D}\mathbf{u}_1 + \mathbf{L}\mathbf{u}_2 = \mu\mathbf{D}\mathbf{u}_1, \tag{7.51}$$

$$\mathbf{L}^T\mathbf{u}_1 - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{u}_2 = \mu\widetilde{\mathbf{S}}\mathbf{u}_2. \tag{7.52}$$

Assume that $\mu = 1$, then from (7.51) $\mathbf{L}\mathbf{u}_2 = \mathbf{0}$ and hence $\mathbf{u}_2 = \mathbf{0}$. Then from (7.52) $\mathbf{L}^T\mathbf{u}_1 = \mathbf{0}$, hence $\mathbf{u}_1 = \mathbf{0}$ and $\mathbf{u} = \mathbf{0}$. Thus, $\mu \neq 1$.

Now, consider the characteristic polynomial

$$0 = det(\mathcal{P}_{B,2}^{-1}\mathcal{A}_2 - \mu\mathbf{I}) \tag{7.53}$$

$$= det\begin{pmatrix} (1-\mu)\mathbf{I} & \mathbf{D}^{-1}\mathbf{L} \\ \widetilde{\mathbf{S}}^{-1}\mathbf{L}^T & -\widetilde{\mathbf{S}}^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mu\mathbf{I} \end{pmatrix}. \tag{7.54}$$

$(1-\mu)\mathbf{I}$ is invertible, because $\mu \neq 1$. Using Theorem 7.2,

$$0 = det((1-\mu)\mathbf{I})det(-\widetilde{\mathbf{S}}^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mu\mathbf{I} - (1-\mu)^{-1}\widetilde{\mathbf{S}}^{-1}\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}) \tag{7.55}$$

and

$$0 = det(-\widetilde{\mathbf{S}}^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mu\mathbf{I} - (1-\mu)^{-1}\widetilde{\mathbf{S}}^{-1}\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}). \tag{7.56}$$

If $\widetilde{\mathbf{S}}^{-1} = (\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}$, then

$$\begin{aligned} 0 &= det(-(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} - \mu\mathbf{I} - (1-\mu)^{-1}\mathbf{I}) && (7.57)\\ &= det((-\mu - (1-\mu)^{-1})\mathbf{I} - (\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}) && (7.58) \end{aligned}$$

and $-\mu - (1-\mu)^{-1}$ is an eigenvalue of $(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$, that is,

$$-\mu - (1-\mu)^{-1} = \theta \tag{7.59}$$

and

$$\mu = \frac{1}{2}\left(1 - \theta \pm \sqrt{\theta^2 + 2\theta + 5}\right). \tag{7.60}$$

□

Notice that $(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is similar to a symmetric positive semi-definite matrix $\mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}$, that is,

$$\mathbf{D}^{-1/2}\mathbf{L}((\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})\mathbf{L}^{-1}\mathbf{D}^{1/2} = \mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}. \tag{7.61}$$

Hence, θ are either positive, or $\theta = 0$ and $\mu = \frac{1}{2}\left(-1 \pm \sqrt{5}\right)$ with multiplicity of at least $n(N+1) - q$.

7.5.4 Change in eigenvalues due to new observations

We explore how the eigenvalues of $\mathcal{A}_f$, $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ and $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ change when the number of observations of the dynamical system is increased. The theoretical estimates hold for diagonal $\mathbf{R}$. In this section, the subscript k denotes the dynamical system with k observations and $k+1$ indicates that a new observation has been added to the system with k observations. The variance $\alpha \in \mathbb{R}$, $\alpha > 0$, and $\mathbf{h}_{k+1} \in \mathbb{R}^{(N+1)n}$ correspond to the new observation.

Theorem 7.7. *If the observation errors are uncorrelated, i.e.,* $\mathbf{R}$ *is diagonal, then the largest eigenvalues of* $\mathcal{A}_f$ *in* (7.13) *either move away from zero or are unchanged when a new observation is added. The smallest eigenvalues of* $\mathcal{A}_f$ *are equal to one when the dynamical system is not fully observed.*

Proof. The smallest eigenvalue is equal to one for any number of observations $q < n(N+1)$, because $\mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}$ is symmetric positive semi-definite.

We can write

$$\mathbf{H}_{k+1}^T\mathbf{R}_{k+1}^{-1}\mathbf{H}_{k+1} = \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k + \alpha^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T \tag{7.62}$$

and

$$(\mathcal{A}_f)_{k+1} = (\mathcal{A}_f)_k + \alpha^{-1}\mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T\mathbf{L}^{-1}\mathbf{D}^{1/2} = (\mathcal{A}_f)_k + \mathcal{E}. \tag{7.63}$$

$\mathcal{E}$ is symmetric positive semi-definite, hence $\lambda_{min}(\mathcal{E}) = 0$. Then from Theorem 7.4

$$\lambda_{max}((\mathcal{A}_f)_k) + \lambda_{min}(\mathcal{E}) \leq \lambda_{max}((\mathcal{A}_f)_{k+1}), \tag{7.64}$$

$$\Rightarrow \lambda_{max}((\mathcal{A}_f)_k) \leq \lambda_{max}((\mathcal{A}_f)_{k+1}). \tag{7.65}$$

□

Theorem 7.8. *If the observation errors are uncorrelated, i.e.,* $\mathbf{R}$ *is diagonal, and* $\widetilde{\mathbf{S}}$ *does not include information on observations, then the smallest and largest negative, and largest positive eigenvalues of* $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ *either move away from zero or are unchanged when new observations are introduced. The smallest positive eigenvalues are equal to one.*

Proof. The smallest positive eigenvalue of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ is equal to one independently of the number of observations by Theorem 7.5. The proof for the change of other eigenvalues is equivalent to the proof of Theorem 3 in [Daužickaitė et al., 2020] regarding the change in eigenvalues of $\mathcal{A}_3$. □

Note that results in Theorems 7.7 and 7.8 when $\widetilde{\mathbf{S}} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$ are consistent.

Theorem 7.9. *If the observation errors are uncorrelated, that is,* $\mathbf{R}$ *is diagonal, and* $\widetilde{\mathbf{S}}$ *does not include information on the observations, then the extreme eigenvalues of* $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ *change in the same way as eigenvalues of* $\mathcal{A}_2$ *when new observations are added. That is, the smallest and largest negative eigenvalues of* $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ *either move away from zero or are unchanged. Contrarily, the smallest and largest positive eigenvalues of* $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ *approach zero or are unchanged.*

Proof. We can write

$$\mathcal{P}_{B,2}^{-1}\mathcal{A}_{2,k+1} = \mathcal{A}_{2,k} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\alpha^{-1}\widetilde{\mathbf{S}}^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T \end{pmatrix} = \mathcal{A}_{2,k} + \widetilde{\mathcal{E}}. \tag{7.66}$$

$\widetilde{\mathbf{S}}^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T$ is similar to $\widetilde{\mathbf{S}}^{1/2}(\widetilde{\mathbf{S}}^{-1}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T)\widetilde{\mathbf{S}}^{-1/2} = \widetilde{\mathbf{S}}^{-1/2}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T\widetilde{\mathbf{S}}^{-1/2}$, where $\widetilde{\mathbf{S}}^{1/2}$ is the symmetric positive definite square root of $\widetilde{\mathbf{S}}$. Matrix $\widetilde{\mathbf{S}}^{-1/2}\mathbf{h}_{k+1}\mathbf{h}_{k+1}^T\widetilde{\mathbf{S}}^{-1/2}$ is symmetric positive semi-definite, hence $\widetilde{\mathcal{E}}$ has negative and zero eigenvalues. The rest of the proof is analogous to the proof of Theorem 5 in [Daužickaitė et al., 2020]. □

7.6 Numerical example

We perform numerical experiments to illustrate the theoretical results in Section 7.5 and explore the performance of $\mathcal{P}_{B,3}$ and $\mathcal{P}_{B,2}$ that include the randomised LMP, that is $\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_k$. Identical twin experiments are performed, where we generate the reference trajectory $\mathbf{x}_0^t, \mathbf{x}_1^t, \ldots, \mathbf{x}_N^t$. The background is obtained by adding random, Gaussian noise with covariance $\mathbf{B}$ to $\mathbf{x}_0^t$ and direct observations are generated by adding random, Gaussian noise with covariance $\mathbf{R}_i$ to $\mathcal{H}_i(\mathbf{x}_i^t)$.

The Lorenz 96 model [Lorenz, 1996] is used, where the dynamics of variables X^j, where $\mathbf{x}_i^T = \left(X^1, X^2, \ldots, X^n\right)$, are described by n coupled ODEs:

$$\frac{dX^j}{dt} = -X^{j-2}X^{j-1} + X^{j-1}X^{j+1} - X^j + F, \tag{7.67}$$

where $F = 8$ and $X^{-1} = X^{n-1}, X^0 = X^n$ and $X^{n+1} = X^1$. We use the fourth-order Runge-Kutta scheme to integrate (7.67) [Butcher, 1987]. The gridpoint distance is $\Delta X = 1/n$ and the time step is $\Delta t = 2.5 \times 10^{-2}$.

