UNIVERSITY OF READING
SCHOOL OF MATHEMATICAL, PHYSICAL AND COMPUTATIONAL SCIENCES

Novel optimisation methods for data assimilation

Maha Hussein Kaouri

Thesis submitted for the degree of
Doctor of Philosophy

March 2021

Abstract

Data assimilation (DA) is a technique used to estimate the state of a dynamical system. In DA, a prior estimate (background state) is combined with observations to estimate the initial state of a dynamical system over a given time-window. This estimate is known as the 'analysis' in DA. In variational data assimilation (VarDA), the DA problem is formulated as a nonlinear least-squares problem, usually solved using a variant of the classical Gauss-Newton (GN) optimisation method known as the incremental method. In the incremental method, the iterative minimisation of the nonlinear objective function and the linearised subproblem are referred to as the 'outer loop' and the 'inner loop' respectively.

Within this thesis, we show how the convergence of GN can be improved through the use of safeguards that, unlike GN, guarantee convergence to the analysis from an arbitrary background state, while considering the limited time and cost available in DA. In particular, we consider GN equipped with line search (LS) and GN equipped with quadratic regularisation (REG), both of which achieve global convergence by guaranteeing a reduction in the VarDA objective function at each outer loop iteration. We prove global convergence of LS and REG and use idealised numerical experiments to show that these methods are able to improve the current estimate of the DA analysis even if the initial estimate of the solution is poor and a long assimilation time-window is used to include more observations. Furthermore, when GN performs poorly, a suitable choice of the initial regularisation parameter is critical in enhancing the performance of the REG method. We study the interaction between the REG parameter and the VarDA inner loop problem and use numerical experiments to show that choosing the initial REG parameter according to components of the VarDA problem results in REG locating a more accurate DA analysis than that obtained by GN, LS or the standard REG method.

Various simplifications are made to solve the variational problem within the time and computational cost available in practice. One of these simplifications is the use of a reduced resolution spatial grid for use within the inner loop. It is known that the accuracy with which the inner loop is solved affects the convergence of the outer loop. The condition number of the Hessian is a measure of the sensitivity of the solution of the inner loop problem to perturbations and also influences the speed of convergence of the VarDA inner loop minimisations. We derive an upper bound on the condition number of the preconditioned VarDA Hessian that accounts for different inner loop resolutions of the incremental method. This bound provides a theoretical insight into how the level of resolution interacts with various components of the incremental method to influence its convergence.

Dedication

To my family.

Declaration

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

Maha Hussein Kaouri

Publications

The work in Chapter 5 of this thesis contains a draft of the following publication:

C. Cartis, M. H. Kaouri, A. S. Lawless and N. K. Nichols. *Convergent least-squares optimisation methods for variational data assimilation.*

All the research within this draft was carried out by Maha H. Kaouri, with the coauthors providing guidance and review.

Acknowledgements

بِسْمِ آللَهِ آلرَحمَن آلرَحيمِ

First and foremost, I would like to express my sincere gratitude to my supervisors Dr Amos Lawless, Prof. Nancy Nichols and Dr Coralia Cartis for the tremendous level of support and guidance I have received from them throughout my PhD research.

I would also like to acknowledge the funding provided by the UK Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Mathematics of Planet Earth, the University of Reading EPSRC studentship (part of Grant/Award Number: EP/N509723/1) and by the NERC National Centre for Earth Observation that made this research possible.

Thank you to the students and staff of the Mathematics and Statistics department at the University of Reading for their support throughout my research. In particular, to the dear friends that I have met during my time at Reading, Aamena, Ning, Jemima, Hasen, James, György and André. Thank you for all the tea breaks, lunches, meals out and time spent in meaningful discussions. I am grateful to my monitoring committee, Dr Cláudia Neves and Prof. Simon Chandler-Wilde for their guidance over the years and to Dr Adam El-Said for introducing me to data assimilation and for his support.

Finally, thank you to my parents, siblings and husband for the never ending support, patience, encouragement and flexibility over these years and for always putting my work above anything else.

Contents

List of Figures

List of Tables

Mathematical Notation

n number of state variables

p number of observations

$\mathbf{x}$ a real-valued model state vector

$\mathbf{x}^*$ the truth vector

$\mathbf{v}$ the control variable

$\delta\mathbf{x}$ increment

k the number of outer loop iterations in an optimisation algorithm

k_J the number of Jacobian evaluations in an optimisation algorithm

l the number of function evaluations in an optimisation algorithm

$\mathbf{B}$ Background error covariance matrix

$\mathbf{C}_B$ Background error correlation matrix

$\mathbf{R}$ Observation error covariance matrix

σ_b^2 Background error variance

σ_o^2 Observation error variance

ϵ_b Background error vector

ϵ_o Observation error vector

$\mathcal{J}$ Data assimilation cost function

$\mathbf{I}$ Identity matrix

$\mathcal{M}$ Nonlinear model operator

$\mathbf{M}$ Linearised model operator

$\mathbf{r}$ Residual vector

$\mathbf{J}$ Jacobian matrix of the residual vector

$\mathbf{x}^b$ Background state vector

$\mathbf{y}$ Vector of observations

$\mathcal{H}$ Nonlinear observation operator

$\mathbf{H}$ Linearised observation operator

$\mathbf{C}$ Correlation matrix

$\|\cdot\|$ 2-norm

$\mathbf{s}^{(k)}$ search direction at the k^{th} iteration of an optimisation algorithm

$\alpha^{(k)}$ line search parameter at the k^{th} iteration of the LS method

$\gamma^{(k)}$ Regularisation parameter at the k^{th} iteration of the REG method

g Resolution parameter

Abbreviations

3D-Var Three-Dimensional Variational Data Assimilation

4D-Var Four-Dimensional Variational Data Assimilation

bArmijo Backtracking Armijo

DA Data Assimilation

ECMWF European Centre for Medium-Range Weather Forecasts

GN Gauss-Newton

LM Levenberg-Marquardt method

LS Gauss-Newton with bArmijo line search

NLLSP Nonlinear Least-Squares Problem

NWP Numerical Weather Prediction

REG Gauss-Newton with quadratic regularisation

VarDA Variational Data Assimilation

Chapter 1

Introduction

Data assimilation (DA) is a technique used to estimate the state of a dynamical system. In DA, a prior estimate of the state from a previous forecast, known as the background state, is combined with observations using an optimisation method to obtain an estimate of the evolving state of the system. The state at the starting time, or the 'initial state' is often referred to as the *initial conditions* of a system. There are two commonly used approaches to DA; sequential and variational [95]. Within our research, we focus on the variational approach, which is adopted by many weather forecasting centres such as ECMWF [63, 80, 105] and the Met Office [109], where the DA problem is formulated as a least-squares problem and solved using numerical optimisation methods.

In Numerical Weather Prediction (NWP), four-dimensional variational data assimilation (4D-Var) is used to estimate the initial conditions for a weather forecast [68]. The 4D-Var scheme is able to incorporate information from a prior forecast along with observations over both temporal and spatial domains in the form of a nonlinear least-squares objective function, which is then minimised using an iterative method. Three-dimensional variational data assimilation (3D-Var) only makes use of observations at the start of the assimilation time-window and was first implemented operationally by ECMWF from 1996 until 1997 when it was replaced by 4D-Var [105]. From a Bayesian point of view, minimising the 4D-Var objective function is equivalent to maximising the posterior probability to obtain the maximum a posteriori estimate [95]. Within our work, we focus on the strong-constraint 4D-Var problem where we assume the numerical model of the system perfectly represents the true dynamics of the system, or the model errors are small enough to be neglected. This formulation has been commonly used operationally in many meteorological centres [103], including the Meteorological Service of Canada [43], ECMWF [63, 80, 105] and the Met Office [109]. It is important to note that the DA problem is built on a Bayesian framework and so is not specific to numerical weather prediction (NWP). In fact, the technique(s) developed to solve such problems are popularly implemented in other applications such as oceanography [132, 128].

In practice, the variational data assimilation (VarDA) problem is a nonlinear least-squares problem, which can be viewed as a large-scale unconstrained optimisation problem [68]. The variational problem is solved as a sequence of linear least-squares problems using an incre-

mental method, which has been shown to be equivalent to the Gauss-Newton (GN) method under certain conditions [65]. In the incremental method, the minimisation of the nonlinear objective function and the linearised subproblem are referred to as the 'outer loop' and the 'inner loop' respectively. It is known that the accuracy with which the inner loop is solved affects the convergence of the outer loop [64, 65]. Within our work, we assume that either the original or approximate inner loop problem is solved and use a variable transformation commonly used in operational VarDA to precondition the variational problem.

The quality of the estimate and the subsequent forecast depends on how accurately the variational problem is solved within the time and computational cost available. The desire to improve the current methods and develop new methods for this class of problems is what drives the work in this thesis. In the following section, we detail the motivations behind our work.

1.1 Motivation

The GN method used in practice does not require the use of high order second derivatives, thus alleviating the issue of calculating and storing them. A drawback of the GN method is that it does not guarantee convergence to an estimate of the initial conditions given poor initialisation (a poor choice of initial guess) for the minimisation [33]. In NWP, the initial guess for the minimisation is generally chosen to be the predicted initial state from a previous forecast, known as the background state. However, for some applications of VarDA, this choice may not be a good enough estimate of the true initial state. Therefore, GN may fail to converge. Furthermore, the use of long assimilation time-windows (in the order of days) has recently become of interest in global NWP as it enables the use of more observations, improving the quality of the estimate of the initial state of the system, known as the 'analysis' in DA [69]. However, when the NWP system is sensitive to small changes in the initial conditions, the errors in the initial conditions are amplified over time through the use of the model, and more so when a long assimilation time-window is used. This is the motivation behind the investigation of alternative numerical optimisation methods such as those that use safeguards to guarantee convergence from an arbitrary starting point. The use of such methods could enable us to obtain an improvement on the estimate of the initial conditions within the limited time and computational cost available.

Another aspect of solving the DA problem that is important to consider is the effect of the simplifications made in operational implementations of VarDA on accuracy of the estimate of the initial conditions. Both the efficiency and accuracy of the numerical methods used to solve the DA problem is important in operational centres in order for the systems to be scalable and to keep computational time within a reasonable limit. Simplifications are made within the inner loop to reduce the computational cost in DA systems and to solve the DA problem in real time, so that the forecast is generated in a time when it is still useful. One such simplification is the use of a reduced resolution inner loop. This is where the inner loop is run at a lower spatial resolution than the nonlinear model used to calculate the innovation vectors (the mismatch between the observations and the model prediction). This simplifica-

tion is widely used in meteorological centres including ECMWF [63, 80, 105] and the Met Office [109]. However, its effect on the convergence of the incremental method has yet to be understood theoretically.

In the following section, we outline the research questions addressed in this thesis.

1.2 Aims and Contributions

The main aim of this thesis is to perform a convergence study on nonlinear least-squares optimisation methods for the variational problem. We aim to provide greater mathematical insight into operational implementations of variational data assimilation, as well as to investigate the convergence properties of alternative techniques used to solve the variational problem. This is achieved by developing theory on the effect of the use of reduced resolution inner loops on the outer loop iterates as well as conducting rigorous numerical experiments in a comparative study using the classical GN, GN with line search (LS) and GN with quadratic regularisation (REG) methods. The latter two of which are referred to as 'globally convergent' methods as, unlike GN, they guarantee convergence from an arbitrary background state vector. In both the globally convergent methods and reduced resolution inner loop chapters, we focus on the convergence of the VarDA outer loop.

The two main research questions we aim to answer within this thesis, along with their subquestions, are given as follows.

RQ1. **Is the use of globally convergent strategies within GN beneficial in variational data assimilation?**
We compare the performance of three optimisation methods that may be used to solve the variational problem; GN, LS and REG, when applied to both the 3D-Var and 4D-Var problems where we account for the computational limits that exist in practical implementations and consider different practical scenarios, such as the use of correlated background error and a long assimilation time-window.

(a) **How can the GN method benefit from the use of globally convergent strategies?**
Unlike with GN, the globally convergent methods use safeguards to guarantee convergence to the analysis from an arbitrary background state vector by ensuring monotonic/strict and sufficient decrease of the error in the objective function. We outline and prove the theorems of global convergence for the specific globally convergent methods used within our work (LS and REG) and discuss how the assumptions for the convergence of the globally convergent methods relate to DA.

(b) **How do the globally convergent method parameters interact with the variational problem?**
We study how the parameters interact with the DA Hessian and show numerically how this affects convergence. We identify the cases where LS and REG

outperform GN using the 3D-Var problem. Focusing on the REG method, we research alternative choices of the initial REG parameter that we show improve convergence of the REG method.

(c) **How do GN, LS and REG behave if the initial guess of the minimisation (the background) is highly inaccurate compared to the observations?**
We show the effect that poor background information has on the quality of the estimate obtained for each of the three methods we consider. We find that in the case where the background information is highly inaccurate compared to the observations, the convergence of all three methods is improved when more observations are included along the time-window.

(d) **In what situations are the globally convergent methods a better option then GN in the presence of a long assimilation time-window?**
Within the long time-window 4D-Var framework, we use two test models to show that when there is more uncertainty in the background information compared to the observations, the GN method may diverge, yet LS and REG, are able to improve the estimate of the analysis and solve more problems than GN in the limited cost available.

RQ2. **What is the effect of using a reduced resolution inner loop on the convergence of the incremental method?**
We analyse the structure of the resolution matrices used to map between the outer loop and inner loop resolutions and how they interact with the components of the incremental method. We then derive a theoretical bound on the condition number of the preconditioned 3D-Var Hessian that accounts for the inner loop resolution. To understand the effect of using a reduced resolution inner loop on the accuracy of the analysis, we use error profiles for the numerical values and apply theory from the discrete Fourier transform (DFT) to understand the implications of using reduced resolution inner loops on the resolution of wavenumbers in the analysis.

(a) **How does the level of resolution reduction affect the convergence of the incremental method?**
We show how various components of the incremental scheme depend on the difference between the inner and outer loop resolution using an expression we derive for the norm of the linear and cubic interpolation extension matrices used to map to the outer loop (full) resolution.

(b) **What is the effect of using a linear and cubic interpolation matrix as the extension operator on the different components of the incremental method?**
We show that we can exploit the structure of the interpolation matrices in order

to bound various quantities in the incremental method. We prove that the 2-norm of the linear and cubic interpolation matrices are equal to the square root of the inverse of the difference between the inner and outer loop resolutions and discuss how this result applies to higher orders of interpolation. These results can then be used to bound various quantities of the incremental method.

(c) **What effect does the use of reduced resolution operators have on the accuracy to which we could be able to solve the inner loop problem in practice?**
We derive a bound in terms of the difference between the inner and outer loop resolution to understand the impact the use of the reduced resolution components has on the conditioning of the inner loop problem and thus, the rate of convergence of the iterative method used to solve it in practice. We conduct a series of numerical experiments where we use the incremental method to solve the 3D-Var problem and identify cases where the bound on the condition number performs well.

(d) **How does the reduced resolution algorithmic output compare to the full resolution algorithmic output?**
We use assimilation experiments to compare the analysis error generated by the incremental method using different inner loop resolutions. We consider the use of both linear and nonlinear observation operators as well as correlated background error covariance matrices, as in practice, and produce error profiles to represent our results and discuss our findings in relation to our theoretical bounds.