The error covariance matrices are set to $\mathbf{B} = 0.2^2\mathbf{C}_b$, $\mathbf{Q}_i = 0.05^2\mathbf{C}_q$ and $\mathbf{R}_i = 0.15^2\mathbf{I}$, where $\mathbf{C}_b$ is a a second-order auto-regressive (SOAR, [Daley, 1993]) matrix with length scale $2\Delta X$ and $\mathbf{C}_q$ is a Laplacian [Johnson et al., 2005] correlation matrix with length scale $0.75\Delta X$. We vary the number of observations.

We consider $\mathcal{P}_{B,3}$ and $\mathcal{P}_{B,2}$ with three choices of $\widetilde{\mathbf{S}}^{-1}$:

- $\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$, that is, $\widetilde{\mathbf{M}}_i = \mathbf{0}$;
- $\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$, that is, $\widetilde{\mathbf{M}}_i = \mathbf{M}_i$;
- $\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_{30}$, that is, using the randomised LMP in (7.32).

When the randomised LMP is used, then $\widetilde{\mathbf{S}}^{-1}$ includes some information on the observations. This is not the case with the other two choices of $\widetilde{\mathbf{S}}^{-1}$. Using the exact linearised model $\mathbf{M}_i$ in the preconditioner is not considered as a practical choice, because it makes the application of preconditioner sequential in time; it is used only for comparison. The randomised LMP $\mathbf{P}_{30}$ is constructed with thirty eigenpairs of $\mathcal{A}_s$ obtained using Algorithm 13. We set the oversampling parameter $l = 5$.

7.6.1 Eigenvalues of the preconditioned systems

We set $n = 40$ and $N = 79$, so that we can compute all of the eigenvalues of the preconditioned systems; the matrices are formed and the Matlab function *eig* is used. The following observation networks are considered with the total number of observations q:

a) $q = 60$, observing every sixteenth model variable at every fourth time step (at times t_3, t_7, …, t_{75}, t_{79}),

b) $q = 200$, observing every eighth model variable at every second time step (at times t_1, t_3, …, t_{77}, t_{79}),

c) $q = 800$, observing every fourth model variable at every time step.

We explore the change of the extreme eigenvalues of the preconditioned systems. Our choice of diagonal $\mathbf{R}$ means that the assumptions of Theorems 7.8 and 7.9 hold when $\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$ and $\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$. We report the computed largest and smallest positive and negative eigenvalues of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ and $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ in Tables 7.1 and 7.2, respectively. All computed eigenvalues for observation networks a) and c) are shown in Figures 7.1 and 7.2. The change of the extreme eigenvalues of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ with $\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$ and $\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$

O.n.	$\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$	$\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$	$\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_{30}$
a)	$[-8.84, -1.10 \times 10^{-5}]$ $[1, 9.84]$	$[-82.92, -6.18 \times 10^{-1}]$ $[1; 83.92]$	$[-59.16, -5.5 \times 10^{-3}]$ $[1; 60.16]$
b)	$[-8.84, -7.12 \times 10^{-5}]$ $[1, 9.84]$	$[-149.03, -6.18 \times 10^{-1}]$ $[1, 150.03]$	$[-59.18, -3.30 \times 10^{-2}]$ $[1, 60.18]$
c)	$[-9.03, -5.00 \times 10^{-4}]$ $[1, 10.03]$	$[-411.10, -6.18 \times 10^{-1}]$ $[1, 412.10]$	$[-59.26, -1.98 \times 10^{-1}]$ $[1, 60.26]$

Table 7.1: Computed eigenvalue intervals of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ with three choices of $\widetilde{\mathbf{S}}^{-1}$ in $\mathcal{P}_{B,3}^{-1}$ for different observation networks (O.n.).

O.n.	$\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$	$\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$	$\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_{30}$
a)	$[-8.84, -1.10 \times 10^{-5}]$ $[1 + 2 \times 10^{-7}, 9.84]$	$[-6.96 \times 10^{3}, -6.18 \times 10^{-1}]$ $[1 + 10^{-4}, 1.62]$	$[-65.87, -5.51 \times 10^{-3}]$ $[1 + 10^{-4}, 60.10]$
b)	$[-8.84, -7.12 \times 10^{-5}]$ $[1 + 2 \times 10^{-7}, 9.84]$	$[-2.24 \times 10^{4}, -6.18 \times 10^{-1}]$ $[1 + 4 \times 10^{-5}, 1.62]$	$[-68.40, -3.31 \times 10^{-2}]$ $[1 + 3 \times 10^{-5}, 59.62]$
c)	$[-11.02, -5.00 \times 10^{-4}]$ $[1 + 10^{-7}, 8.27]$	$[-1.69 \times 10^{5}, -6.18 \times 10^{-1}]$ $[1 + 6 \times 10^{-6}, 1.62]$	$[-75.29, -2.12 \times 10^{-1}]$ $[1 + 6 \times 10^{-6}, 57.33]$

Table 7.2: As in Table 7.1, but for $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$.

when the number of observations increases is as described in Theorem 7.8. That is, the negative and largest positive eigenvalues move away from zero or stay unchanged while the smallest positive eigenvalue is equal to one. The change of the extreme eigenvalues of $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$ agrees with Theorem 7.9; the negative eigenvalues move away from zero or stay unchanged and the positive ones approach zero or stay unchanged. The change when using $\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_k$ agrees with the results in Theorems 7.8 and 7.9 even though $\widetilde{\mathbf{S}}^{-1}$ is influenced by the observation information.

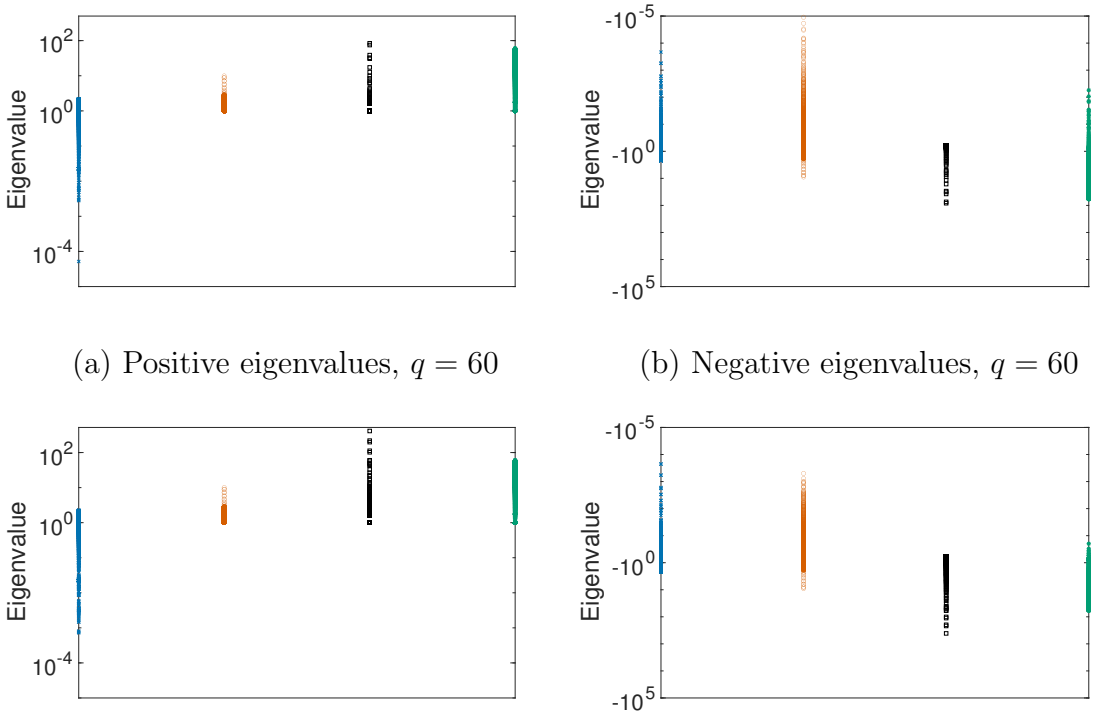
In the case of the preconditioner with the exact linearised model, the modulus of the largest positive and negative eigenvalues of $\mathcal{P}_{B,3}^{-1}\mathcal{A}_3$ grows more than with other choices of $\widetilde{\mathbf{S}}^{-1}$ when the number of observations is increased. The same is observed for the modulus largest negative eigenvalues of $\mathcal{P}_{B,2}^{-1}\mathcal{A}_2$. This may result in poor performance of the preconditioner that uses the exact linearised model when the dynamical system has many observations.

7.6.2 Solving the preconditioned systems

To explore the performance of the preconditioners we consider larger systems and set $n = 120$ and $N = 149$. The observations are taken at the same frequencies as in the previous section giving the the total number of observations q:

a) $q = 304$,

b) $q = 1125$,



(a) Positive eigenvalues, $q = 60$

(b) Negative eigenvalues, $q = 60$

(c) Positive eigenvalues, $q = 800$

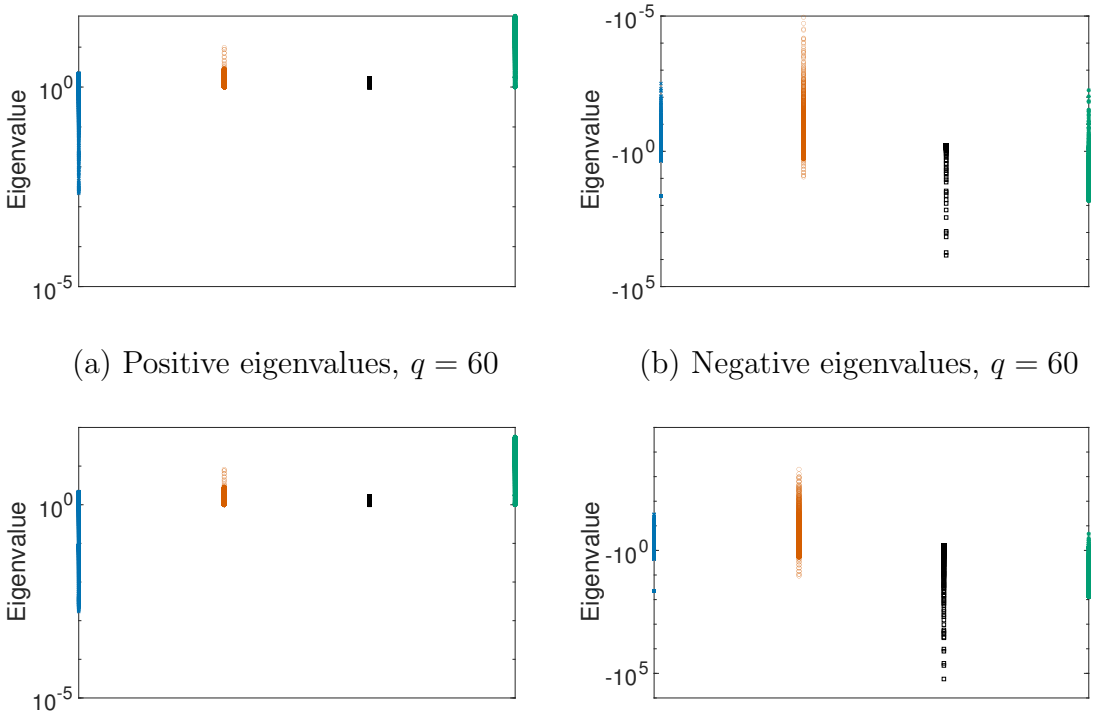
(d) Negative eigenvalues, $q = 800$

Figure 7.1: Eigenvalues of the unpreconditioned (blue) and preconditioned 3×3 block systems that have q observations. Preconditioner $\mathcal{P}_{B,3}^{-1}$ is constructed using $\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$ (vermilion), $\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$ (black), and $\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_{30}$ (green).

c) $q = 4500$.

MINRES is run for 1000 iterations and the norm of the relative residual $\|\mathbf{r}_j\|/\|\mathbf{r}_0\|$, where $\mathbf{r}_j$ is the residual at jth iteration and $\|\cdot\|$ is the L_2 norm, stays above 10^{-4} for all the systems. We choose 1000 iterations to explore how the preconditioning affects the early iterations, that are the most important if the solvers is stopped after a fixed number of iterations, and to see the effect on the later iterations when the quadratic cost function is expected to approach its minimum value. REVD_*ritzit* is run five times with different initialisations.

The change of the quadratic cost function value during the solution process is reported in Figure 7.3. Preconditioning using $\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_{30}$, that is, the randomised LMP constructed with thirty eigenpairs, is useful for both 3×3 block and 2×2 block systems with all three observation networks compared to the unpreconditioned case. Notice that the performance is not sensitive to the initialisation of REVD_*ritzit*. Preconditioners constructed using $\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$ are in general detrimental to the convergence or perform similarly to the unpreconditioned case. The performance of the preconditioner that uses the exact linearised model, i.e., $\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$, depends on the total number of observations. It outperforms other preconditioners and no preconditioning when there are very few observations, but performs worse than these when the number of observations is increased.



(a) Positive eigenvalues, $q = 60$

(b) Negative eigenvalues, $q = 60$

(c) Positive eigenvalues, $q = 800$

(d) Negative eigenvalues, $q = 800$

Figure 7.2: As in Figure 7.1, but for the 2×2 block system using $\mathcal{P}_{B,2}^{-1}$.

This agrees with the results in the previous section, where we saw the large growth of the modulus largest eigenvalues when using $\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$.

Notice the initial jump in the quadratic cost function value in the unpreconditioned cases for both 3×3 block and 2×2 block systems and when using $\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_{30}$ for the 2×2 block system in the $p = 304$ and $p = 1125$ cases. It reinforces the need for caution if early termination of MINRES is needed.

7.7 Conclusions

We have introduced a new way to include the observation information in the approximation of the inverse Schur complement, which is used in a block diagonal preconditioner for the saddle point systems in incremental weak constraint 4D-Var. This information has been excluded in the previously used preconditioners [Gratton et al., 2018a, Freitag and Green, 2018, Tabeart and Pearson, 2021]. The new approximation of the inverse Schur complement is constructed by employing randomised limited memory preconditioners, which have been used to precondition the symmetric positive definite forcing formulation [Daužickaitė et al., 2021b]. These preconditioners are cheap to construct and apply. Numerical experiments have shown that the randomised preconditioner is useful and outperforms other block diagonal preconditioners, especially when the number of observations is high.

We have provided theoretical results that show the relationship between the eigenvalues of preconditioned saddle point systems and those of the symmetric positive definite

formulations. The results depend on the choice of the inverse Schur complement approximation and can be used to better understand the sensitivities of the preconditioned systems. The change of the extreme eigenvalues due to introduction of new observations has also been examined in the case of uncorrelated observation errors and when the Schur complement approximation excludes observation information. The extreme eigenvalues of the preconditioned 3×3 block system either move away from zero or stay unchanged when new observations are introduced. This is also true for the extreme negative eigenvalues of the preconditioned 2×2 block system, but the extreme positive eigenvalues may approach zero. Further preconditioning for the small positive eigenvalues may be needed.

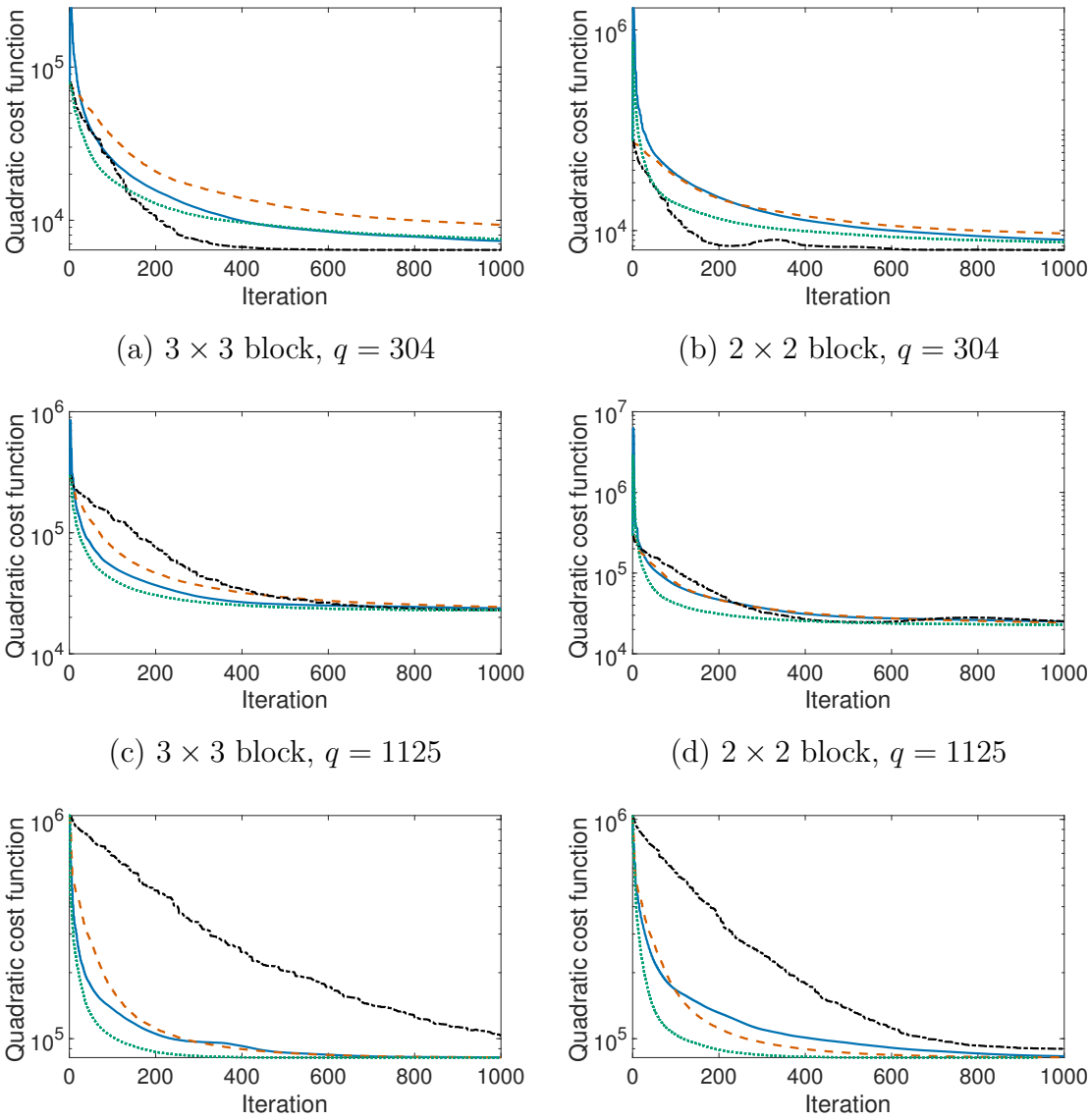
The block diagonal preconditioners are advantageous, because MINRES can be used to solve the preconditioned systems. It has been shown that inexact constraint preconditioners, which are not symmetric positive definite and require using other solvers, for example, GMRES instead of MINRES, may accelerate the convergence when solving the 3×3 block saddle point system [Gratton et al., 2018a, Tabeart and Pearson, 2021]. Using randomised methods to improve the approximations in these preconditioners is the scope of future research.