(e) **How does the use of restriction/extension matrices affect the convergence of the wavenumbers in the analysis?**
We use the power spectra of the DFT of the 3D-Var analysis to understand how accurately the non-zero wavenumbers of the true solution of the variational problem can be resolved. We generate error profiles for the error in the amplitudes of the non-zero wavenumbers for different resolution choices and discuss our findings in relation to our theoretical results from earlier in the chapter.

In the following section, we outline the structure of this thesis.

1.3 Overview of Thesis

This thesis is organised as follows. In Chapter 2 we begin by introducing some key definitions and theorems used throughout this thesis. We then outline the 4D-Var problem, noting that the 3D-Var equations can be derived from the 4D-Var equations, along with the framework for the incremental method with a reduced resolution inner loop. We introduce the preconditioned formulation using the square-root of the background error covariance

matrix; a preconditioner commonly used in practice for the VarDA inner loop, and outline the numerical models used in the 4D-Var experimental work of this thesis. In Chapter 3, we outline some basic optimisation theory along with a literature review of the optimisation methods used to solve the VarDA outer loop problem. We also conduct a literature review of globally convergent optimisation strategies for least-squares optimisation, specifying the strategies we choose to equip GN with, namely, LS and REG.

Research questions RQ1(a) and RQ1(b) are addressed in Chapter 4. We derive the global convergence proofs and discuss the assumptions in the DA context. We then apply the GN method along with the two globally convergent methods to general least-squares and 3D-Var problems. We analyse how the globally convergent optimisation strategies interact with various components of the 3D-Var problem. We outline ways to improve the REG method by choosing the initial regularisation parameter according to the background error covariance matrix. We also partially address research question RQ1(c) using the 3D-Var problem by initialising the minimisation methods with an increasingly noisier background state vector and discussing the results.

Research questions RQ1(c) and RQ1(d) are addressed in Chapter 5. Here, we extend the use of GN, LS and REG to 4D-Var problems where a numerical model is used. We first include a draft paper with our main research findings when applying the three methods to the 4D-Var problem. Within this paper, we vary various components in the 4D-Var problem and discuss both the theoretical and experimental effects on the convergence of the three methods. At the end of this chapter, we include some of our findings that were omitted from the paper where correlated background errors are considered. We then redirect our focus to the second main aim of this thesis.

Research questions RQ2(a)-RQ2(e) are addressed in Chapter 6. We outline the grid-point framework and the restriction and extension operators. We derive theoretical bounds on various components of the incremental algorithm when using different resolutions and extension operators. We conduct numerical experiments using the incremental 3D-Var method to gain an insight into how our theoretical findings work in practice. We use the DFT theory in our 3D-Var numerical experiments to analyse the impact the use of various levels of reduced resolution inner loops has on which wavenumbers are resolved in the reduced resolution framework. Finally, we conclude this thesis in Chapter 7, reflecting on our research findings and suggesting directions for future research.

Chapter 2

Variational Data Assimilation

Variational Data Assimilation (VarDA) problems can be formulated as optimal control problems, see [68, 71]. Within our work, we focus on the least-squares formulation of the variational problem where information from a prior forecast is incorporated with observations in the form of a nonlinear least-squares objective function, which is then minimised using an iterative method to obtain an estimate of the true solution. In a Bayesian framework, this is equivalent to maximising the posterior probability to obtain the maximum a posteriori estimate of the true solution [95].

In Section 2.1, we introduce some key mathematical notation, theorems and definitions used within this thesis. In Section 2.2, we outline the least-squares formulation of two commonly used variational data assimilation schemes, three-dimensional variational data assimilation (3D-Var), where all observations are assumed to be taken at the beginning of the time-window and four-dimensional variational data assimilation (4D-Var), where observations can be taken over a period of time. In Section 2.3, we outline how the variational problem is preconditioned using a variable transformation. In Section 2.4, we outline two numerical models used for our 4D-Var experimental work. Finally, we conclude this chapter in Section 2.5.

2.1 Mathematical preliminaries

In the following section, we outline some key definitions on vector norms, matrix properties and matrix structures used throughout this thesis.

2.1.1 Vector and matrix norms, properties and structures

The eigenvalues and eigenvectors of a matrix are defined in the following.

Definition 2.1.1 (Eigenvalues and eigenvectors of a matrix (see Definition 2.11 of [36]))**.** *Let* $\mathbf{A} \in \mathbb{R}^{n\times n}$*. A scalar* $\lambda \in \mathbb{C}$ *is called an eigenvalue of* $\mathbf{A}$ *if it satisfies*

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \tag{2.1}$$

where the solution $\mathbf{v} \in \mathbb{C}\backslash\{\mathbf{0}\}$ *is the corresponding eigenvector of* $\mathbf{A}$*.*

Within our work, the set of all eigenvalues of a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is denoted by $\{\lambda_i\}$, where $i = 1, 2, \ldots, n$ and

$$\lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \cdots \geq \lambda_n(\mathbf{A}). \tag{2.2}$$

For simplicity, we denote the maximum eigenvalue of the matrix $\mathbf{A}$ by

$$\lambda_{\max}(\mathbf{A}) = \lambda_1(\mathbf{A}) \tag{2.3}$$

and the minimum eigenvalue by

$$\lambda_{\min}(\mathbf{A}) = \lambda_n(\mathbf{A}). \tag{2.4}$$

If at least one of λ_i is zero, the matrix $\mathbf{A}$ is singular and its inverse does not exist, otherwise, the matrix $\mathbf{A}$ is nonsingular [96].

If a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric, $\lambda_i \in \mathbb{R}$, for all i [96], otherwise, if $\mathbf{A}$ is nonsymmetric, its eigenvalues may be complex. Many of the matrices that we use within this thesis are classed as symmetric positive definite matrices, as defined in the following.

Definition 2.1.2 (Positive definite matrix)**.** *A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is said to be positive definite matrix if and only if all of its eigenvalues are positive.*

Some of the matrices we use are classed as positive semidefinite matrices, defined in the following.

Definition 2.1.3 (Positive semidefinite matrix)**.** *A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is said to be a positive semidefinite matrix if and only if all of its eigenvalues are non-negative.*

We next outline the vector and matrix norms and their properties that are used heavily within our computations. We first define the 2-norm of a vector in the following.

Definition 2.1.4 (2-norm of a vector (see [36] Definition 2.8))**.** *The 2-norm of a vector $\mathbf{x} \in \mathbb{R}^n$ is given by*

$$\|\mathbf{x}\|_2 = \left\{ \sum_{i=1}^{n} |x_i|^2 \right\}^{1/2} = \sqrt{\mathbf{x}^T\mathbf{x}}. \tag{2.5}$$

Note that the 2-norm of a vector is also referred to as the Euclidean norm.

An important property of vector norms is the *triangle inequality*, defined in the following.

Definition 2.1.5 (The triangle inequality (see [47] Section 2.2.1))**.** *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. The triangle inequality is given by*

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|. \tag{2.6}$$

Here, the equality holds if and only if one of the vectors $\mathbf{x}$ and $\mathbf{y}$ is a non-negative scalar multiple of the other [96].

The *Cauchy-Schwarz inequality* is an important property of the 2-norm and is outlined in the following.

Theorem 2.1.1 (Cauchy-Schwarz inequality (see [47] Section 2.2.2)). *Let* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. *The Cauchy-Schwarz inequality is given by*

$$|\mathbf{x}^T\mathbf{y}| \leq \|\mathbf{x}\|_2\|\mathbf{y}\|_2. \tag{2.7}$$

Proof. See [47] Section 2.2.2.

As with the vector norm, within our work we are concerned with the 2-norm of a matrix defined in the following.

Definition 2.1.6 (2-norm of a matrix (see [47] Section 2.3.1)). *The 2-norm of a matrix* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *is given by*

$$\|\mathbf{A}\|_2 = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}, \tag{2.8}$$

where $\mathbf{x} \in \mathbb{R}^n$.

For symmetric positive definite matrices, the 2-norm of a matrix, and its inverse can be calculated as in the following definition.

Theorem 2.1.2 (Symmetric positive definite matrices and the 2-norm (see [36] Theorem 2.9)). *Let* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *be a symmetric positive definite matrix with eigenvalues* $\lambda_i \in \mathbb{R}$, *where* $i = 1, \ldots, n$. *We have,*

$$\|\mathbf{A}\|_2 = \lambda_{max}(\mathbf{A}) \tag{2.9}$$

and

$$\|\mathbf{A}^{-1}\|_2 = \frac{1}{\lambda_{min}(\mathbf{A})}. \tag{2.10}$$

Proof. See [36] Theorem 2.9.

All matrix norms satisfy the following inequality.

Definition 2.1.7 (Matrix norm property (see [47] Section 2.3.1)). *Let* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$. *By definition, a matrix norm satisfies the following inequality*

$$\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|. \tag{2.11}$$

The 2-norm is classed as a submultiplicative norm, that is, it satisfies the submultiplicative property defined as follows.

Theorem 2.1.3 (Submultiplicative property of the 2-norm (see [36] Theorem 2.10)). *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times q}$. *Then the following property, known as the submultiplicative property, holds*

$$\|\mathbf{A}\mathbf{B}\|_2 \leq \|\mathbf{A}\|_2\|\mathbf{B}\|_2. \tag{2.12}$$

Proof. See [36] Theorem 2.10.

The spectral radius of a matrix is used in the convergence theorems of the optimisation methods in our work. If $\mathbf{A}$ is a square matrix, the spectral radius of a matrix is simply the largest absolute eigenvalue of the matrix $\mathbf{A}$, as defined in the following.

Definition 2.1.8 (Spectral radius of a matrix (see [98] section 2.2)). *Let* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *with eigenvalues* $\lambda_i \in \mathbb{R}$*, where* $i = 1, \ldots, n$*, then the spectral radius of* $\mathbf{A}$ *is given by*

$$\rho(\mathbf{A}) = \max |\lambda_i|. \tag{2.13}$$

The condition number of a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ allows us to understand how sensitive the solution $\mathbf{x} \in \mathbb{R}^n$ of linear problem $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^n$ is the data, would be to small perturbations in $\mathbf{b}$. A problem with a low condition number signifies that small perturbations in $\mathbf{b}$ results in small changes in the solution. In this case, the problem is said to be *well-conditioned*. An *ill-conditioned* problem is one where small perturbations in $\mathbf{b}$ results in large changes in the solution. The condition number of a matrix is defined in the following.

Definition 2.1.9 (Condition number of a matrix (see [36] Definition 2.12)). *Let* $\mathbf{A}$ *be a nonsingular matrix, then the condition number of* $\mathbf{A}$*, in any norm, is given by*

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|. \tag{2.14}$$

More specifically, the condition number of a symmetric, positive semidefinite matrix in the 2-norm can be expressed as in the following theorem.

Theorem 2.1.4 (Condition number of a symmetric, positive semidefinite matrix (see [47] Section 2.7.2)). *If* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *is a symmetric, positive semidefinite matrix, then the condition number in the 2-norm can be expressed as the ratio of the largest and smallest eigenvalues and is given by*

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|_2\|\mathbf{A}^{-1}\|_2 \equiv \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}. \tag{2.15}$$

If $\mathbf{A}$ *is singular then* $\kappa(\mathbf{A})$ *is infinite.*

Proof. See [47] Section 2.7.2.

In the case where $\mathbf{A}$ is singular, its inverse does not exist. However, every matrix has a generalised (or pseudo) inverse. A common choice of generalised inverse is the Moore-Penrose pseudoinverse, which was independently discovered by [88] in 1920, [12] in 1951 and [100] in 1955, and is defined as follows.

Definition 2.1.10 (Moore-Penrose conditions (See [47] Section 5.5.4)). *A matrix* $\mathbf{A}^+ \in \mathbb{R}^{n\times r}$ *is said to be a pseudoinverse of* $\mathbf{A} \in \mathbb{R}^{r\times n}$ *if it satisfies the following conditions*

- $\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}$
- $\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+$
- $(\mathbf{A}\mathbf{A}^+)^T = \mathbf{A}\mathbf{A}^+$
- $(\mathbf{A}^+\mathbf{A})^T = \mathbf{A}^+\mathbf{A}$

Provided that the matrix $\mathbf{A} \in \mathbb{R}^{r\times n}$ is full rank, i.e. $rank(\mathbf{A}) = \min(r, n)$, the Moore-Penrose pseudoinverse of $\mathbf{A}$, $\mathbf{A}^+ \in \mathbb{R}^{n\times r}$ can be calculated as in the following.

$$\mathbf{A}^+ = \begin{cases} \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}, & \text{if } r < n \text{ (right inverse)} \\ (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T, & \text{if } r > n \text{ (left inverse).} \end{cases} \tag{2.16}$$

Some of the matrices used within our work have special properties arising from their circulant structure. We define a circulant matrix in the following.

Definition 2.1.11 (Circulant matrices (see [31])). *A circulant matrix* $\mathbf{C} \in \mathbb{R}^{r\times r}$ *is of the form*

$$\mathbf{C} = \begin{pmatrix} c_0 & c_1 & \dots & c_{r-2} & c_{r-1} \\ c_{r-1} & c_0 & c_1 & & c_{r-2} \\ \vdots & c_{r-1} & c_0 & \ddots & \vdots \\ c_2 & & \ddots & \ddots & c_1 \\ c_1 & c_2 & \dots & c_{r-1} & c_0 \end{pmatrix} \tag{2.17}$$

The eigenvalues of a circulant matrix can be expressed as in the following theorem.

Theorem 2.1.5 (Eigenvalues of a circulant matrix (see [53] Section 3.1)). *Let* $\mathbf{C} \in \mathbb{R}^{r\times r}$ *be a circulant matrix. Then the eigenvalues of* $\mathbf{C}$ *are given by*

$$\lambda_\kappa^C = \sum_{j=0}^{r-1} c_j \omega^{\kappa j}, \tag{2.18}$$

with eigenvectors

$$\mathbf{v}_\kappa^C = \frac{1}{\sqrt{r}}(1, \omega^\kappa, \dots, \omega^{\kappa(r-1)}), \tag{2.19}$$

where $\omega = e^{-2\pi i/r}$ *and* $\kappa = 0, 1, \dots, r-1$ *is the wavenumber.*

Proof. See [53] Section 3.1.

Note that in Theorem 2.1.5, the eigenvalues of $\mathbf{C}$ are ordered according to the wavenumbers. The circulant matrices we are concerned with are symmetric. The eigenvalues of symmetric circulant matrices are real-valued, some of which are repeated as outlined in the following [97].

Corollary 2.1.6 (Eigenvalues of symmetric circulant matrices (see [97] Corollary 17)). *If* $\mathbf{C} \in \mathbb{R}^{r\times r}$ *is symmetric circulant, then* (2.18) *reduces to*

$$\lambda_\kappa^C = \begin{cases} c_0 + 2\sum_{j=1}^{(r-2)/2} c_j \omega^{\kappa j} + (-1)^\kappa c_{r/2} & r \text{ even} \\ c_0 + 2\sum_{j=1}^{(r-1)/2} c_j \omega^{\kappa j} & r \text{ odd} \end{cases}, \tag{2.20}$$

where $\omega = e^{-2\pi i/r}$ *and* $\kappa = 0, 1, \dots, r-1$ *is the wavenumber.*

Proof. See [97] Corollary 17.

Corollary 2.1.6 implies that for a symmetric circulant matrix $\mathbf{C} \in \mathbb{R}^{r\times r}$, λ_0^C is distinct, $\lambda_i = \lambda_{r-i}^C$ for $i = 1, 2, \dots, r-1$ are repeated pairs and if r is even, $\lambda_{r/2}^C$ is also distinct. In the

case that $\mathbf{C} \in \mathbb{R}^{r\times r}$ is a positive symmetric circulant matrix, its eigenvalues decrease as the wavenumber increases until the minimum at $\kappa = r/2$ if r is even and $\kappa = (r-1)/2$ if r is odd [53]. Therefore, the maximum eigenvalue can be calculated explicitly using the following,

$$\lambda_0^C = \sum_{j=0}^{r-1} c_j. \tag{2.21}$$

Equation (2.21) is simply the row/column sum of a non-negative circulant matrix.

We refer the reader to Chapter 3 of [31] for more details on the properties of circulant matrices, including the proofs that

- the sum of two circulant matrices is circulant,
- the product of two circulant matrices is circulant,
- the transpose of a circulant matrix is circulant,
- the inverse of a circulant matrix is circulant,

all of which are relevant to our work.

The variational problem can be written in form of a least-squares problem and solved using nonlinear optimisation methods, outlined later in Chapter 3. In the following section, we outline the theory of least-squares problems.

2.1.2 Least-Squares problems

We begin by outlining some key definitions of a function $f : \mathbb{R}^n \to \mathbb{R}^m$.

A common type of differentiability of a function is known as Frechét differentiability and is defined as follows.

Definition 2.1.12 (A Frechét differentiable function (see [96] Appendix A))**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be Frechét differentiable at $\mathbf{x} \in \mathbb{R}^n$ if there exists a vector $\mathbf{g} \in \mathbb{R}^n$ such that*

$$\lim_{\mathbf{y}\to 0} \frac{f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}) - \mathbf{g}^T\mathbf{x}}{\|\mathbf{y}\|} = 0, \tag{2.22}$$

where $\mathbf{y} \in \mathbb{R}^n$.

If $\mathbf{g}$ exists, it is known as the gradient of f at $\mathbf{x}$, denoted by f'. We first define a continuous differentiable function, followed by the definitions of the gradient and Hessian of f.

Definition 2.1.13 (A continuously differentiable function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be continuously differentiable and is denoted by $f \in \mathcal{C}^1(\mathbb{R}^n)$ if its derivative f' exists and is continuous. We say that f is twice continuously differentiable and is denoted by $f \in \mathcal{C}^2(\mathbb{R}^n)$ if its second derivative f'' exists and is continuous.*

Definition 2.1.14 (The gradient and Hessian of a function of several variables (see [96] Appendix A)). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. Then the gradient of f at $\mathbf{x}$ is given by*

$$\nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \tag{2.23}$$

and the matrix of second partial derivative of f, known as the Hessian, is given by

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \tag{2.24}$$

The Lipschitz continuity property is used in the convergence proofs later in our work and compares the difference between changes in the function value against changes in the dependent variable. We define Lipschitz continuity of a function in the following.

Definition 2.1.15 (Lipschitz continuous function (see [96] A.42)). *Let f be a function where $f : \mathbb{R}^n \to \mathbb{R}^m$ for general n and m. The function f is said to be Lipschitz continuous on some set $\mathcal{N} \subset \mathbb{R}^n$ if there exists a constant $L > 0$ such that,*

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathcal{N}. \tag{2.25}$$

The following lemma states that a Lipschitz continuous gradient of a function implies that its Hessian is bounded above.

Lemma 2.1.7 (Lipschitz continuity implies bounded Hessian (see [92] Lemma 1.2.2)). *Let f be a function where $f : \mathbb{R}^n \to \mathbb{R}^m$ for general n and m and $f \in \mathcal{C}^2(\mathbb{R}^n)$, then its gradient ∇f is Lipschitz continuous on $\mathbb{R}^n$ with Lipschitz constant $L > 0$ such that,*

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\| \leq L\|\mathbf{y} - \mathbf{x}\|, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \tag{2.26}$$

if and only if its Hessian $\nabla^2 f(\mathbf{x})$ is uniformly bounded above on $\mathbb{R}^n$, that is,

$$\|\nabla^2 f(\mathbf{x})\| \leq L, \qquad \forall \mathbf{x} \in \mathbb{R}^n. \tag{2.27}$$

Proof. See [92] Lemma 1.2.2.

In a least-squares problem, the function $f : \mathbb{R}^n \to \mathbb{R}$ has a special form, as defined in the following.

Definition 2.1.16 (A least squares function (see [96] Chapter 10)). *Let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$. A nonlinear least-squares (NLLS) function is of the following form,*

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{r}(\mathbf{x})\|_2^2, \tag{2.28}$$

where $\mathbf{r}(\mathbf{x}) = [r_1(\mathbf{x}), ..., r_m(\mathbf{x})]^T$ and each $r_j : \mathbb{R}^n \to \mathbb{R}$, for $j = 1, 2, \ldots, m$, is referred to as a residual.

The gradient and Hessian of f can also be represented using $\mathbf{r}(\mathbf{x})$ and its derivatives. The matrix of partial derivatives of residual vector $\mathbf{r}(\mathbf{x})$, known as the Jacobian, is defined as follows.

Definition 2.1.17 (The Jacobian matrix (see [96] Chapter 10))**.** *Let $\mathbf{x} \in \mathbb{R}^n$, where $\mathbf{x} = [x_1, \ldots, x_n]^T$. Furthermore, let $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$, where $\mathbf{r} = [r_1(\mathbf{x}), \ldots, r_m(\mathbf{x})]^T$. The Jacobian matrix of partial derivatives of the residual vector $\mathbf{r}(\mathbf{x})$ is given by,*

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial r_1}{\partial x_1} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_m}{\partial x_1} & \cdots & \frac{\partial r_m}{\partial x_n} \end{pmatrix}. \tag{2.29}$$

The gradient of f in (2.28) is defined as follows.

Definition 2.1.18 (The gradient of a least-squares function)**.** *Let $\mathbf{x} \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ be a least-squares function of the form* (2.28)*, then the first derivative of f is given by*

$$\nabla f(\mathbf{x}) = \sum_{j=1}^{m} r_j(\mathbf{x})^T \nabla r_j(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{r}(\mathbf{x}), \tag{2.30}$$

where $\mathbf{r}(\mathbf{x})$ is the residual vector and $\mathbf{J}(\mathbf{x})$ is its Jacobian matrix.

The Hessian of f in (2.28) is defined as follows.

Definition 2.1.19. *Let $\mathbf{x} \in \mathbb{R}^n$ and $f \in \mathcal{C}^2$ be a least-squares function of the form* (2.28)*, then the second derivative of f is given by*

$$\nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) + \mathbf{Q}(\mathbf{x}), \tag{2.31}$$

where $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix of the residual vector $\mathbf{r}(\mathbf{x})$ and $\mathbf{Q}(\mathbf{x})$ denote the higher-order terms given by

$$\mathbf{Q}(\mathbf{x}) = \sum_{j=1}^{m} r_j \nabla^2 r_j. \tag{2.32}$$

In optimisation, we seek to find an $\mathbf{x}$ that minimises the least-squares function (2.28), often referred to as the objective function or cost function, by solving the *least-squares problem*

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|_2^2, \tag{2.33}$$

A least-squares problem is said to be nonlinear if the residual terms r_j are nonlinear functions of the variables $\mathbf{x}$.

Within our work, we solve VarDA problems of the form (2.33) using nonlinear optimisation algorithms, outlined later in Chapter 3, where $\mathbf{x}$ is defined using points on a discrete spatial grid. The output from these algorithms can be analysed in grid-point space, or transformed to spectral space, which instead models the variables using a series of waves. In the following section, we outline the Discrete Fourier transform used in Chapter 6 to transform from grid-point to spectral space.

2.1.3 Discrete Fourier transform

The Discrete Fourier transform is based on the Fourier transform except the indefinite integral is replaced by a finite sum, making the DFT computationally more relevant for finite samples [117], such as the quantities in DA. From a linear algebra standpoint, the DFT can be seen as a change of basis from the standard basis to the discrete Fourier basis.

Over a series of discrete, equally spaced points, sine, cosine and imaginary functions have the orthogonality property. We first define the Euclidean inner product.

Definition 2.1.20 (Euclidean inner product)**.** *Let $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$. The Euclidean inner product is given by*

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i^*, \tag{2.34}$$

where y_i^* denotes the complex conjugate of y_i.

We use (2.34) in the following lemma for the orthogonality condition.

Lemma 2.1.8 (Orthogonality condition (see [120] Chapter 2))**.** *Let $m, l = 0, 1, \ldots, n-1$. The inner product of $e^{i\frac{2\pi}{n}jm}$ and $e^{i\frac{2\pi}{n}jl}$ is given by*

$$\sum_{j=0}^{n-1} e^{i\frac{2\pi}{n}j(m-l)} = \begin{cases} n, & \text{for } m = l \\ 0, & \text{for } m \neq l. \end{cases} \tag{2.35}$$

This property allows us to define the 1-dimensional DFT of a function of discrete, equally spaced points, as follows.

Definition 2.1.21 (Discrete Fourier transform (see [1] Section 20.6))**.** *Let f be a function of discrete complex values defined only at the set of n equally spaced points on the interval $[0, 2\pi]$ given by*

$$\mu_j = \frac{2\pi}{n} j, \tag{2.36}$$

where $j = 0, 1, \ldots, n$. The discrete Fourier transform of f_j, where $f_j = f(\mu_j)$ is given by

$$DFT_n(f_j) = \mathcal{F}_\kappa = \sum_{j=0}^{n-1} f_j e^{-i\mu_j \kappa} \tag{2.37}$$

where $\kappa = 0, 1, \ldots, n-1$ is the wavenumber.

The subscript of DFT_n in Definition 2.1.21 is used to highlight the number of grid points. This is important within our work as we will often be considering the DFT of reduced grids. We assume that $DFT = DFT_n$ unless explicitly stated otherwise. In Chapter 6, we outline our chosen scaling factor for (2.37) which comes from our desire to analyse wave amplitudes.

We can represent the DFT as a linear transformation matrix. Let $\mathcal{F} = [\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_{n-1}]^T$ and $\mathbf{f} = [f_0, f_1, \ldots, f_{n-1}]^T$. Then the DFT matrix is given by

$$\mathbf{W} = (e^{i\mu_j \kappa})_{j,\kappa=0,1,\ldots,n-1}, \tag{2.38}$$

where $\mathbf{W} \in \mathbb{R}^{n\times n}$. The matrix transformation is given by

$$\mathcal{F} = \mathbf{W}\mathbf{f}. \tag{2.39}$$

When the DFT input is real, the complex conjugate of the κ^{th} DFT output is equal to the $n-\kappa^{\text{th}}$ DFT output, see [79] Section 3.2. Thus, for a real DFT input sequence, we have

$$\mathcal{F}_{n-\kappa} = \mathcal{F}_\kappa^* \tag{2.40}$$

where $\mathcal{F}_\kappa^*$ is the complex conjugate of $\mathcal{F}_\kappa$. Therefore, we only need to calculate the first $n/2+1$ values of $\mathcal{F}_\kappa$ as the remaining terms do not provide any additional information. In DA, the quantities we consider are real numbers. Therefore, the symmetry property of the DFT (2.40) will hold.

The DFT of a function f_j can be written in terms of its real parts $\text{Re}(\mathcal{F}_\kappa)$ and its imaginary parts $\text{Im}(\mathcal{F}_\kappa)$ as follows,

$$\mathcal{F}_\kappa = \text{Re}(\mathcal{F}_\kappa) + i\text{Im}(\mathcal{F}_\kappa). \tag{2.41}$$

For a real DFT input f_j, as in our application, we have

$$\begin{aligned} \text{Re}(\mathcal{F}_\kappa) &= \sum_{j=0}^{n-1} f_j \cos(\mu_j \kappa) \\ \text{Im}(\mathcal{F}_\kappa) &= \sum_{j=0}^{n-1} f_j \sin(\mu_j \kappa). \end{aligned} \tag{2.42}$$

We recall that a discrete function f_j with $j = 0, 1, \ldots, n-1$ is said to be even if $f_j = f_{-j}$ and odd if $f_{-j} = -f_j$. If f_j are discrete points from an even function, then,

$$\mathcal{F}_\kappa = \text{Re}(\mathcal{F}_\kappa). \tag{2.43}$$

If f_j are discrete points from an odd function, then,

$$\mathcal{F}_\kappa = \text{Im}(\mathcal{F}_\kappa). \tag{2.44}$$

A special case occurs for wavenumber $\kappa = 0$, whereby

$$\mathcal{F}_0 = \sum_{j=0}^{n-1} f_j. \tag{2.45}$$

That is, the DFT coefficient corresponding to wavenumber 0 is the sum of the DFT input.

The following reciprocity relation relates the spatial and frequency grid domains,

$$A\Omega = n, \tag{2.46}$$

where $A = n\Delta x$ is the size of the spatial domain with spatial grid length Δx and $\Omega = n\Delta\omega$ is the size of the frequency domain with frequency grid length $\Delta\omega$. Equation (2.46) shows that the lengths of the spatial and frequency domains vary inversely with each other [18].

The largest wave that can be represented within the interval $[0, A]$ has a period of A units and a frequency of $1/A$ periods per unit length [18]. This is the lowest frequency associated with this interval. The following reciprocity relation relates the spatial and frequency grid lengths,

$$\Delta x \Delta\omega = 1/n. \tag{2.47}$$

Therefore, by increasing the spatial grid length Δx, we are decreasing the number of frequencies that can be represented using the corresponding frequency grid.

Explained in detail in Section 5 of [111] and Appendix H of [30], aliasing is the phenomenon that occurs when high frequency (small-scale) waves are misinterpreted (aliased) as lower frequency (large-scale) waves. Such a phenomenon may occur when there are not enough grid-points to represent the structure of a wave. This phenomenon may occur in VarDA as the number of grid-points used to represent the variational problem is restricted so as to be able to solve the problem in the limited time and computational cost available.

Aliasing is illustrated in the following example, similar to that in Section 5 of [111]. Figure 2.1 is a plot of the high frequency continuous function $g(x) = \sin(2\pi 5x)$ and the lower frequency continuous function $l(x) = \sin(2\pi x)$ along with the discrete function of 9 evenly spaced points that happen to fit both curves. From this plot, we can see that unless we take a larger sample of points, we are unable to know whether the sample comes from the function $g(x)$ or $l(x)$. Therefore, we may incorrectly attribute the points to the lower frequency function.

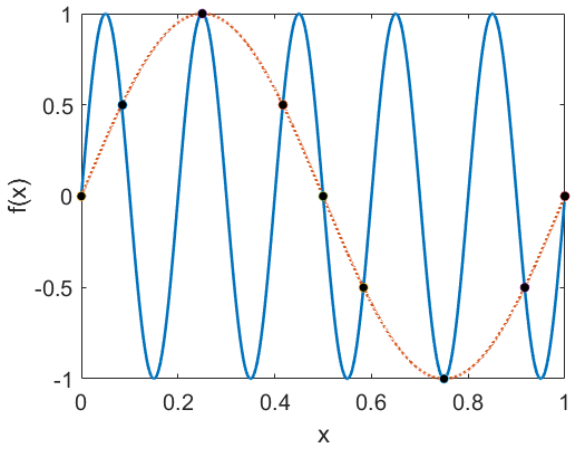


Figure 2.1: Plot of $f(x) = g(x)$ (red dotted curve), $f(x) = l(x)$ (blue solved curve) and the discrete points (black solid dots).