7.8 Summary

We proposed a new way to include the observation information in the block diagonal Schur preconditioners for the saddle point matrices and the numerical experiments with an idealised model showed that this technique is effective. The improvement is greater when the number of observations of the dynamical system is increased.

Theoretical results regarding the eigenvalues of the preconditioned matrices were presented. We showed how the eigenvalues of the preconditioned 3×3 block saddle point matrix are connected to the eigenvalues of the SPD state matrix. Similar sensitivities thus can be expected when solving both systems and this can be exploited when designing alternative preconditioning strategies. We explored how the extreme eigenvalues of the preconditioned systems change when new observations are added. The extreme eigenvalues of the preconditioned 3×3 block saddle point matrix either move away from zero or stay unchanged, which agrees with the results for the SPD state system in Chapter 5. Note that this solves the problem of the small positive eigenvalues of the unpreconditioned 3×3 block matrix. This problem is not solved for the preconditioned 2×2 block system though: the negative eigenvalues move away from zero or stay unchanged whereas the positive eigenvalues move towards zero or stay unchanged. Further preconditioning to address the small positive eigenvalues may be necessary. The largest eigenvalues of the SPD matrix in the forcing formulation move away from zero or stay unchanged and the CVT ensures that the smallest eigenvalues remain equal to one, hence expanding the spectrum and increasing the condition number. These results for SPD and saddle point systems hold when the observation error matrix is diagonal and the Schur approximation does not include observation information.

We conclude the thesis and propose areas of further research in the following chapter.



(a) 3×3 block, $q = 304$

(b) 2×2 block, $q = 304$

(c) 3×3 block, $q = 1125$

(d) 2×2 block, $q = 1125$

(e) 3×3 block, $q = 4500$

(f) 2×2 block, $q = 4500$

Figure 7.3: Quadratic cost function value at every MINRES iteration when solving the unpreconditioned (blue solid line) and preconditioned 3×3 block and 2×2 block systems that have q observations. Preconditioners $\mathcal{P}_{B,3}^{-1}$ and $\mathcal{P}_{B,2}^{-1}$ are constructed using $\widetilde{\mathbf{S}}^{-1} = \mathbf{D}$ (vermilion dashed), $\widetilde{\mathbf{S}}^{-1} = \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}$ (black dash-dotted), and $\widetilde{\mathbf{S}}^{-1} = \mathbf{P}_{30}$ (green dotted) for five different initialisations of the randomised method.

Chapter 8

Conclusions and future work

8.1 Conclusions

The data assimilation method weak constraint 4D-Var can be advantageous compared to other variational methods, because it accounts for the model error [Sasaki, 1970, Trémolet, 2006, Laloyaux et al., 2020a] and its so-called state formulation has potential for time-parallelism [Fisher and Gürol, 2017]. If the model error is accounted for, then longer assimilation window can be used and hence more observations can be assimilated [Trémolet, 2006]. The time-parallelism for large systems is important, because the resolution of the models is ever increasing and a smaller time step is needed for numerical stability, and the arising additional computations need to be parallelised to finish the assimilation process in the given wall-clock time [Fisher and Gürol, 2017].

A series of quadratic cost functions have to be minimised to approximate the solution of the weak constraint 4D-Var. Each minimisation is called an inner loop. In this thesis, we considered the state and forcing formulations of the method, and four different systems of linear equations that can be solved to obtain the minimisers of the cost functions, namely the standard symmetric positive definite (SPD) systems in the forcing and state formulations, and the 3×3 block saddle point system of [Fisher and Gürol, 2017] and the reduced 2×2 block saddle point system in the state formulation. The choice of the linear system should depend on how much computational resources for parallel computations are available; the systems ordered by the increasing potential for parallelism are

- forcing SPD system;
- state SPD system;
- 2×2 block saddle point system;
- 3×3 block saddle point system.

The SPD systems are of the same size in the forcing and state formulation, but the 2×2 block system is twice the size of these and the 3×3 block system grows even larger. Availability of estimates of the inverses of error covariance matrices can be taken into account too: the SPD systems include inverses of the error covariance matrices, the 2×2

block system requires inverse of only the observation error covariance matrix, the 3×3 block system does not need any inverses. The systems show different sensitivities to adding new observations [El-Said, 2015].

Independently of which system is chosen, iterative solution methods are used and they require preconditioning for efficient performance (e.g., [Trefethen and Bau, III, 1997, Saad, 2003, Wathen, 2015]). The design of suitable preconditioners is a challenging and important task in data assimilation [Fisher, 1998, Tshimanga et al., 2008, Fisher and Gürol, 2017, Gratton et al., 2018a, Gratton et al., 2018b, Fisher et al., 2018, Freitag and Green, 2018, Tabeart and Pearson, 2021]. In our work, we employed randomised methods for low-rank matrix approximations [Halko et al., 2011, Martinsson and Tropp, 2020] to suggest new preconditioning approaches, where the construction and application of the preconditioner can be performed in parallel and hence efficiently on current computers. We further discuss how we addressed each research question, which was presented in Chapter 1.

Research question 1: How can we precondition the linear systems of equations arising in the forcing formulation independently of the previously solved systems?
The systems in different inner loops change because of a different linearisations of the model and the observation operator, but they can also be affected by other changes, such as introducing new observations and increasing the resolution of the model in the later inner loops as is done in ECMWF [Lean et al., 2021]. If the changes to the subsequent systems are large enough, the usefulness of limited memory preconditioners (LMPs) [Fisher, 1998, Tshimanga et al., 2008, Gratton et al., 2011] can be affected, because they are constructed using cheaply obtained estimates of the eigenpairs from the previous inner loops. LMPs are currently used in, for example, ocean data assimilation [Mogensen et al., 2012, Moore et al., 2011] and climate reanalysis [Laloyaux et al., 2018].

We addressed this in Chapter 4 by proposing to use randomised eigenvalue decomposition (REVD) to construct LMPs, which we then called randomised LMPs. The eigenpairs can be obtained at the beginning of each inner loop independently of the previous inner loops. We performed idealised numerical experiments and observed the following.

- The randomised LMPs improve the minimisation compared to using no preconditioning and are more effective than the LMPs constructed with eigenpair estimates obtained in the previous loop (deterministic LMPs). This holds even if the randomised LMPs are constructed with fewer eigenpairs than the deterministic LMPs.

- The effectiveness of the randomised LMPs grows when they are constructed using more eigenpairs.

- Out of three REVD methods tested, REVD_*ritzit* performed the best.

- Large oversampling is not necessary for the randomised methods, although it can further reduce the small variation of the LMPs performance in the first iterations of the iterative solver.

Research question 2: How do the extreme eigenvalues of the coefficient matrices change when new observations are introduced?

The convergence of Krylov subspace solvers CG and MINRES can be described by the eigenvalue distribution of the coefficient matrix [Trefethen and Bau, III, 1997]. We analysed how the extreme eigenvalues of the SPD matrices and saddle point matrices change when new observations are added in Chapters 5 and 7. The change for the saddle point matrices preconditioned using a block diagonal preconditioner was also investigated in the latter. We provided bounds for the eigenvalues of the unpreconditioned coefficient matrices in the state formulation, and showed the relationship between the eigenvalues of the preconditioned saddle matrices and SPD coefficient matrices. We showed the following, where the results for the unpreconditioned 3×3 block coefficient matrix hold for general observation error covariance matrix $\mathbf{R}$, and a diagonal $\mathbf{R}$, that is, uncorrelated observation errors, is assumed for other systems. The Schur complement approximation in the block diagonal preconditioner is assumed to contain no information on the observations.