Aliasing can be avoided by choosing the right sampling rate. More precisely, the Nyquist-

–Shannon sampling theorem states that in order to correctly reconstruct the waves, they must be sampled at a rate greater than twice their highest frequency.

Theorem 2.1.9 (Nyquist–Shannon sampling theorem (see [114] Theorem 1))**.** *If a function $f : \mathbb{R} \to \mathbb{R}$ contains no frequencies higher than $f_{\max}$, then it can be completely determined by a sequence of discrete points that are spaced $1/(2f_{\max})$ apart, where $2f_{\max}$ is called the Nyquist rate. Therefore, for a given sampling rate f_s, we require*

$$f_s > 2f_{\max}, \tag{2.48}$$

in order to completely determine the function.

Proof. See [114] Section 2.

The power spectrum of a DFT output is able to tell us which wavenumbers are present in the function being transformed and how important they are and thus allow us to effectively analyse the frequencies of the algorithmic output in our work.

The power spectrum is a plot of the square of the modulus of the DFT coefficients $\mathcal{F}_\kappa$ given in (2.37), known as the power, against the corresponding wavenumber $\kappa = 0, 1, \ldots n-1$. We scale the DFT output $\mathcal{F}_\kappa$ using a factor $2/n$ so that the amplitude of a sine wave and the corresponding power are both equal to 1. Thus, the power ρ of the (scaled) DFT is given by

$$\rho\Big(\frac{2}{n}\mathcal{F}_\kappa\Big) = \Big|\frac{2}{n}\mathcal{F}_\kappa\Big|^2, \tag{2.49}$$

where $\kappa = 0, 1, \ldots n-1$ is the wavenumber. Due to the symmetry property of the DFT, the power spectrum will have the same magnitudes at wavenumbers n and $n - \kappa$ for all κ. The knowledge that a scaled function corresponds to a scaled magnitude in its power spectrum can help us identify scaling factors in the algorithmic output.

Now that we have outlined the key notation, definitions and theorems used within this thesis, we can move onto outlining the variational problem.

2.2 3D and 4D-Var

A VarDA scheme incorporates information from a prior forecast along with observations over a spatial domain in the form of a nonlinear least-squares objective function, which is then minimised using an iterative method. In VarDA, the nonlinear least-squares problem is solved as a sequence of linear least-squares problems using an incremental method, which has been shown to be equivalent to the Gauss-Newton method under certain conditions [65]. In the incremental method, the minimisation of the nonlinear objective function and the linearised subproblem are referred to as the 'outer loop' and the 'inner loop' respectively.

Within this thesis, we focus on two VarDA schemes; 3D-Var and 4D-Var. The 3D-Var scheme only makes use of observations at the start of the assimilation time-window and

was first implemented operationally by ECMWF from 1996 until 1997 when it was replaced by 4D-Var [105]. The 4D-Var scheme, an extension to 3D-Var, also considers the temporal location of observations. In this section, we outline the 4D-Var scheme with assimilation time-window length N, while keeping in mind that the same equations apply for the 3D-Var scheme where $N = 0$ [27].

2.2.1 Standard formulation

In four-dimensional variational data assimilation (4D-Var), the analysis $\mathbf{x}_0^a \in \mathbb{R}^n$ is obtained by minimising a objective function consisting of two terms: the background term and the observation term, namely;

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)). \tag{2.50}$$

The background term measures the difference between the initial state of the system and the background state vector $\mathbf{x}_0^b \in \mathbb{R}^n$, which contains prior information. The observation term measures the difference between information from observations at times t_i in the observation vector $\mathbf{y}_i \in \mathbb{R}^{p_i}$ and the model state vector $\mathbf{x}_i \in \mathbb{R}^n$ at the same time through use of the observation operator $\mathcal{H}_i : \mathbb{R}^n \to \mathbb{R}^{p_i}$ that maps from the model state space to the observation space. Both terms are weighted by their corresponding covariance matrices to represent the uncertainty in the respective measures, the background error covariance matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ and the observation error covariance matrices at times t_i, $\mathbf{R}_i \in \mathbb{R}^{p_i \times p_i}$, which are assumed to be symmetric positive definite. We note that observations are distributed both in time and space and there are usually fewer observations available than there are state variables so $p < n$, where $p = \sum_{i=0}^{N} p_i$.

The 4D-Var objective function (2.50) is subject to the nonlinear dynamical model equations which contain the physics of the system

$$\mathbf{x}_i = \mathcal{M}_{0,i}(\mathbf{x}_0), \tag{2.51}$$

where the nonlinear model $\mathcal{M}_{0,i} : \mathbb{R}^n \to \mathbb{R}^n$ evolves the state vector from the initial time point t_0 to the time point t_i.

By including the model information within the objective function, we are able to write the constrained optimisation problem (2.50)-(2.51) in the form of an unconstrained optimisation problem and apply the minimisation methods described later in this thesis.

$$\begin{aligned}\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) \\ + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathcal{M}_{0,i}(\mathbf{x}_0)))^T \mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathcal{M}_{0,i}(\mathbf{x}_0))).\end{aligned} \tag{2.52}$$

We note that the function (2.52) is continuously differentiable if the operators $\mathcal{H}_i$ and $\mathcal{M}_{0,i}$ are continuously differentiable. To save both computational cost and time in VarDA, the

nonlinear operators in (2.52) are often replaced by their Jacobians, the tangent linear model (TLM) and tangent linear observation operator, for use in the inner loop [27].

Equation (2.52) is equivalent to a nonlinear least-squares function of the form (2.28) where the residual vector $\mathbf{r}(\mathbf{x}_0) \in \mathbb{R}^{(n+p)}$ and its Jacobian $\mathbf{J}(\mathbf{x}_0)$ are given by

$$\mathbf{r}(\mathbf{x}_0) = \begin{pmatrix} \mathbf{B}^{-1/2}(\mathbf{x}_0 - \mathbf{x}_0^b) \\ \mathbf{R}_0^{-1/2}(\mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0) \\ \mathbf{R}_1^{-1/2}(\mathbf{y}_1 - \mathcal{H}_1(\mathcal{M}_{0,1}(\mathbf{x}_0))) \\ \vdots \\ \mathbf{R}_N^{-1/2}(\mathbf{y}_N - \mathcal{H}_N(\mathcal{M}_{0,N}(\mathbf{x}_0))) \end{pmatrix} \text{ and } \mathbf{J}(\mathbf{x}_0) = \begin{pmatrix} \mathbf{B}^{-1/2} \\ -\mathbf{R}_0^{-1/2}\mathbf{H}_0 \\ -\mathbf{R}_1^{-1/2}\mathbf{H}_1\mathbf{M}_{0,1} \\ \vdots \\ -\mathbf{R}_N^{-1/2}\mathbf{H}_N\mathbf{M}_{0,N} \end{pmatrix}, \tag{2.53}$$

where

$$\mathbf{M}_{0,i} = \frac{\partial \mathcal{M}_{0,i}}{\partial \mathbf{x}_0}\Big|_{\mathcal{M}_{0,i}(\mathbf{x}_0)} \text{ and } \mathbf{H}_i = \frac{\partial \mathcal{H}_i}{\partial \mathbf{x}_0}\Big|_{\mathcal{M}_{0,i}(\mathbf{x}_0)} \tag{2.54}$$

are the Jacobian matrices of the model operator $\mathcal{M}_{0,i}$ and observation operator $\mathcal{H}_i$ respectively, $\mathbf{M}_{0,i} \in \mathbb{R}^{n\times n}$ is the tangent linear of $\mathcal{M}_{0,i}$ and $\mathbf{H}_i \in \mathbb{R}^{p_i \times n}$ is the tangent linear of $\mathcal{H}_i$ [95].

Consider a perturbation of $\mathbf{x}_i$, $\delta\mathbf{x}_i = \mathcal{M}_{0,i}(\delta\mathbf{x}_0)$. Using the Taylor series expansion, we have the following relation

$$\begin{aligned} \mathcal{M}_{0,i}(\mathbf{x}_0 + \delta\mathbf{x}_0) &= \mathcal{M}_{0,i}(\mathbf{x}_0) + \mathbf{M}_{0,i}(\delta\mathbf{x}_0) + h.o.t. \\ &\approx \mathbf{M}_{0,i}(\delta\mathbf{x}_0) \end{aligned} \tag{2.55}$$

The following measure is used to check whether the TLM $\mathbf{M}_{0,i}$ of the nonlinear model $\mathcal{M}_{0,i}$ is correct,

$$\frac{\|\mathcal{M}_{0,i}(\mathbf{x}_0 + \alpha\delta\mathbf{x}_0) - \mathcal{M}_{0,i}(\mathbf{x}_0) - \mathbf{M}_{0,i}\alpha\delta\mathbf{x}_0\|}{\|\mathbf{M}_{0,i}\alpha\delta\mathbf{x}_0\|}. \tag{2.56}$$

The measure (2.56) is used to calculate the relative error between the nonlinear perturbation and the linearised perturbation at each time point i [104].

For the nonlinear observation operator, we have the relation

$$\mathcal{H}_i(\mathbf{x}_i + \delta\mathbf{x}_i) \approx \mathcal{H}_i(\mathbf{x}_i) + \mathbf{H}_i\delta\mathbf{x}_i. \tag{2.57}$$

The iterative methods used to minimise the objective function (2.52) in VarDA practice and in our work require the use of the gradient and approximate Hessian of (2.52). In practice, an adjoint method is used to calculate the gradient of (2.52) (see Section 4.1 of [95]), defined as

$$\nabla\mathcal{J}(\mathbf{x}_0) = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) - \sum_{i=0}^{N} \mathbf{M}_{0,i}^T\mathbf{H}_i^T\mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathcal{M}_{0,i}(\mathbf{x}_0))), \tag{2.58}$$

where $\mathbf{M}_{0,i}^T$ and $\mathbf{H}_i^T$ are the adjoint operators. Due to the model complexity, implementing and maintaining the adjoint can be time consuming as when the model is changed, the adjoint must also be updated. Within our work, we calculate the gradient directly as in (2.58).

In DA, the second-order terms (2.32) in (2.31) are often difficult to calculate in the time and cost available and too large to store. Therefore, a first-order approximation to the Hessian of the objective function (2.52) is used and is given by

$$\nabla^2 \mathcal{J}(\mathbf{x}_0) = \mathbf{B}^{-1} + \sum_{i=0}^{N} \mathbf{M}_{0,i}^T \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \mathbf{M}_{0,i}, \tag{2.59}$$

which is symmetric positive definite when $\mathbf{J}(\mathbf{x}_0)$ has full column rank, which is the case if $\mathbf{B}$ is full rank. When this assumption holds, the condition number in the 2-norm of (2.59) $\kappa(\nabla^2 \mathcal{J}(\mathbf{x}_0))$ can be calculated using (2.15) and is related to the number of iterations used for the linear minimisation problems in VarDA and how sensitive the estimate of the initial state is to perturbations of the data. We can use $\kappa(\nabla^2 \mathcal{J}(\mathbf{x}_0))$ to indicate how quickly and accurately the optimisation problem can be solved [47].

In the following section, we outline the incremental formulation of the variational problem, the formulation commonly used in operational implementations, see for example [27].

2.2.2 Incremental formulation

For the incremental 4D-Var formulation, we first define the full resolution increment $\delta\mathbf{x}_0 \in \mathbb{R}^n$ by

$$\delta\mathbf{x}_0 = \mathbf{x}_0 - \mathbf{x}_0^b \tag{2.60}$$

and the innovation vector by

$$\mathbf{d}_i = \mathbf{y}_i - \mathcal{H}_i(\mathcal{M}_{0,i}(\mathbf{x}_0)). \tag{2.61}$$

To save computational time and cost, in January 2003, ECMWF implemented the use of an algorithm that calculates the innovation vectors at high resolution, to maximise the use of observations, and interpolates the background (initial guess for the algorithm) from the high resolution to the inner loop resolution using a restriction operator [107]. After the inner loop minimisation, the increments are projected to the high resolution and used to update the outer loop. The observation operator is also linearised about the high resolution nonlinear state interpolated to the inner loop resolution.

To restrict the increment $\delta\mathbf{x}_0$ to the reduced dimensional space $\mathbb{R}^r$ where $r < n$, we define the restriction operator $\mathbf{S}_l : \mathbb{R}^n \to \mathbb{R}^r$ such that the reduced resolution increment $\delta\hat{\mathbf{x}}_0 \in \mathbb{R}^r$ is given by

$$\delta\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_0 - \hat{\mathbf{x}}_0^b, \tag{2.62}$$

where $\hat{\mathbf{x}}_0 = \mathbf{S}_l\mathbf{x}_0$ and $\hat{\mathbf{x}}_0^b = \mathbf{S}_l\mathbf{x}_0^b$. We also define the extension operator $\mathbf{S}_h : \mathbb{R}^r \to \mathbb{R}^n$ that is used to interpolate the reduced space increment $\delta\hat{\mathbf{x}}_0$ to the outer loop space $\mathbb{R}^n$ at the end of each series of inner loop minimisations. Using this formulation, the linearised 4D-Var objective function is given by

$$\begin{aligned}\hat{\mathcal{J}}(\delta\hat{\mathbf{x}}_0) = \frac{1}{2}(\delta\hat{\mathbf{x}}_0 + \mathbf{S}_l(\mathbf{x}_0 - \mathbf{x}_0^b))^T \hat{\mathbf{B}}^{-1}(\delta\hat{\mathbf{x}}_0 + \mathbf{S}_l(\mathbf{x}_0 - \mathbf{x}_0^b)) \\ + \frac{1}{2}\sum_{i=0}^{N}(\hat{\mathbf{H}}_i\hat{\mathbf{M}}_{0,i}\delta\hat{\mathbf{x}}_0 - \mathbf{d}_i)^T \mathbf{R}_i^{-1}(\hat{\mathbf{H}}_i\hat{\mathbf{M}}_{0,i}\delta\hat{\mathbf{x}}_0 - \mathbf{d}_i)\end{aligned} \tag{2.63}$$

where $\hat{\mathcal{J}}$ denotes the reduced resolution inner loop cost function,

$$\hat{\mathbf{B}} = \mathbf{S}_l \mathbf{B} \mathbf{S}_l^T \tag{2.64}$$

is the reduced space background error covariance matrix and

$$\hat{\mathbf{H}}_i = \mathbf{H}_i \mathbf{S}_h, \tag{2.65}$$

is the reduced space observation operator where $\hat{\mathbf{B}} \in \mathbb{R}^{r \times r}$ and $\hat{\mathbf{H}}_i \in \mathbb{R}^{p \times r}$. The linearised model operator in the reduced resolution is given by

$$\hat{\mathbf{M}}_{0,i} = \mathbf{S}_l \mathbf{M}_{0,i} \mathbf{S}_h, \tag{2.66}$$

as in [127]. The reduced resolution residual vector and its Jacobian are given by

$$\hat{\mathbf{r}}(\mathbf{x}_0) = \begin{pmatrix} \hat{\mathbf{B}}^{-1/2} \mathbf{S}_l (\mathbf{x}_0 - \mathbf{x}_0^b) \\ \mathbf{R}_0^{-1/2} (\mathbf{y}_0 - \mathcal{H}_0(\mathbf{x}_0) \\ \mathbf{R}_1^{-1/2} (\mathbf{y}_1 - \mathcal{H}_1(\mathcal{M}_{0,1}(\mathbf{x}_0))) \\ \vdots \\ \mathbf{R}_N^{-1/2} (\mathbf{y}_N - \mathcal{H}_N(\mathcal{M}_{0,N}(\mathbf{x}_0))) \end{pmatrix} \text{ and } \hat{\mathbf{J}}(\mathbf{x}_0) = \begin{pmatrix} \hat{\mathbf{B}}^{-1/2} \\ -\mathbf{R}_0^{-1/2} \hat{\mathbf{H}}_0 \\ -\mathbf{R}_1^{-1/2} \hat{\mathbf{H}}_1 \hat{\mathbf{M}}_{0,1} \\ \vdots \\ -\mathbf{R}_N^{-1/2} \hat{\mathbf{H}}_N \hat{\mathbf{M}}_{0,N} \end{pmatrix}, \tag{2.67}$$

where $\hat{\mathbf{r}}(\mathbf{x}) \in \mathbb{R}^{(r+p)}$ and its Jacobian $\hat{\mathbf{J}}(\mathbf{x}) \in \mathbb{R}^{(r+p) \times r}$, so the VarDA inner loop minimisation problem has a reduced dimension.