- The change when new observations are introduced:
  - The largest positive eigenvalues of the SPD forcing coefficient matrix with the control variable transform (CVT) move away from zero or stay unchanged, and the smallest positive eigenvalue stays at one when the system is not fully observed.
  - The extreme eigenvalues of the SPD state coefficient matrix move away from zero or stay unchanged.
  - The extreme negative eigenvalues of the unpreconditioned 2×2 block coefficient matrix move away from zero or stay unchanged, and the extreme positive eigenvalues move towards zero or stay unchanged.
  - The extreme negative and largest positive eigenvalues of the unpreconditioned 3×3 block coefficient matrix move away from zero or stay unchanged, while the smallest positive eigenvalues move towards zero or stay unchanged.
  - The extreme eigenvalues of the preconditioned 2×2 block coefficient matrix change in the same way as in the unpreconditioned case.
  - The extreme eigenvalues of the preconditioned 3×3 block coefficient matrix move away from zero or stay unchanged.
- Eigenvalue bounds:
  - The provided bounds depend on the extreme eigenvalues of the error covariance matrices and the singular values of the matrix including the linearised model and the linearised observation operator.
  - The bounds for the saddle point coefficient matrices are tight, whereas the bounds for the state SPD coefficient matrix may be too pessimistic.
- We showed the direct relationship between the eigenvalues of the preconditioned 3×3 block matrix and the preconditioned SPD state coefficient matrix.

- We showed the direct relationship between the eigenvalues of the preconditioned 2×2 block matrix and the eigenvalues of a matrix involving the interaction between the model and observation terms.

Research question 3: How can we precondition the linear systems in the state formulation so that the potential for time-parallel computations is preserved?
Potential for time-parallel model integration is embedded in the state formulation [Fisher and Gürol, 2017]. Fisher and Gürol suggested approximating the CVT technique for the SPD state system, but did not find an effective and parallelism preserving approximation. We proposed using a randomised singular value decomposition to approximate the matrix involving the linearised model, and this matrix in interaction with the background and model error covariance matrices in Chapter 6. Although the construction of these preconditioners is not time-parallel, their application is cheap and preserves the time-parallelism when solving the systems. Our numerical experiments with idealised system showed the following.

- The exact CVT technique is not always useful compared to using no preconditioning, especially if there are many high quality observations of the dynamical system.

- If the exact CVT techniques is useful, then its randomised approximation improves the minimisation in the first iterations.

- Including the background and model error covariance matrices in the CVT approximation is particularly useful when the model error is large.

- Including the background and model error covariance matrices in the CVT approximation may result in preconditioner that is less sensitive to the random initialisation of RSVD.

Research question 4: How can we include more information about the observations when preconditioning the saddle point systems?
Block diagonal Schur preconditioners were used previously to precondition the 3×3 block system, but they included the approximation to the inverse Schur complement without the observation term [Gratton et al., 2018a, Freitag and Green, 2018, Tabeart and Pearson, 2021]. We proposed a new way to approximate the inverse Schur complement, namely using the randomised LMP to construct an approximation that does not exclude the observation term in Chapter 7. Such an approximation can be also used when preconditioning the 2×2 block system. The randomised LMP can be generated in a time-parallel way and the application is cheap. We performed experiments with a simple system and observed the following.

- Our proposed preconditioner is more effective than using no preconditioning.

- The new preconditioner outperforms other preconditioners when the number of observations is high.

- Using the exact model in the preconditioner may be detrimental when there are many observations.

We discuss the relevance of the results. The fact that the largest eigenvalues of the forcing coefficient matrix with CVT can grow when new observation are added, reinforces the importance of generating the LMPs independently of the previous inner loops. Note that these preconditioners are constructed with estimates of the largest eigenvalues and the relevance of the estimates obtained in the previous inner loop can hence decrease if new observations are added in the later loops. The increase of the largest eigenvalue while the smallest eigenvalue stays equal to one means that the spectrum of the forcing coefficient matrix with CVT expands and thus its condition number grows. This agrees with our observation that using the CVT for the state SPD system, and in this way converting it to the forcing formulation with CVT, may be detrimental when the number of observations is increased.

The bounds for the eigenvalues of the unpreconditioned saddle point matrices suggest that observation information should be included in the preconditioning approaches. The effectiveness of our new preconditioner with observation information in the Schur complement supports this. Our preconditioner is very effective when the number of observations is high. This is useful because the theoretical results show that the unpreconditioned systems may be harder to solve when the observation number is increased because of the small positive eigenvalues approaching zero. Note that we showed that when block diagonal preconditioner is used, this problem is eliminated for the 3×3 block coefficient matrix. The relationships between the eigenvalues of the preconditioned saddle point matrices and the SPD matrices can be helpful to better understand other sensitivities of the systems.

8.2 Future work

We present research questions for future work.

- When is the CVT technique useful? Experiments with simple systems showed that this technique may not be useful in the case of many accurate observations (Chapter 6), and this is confirmed by the theoretical result on the eigenvalue change when CVT is used and the observation error covariance matrix is diagonal (Chapter 7). Further theoretical and experimental explorations with more realistic systems are needed.

- How should the state SPD formulation be preconditioned when the number of observations is high? Our suggested randomised approximation to the CVT is useful in the cases when the exact CVT is useful, that is, when the number of observations of the dynamical system is low (Chapter 6). The increasing number of observations may require different preconditioning approaches that still allow the time-parallelism.

- How should we precondition the state SPD formulation if a large number of iterations can be run? The randomised approach in Chapter 6 proved to be useful in the first

iterations, which are the most important, but the iterative solver stagnates in the later iterations. In case a very large number of iterations has to be run, the solver may require restarting with a different preconditioner.

- How can the 2×2 block system be preconditioned so that the positive eigenvalues do not approach zero when new observations are added? The block diagonal preconditioners (Chapter 7) did not fix this problem and the small eigenvalues can cause convergence issues.

- How do the extreme eigenvalues of the coefficient matrices change when the observation error covariance matrix is correlated? Our theoretical results consider a diagonal observation error covariance matrix (Chapters 5 and 7), but there is a growing amount of work on using the correlated matrices (e.g., [Stewart et al., 2008, Weston et al., 2014, Tabeart et al., 2020]). Bounds for the eigenvalues of the 3×3 block system preconditioned using a block diagonal preconditioner with approximation to the correlated observation error covariance matrix are presented by [Tabeart and Pearson, 2021].

- Finally, are the proposed preconditioners useful for realistic systems? Our numerical tests considered idealised systems in a sequential environment, and did not take into account the cost of generating and applying the preconditioners. Further tests on large systems in parallel environments are needed to evaluate the effects on the run time and energy consumption (e.g., [Carson and Strakoš, 2020, Bousserez et al., 2020]).

Bibliography

[Allen et al., 2006] Allen, M., Frame, D., Kettleborough, J., and Stainforth, D. (2006). Model error in weather and climate forecasting. In Palmer, T. and Hagedorn, R., editors, *Predictability of Weather and Climate*, pages 391 – 427. Cambridge University Press, Cambridge, UK.

[Andersson and Thépaut, 2010] Andersson, E. and Thépaut, J.-N. (2010). Assimilation of operational data. In Lahoz, W., Khattatov, B., and Menard, R., editors, *Data Assimilation. Making Sense of Observations*, pages 283 – 299. Springer-Verlag, Berlin, Heidelberg/Germany.

[Arcucci et al., 2018] Arcucci, R., Pain, C., and Guo, Y. (2018). Effective variational data assimilation in air-pollution prediction. *Big Data Mining and Analytics*, 1(4):297 – 307.

[Axelsson and Neytcheva, 2006] Axelsson, O. and Neytcheva, M. (2006). Eigenvalue estimates for preconditioned saddle point matrices. *Numerical Linear Algebra with Applications*, 13(4):339 – 360.

[Bannister, 2008a] Bannister, R. N. (2008a). A review of forecast error covariance statistics in atmospheric variational data assimilation. I: Characteristics and measurements of forecast error covariances. *Quarterly Journal of the Royal Meteorological Society*, 134(637):1951 – 1970.

[Bannister, 2008b] Bannister, R. N. (2008b). A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics. *Quarterly Journal of the Royal Meteorological Society*, 134(637):1971 – 1996.

[Bauer et al., 2021] Bauer, P., Dueben, P. D., Hoefler, T., Quintino, T., Schulthess, T. C., and Wedi, N. P. (2021). The digital revolution of Earth-system science. *Nature Computational Science*, 1:104 – 113.

[Bauer et al., 2015] Bauer, P., Thorpe, A., and Brunet, G. (2015). The quiet revolution of numerical weather prediction. *Nature*, 525:47 – 55.

[Benzi, 2002] Benzi, M. (2002). Preconditioning techniques for large linear systems: A survey. *Journal of Computational Physics*, 182:418 – 477.

[Benzi et al., 2005] Benzi, M., Golub, G. H., and Liesen, J. (2005). Numerical solution of saddle point problems. *Acta Numerica*, 14:1 – 137.

[Benzi and Wathen, 2008] Benzi, M. and Wathen, A. J. (2008). Some preconditioning techniques for saddle point problems. In Schilders, W. H. A., van der Vorst, H. A., and Rommes, J., editors, *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry (The European Consortium for Mathematics in Industry)*, pages 195 – 212. Springer, Berlin, Heidelberg/Germany.

[Bergamaschi et al., 2007] Bergamaschi, L., Gondzio, J., Venturin, M., and Zilli, G. (2007). Inexact constraint preconditioners for linear systems arising in interior point methods. *Computational Optimization and Applications*, 36(2):137 – 147.

[Bergamaschi et al., 2011] Bergamaschi, L., Gondzio, J., Venturin, M., and Zilli, G. (2011). Erratum to: Inexact constraint preconditioners for linear systems arising in interior point methods. *Computational Optimization and Applications*, 49:401 – 406.

[Berry et al., 2005] Berry, M. W., Mezher, D., and Ahmed Sameh, B. P. (2005). Parallel algorithms for the singular value decomposition. In Kontoghiorghes, E. J., editor, *Handbook of Parallel Computing and Statistics*, pages 117 – 164. Chapman and Hall/CRC, New York, NY.

[Bonavita and Lean, 2021] Bonavita, M. and Lean, P. (2021). 4D-Var for numerical weather prediction. *Weather*, 76(2):65 – 66.

[Bonavita et al., 2017] Bonavita, M., Trémolet, Y., Hólm, E., Lang, S. T. K., Chrust, M., Janiskova, M., Lopez, P., Laloyaux, P., de Rosnay, P., Fisher, M., Hamrud, M., and English, S. (2017). A strategy for data assimilation. *ECMWF Technical Memoranda*, (800).

[Bousserez et al., 2020] Bousserez, N., Guerrette, J. J., and Henze, D. K. (2020). Enhanced parallelization of the incremental 4D-Var data assimilation algorithm using the Randomized Incremental Optimal Technique. *Quarterly Journal of the Royal Meteorological Society*, 146(728):1351 – 1371.

[Bousserez and Henze, 2018] Bousserez, N. and Henze, D. K. (2018). Optimal and scalable methods to approximate the solutions of large-scale Bayesian problems: theory and application to atmospheric inversion and data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(711):365 – 390.

[Bowler, 2017] Bowler, N. E. (2017). On the diagnosis of model error statistics using weak-constraint data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(705):1916 – 1928.

[Butcher, 1987] Butcher, J. C. (1987). *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods.* Wiley-Interscience, Chichester, UK.

[Carson and Strakoš, 2020] Carson, E. and Strakoš, Z. (2020). On the cost of iterative computations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378(2166):20190050.

[Chen et al., 2013] Chen, H., Yang, D., Hong, Y., Gourley, J. J., and Zhang, Y. (2013). Hydrological data assimilation with the Ensemble Square-Root-Filter: Use of streamflow observations to update model states for real-time flash flood forecasting. *Advances in Water Resources*, 59:209 – 220.

[Clayton et al., 2013] Clayton, A. M., Lorenc, A. C., and Barker, D. M. (2013). Operational implementation of a hybrid ensemble/4D-Var global data assimilation system at the Met Office. *Quarterly Journal of the Royal Meteorological Society*, 139(675):1445 – 1461.

[Coiffier, 2011] Coiffier, J. (2011). *Fundamentals of Numerical Weather Prediction*. Cambridge University Press, Cambridge, UK.

[Courtier et al., 1994] Courtier, P., Thépaut, J.-N., and Hollingsworth, A. (1994). A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367 – 1387.

[Daley, 1993] Daley, R. (1993). *Atmospheric Data Analysis*, volume 2. Cambridge University Press, Cambridge, UK.

[Daužickaitė et al., 2020] Daužickaitė, I., Lawless, A. S., Scott, J. A., and van Leeuwen, P. J. (2020). Spectral estimates for saddle point matrices arising in weak constraint four-dimensional variational data assimilation. *Numerical Linear Algebra with Applications*, 27(5):e2313.

[Daužickaitė et al., 2021a] Daužickaitė, I., Lawless, A. S., Scott, J. A., and van Leeuwen, P. J. (2021a). On time-parallel preconditioning for the state formulation of incremental weak constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 147(740):3521 – 3529.

[Daužickaitė et al., 2021b] Daužickaitė, I., Lawless, A. S., Scott, J. A., and van Leeuwen, P. J. (2021b). Randomised preconditioning for the forcing formulation of weak constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 147(740):3719 – 3734.

[Dee et al., 2011] Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M. A., Balsamo, G., Bauer, P., Bechtold, P., Beljaars, A. C. M., van de Berg, L., Bidlot, J., Bormann, N., Delsol, C., Dragani, R., Fuentes, M., Geer, A. J., Haimberger, L., Healy, S. B., Hersbach, H., Hólm, E. V., Isaksen, L., Kållberg, P., Köhler, M., Matricardi, M., McNally, A. P., Monge-Sanz, B. M., Morcrette, J.-J., Park, B.-K., Peubey, C., de Rosnay, P., Tavolato, C., Thépaut, J.-N., and Vitart, F. (2011). The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137(656):553 – 597.

[Demmel et al., 2012] Demmel, J., Grigori, L., Hoemmen, M., and Langou, J. (2012). Communication-optimal parallel and sequential QR and LU factorizations. *SIAM Journal on Scientific Computing*, 34(1):A206 – A239.

[Doong et al., 2012] Doong, D.-J., Chuang, L. Z.-H., Wu, L.-C., Fan, Y.-M., Kao, C. C., and Wang, J.-H. (2012). Development of an operational coastal flooding early warning system. *Natural Hazards and Earth System Sciences*, 12(2):379 – 390.

[Duff and Scott, 1993] Duff, I. S. and Scott, J. A. (1993). Computing selected eigenvalues of sparse unsymmetric matrices using subspace iteration. *ACM Transactions on Mathematical Software*, 19(2):137 –159.

[ECMWF, 2020] ECMWF (2020). *Part II: Data Assimilation.* Number 2 in IFS Documentation. European Centre for Medium Range Weather Forecasts. Available on `https://www.ecmwf.int/node/19746`.

[El-Said, 2015] El-Said, A. (2015). *Conditioning of the weak-constraint variational data assimilation problem for numerical weather prediction.* PhD thesis, Department of Mathematics and Statistics, University of Reading.

[Elbern et al., 1997] Elbern, H., Schmidt, H., and Ebel, A. (1997). Variational data assimilation for tropospheric chemistry modeling. *Journal of Geophysical Research: Atmospheres*, 102(D13):15967 – 15985.

[Evensen et al., 2021] Evensen, G., Amezcua, J., Bocquet, M., Carrassi, A., Farchi, A., Fowler, A., Houtekamer, P. L., Jones, C. K., de Moraes, R. J., Pulido, M., Sampson, C., and Vossepoel, F. C. (2021). An international initiative of predicting the Sars-Cov-2 pandemic using ensemble data assimilation. *Foundations of Data Science*, 3(3):413 – 477.

[Ferronato, 2012] Ferronato, M. (2012). Preconditioning for sparse linear systems at the dawn of the 21st century: History, current developments, and future perspectives. *ISRN Applied Mathematics*, 2012.

[Fisher, 1998] Fisher, M. (1998). Minimization algorithms for variational data assimilation. In *Proceedings of the Seminar on Recent Developments in Numerical Methods for Atmospheric Modelling*, pages 364 – 385, Reading, UK. European Centre for Medium Range Weather Forecasts.

[Fisher et al., 2018] Fisher, M., Gratton, S., Gürol, S., Trémolet, Y., and Vasseur, X. (2018). Low rank updates in preconditioning the saddle point systems arising from data assimilation problems. *Optimization Methods and Software*, 33(1):45 – 69.

[Fisher and Gürol, 2017] Fisher, M. and Gürol, S. (2017). Parallelisation in the time dimension of four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):1136 – 1147.

[Fisher et al., 2009] Fisher, M., Nocedal, J., Trémolet, Y., and Wright, S. J. (2009). Data assimilation in weather forecasting: a case study in PDE-constrained optimization. *Optimization and Engineering*, 10:409 – 426.

[Freitag, 2020] Freitag, M. A. (2020). Numerical linear algebra in data assimilation. *GAMM-Mitteilungen*, 43(3):e202000014.

[Freitag and Green, 2018] Freitag, M. A. and Green, D. L. H. (2018). A low-rank approach to the solution of weak constraint variational data assimilation problems. *Journal of Computational Physics*, 357:263 – 281.

[García-Pintado et al., 2015] García-Pintado, J., Mason, D. C., Dance, S. L., Cloke, H. L., Neal, J. C., Freer, J., and Bates, P. D. (2015). Satellite-supported flood forecasting in river networks: A real case study. *Journal of Hydrology*, 523:706 – 724.

[Gauthier et al., 2007] Gauthier, P., Tanguay, M., Laroche, S., Pellerin, S., and Morneau, J. (2007). Extension of 3Dvar to 4Dvar: Implementation of 4Dvar at the meteorological service of Canada. *Monthly Weather Review*, 133(637):2339 – 2354.