In operational DA, the inverse of the background error covariance matrix is only specified on the inner loop level and due to its size, the full resolution $\mathbf{B}^{-1}$ is not computed explicitly. Furthermore, it is known that the use of realistic background error statistics is important in VarDA as it has a profound impact on the analysis [4]. The Met Office commonly model their background error correlations using the second-order auto-regressive (SOAR) distribution, see for example [59, 74, 75, 115] and update their $\mathbf{B}$ using flow-dependent information from ensembles in their hybrid 4D-Var scheme [25]. The ECMWF also update their static $\mathbf{B}$ model (see [5, 38]) using flow-dependent ensemble information from their ensemble of data assimilations (EDA) scheme [14, 15]. In addition, the variational problem is preconditioned to simplify calculations and solve the problem in real-time. This is often achieved by the use of a variable transformation and is outlined in the following section.

2.3 Control variable transform

Preconditioning the 4D-Var problem using a variable transform has been shown to improve the conditioning of the variational optimisation problem [56, 57]. By preconditioning using the square root of $\mathbf{B}$, we are able to avoid (explicitly or by the use of matrix-vector products) the calculation of $\mathbf{B}$. This is particularly important when solving high-dimensional problems such as those in operational settings.

To be able to use the negative square root of $\hat{\mathbf{B}}$ in our variable transformation, we first require the assumption that the matrix $\hat{\mathbf{B}}$ is full rank. This assumption is satisfied for the

choices of $\mathbf{B}$ used in our numerical experiments in later chapters. We define a new variable in the reduced resolution space $\delta\hat{\mathbf{z}}_0 \in \mathbb{R}^r$ to be,

$$\delta\hat{\mathbf{z}}_0 = \hat{\mathbf{B}}^{-1/2}\delta\hat{\mathbf{x}}_0, \tag{2.68}$$

which is known as the 'control variable' in DA terms. The 4D-Var objective function can then be written in terms of $\delta\hat{\mathbf{z}}_0$ and minimised with respect to this instead. The preconditioned linearised 4D-Var objective function is given by

$$\begin{aligned}\hat{\mathcal{J}}_p(\delta\hat{\mathbf{z}}_0) = \frac{1}{2}(\delta\hat{\mathbf{z}}_0 + \mathbf{S}_l(\mathbf{z}_0 - \mathbf{z}_0^b))^T(\delta\hat{\mathbf{z}}_0 + \mathbf{S}_l(\mathbf{z}_0 - \mathbf{z}_0^b)) \\ + \frac{1}{2}\sum_{i=0}^{N}(\hat{\mathbf{H}}_i\hat{\mathbf{M}}_{0,i}\hat{\mathbf{B}}^{1/2}\delta\hat{\mathbf{z}}_0 - \mathbf{d}_i)^T\mathbf{R}_i^{-1}(\hat{\mathbf{H}}_i\hat{\mathbf{M}}_{0,i}\hat{\mathbf{B}}^{1/2}\delta\hat{\mathbf{z}}_0 - \mathbf{d}_i).\end{aligned} \tag{2.69}$$

The use of the variable transform (2.68) results in the first term of the reduced resolution preconditioned Hessian being the identity matrix, as in the following

$$\nabla^2\hat{\mathcal{J}}_p = \mathbf{I} + \sum_{i=0}^{N}\hat{\mathbf{B}}^{1/2}\hat{\mathbf{M}}_{0,i}^T\hat{\mathbf{H}}_i^T\mathbf{R}_i^{-1}\hat{\mathbf{H}}_i\hat{\mathbf{M}}_{0,i}\hat{\mathbf{B}}^{1/2}. \tag{2.70}$$

By construction, the preconditioned 4D-Var Hessian (2.70) is full rank and symmetric positive definite.

Within this thesis, we focus on the strong-constraint 4D-Var problem where we assume the numerical model of the system perfectly represents the true dynamics of the system, or the model errors are small enough to be neglected. This formulation has been commonly used operationally in many meteorological centres [103], including the Meteorological Service of Canada [43], ECMWF [63, 80, 105] and the Met Office [109]. In the following section, we outline the two numerical models used within the numerical experiments of this thesis.

2.4 Numerical models

Within this section, we outline two numerical models used within our work, namely, the Lorenz 1963 and 1996 models. We acknowledge in this work that the code for the Lorenz 1963 model was developed by Amos S. Lawless and the code for the Lorenz 1996 model was developed by Adam El-Said.

2.4.1 Lorenz 1963 model

Proposed in [77], the Lorenz 63 model (L63) is a popular experimental dynamical system that represents meteorological processes using a simple model. The model consists of three

nonlinear, ordinary differential equations given as

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x),\\ \frac{dy}{dt} &= x(\rho-z)-y,\\ \frac{dz}{dt} &= xy-\beta z,\end{aligned} \tag{2.71}$$

where the state vector consists of $n=3$ time-dependent variables $\mathbf{x}=[x(t),y(t),z(t)]^T\in\mathbb{R}^3$. The scalar parameters are chosen to be $\sigma=10$, $\rho=\frac{8}{3}$ and $\beta=28$, making the system chaotic.

To discretise the Lorenz 1963 model equations, we use a modified Euler scheme known as the second-order Runge-Kutta scheme for ordinary differential equations (see [119]), resulting in the following set of discretised equations

$$\begin{aligned}x^{i+1} &= x^i+\frac{\Delta t}{2}\sigma\Big[2(y^i-x^i)+\Delta t(\rho x^i-y^i-x^iz^i)-\Delta t\sigma(y^i-x^i)\Big]\\ y^{i+1} &= y^i+\frac{\Delta t}{2}\Big[\rho x^i-y^i-x^iz^i+\rho(x^i+\Delta t\sigma(y^i-x^i))-y^i-\Delta t(\rho x^i-y^i-x^iz^i)\\ &\quad-(x^i+\Delta t\sigma(y^i-x^i))(z^i+\Delta t(x^iy^i-\beta z^i))\Big]\\ z^{i+1} &= z^i+\frac{\Delta t}{2}\Big[(x^iy^i-\beta z^i)+(x^i+\Delta t\sigma(y^i-x^i))(y^i+\Delta t(\rho x^i-y^i-x^iz^i))\\ &\quad-\beta z^i-\Delta t(x^iy^i-\beta z^i)\Big],\end{aligned} \tag{2.72}$$

where i denotes the time step index and Δt denotes the model time step. Within our work, we set $\Delta t=0.025$.

The tangent linear code is obtained by forming $\mathcal{M}(\mathbf{x}+\delta\mathbf{x})-\mathcal{M}(\mathbf{x})$ as follows

$$\mathcal{M}(\mathbf{x}+\delta\mathbf{x})-\mathcal{M}(\mathbf{x})=\begin{pmatrix}-\sigma & \sigma & 0\\ \rho-z & -1 & -x\\ y & x & -\beta\end{pmatrix}\begin{pmatrix}\delta x\\ \delta y\\ \delta z\end{pmatrix}+\begin{pmatrix}0\\ -\delta x\delta z\\ \delta x\delta y\end{pmatrix} \tag{2.73}$$

and neglecting the second order terms $[0,-\delta x\delta z,\delta x\delta y]^T$. The linearised equations are then discretised using a second-order Runge-Kutta scheme. The TLM, $\mathbf{M}_{0,N}$, of (2.71) satisfies

$$\begin{pmatrix}\delta x\\ \delta y\\ \delta z\end{pmatrix}_N=\mathbf{M}_{0,N}\begin{pmatrix}\delta x\\ \delta y\\ \delta z\end{pmatrix}_0, \tag{2.74}$$

where $[\delta x,\delta y,\delta z]_i^T$ is the vector of perturbations at time i. For our experiments, we calculate the TLM, $\mathbf{M}_{0,N}$, of (2.71) numerically by applying the tangent linear code to the unit vectors, so the columns of $\mathbf{M}_{0,N}$ are the evolved perturbations.

In this section, we have outlined the Lorenz 1963 model, its discretisation and TLM. In the following section, we outline the Lorenz 1996 model.

2.4.2 Lorenz 1996 model

Another popular experimental system is the one-dimensional atmospheric Lorenz 96 model (L96) [78] given by the following n equations,

$$\frac{dx_j}{dt} = -x_{j-2}x_{j-1} + x_{j-1}x_{j+1} - x_j + F, \tag{2.75}$$

where $j = 1, 2, \ldots, n$ is a spatial coordinate. For a forcing term $F = 8$ and $n = 40$ state variables, the system is chaotic [78]. The variables are evenly distributed over a circle of latitude of the Earth with n points with a cyclic domain and a single time unit is equivalent to approximately 5 atmospheric days. A fourth-order Runge-Kutta method is used to discretise the model equations using a time step $\Delta t = 0.025$ (approximately 3 hours). We omit the discretisation of (2.75) here due to the dimension of the model. We instead refer the reader to [78] for more details.

The tangent linear code is obtained by first linearising the nonlinear model (2.75),

$$\mathcal{M}(\mathbf{x} + \delta\mathbf{x}) - \mathcal{M}(\mathbf{x}) = -\delta x_{j-2}x_{j-1} + \delta x_{j-1}(x_{j+1} - x_{j-2}) - \delta x_j + \delta x_{j+1}x_{j-1} + F, \tag{2.76}$$

where $j = 1, 2, \ldots n$. As with the nonlinear model, the linearised equations are then discretised using a fourth-order Runge-Kutta scheme.

Now that we have introduced the two models used in our experimental work, we conclude this chapter in the following.

2.5 Conclusion

Within this chapter, we have introduced the mathematical preliminaries used throughout this thesis. We then outlined both the standard and incremental formulations of the 4D-Var method, noting that the same equations can be used for the 3D-Var method where $N = 0$. Next, we introduced the preconditioned formulation using the square-root of the background error covariance matrix; a preconditioner commonly used in practice for the VarDA inner loop. Finally, we outlined the numerical models used in the 4D-Var experimental work of this thesis.

In the following chapter, we provide a review of the optimisation methods used to solve nonlinear least-squares problems, including the variational problem.

Chapter 3

Numerical Optimisation

Within this chapter, we outline the theory behind the optimisation methods used to solve nonlinear least-squares problems (NLLSPs) of the form (2.33). We focus specifically on Gauss-Newton methods as they are what are used in practice to minimise the VarDA objective function (2.2) and in our work in Chapters 4, 5 and 6. We outline and discuss the (standard) Gauss-Newton method (GN) along with Gauss-Newton equipped with strategies to guarantee convergence: backtracking Armijo line search (LS) and regularisation (REG). We provide a review of the methods used in VarDA practice and discuss the simplifications made to be able to solve the variational problem in the time and computational cost available, specifically the use of a reduced resolution inner loop.

We begin by outlining the theory of NLLSPs in Section 3.1. In Section 3.2, we outline the fundamental optimisation methods used to solve NLLSPs, namely, the Steepest Descent method, Newton's method and GN. In Section 3.3, we focus on two optimisation methods that use strategies to guarantee convergence, namely LS and REG. In Section 3.4, we review the optimisation methods that have been applied to the variational problem in practice. In Section 3.5, we outline some algorithmic considerations we make within our work. We conclude this chapter in Section 3.6.

3.1 Nonlinear least-squares problems

Let $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{x} \in \mathbf{R}^n$. In unconstrained optimisation, one aims to find the minimum (or maximum) of $f(\mathbf{x})$, referred to as the objective function, with no restriction on the values the variables $\mathbf{x}$ can take. In VarDA, it is common to assume that f is twice continuously differentiable, as defined in Definition 2.1.13. The optimisation algorithms we will be focusing on are specific to solving nonlinear least-squares problems where the function f is of the form (2.28).

In numerical optimisation, algorithms must first be *initialised* using an initial vector of variables, $\mathbf{x}^{(0)}$, referred to as an initial guess. The initial guess can then be used to generate a sequence of iterates $\left\{\mathbf{x}^{(k)}\right\}$ where $k = 0, 1, 2, \ldots,$. Each iteration should improve the estimate of the solution to the problem, aiming to terminate with a solution $\mathbf{x}^*$ to the problem

(2.33). Such algorithms are referred to as iterative methods, defined as follows.

Definition 3.1.1 (An iterative method)**.** *Let* $\mathbf{x} \in \mathbb{R}^n$*. Given an initial guess* $\mathbf{x}^{(0)}$*, an iterative method generates a sequence of iterates* $\{\mathbf{x}^{(k)}\}$ *where* $k = 1, 2, \ldots$.

An iterative method is terminated when it meets a user specified criteria, this is discussed in detail later in Section 3.5.1. Furthermore, the sequence of iterates of an iterative method is generated using information about the function at the current and/or past iterates in each iteration k of an iterative method. This information is in the form of a vector, referred to as the search direction and is obtained by a line search method, defined in the following.

Definition 3.1.2 (A line search method (See [96] Section 2.2))**.** *Let* $f : \mathbb{R}^n \to \mathbb{R}$*. A line search method is an iterative method that aims to find a search direction* $\mathbf{s} \in \mathbb{R}^n$ *at each iteration* k*, where* $k = 0, 1, 2, \ldots$*. The search direction is used to update the iterate* $\mathbf{x}^{(k)}$*, as in the following*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}. \tag{3.1}$$

Within our work, we are concerned with those methods that choose a search direction $\mathbf{s}^{(k)}$ that results in either a monotonic or strict decrease in f [96]. We discuss such choices in Sections 3.2.1, 3.2.2 and 3.2.3.

An iterative method is said to be convergent if its sequence of iterates converges, as defined in the following.

Definition 3.1.3 (Convergence of a sequence of iterates (See [33] Definition 2.3.1))**.** *Let* $\mathbf{x}^* \in \mathbb{R}^n$ *and* $\mathbf{x}^{(k)} \in \mathbb{R}^n$*, where* $k = 0, 1, 2, \ldots$*. Then the sequence of iterates* $\{\mathbf{x}^{(k)}\}$ *is said to converge to* $\mathbf{x}^*$ *if*

$$\lim_{k \to \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\| = 0. \tag{3.2}$$

We next define a one-step stationary iteration, a point of attraction and a fixed point, all of which are used to prove convergence of some of the methods we are concerned with within this thesis.