[Giraud and Gratton, 2006] Giraud, L. and Gratton, S. (2006). On the sensitivity of some spectral preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 27(4):1089 – 1105.

[Golub and Van Loan, 2013] Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. JHU Press, Baltimore, MD, 4th edition.

[Gratton et al., 2018a] Gratton, S., Gürol, S., Simon, E., and Toint, P. L. (2018a). Guaranteeing the convergence of the saddle formulation for weakly-constrained 4D-Var data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(717):2592 – 2602.

[Gratton et al., 2018b] Gratton, S., Gürol, S., Simon, E., and Toint, P. L. (2018b). A note on preconditioning weighted linear least-squares, with consequences for weakly constrained variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(712):934 – 940.

[Gratton et al., 2007] Gratton, S., Lawless, A. S., and Nichols, N. K. (2007). Approximate Gauss-Newton methods for nonlinear least squares problems. *SIAM Journal on Optimization*, 18(1):106 – 132.

[Gratton et al., 2011] Gratton, S., Sartenaer, A., and Tshimanga, J. (2011). On a class of limited memory preconditioners for large scale linear systems with multiple right-hand sides. *SIAM Journal on Optimization*, 21(3):912 – 935.

[Greenbaum, 1997] Greenbaum, A. (1997). *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, PA.

[Greenbaum et al., 1996] Greenbaum, A., Pták, V., and Strakoš, Z. (1996). Any nonincreasing convergence curve is possible for GMRES. *SIAM Journal on Matrix Analysis and Applications*, 17(3):465 – 469.

[Greif et al., 2014] Greif, C., Moulding, E., and Orban, D. (2014). Bounds on eigenvalues of matrices arising from interior-point methods. *SIAM Journal on Optimization*, 24(1):49 – 83.

[Griffith and Nichols, 2000] Griffith, A. and Nichols, N. K. (2000). Adjoint methods in data assimilation for estimating model error. *Flow, Turbulence and Combustion*, 65(3 - 4):469 – 488.

[Gu, 2015] Gu, M. (2015). Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):A1139 – A1173.

[Haben, 2011] Haben, S. A. (2011). *Conditioning and preconditioning of the minimisation problem in variational data assimilation.* PhD thesis, Department of Mathematics and Statistics, University of Reading.

[Haben et al., 2011a] Haben, S. A., Lawless, A. S., and Nichols, N. K. (2011a). Conditioning and preconditioning of the variational data assimilation problem. *Computers and Fluids*, 46:252 – 256.

[Haben et al., 2011b] Haben, S. A., Lawless, A. S., and Nichols, N. K. (2011b). Conditioning of incremental variational data assimilation, with application to the Met Office system. *Tellus A: Dynamic Meteorology and Oceanography*, 64(4):782 – 792.

[Halko et al., 2011] Halko, N., Martinsson, P., and Tropp, J. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217 – 288.

[Hestenes and Stiefel, 1952] Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409 – 436.

[Horn and Johnson, 2012] Horn, R. A. and Johnson, C. R. (2012). *Matrix Analysis.* Cambridge University Press, Cambridge, UK, 2 edition.

[Janjić et al., 2018] Janjić, T., Bormann, N., Bocquet, M., Carton, J. A., Cohn, S. E., Dance, S. L., Losa, S. N., Nichols, N. K., Potthast, R., Waller, J. A., and Weston, P. (2018). On the representation error in data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(713):1257 – 1278.

[Johnson et al., 2005] Johnson, C., Hoskins, B. J., and Nichols, N. K. (2005). A singular vector perspective of 4D-Var: Filtering and interpolation. *Quarterly Journal of the Royal Meteorological Society*, 131(605):1 – 19.

[Kalnay, 2002] Kalnay, E. (2002). *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, Cambridge, UK.

[Kostka et al., 2014] Kostka, P. M., Weissmann, M., Buras, R., Mayer, B., and Stiller, O. (2014). Observation operator for visible and near-infrared satellite reflectances. *Journal of Atmospheric and Oceanic Technology*, 31(6):1216 – 1233.

[Kovats and Ebi, 2006] Kovats, R. S. and Ebi, K. L. (2006). Heatwaves and public health in Europe. *European Journal of Public Health*, 16(6):592 – 599.

[Laloyaux et al., 2020a] Laloyaux, P., Bonavita, M., Chrust, M., and Gürol, S. (2020a). Exploring the potential and limitations of weak-constraint 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 146(733):4067 – 4082.

[Laloyaux et al., 2020b] Laloyaux, P., Bonavita, M., Dahoui, M., Farnan, J., Healy, S., Hólm, E., and Lang, S. T. K. (2020b). Towards an unbiased stratospheric analysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):2392 – 2409.

[Laloyaux et al., 2018] Laloyaux, P., Frolov, S., Ménétrier, B., and Bonavita, M. (2018). Implicit and explicit cross-correlations in coupled data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(715):1851 – 1863.

[Laroche and Gauthier, 1998] Laroche, S. and Gauthier, P. (1998). A validation of the incremental formulation of 4D variational data assimilation in a nonlinear barotropic flow. *Tellus A*, 50(5):557 – 572.

[Lawless, 2013] Lawless, A. S. (2013). Variational data assimilation for very large environmental problems. In Cullen, M., Freitag, M. A., Kindermann, S., and Scheichl, R., editors, *Large Scale Inverse Problems: Computational Methods and Applications in the Earth Sciences*, Radon Series on Computational and Applied Mathematics 13, pages 55 – 90. De Gruyter, Berlin, Germany.

[Lawless and Nichols, 2006] Lawless, A. S. and Nichols, N. K. (2006). Inner-loop stopping criteria for incremental four-dimensional variational data assimilation. *Monthly Weather Review*, 134(11):3425 – 3435.

[Lawless et al., 2008] Lawless, A. S., Nichols, N. K., Boess, C., and Bunse-Gerstner, A. (2008). Using model reduction methods within incremental four-dimensional variational data assimilation. *Monthly Weather Review*, 136:1511 – 1522.

[Lean et al., 2021] Lean, P., Hólm, E. V., Bonavita, M., Bormann, N., McNally, A. P., and Järvinen, H. (2021). Continuous data assimilation for global numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 147(734):273 – 288.

[Leutbecher et al., 2017] Leutbecher, M., Lock, S.-J., Ollinaho, P., Lang, S. T. K., Balsamo, G., Bechtold, P., Bonavita, M., Christensen, H. M., Diamantakis, M., Dutra,

E., English, S., Fisher, M., Forbes, R. M., Goddard, J., Haiden, T., Hogan, R. J., Juricke, S., Lawrence, H., MacLeod, D., Magnusson, L., Malardel, S., Massart, S., Sandu, I., Smolarkiewicz, P. K., Subramanian, A., Vitart, F., Wedi, N., and Weisheimer, A. (2017). Stochastic representations of model uncertainties at ECMWF: state of the art and future vision. *Quarterly Journal of the Royal Meteorological Society*, 143(707):2315 – 2339.

[Liesen and Strakoš, 2013] Liesen, J. and Strakoš, Z. (2013). *Krylov subspace methods: principles and analysis. Numerical Mathematics and Scientific Computation.* Oxford University Press, Oxford, UK.

[Lorenc, 1992] Lorenc, A. C. (1992). Iterative analysis using covariance functions and filters. *Quarterly Journal of the Royal Meteorological Society*, 118:569 – 591.

[Lorenc et al., 2000] Lorenc, A. C., Ballard, S. P., Bell, R. S., Ingleby, N. B., Andrews, P. L. F., Barker, D. M., Bray, J. R., Clayton, A. M., Dalby, T., Li, D., Payne, T. J., and Saunders, F. W. (2000). The Met Office global three-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 126(570):2991 – 3012.

[Lorenz, 1996] Lorenz, E. (1996). Predictability - a problem partly solved. In *Proceedings of the Seminar on Predictability*, volume 1, pages 1 – 18, Reading, UK. European Centre for Medium Range Weather Forecasts.

[Lütkepohl, 1996] Lütkepohl, H. (1996). *Handbook of Matrices.* Wiley, Chichester, UK.

[Martinsson and Tropp, 2020] Martinsson, P. G. and Tropp, J. A. (2020). Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403 – 572.

[Mogensen et al., 2012] Mogensen, K., Alonso Balmaseda, M., and Weaver, A. (2012). The NEMOVAR ocean data assimilation system as implemented in the ECMWF ocean analysis for System 4. *ECMWF Technical Memoranda*, (668):59.

[Moore et al., 2011] Moore, A. M., Arango, H. G., Broquet, G., Powell, B. S., Weaver, A. T., and Zavala-Garay, J. (2011). The regional ocean modeling system (ROMS) 4-dimensional variational data assimilation systems: Part I - system overview and formulation. *Progress in Oceanography*, 91(1):34 – 49.

[Morini et al., 2016] Morini, B., Simoncini, V., and Tani, M. (2016). Spectral estimates for unreduced symmetric KKT systems arising from interior point methods. *Numerical Linear Algebra with Applications*, 23(5):776 – 800.

[Morini et al., 2017] Morini, B., Simoncini, V., and Tani, M. (2017). A comparison of reduced and unreduced KKT systems arising from interior point methods. *Computational Optimization and Applications*, 68(1):1 – 27.

[Morton and Mayers, 1994] Morton, K. W. and Mayers, D. (1994). *Numerical solution of partial differential equations*, volume 2. Cambridge University Press, Cambridge, UK.

[Moye and Diekman, 2018] Moye, M. J. and Diekman, C. O. (2018). Data assimilation methods for neuronal state and parameter estimation. *The Journal of Mathematical Neuroscience*, 8(11).

[Nakatsukasa, 2020] Nakatsukasa, Y. (2020). Fast and stable randomized low-rank matrix approximation. Available on `https://arxiv.org/abs/2009.11392`.

[Nichols, 2010] Nichols, N. K. (2010). Mathematical concepts of data assimilation. In Lahoz, W., Khattatov, B., and Menard, R., editors, *Data Assimilation. Making Sense of Observations*, pages 13 – 39. Springer-Verlag, Berlin, Heidelberg/Germany.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. World Scientific, New York, NY, 2nd edition.

[Paige and Saunders, 1975] Paige, C. C. and Saunders, M. A. (1975). Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617 – 629.

[Parlett, 1998] Parlett, B. N. (1998). *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia, PA.

[Pearson and Pestana, 2020] Pearson, J. W. and Pestana, J. (2020). Preconditioners for Krylov subspace methods: An overview. *GAMM-Mitteilungen*, 43(4).

[Rabier, 2005] Rabier, F. (2005). Overview of global data assimilation developments in numerical weather-prediction centres. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3215 – 3233.

[Rabier et al., 2000] Rabier, F., Järvinen, H., Klinker, E., Mahfouf, J.-F., and Simmons, A. (2000). The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1143 – 1170.

[Rawlins et al., 2007] Rawlins, F., Ballard, S. P., Bovis, K. J., Clayton, A. M., Li, D., Inverarity, G. W., Lorenc, A. C., and Payne, T. J. (2007). The Met Office global four-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 133(623):347 – 362.

[Rees and Wathen, 2009] Rees, T. and Wathen, A. J. (2009). Preconditioning iterative methods for the optimal control of the Stokes equation. *SIAM Journal on Scientific Computing*, 33(5):2903 – 2926.

[Rood, 2010] Rood, R. (2010). The role of the model in the data assimilation system. In Lahoz, W., Khattatov, B., and Menard, R., editors, *Data Assimilation. Making Sense of Observations*, pages 351 – 379. Springer-Verlag, Berlin, Heidelberg/Germany.

[Rusten and Winther, 1992] Rusten, T. and Winther, R. (1992). A preconditioned iterative method for saddlepoint problems. *SIAM Journal on Matrix Analysis and Applications*, 13(3):887 – 904.

[Rutishauser, 1971] Rutishauser, H. (1971). Contribution II/9: Simultaneous iteration method for symmetric matrices. In Wilkinson, J. H. and Reinsch, C., editors, *Handbook for Automatic Computation: Volume II: Linear Algebra*, Die Grundlehren der mathematischen Wissenschaften 186, pages 284 – 302. Springer-Verlag, Berlin, Heidelberg/Germany.

[Saad, 2003] Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, 2nd edition.

[Saad, 2011] Saad, Y. (2011). *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. SIAM, Philadelphia, PA.

[Saad and Schultz, 1986] Saad, Y. and Schultz, M. H. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856 – 869.

[Sasaki, 1970] Sasaki, Y. (1970). Some basic formalisms in numerical variational analysis. *Monthly Weather Review*, 98(12):875 – 883.

[Schnabel, 1983] Schnabel, R. (1983). Quasi-newton methods using multiple secant equations. Technical Report CU-CS-247-83, Department of Computer Science, University of Colorado at Boulder USA.

[Shaw and Daescu, 2017] Shaw, J. A. and Daescu, D. N. (2017). Sensitivity of the model error parameter specification in weak-constraint four-dimensional variational data assimilation. *Journal of Computational Physics*, 343:115 – 129.

[Silvester and Wathen, 1994] Silvester, D. and Wathen, A. (1994). Fast iterative solution of stabilised Stokes systems. Part II: using general block preconditioners. *SIAM Journal on Numerical Analysis*, 31(5):1352 – 1367.

[Silvester, 2000] Silvester, J. (2000). Determinants of block matrices. *Mathematical Gazette*, 84(501):460 – 467.

[Simoncini and Szyld, 2013] Simoncini, V. and Szyld, D. B. (2013). On the superlinear convergence of MINRES. In Cangiani, A., Davidchack, R., Georgoulis, E., Gorban, A., Levesley, J., and Tretyakov, M., editors, *Numerical Mathematics and Advanced Applications 2011. Proceedings of ENUMATH 2011*, pages 733 – 740. Springer.

[Stewart, 2001] Stewart, G. W. (2001). *Matrix Algorithms: Volume II: Eigensystems*. SIAM, Philadelphia, PA.

[Stewart and Sun, 1990] Stewart, G. W. and Sun, J.-G. (1990). *Matrix Perturbation Theory*. Computer Science and Scientific Computing. Academic Press, London, UK.

[Stewart et al., 2008] Stewart, L. M., Dance, S. L., and Nichols, N. K. (2008). Correlated observation errors in data assimilation. *International Journal for Numerical Methods in Fluids*, 56:1521 – 1527.

[Swinbank, 2010] Swinbank, R. (2010). Numerical weather prediction. In Lahoz, W., Khattatov, B., and Menard, R., editors, *Data Assimilation. Making Sense of Observations*, pages 381 – 406. Springer-Verlag, Berlin, Heidelberg/Germany.

[Tabeart et al., 2020] Tabeart, J. M., Dance, S. L., Lawless, A. S., Migliorini, S., Nichols, N. K., Smith, F., and Waller, J. A. (2020). The impact of using reconditioned correlated observation-error covariance matrices in the Met Office 1D-Var system. *Quarterly Journal of the Royal Meteorological Society*, 146(728):1372 – 1390.

[Tabeart and Pearson, 2021] Tabeart, J. M. and Pearson, J. W. (2021). Saddle point preconditioners for weak-constraint 4D-Var. Available on `https://arxiv.org/abs/2105.06975`.

[Talagrand, 2010] Talagrand, O. (2010). Variational assimilation. In Lahoz, W., Khattatov, B., and Menard, R., editors, *Data Assimilation. Making Sense of Observations*, pages 41 – 68. Springer-Verlag, Berlin, Heidelberg/Germany.

[Tang et al., 2009] Tang, J. M., Nabben, R., Vuik, C., and Erlangga, Y. A. (2009). Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *Journal of Scientific Computing volume*, 39:340 – 370.

[Trefethen and Bau, III, 1997] Trefethen, L. N. and Bau, III, D. (1997). *Numerical Linear Algebra*. SIAM, Philadelphia, PA.

[Trémolet, 2004] Trémolet, Y. (2004). Diagnostics of linear and incremental approximations in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 130(601):2233 – 2251.

[Trémolet, 2006] Trémolet, Y. (2006). Accounting for an imperfect model in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483 – 2504.

[Trémolet, 2007] Trémolet, Y. (2007). Model-error estimation in 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 133(626):1267 – 1280.

[Tshimanga, 2007] Tshimanga, J. (2007). *On a Class of Limited Memory Preconditioners for Large-Scale Nonlinear Least-Squares Problems (with Application to Variational Ocean Data Assimilation)*. PhD thesis, Department of Mathematics, University of Namur, Belgium.

[Tshimanga et al., 2008] Tshimanga, J., Gratton, S., Weaver, A. T., and Sartenaer, A. (2008). Limited-memory preconditioners, with application to incremental four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 134(632):751 – 769.

[Veerse and Thépaut, 1998] Veerse, F. and Thépaut, J.-N. (1998). Multiple-truncation incremental approach for four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1889 – 1908.

[Vlček and Lukšan, 2019] Vlček, J. and Lukšan, L. (2019). Properties of the block BFGS update and its application to the limited-memory block BNS method for unconstrained minimization. *Numerical Algorithms*, 80:957 – 987.

[Warner, 2010] Warner, T. T. (2010). *Numerical Weather and Climate Prediction*. Cambridge University Press, Cambridge, UK.

[Wathen, 2015] Wathen, A. J. (2015). Preconditioning. *Acta Numerica*, 24:329 – 376.

[Weston et al., 2014] Weston, P. P., Bell, W., and Eyre, J. R. (2014). Accounting for correlated error in the assimilation of high-resolution sounder data. *Quarterly Journal of the Royal Meteorological Society*, 140(685):2420 – 2429.

[Zhang et al., 2019] Zhang, L., Liu, Y., Liu, Y., Gong, J., Lu, H., Jin, Z., Tian, W., Liu, G., Zhou, B., and Zhao, B. (2019). The operational global four-dimensional variational data assimilation system at the China Meteorological Administration. *Quarterly Journal of the Royal Meteorological Society*, 145(722):1882 – 1896.