Definition 3.1.4 (Stationary iterations (see [98] section 10.1))**.** *One-step stationary iterations are of the form*

$$\mathbf{x}^{(k+1)} = \mathbf{G}(\mathbf{x}^{(k)}), \qquad k = 0, 1, \ldots, \tag{3.3}$$

where $\mathbf{G} : \mathbf{D} \subset \mathbb{R}^n \to \mathbb{R}^n$*.*

Definition 3.1.5 (Point of attraction (see [98] Definition 10.1.1))**.** *Let* $\mathbf{G} : \mathbf{D} \subset \mathbb{R}^n \to \mathbb{R}^n$*. Then* $\mathbf{x}^*$ *is a point of attraction of the iteration* (3.3) *if there is an open neighbourhood* $\mathbf{S}$ *of* $\mathbf{x}^*$ *such that* $\mathbf{S} \subset \mathbf{D}$ *and, for any* $\mathbf{x}^{(\mathbf{0})} \in \mathbf{S}$*, the iterates* $\{\mathbf{x}^{(k)}\}$ *defined by* (3.3) *all lie in* $\mathbf{D}$ *and converge to* $\mathbf{x}^*$*.*

Definition 3.1.6 (Fixed point (see [98] section 5.1))**.** *Let* $\mathbf{G} : \mathbf{D} \subset \mathbb{R}^n \to \mathbb{R}^n$ *where* $\mathbf{D}$ *is an open set and let* $\mathbf{x}^* \in \mathbf{D}$ *be a point of attraction.* $\mathbf{G}$ *is said to have a fixed point at* $\mathbf{x}^*$ *iff*

$$\mathbf{G}(\mathbf{x}^*) = \mathbf{x}^*. \tag{3.4}$$

The performance of an iterative method is often gauged by its rate of convergence to a solution $\mathbf{x}^*$. The following definitions state different rates of convergence of an iterative method.

Definition 3.1.7 (q-linear convergence (See [33] Definition 2.3.1))**.** *In addition to Definition 3.1.3, the sequence of iterates $\{\mathbf{x}^{(k)}\}$ is said to be q-linearly convergent to $\mathbf{x}^*$ if there exists a constant $c \in [0, 1)$ and an integer k_1 such that $\forall k \geq k_1$,*

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq c\|\mathbf{x}^{(k)} - \mathbf{x}^*\|. \tag{3.5}$$

Linear convergence is the minimal requirement for an algorithm as in this case, the error will decrease by a factor $c < 1$ [32]. A more desirable rate of convergence is a quadratic rate, defined as follows.

Definition 3.1.8 (q-quadratic convergence (See [33] Definition 2.3.1))**.** *In addition to Definition 3.1.3, the sequence of iterates $\{\mathbf{x}^{(k)}\}$ is said to be q-quadratically convergent to $\mathbf{x}^*$ if there exist constants $p > 1, c \geq 0$, and an integer k_1 such that the sequence of iterates $\{\mathbf{x}^{(k)}\}$ converges to $\mathbf{x}^*$ and $\forall k \geq k_1$,*

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\| \leq c\|\mathbf{x}^{(k)} - \mathbf{x}^*\|^2. \tag{3.6}$$

There are a variety of solution types $\mathbf{x}^*$ that an optimisation algorithm may converge to. Figure 3.1 is a visual interpretation of types of global and local minima (and maxima). Within our work, we are mostly concerned with three of these solution types, namely, (strict) global minima, strict local minima and weak local minima.

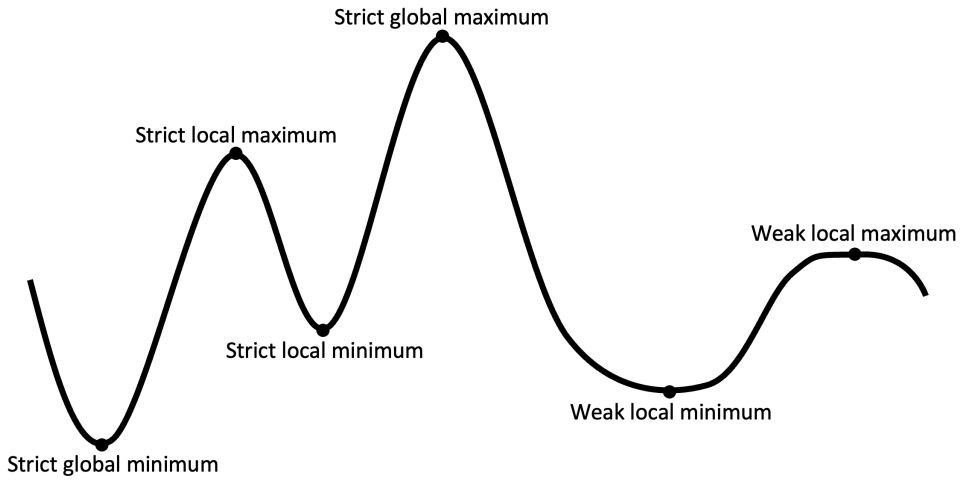


Figure 3.1: Types of minima and maxima schematic.

We would ideally like to find *a global minimiser*; a solution that gives us the lowest value of the cost function amongst all feasible points. A global minimiser is defined as follows.

Definition 3.1.9 (Global minimiser (see [96], Section 2.1))**.** *A point $\mathbf{x}^* \in \mathbb{R}^n$ is a (strict/strong) global minimiser if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$.*

However, a global minimiser is often difficult to locate in most cases due to the nonlinearity of the problems. Therefore, *a local minimiser*, where the objective function is the lowest amongst the nearby points, is often sought by algorithms for nonlinear optimisation. A local minimiser is defined as follows.

Definition 3.1.10 (Local minimiser (see [96], Section 2.1)). *A point $\mathbf{x}^*$ is a local minimiser if there is a neighbourhood $\mathcal{N}$ of $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}$.*

Definition 3.1.11 (Strict and weak local minimisers (see [96], Section 2.1)). *A point $\mathbf{x}^*$ is said to be a strict (or strong) local minimiser if there is a neighbourhood $\mathcal{N}$ of $\mathbf{x}^*$ such that $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}$ where $\mathbf{x} \neq \mathbf{x}^*$. Otherwise, $\mathbf{x}^*$, as defined in Definition 3.1.10, is referred to as a weak local minimiser.*

Some optimisation algorithms can only guarantee convergence to a local minimum under certain conditions and not necessarily convergence to a global minimum. This is dependent on how close the initial guess is from the local minimum the algorithm locates.

Definition 3.1.12 (Locally convergent method). *Let $\mathbf{x}^* \in \mathbb{R}^n$, $\mathbf{x}^{(k)} \in \mathbb{R}^n$, $k = 0, 1, 2, \ldots$. An iterative method is said to be locally convergent if it produces a sequence of iterates $\{\mathbf{x}^{(k)}\}$ that converge to a local minimum $\mathbf{x}^*$ given that the initial guess $\mathbf{x}^{(0)} \in \mathcal{N}$ where $\mathcal{N} \subset \mathbb{R}^n$ is some neighbourhood around $\mathbf{x}^*$.*

Definition 3.1.13 (Globally convergent method). *Let $\mathbf{x}^* \in \mathbb{R}^n$, $\mathbf{x}^{(k)} \in \mathbb{R}^n$, $k = 0, 1, 2, \ldots$. An iterative method is said to be globally convergent if it produces a sequence of iterates $\{\mathbf{x}^{(k)}\}$ that converge to a local minimum $\mathbf{x}^*$ given any initial guess $\mathbf{x}^{(0)}$.*

Furthermore, the following conditions follow from $\mathbf{x}^*$ being a local minimiser of the function f. These conditions, are based on $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$, which denote the first and second derivative of f at a point $\mathbf{x} \in \mathbb{R}^n$ respectively.

Theorem 3.1.1 (First-Order Necessary Conditions (see [96] Theorem 2.2)). *If $\mathbf{x}^*$ is a local minimiser and f is continuously differentiable in an open neighbourhood of $\mathbf{x}^*$, then $\nabla f(\mathbf{x}^*) = 0$.*

Proof. See [96] Section 2.1.

Theorem 3.1.2 (Second-Order Necessary Conditions (see [96] Theorem 2.3)). *If $\mathbf{x}^*$ is a local minimiser of f and $\nabla^2 f(\mathbf{x}^*)$ exists and is continuous in an open neighbourhood of $\mathbf{x}^*$, then $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is positive semidefinite.*

Proof. See [96] Section 2.1.

Theorem 3.1.3 (Second-Order Sufficient Conditions (see [96] Theorem 2.4)). *Suppose that $\nabla^2 f(\mathbf{x}^*)$ is continuous in an open neighbourhood of $\mathbf{x}^*$ and that $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is positive definite. Then $\mathbf{x}^*$ is a strict local minimiser of f.*

Proof. See [96] Section 2.1.

The size of the residual vector in (2.28) at the solution $\mathbf{x}^*$ greatly impacts the convergence properties of an optimisation method. There are two types of problems that we consider in our work, which are defined in the following.

Definition 3.1.14 (Zero and nonzero residual problems)**.** *Let* $\mathbf{x}^* \in \mathbb{R}^n$ *and* $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$. *A nonlinear least-squares (NLLS) problem is said to be a zero residual problem if*

$$\mathbf{r}(\mathbf{x}^*) = 0, \tag{3.7}$$

otherwise, it is said to be a nonzero residual problem.

In the following section, we outline three fundamental methods used to solve NLLS problems, namely, the Steepest Descent method, the Newton method and the Gauss-Newton method.

3.2 Basic methods

3.2.1 Steepest Descent Method

The Steepest Descent (SD) method, also known as gradient descent, is a gradient-based minimisation method in its simplest form. It is used to solve nonlinear least-squares problems and chooses the search direction as the negative gradient such that,

$$\mathbf{s}_{SD}^{(k)} = -\nabla f(\mathbf{x}^{(k)}), \tag{3.8}$$

at each iteration k (see [96] Section 2.2). A summary of the SD method is given by the following.

Algorithm 3.2.1: SD algorithm applied to (2.33) **[33].**

Step 0: Initialisation. Given $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and some stopping criteria. Set $k = 0$.

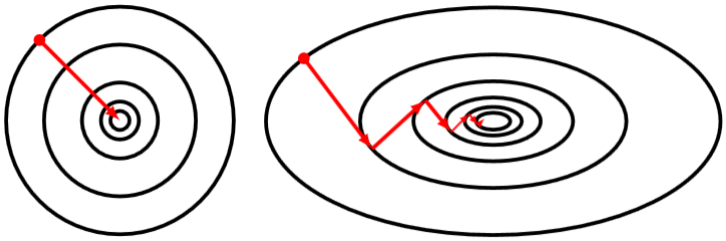
Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation. Compute a step $\mathbf{s}_{SD}^{(k)}$ that satisfies Equation (3.8).

Step 3: Iterate update. Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_{SD}^{(k)}$, $k := k + 1$ and go to Step 1.

Although the SD method only requires the use of first-order derivatives, its convergence to a solution in complex problems can be very slow. When the iso-surfaces of the cost function are almost spherical as in Figure 3.2a, the method works well as the problem is well-conditioned and the descent direction is towards the minimum. When the iso-surfaces are ellipsoidal as in Figure 3.2b, the problem is ill-conditioned and the convergence of the SD method to a solution can be very slow.

The performance of the SD method depends on the scaling of the problem, that is, how f varies if $\mathbf{x}$ was changed. A poorly scaled problem results in the method performing poorly.



(a) Spherical (b) Ellipsoidal

Figure 3.2: Diagram 3.2a represents a cost function with spherical iso-surfaces. Diagram 3.2b represents a cost function with ellipsoidal iso-surfaces (Courtesy of [85]).

This is illustrated in [96] Chapter 2. The use of second derivative information, as in Newton's method, may be necessary to aid convergence as this gives some indication of the curvature of the cost function and its performance is unaffected by poorly scaled problems, [85]. The theory behind Newton's method will be outlined in the following section.

3.2.2 Newton's Method

Unlike the SD method, Newton's method makes use of second-order information to find a descent direction by solving

$$\nabla^2 f(\mathbf{x}^{(k)})\mathbf{s}_{NEW}^{(k)} = -\nabla f(\mathbf{x}^{(k)}), \tag{3.9}$$

for the Newton search direction $\mathbf{s}_{NEW}^{(k)}$. A summary of the Newton method is given by the following.

Algorithm 3.2.2: Newton's algorithm applied to (2.33) **[33].**

Step 0: Initialisation. Given $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and some stopping criteria. Set $k = 0$.

Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation.
Compute a step $\mathbf{s}_{NEW}^{(k)}$ that satisfies Equation (3.9).

Step 3: Iterate update. Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_{NEW}^{(k)}$, $k := k+1$ and go to Step 1.

A theorem for local convergence of the Newton method can be found in [98] Theorem 10.2.2 and [96] Theorem 3.5. In large-scale problems, the second derivative of the cost function can be complicated and hence, it can be highly computationally expensive to calculate the higher order terms (2.32). The Gauss-Newton method instead approximates the Hessian of the cost function by neglecting (2.32) and is outlined in the following section.

3.2.3 Gauss-Newton Method

The Gauss-Newton method can be seen as a compromise between the SD method and the Newton method as it uses an approximation of the second derivative of f by removing the nonlinear second-order terms. The Gauss-Newton search direction $\mathbf{s}^{(k)}$ is found by solving

$$\tilde{\nabla}^2 f(\mathbf{x}^{(k)})\mathbf{s}_{GN}^{(k)} = -\nabla f(\mathbf{x}^{(k)}), \tag{3.10}$$

where $\tilde{\nabla}^2 f(\mathbf{x}^{(k)})$ is the first-order approximation of (2.31) where (2.32) is neglected as given by

$$\tilde{\nabla}^2 f(\mathbf{x}^{(k)}) = \mathbf{J}(\mathbf{x}^{(k)})^{\mathbf{T}}\mathbf{J}(\mathbf{x}^{(k)}). \tag{3.11}$$

A summary of the GN method is given by the following.

Algorithm 3.2.3: GN algorithm applied to (2.33) **[33].**

Step 0: Initialisation. Given $\mathbf{x}^{(0)} \in \mathbb{R}^n$ and some stopping criteria. Set $k = 0$.

Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation. Compute a step $\mathbf{s}_{GN}^{(k)}$ that satisfies Equation (3.10).

Step 3: Iterate update. Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}_{GN}^{(k)}$, $k := k + 1$ and go to Step 1.

The linear system (3.10) is equivalent to solving the linear least-squares problem

$$\min_{\mathbf{s}_{GN}^{(k)}} \frac{1}{2}\left\|\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}_{GN}^{(k)} + \mathbf{r}(\mathbf{x}^{(k)})\right\|_2^2. \tag{3.12}$$

We note that the step calculation (3.10) uniquely defines $\mathbf{s}^{(k)}$, and $\mathbf{s}^{(k)}$ is a descent direction when $\mathbf{J}(\mathbf{x}^{(k)})$ is full column rank so that $\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{J}(\mathbf{x}^{(k)})$ is nonsingular. This ensures the GN step obtained by solving (3.10) is a descent direction [33]. Alternatively, to reduce the computational cost in high-dimensional problems and to solve the problem in real time, the series of problems (3.10) can be solved approximately as a series of linearised least-squares problems in the inner loop. The inner loop can be solved using iterative optimisation methods such as Conjugate Gradient (CG) where a limited number of CG iterations are allowed and an exact or approximate $\mathbf{J}$ is used [52].

A local convergence result can be found in Theorem 4 of [52] where GN is treated as an inexact Newton method. The theorem guarantees convergence of the GN method if for each iteration $k = 0, 1, \ldots$, the norm of the ratio of $\mathbf{Q}(\mathbf{x}^{(k)})$ and $\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{J}(\mathbf{x}^{(k)})$, the second and first terms of (2.59) respectively, is less than or equal to some constant $\hat{\eta}$ where $0 \leq \hat{\eta} \leq 1$. Another local convergence result for the GN method can be found in Theorem 10.2.1 of [33] where the performance of GN is shown to be dependent on whether or not the second-order terms (2.32) in (2.31) evaluated at the solution $\mathbf{x}^*$ are close to zero. This result gives information about the rate of convergence of GN and requires the following assumptions.

A1. *Let* $\mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m$ *and* $f(\mathbf{x}) = \frac{1}{2}\mathbf{r}(\mathbf{x})^T\mathbf{r}(\mathbf{x})$ *be twice continuously differentiable in an open convex set* $\mathbf{D} \subset \mathbb{R}^n$

A2. $\mathbf{J}(\mathbf{x})$ *is Lipschitz continuous with constant* L *and* $\|\mathbf{J}(\mathbf{x})\|_2 \leq \alpha$, $\forall \mathbf{x} \in \mathbf{D}$.

A3. *There exists a stationary point* $\mathbf{x}^* \in \mathbf{D}$

A4. *There exists* $\sigma \geq 0$ *for all* $\mathbf{x} \in \mathbf{D}$, *such that* $\mathbf{J}(\mathbf{x}^*)^T\mathbf{r}(\mathbf{x}^*) = 0$, *and*

$$\|(\mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{x}^*))^T\mathbf{r}(\mathbf{x}^*)\|_2 \leq \sigma\|\mathbf{x} - \mathbf{x}^*\|_2. \tag{3.13}$$

Theorem 3.2.1 (Local convergence of the Gauss-Newton method (see [33] Theorem 10.2.1))**.** *Let Assumptions A1, A2, A3 and A4 hold. Furthermore, let* $\lambda \geq 0$ *denote the smallest eigenvalue of* $\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})$. *If* $\sigma < \lambda$, *then for any* $c \in (1, \lambda/\sigma)$, *there exists a neighbourhood* $\mathcal{N}(\mathbf{x}^*, \varepsilon)$ *with* $\varepsilon > 0$ *such that for any* $\mathbf{x}^{(0)} \in \mathcal{N}(\mathbf{x}^*, \varepsilon)$, *the sequence generated by the Gauss-Newton method*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{J}(\mathbf{x}^{(k)}))^{-1}\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{r}(\mathbf{x}^{(k)}) \tag{3.14}$$

is well-defined, converges linearly to $\mathbf{x}^*$, *and obeys*

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_2 \leq \frac{c\sigma}{\lambda}\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 + \frac{c\alpha L}{2\lambda}\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2^2 \tag{3.15}$$

and

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_2 \leq \frac{c\sigma + \lambda}{2\lambda}\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 < \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2. \tag{3.16}$$

Proof. By induction as in [33] Theorem 10.2.1.

By the following corollary, [33] shows that if $\mathbf{r}(\mathbf{x}^*) = 0$, then the Gauss-Newton method will converge q-quadratically.

Corollary 3.2.2 (See [33] Corollary 10.2.2)**.** *Let the assumptions of Theorem 3.2.1 be satisfied. If* $\mathbf{r}(\mathbf{x}^*) = 0$, *then there exists* $\varepsilon > 0$ *such that for all* $\mathbf{x}^{(0)} \in \mathcal{N}(\mathbf{x}^*, \varepsilon)$, *the sequence generated by the Gauss-Newton method* $\{\mathbf{x}^{(k)}\}$ *is well defined and converges q-quadratically to* $\mathbf{x}^*$.

Proof. See [33] Corollary 10.2.2.

So far, we have outlined two local convergence proofs using the work of [52], [98] and [33]. We next highlight some of the advantages and disadvantages of the GN method.

Although the GN method benefits from local convergence properties, convergence can only be guaranteed if the initial guess $\mathbf{x}^{(0)}$ of the algorithm is in some neighbourhood around an (unknown) local solution $\mathbf{x}^*$, that is, convergence from an arbitrary initial guess is not guaranteed [33]. Even if the GN method does converge, it may not necessarily converge to the global minimum due to the fact that multiple local minima of a nonlinear least-squares objective function may exist.

In 4D-Var we seek to find the most probable estimate of the state $\mathbf{x}$ given the background information and the observations $\mathbf{y}$ by solving (2.33). Recall from Chapter 2, minimising the 4D-Var cost function is equivalent to maximising the posterior probability for the maximum a posteriori estimate. In the presence of local minima in the 4D-Var cost function, the posterior distribution is multimodal and the maximum a posteriori estimate is in fact the global minimiser of (2.52). This is what motivates our choice of finding an algorithm that best minimises the 4D-Var cost function in the computational cost available, while not relying on starting close to a local minimum.

In [33], the authors discuss the cases where Gauss-Newton may be slow to converge or where it may not converge at all using Theorem 3.2.1. In Table 10.2.3 of [33], the authors proceed to outline the advantages and disadvantages of the Gauss-Newton method. We discuss those we are concerned with in the following.

Recall from Section 2.2, the operators $\mathcal{H}_i$ and $\mathcal{M}_{0,i}$ can be linear or nonlinear in practice. In the linear case, the local minimum of the VarDA cost function (2.52) is in fact a global minimum and the GN method is equivalent to the Newton method and so obtains quadratic convergence to the global minimum if the initial starting point is in some neighbourhood around $\mathbf{x}_0^*$ [28].

Although GN has impressive local convergence properties for zero-residual problems and problems with small residuals that are reasonably linear (see [33] for a more detailed explanation of when GN performs well), GN may converge slowly in the case when the residual vector $\mathbf{r}$ at the solution $\mathbf{x}^*$ is non-zero [33]. Due to the presence of error in many problems in practice, including the variational problem, $\mathbf{r}(\mathbf{x}^*)$ is not necessarily zero nor close to zero and so the quadratic convergence rate shown in Corollary 3.2.2 (Corollary 10.2.2 of [33]) may not be achieved.

GN has no way of adjusting the length of the step $\mathbf{s}^{(k)}$ and hence, may take steps that are too long and fail to decrease the objective function value and thus to converge, see Example 10.2.5 in [33]. As GN only guarantees local convergence, we are interested in investigating methods that converge when $\mathbf{x}^{(0)}$ is far away from a local minimiser $\mathbf{x}^*$, as defined in Definition 3.1.13. Mathematical theory on global strategies can be found in [96] and [33].

There are three main strategies that safeguard GN and make it convergent from an arbitrary initial guess: line search, trust-region and regularisation [96]. GN with quadratic regularisation is similar to GN with trust-region (see Lemma 10.2. of [96]), also referred to as the Levenberg-Marquardt method (LM) [26]. Within our work, we focus on GN with quadratic regularisation (REG) and compare its performance to GN with backtracking Armijo line search (LS) and GN alone. The LS and REG methods will be presented in the next section.

3.3 Globally Convergent Methods

In this section, we will outline and discuss two globally convergent optimisation methods used for solving nonlinear least-squares problems of the form (2.33), namely, the Gauss-Newton method with bArmijo line search (LS) and the Gauss-Newton method with quadratic regularisation (REG). These two methods use a strategy within the GN framework to achieve convergence to a stationary point given an arbitrary initial guess by adjusting the GN step (3.10).

3.3.1 Gauss-Newton with line search

An optimisation method can be equipped with a line search strategy to find a stepsize $\alpha^{(k)} > 0$ from the current iterate $\mathbf{x}^{(k)} \in \mathbb{R}^n$, for a search direction $\mathbf{s}^{(k)} \in \mathbb{R}^n$, where $\mathbf{s}^{(k)}$ is set according to the optimisation algorithm to obtain a decrease in the function value. At each iteration k, a minimisation problem of the form

$$\min_{\alpha^{(k)}>0} f(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{s}^{(k)}) \tag{3.17}$$

is solved [96]. Note that when $\alpha^{(k)} = 1$ in (3.17) for all k, we have the standard GN method.

There are two steps in a line search algorithm:

1. Bracketing: find an interval that contains desirable stepsizes.

2. Bisection: compute a stepsize that lies within the interval found.

This algorithm can solve (3.17) exactly, although this can be computationally expensive. Therefore, inexact line search methods can be used that ensure (at minimal cost) that the step taken in the optimisation method is of the acceptable length, determined by the condition set by the user: neither too long (relative to the decrease in f) nor too short [96].

A popular inexact line search method is the backtracking-Armijo (bArmijo) algorithm, which improves the simple line search condition $f(\mathbf{x}^{(k)}+\alpha^{(k)}\mathbf{s}^{(k)}) \leq f(\mathbf{x}^{(k)})$ by including the Armijo condition [2]

$$f(\mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{s}^{(k)}) \leq f(\mathbf{x}^{(k)}) + \beta\alpha^{(k)}\nabla f(\mathbf{x}^{(k)})^T\mathbf{s}^{(k)}, \tag{3.18}$$

where the control parameter $\beta \in (0, 1)$[96]. The right-hand-side of the Armijo condition (3.18) is a function with a negative slope, by choosing $\beta \in (0, 1)$ ensures that this lies above the function on the left-hand-side when α is small and positive.

We can incorporate (3.18) by using the backtracking-Armijo (bArmijo) algorithm within the inner loop of GN to find an $\alpha > 0$ to restrict the step $\mathbf{s}^{(k)}$ in (3.10) so as to guarantee a decrease in the error for f. The Gauss-Newton with backtracking-Armijo line search (LS) method is presented as follows.

Algorithm 3.3.1: LS algorithm applied to (2.33) **[96].**

Step 0: Initialisation. Given $\mathbf{x}^{(0)} \in \mathbb{R}^n$, $\tau \in (0,1)$ and $\beta \in (0,1)$ and $\alpha_0 > 0$ and some stopping criteria. Set $k = 0$.

Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation.
Compute a step $\mathbf{s}^{(k)}$ that satisfies Equation (3.10) and set $\alpha^{(k)} = \alpha_0$.

Step 3: Check Armijo condition.
While the (Armijo) condition (3.18) is not satisfied, do:

Step 4: Shrink stepsize.
Set $\alpha^{(k)} := \tau\alpha^{(k)}$ and go to Step 3.

Step 5: Iterate update.
Set $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha^{(k)}\mathbf{s}^{(k)}$, $k := k + 1$ and go to Step 1.

The step equation in Algorithm 3.3.1 is the same as the GN step in Algorithm 3.3.1; thus when $\alpha^{(k)} = 1$, the GN and LS iterates coincide. Thus, the LS method will have the same local convergence properties as GN.

The Armijo condition essentially states that the reduction in our function f must be proportional to a fixed fraction (β) of the stepsize $\alpha^{(k)}$ and the directional derivative $\nabla f(\mathbf{x}^{(k)})^T\mathbf{s}^{(k)}$ [96]. This ensures we get a sufficient decrease in f proportional to the stepsize given that the following requirement is met,

$$\nabla f(\mathbf{x}^{(k)})^T\mathbf{s}^{(k)} < 0. \tag{3.19}$$

For Gauss-Newton (3.19) holds if $\nabla f(\mathbf{x}^{(k)}) \neq 0$ and $\mathbf{J}(\mathbf{x}^{(k)})$ has full column rank so that $\mathbf{s}^{(k)}$ is the descent direction. If these conditions are satisfied then (3.19) ensures that the GN iterates with bArmijo chosen stepsizes satisfy

$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)}), \quad \forall k. \tag{3.20}$$

That is, the use of condition (3.18) in the GN method ensures that the accepted steps produce a sequence of strictly decreasing function values.

Regarding convergence guarantees of Algorithm 3.3.1, it is important to note that, as discussed in Section 3.2 of [96], we cannot guarantee that a line search method locates a local minimum. The strongest guarantee we can obtain is that the method is attracted to stationary points. That is,

$$\lim_{k\to\infty} \|\nabla f(\mathbf{x}^{(k)})\| \to 0. \tag{3.21}$$

Convergence to a local minimum can only be guaranteed if information from the Hessian of the cost function (2.31) is included.

Despite its global convergence property (outlined later in Chapter 4), the LS method has some disadvantages. The use of the stepsize $\alpha^{(k)}$ may sometimes unnecessarily shorten the

step $\mathbf{s}^{(k)}$, slowing down the convergence. Figure 3.3 is a schematic from [96] and shows the acceptable choices of α where $l(\cdot)$ denotes the negative slope $\beta \nabla f(\mathbf{x}^{(k)})^T \mathbf{s}^{(k)}$. From this figure, one can see that the condition (3.18) is satisfied for all small values of α, which may allow the algorithm to take steps which are unacceptably short.

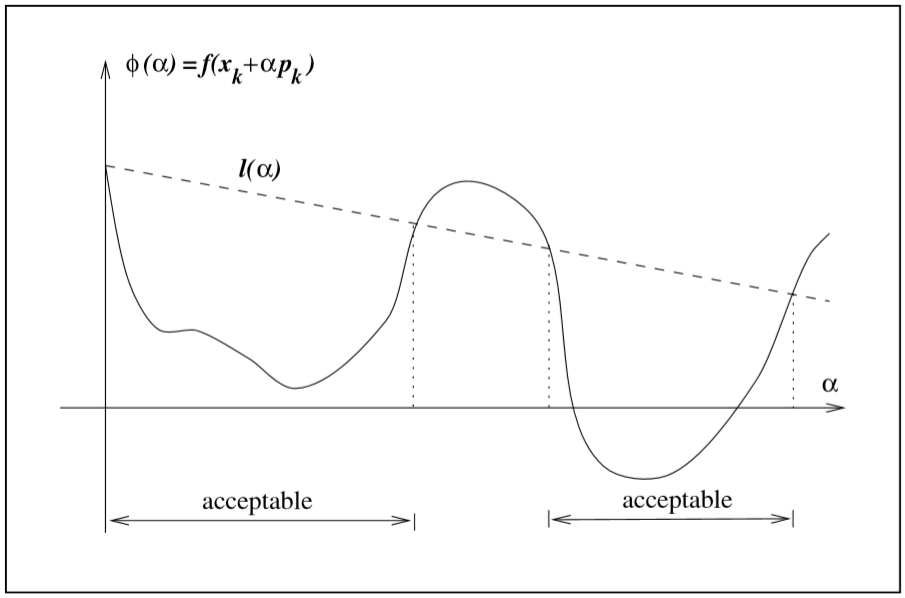


Figure 3.3: Schematic of choices of α that obtain sufficient decrease in f. Reprinted by permission from Springer Nature: Springer, Numerical Optimization by Jorge Nocedal and Stephen Wright, Springer Science+Business Media, LLC. (2006).

Furthermore, LS may be computationally costly in high dimensional problems due to the need to calculate the value of the function f each time $\alpha^{(k)}$ is adjusted, although more sophisticated updating strategies for α may be used to try to reduce this effect. As LS requires the re-evaluation of the outer loop objective function each time it adjusts its line search parameter, its applicability to real DA systems has been in doubt due to the constraint on the computational cost in VarDA [106]. This will be investigated further in the numerical experiments of Chapters 4 and 5.

Other line search strategies are possible such as Wolfe, Goldstein-Armijo and more [96]. For Wolfe line search, an additional curvature condition on α can be included to ensure that the slope of the function at the new iterate is greater than that at the previous iterate by a fixed fraction and is given by,

$$\nabla f(\mathbf{x}^{(k)} + \alpha^{(k)} \mathbf{s}^{(k)})^{\mathbf{T}} \mathbf{s}^{(k)} \geq c_2 \alpha^{(k)} \nabla f(\mathbf{x}^{(k)})^{\mathbf{T}} \mathbf{s}^{(k)}. \tag{3.22}$$

The Armijo condition is commonly used along with this curvature condition (3.22) to avoid unacceptably small steps. These two conditions form the Wolfe conditions (see [96] Section 3.1). However, re-evaluating the gradient to test the curvature condition at each line search iteration is costly and so we will focus on the use of the Armijo condition alone. It is sufficient to do so when using a backtracking algorithm to choose the stepsize using a factor τ

(see [96] Section 3.1). This ensures that the steps are not too short as large stepsize values are checked against the Armijo condition before smaller values.

We note here that Gauss-Newton with line search is often referred to as Gauss-Newton in some texts such as in [96]. We will now go on to describe a method, which benefits from global convergence properties, but does not require an additional inner loop as in the LS method.

A weakness of the GN method is that it fails when the Jacobian is (or is nearly) rank-deficient. Including a regularisation term, $\gamma^{(k)}\mathbf{s}^{(k)}$, in the original problem deals with this rank-deficiency. This is a common variation of the GN method known as the Levenberg-Marquardt method proposed in [70] and [84]. Moré [89] connected the Levenberg-Marquardt method with the trust-region framework, see also Lemma 10.2. of [96]. Details of its convergence in this framework can be found in Chapter 10 of [96]. In the following section, we will look at the Levenberg-Marquardt method as a GN method with a quadratic regularisation term.

3.3.2 Gauss-Newton with regularisation

The GN method may also be equipped with a regularisation term $\gamma^{(k)}\mathbf{s}^{(k)}$ in the step calculation (3.10) to achieve convergence to a stationary point given an arbitrary initial guess. The regularisation parameter $\gamma^{(k)} \in \mathbb{R}$ is adjusted within the given optimisation algorithm in order to find the search direction $\mathbf{s}^{(k)}$ that satisfies

$$\left(\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{J}(\mathbf{x}^{(k)}) + \gamma^{(k)}\mathbf{I}\right)\mathbf{s}^{(k)} = -\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{r}(\mathbf{x}^{(k)}). \tag{3.23}$$

This is equivalent to solving

$$\mathbf{s}^{(k)} = \arg\min_{\mathbf{s}} m^{(k)}(\mathbf{s}), \tag{3.24}$$

where

$$m^{(k)}(\mathbf{s}) = \frac{1}{2}\|\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s} + \mathbf{r}(\mathbf{x}^{(k)})\|_2^2 + \frac{1}{2}\gamma^{(k)}\|\mathbf{s}\|_2^2, \tag{3.25}$$

which can be seen as a second-order linear approximation of the nonlinear function $f(\mathbf{x}^{(k)})$. The initial regularisation parameter, $\gamma^{(0)}$, is usually set to 1 and is updated within each iteration of the algorithm according to the ratio

$$\rho^{(k)} = \frac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})}{m(0) - m(\mathbf{s}^{(k)})} = \frac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})}{f(\mathbf{x}^{(k)}) - m(\mathbf{s}^{(k)})}. \tag{3.26}$$

In Equation (3.26), $f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$ is the actual reduction; the difference between the value of the function at the current iterate and the new iterate and $m(0) - m(\mathbf{s}^{(k)})$ is the predicted reduction; the difference between the function at the current iterate and the model (see [96] Chapter 4).

The use of a regularisation term in this way ensures that the accepted steps produce a sequence of monotonically decreasing function values. That is, the GN with quadratic regularisation iterates satisfy

$$f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)}), \quad \forall k. \tag{3.27}$$

The effect of $\gamma^{(k)}$ is to implicitly control the length of the step $\mathbf{s}^{(k)}$. Increasing $\gamma^{(k)}$ shortens the steps, thus increasing the possibility that the procedure will decrease the objective function in the next iteration. The REG method is presented as follows.

Algorithm 3.3.2: REG algorithm applied to (2.33) **[89].**

Step 0: Initialisation. Given $\mathbf{x}^{(0)} \in \mathbb{R}^n$, $1 > \eta_2 \geq \eta_1 > 0$, $\gamma^{(0)} > 0$ and some stopping criteria. Set $k = 0$.

Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation. Compute a step $\mathbf{s}^{(k)}$ that satisfies Equation (3.23).

Step 3: Iterate update. Compute the ratio (3.26) and set

$$\mathbf{x}^{(k+1)} = \begin{cases} \mathbf{x}^{(k)} + \mathbf{s}^{(k)}, & \text{if } \rho^{(k)} \geq \eta_1 \\ \mathbf{x}^{(k)}, & \text{otherwise.} \end{cases} \tag{3.28}$$

Step 4: Regularisation parameters update. Set

$$\gamma^{(k+1)} = \begin{cases} \frac{1}{2}\gamma^{(k)}, & \text{if } \rho^{(k)} \geq \eta_2 \text{ (very successful iteration)} \\ \gamma^{(k)}, & \text{if } \eta_1 \leq \rho^{(k)} < \eta_2 \text{ (successful iteration)} \\ 2\gamma^{(k)}, & \text{otherwise, (unsuccessful iteration)} \end{cases} \tag{3.29}$$

Let $k := k + 1$ and go to Step 1.

We note that when $\gamma^{(k)} = 0$ in (3.23), the REG step in (3.23) is the same as the GN step in (3.10); thus the GN and REG iterates coincide at the iterate $\mathbf{x}^{(k)}$. By comparing (3.23) with (3.10), we are able to see how the REG step differs from the GN step. The diagonal entries of the Hessian of the objective function (3.11) are increased by the regularisation parameter $\gamma^{(k)}$ at each iteration of the REG method. The method is able to vary its step between a GN and a gradient descent step by adjusting $\gamma^{(k)}$ (see [96]) but may be costly due to the need to calculate the value of the function f on each iteration.

Step 4 of Algorithm 3.3.2 determines whether the use of the regularisation parameter $\gamma^{(k)}$ is successful at each iteration. Note that other choices of the factors $\frac{1}{2}$ and 2 for updating $\gamma^{(k)}$ in (3.29) are possible and even more sophisticated variants for choosing $\gamma^{(k)}$ have been proposed. In general, the choice of $\gamma^{(k)}$ is based on the following criteria [8].

- (very successful) - If the difference between the function, $f(\mathbf{x}^{(k)})$, and the approximate of the function, $m^{(k)}$, is close to the difference between the function, $f(\mathbf{x}^{(k)})$, and the function in the next iterate, $f(\mathbf{x}^{(k+1)})$, then the regularisation term has worked well and is reduced for the next iteration. This is equivalent to increasing the trust-region in a trust region algorithm (see [96] Chapter 4).
- (successful) - If the difference between the function, $f(\mathbf{x}^{(k)})$, and the approximate of the function, $m^{(k)}$, is fairly close to the difference between the function, $f(\mathbf{x}^{(k)})$, and

the function in the next iterate, $f(\mathbf{x}^{(k+1)})$, then the regularisation term has worked fairly well and is kept the same for the next iteration.

- (unsuccessful) - If the difference between the function, $f(\mathbf{x}^{(k)})$, and the approximate of the function, $m^{(k)}$, is not close to the difference between the function, $f(\mathbf{x}^{(k)})$, and the function in the next iterate, $f(\mathbf{x}^{(k+1)})$, then the regularisation term has not worked well and is increased for the next iteration. This is equivalent to decreasing the trust-region in a trust region algorithm (see [96] Chapter 4).

Algorithm 3.3.2 uses quadratic regularisation and adapts the regularisation parameter to safeguard the convergence of the GN method. Other adaptive regularisation safeguards exist, see [23], [54], [93] and [133].

Local convergence of the REG method, found in [33], (where $\mathbf{x}^{(0)} \in \mathcal{N}(\mathbf{x}^*, \varepsilon)$) is similar to that of the GN method, with the following additional assumption.

A5. *Let the sequence* $\{\gamma^{(k)}\}$ *where* $\gamma^{(k)} \in \mathbb{R}^+, \forall k$ *be bounded by* $b > 0$.

Theorem 3.3.1 (Local convergence of the REG method (see [33] Theorem 10.2.6))**.** *Let Assumptions A1, A2, A3, A4 and A5 hold. Furthermore, let* $\lambda \geq 0$ *denote the smallest eigenvalue of* $\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})$. *If* $\sigma < \lambda$, *then for any* $c \in (1, (\lambda + b)/(\sigma + b))$, *there exists a neighbourhood* $\mathcal{N}(\mathbf{x}^*, \varepsilon)$ *with* $\varepsilon > 0$ *such that for al* $\mathbf{x}^{(0)} \in \mathcal{N}(\mathbf{x}^*, \varepsilon)$, *the sequence generated by the REG method*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{J}(\mathbf{x}^{(k)}) + \gamma^{(k)}\mathbf{I})^{-1}\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{r}(\mathbf{x}^{(k)}) \tag{3.30}$$

is well-defined, converges to $\mathbf{x}^*$, *and satisfies*

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_2 \leq \frac{c(\sigma + b)}{(\lambda + b)}\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 + \frac{c\alpha L}{2(\lambda + b)}\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2^2 \tag{3.31}$$

and

$$\|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|_2 \leq \frac{c(\sigma + b) + (\lambda + b)}{2(\lambda + b)}\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 < \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2. \tag{3.32}$$

Furthermore, if $\mathbf{r}(\mathbf{x}^*) = 0$ *and* $\gamma^{(k)} = \mathcal{O}(\|\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{r}(\mathbf{x}^{(k)})\|_2)$, *then the sequence* $\{\mathbf{x}^{(k)}\}$ *converges q-quadratically to* $\mathbf{x}^*$.

Proof. As with Theorem 3.2.1.

If $\mathbf{J}^T\mathbf{J}$ term dominates $\mathbf{Q}$ term in the Hessian (2.31) near the solution $\mathbf{x}^*$, the REG method converges rapidly as regularisation is no longer required and the Gauss-Newton steps are taken (see [96] Section 10.3).

Given that the regularisation parameter is large enough, assumptions that ensure global convergence of the LM method can be found in [45] and [99]. Algorithm 3.3.2 also benefits from global convergence under certain conditions. We derive the proof of global convergence of REG later in Chapter 4.

In the following section, we review the relevant literature on numerical optimisation methods used to solve the variational problem.

3.4 Optimisation methods for VarDA

Recall from Chapter 1, in practice, the variational data assimilation (VarDA) problem is a nonlinear least-squares problem, which can be viewed as a large-scale unconstrained optimisation problem [68], usually solved using a variant of the classical GN method, known as the incremental method. In Chapter 2.2, we outlined the incremental formulation of the variational problem, including the inner loop problem (2.63) and outer loop problem (2.52). It is known that the accuracy with which the inner loop is solved affects the convergence of the outer loop [64, 65]. Within our work, we focus on the convergence of the outer loop and assume that the inner loop is solved exactly. Furthermore, we use a variable transformation commonly used in operational VarDA to precondition the variational problem.

We begin by outlining the incremental method, discussing the conditions under which it is equivalent to the GN method and the various simplifications and approximations made to solve the variational problem quickly and efficiently, focusing specifically on the use of a reduced resolution inner loop problem, which is relevant to Chapter 6. We then focus specifically on globally convergent methods, which are relevant to Chapters 4 and 5, discussing the various attempts made to safeguard the incremental method to ensure it is convergent.

3.4.1 Incremental VarDA

Recall that Algorithm 3.2.3 assumes that the full resolution step equation (3.10) is solved to calculate the GN step. In VarDA practice, the incremental method is used to solve the 4D-Var problem efficiently and in real-time, where the outer loop uses a high resolution model to obtain an estimate of the initial state and the inner loop is solved using a reduced resolution model. The inner loop (GN step calculation (3.10)) is run at a lower spatial resolution than the nonlinear model used to calculate the innovation vectors. This simplification is widely used in meteorological centres including ECMWF [63, 80, 105] and the Met Office [109].

In the incremental approach a series of linear least-squares problems (quadratic functions) are minimised to estimate the nonlinear least-squares objective function (2.52), where each of the sequence of linear least-squares problems are local quadratic approximations of the original NLLSP. The use of incremental 4D-Var algorithm to enable the implementation of 4D-Var in an operational setting was first proposed in [27] and is summarised in the following.

Algorithm 3.4.1: Incremental 4D-Var algorithm [27]

Step 0: Initialisation. Given the background state $\mathbf{x}^b \in \mathbb{R}^n$, some outer loop stopping criteria and some inner loop stopping criteria, set $k = 0$ and

$$\mathbf{x}_0^{(0)} = \mathbf{x}^b. \tag{3.33}$$

Step 1: Check outer loop stopping criteria. While the outer loop stopping criteria are not satisfied, do:

Step 2: Nonlinear model run. Run the nonlinear model $\mathcal{M}_{0,i}$ at high resolution to obtain $\mathbf{x}_i^{(0)}$ at each time point t_i.

Step 3: Innovation calculation. Calculate the innovation vectors at each observational time point using the high resolution nonlinear model run

$$\mathbf{d}_i^{(k)} = \mathbf{y}_i^{(k)} - \mathcal{H}_i(\mathbf{x}_i^{(k)}). \tag{3.34}$$

Step 4: Check inner loop stopping criteria. While the inner loop stopping criteria are not satisfied, do:

Step 5: Step calculation. Let

$$\delta\hat{\mathbf{x}}_0^{(k)} = \mathbf{S}_l(\mathbf{x}_0^{(k+1)} - \mathbf{x}_0^{(k)}), \tag{3.35}$$

be the increment in the reduced resolution and solve the inner loop problem

$$\hat{\mathbf{J}}(\mathbf{x}^{(k)})^T\hat{\mathbf{J}}(\mathbf{x}^{(k)})\delta\hat{\mathbf{x}}^{(k)} = -\hat{\mathbf{J}}(\mathbf{x}^{(k)})^T\hat{\mathbf{r}}(\mathbf{x}^{(k)}), \tag{3.36}$$

for $\delta\hat{\mathbf{x}}_0^{(k)}$.

Step 6: Update outer loop Set

$$\mathbf{x}_0^{(k+1)} = \mathbf{x}_0^{(k)} + \mathbf{S}_h\delta\hat{\mathbf{x}}_0^{(k)} \tag{3.37}$$

and $k := k + 1$, then go to Step 1.

The ECMWF Integrated Forecast System (IFS) is a practical implementation of Algorithm 3.4.1. It has been shown that additional outer loop iterations result in a better analysis [17]. This resulted in four outer loop iterations being used since the IFS update (46r1), instead of the previous three, as well as newer observations. This was made possible by starting the assimilation 10 minutes earlier through the introduction of a Continuous DA in Cycle 46R1 of the ECMWF IFS [34]. For future IFS update cycles, ECMWF are also looking into the possibility of running higher-resolution minimisations. However, for every outer loop iteration performed, many inner loop iterations are performed. Therefore, it is important that simplifications are made on the inner loop level.

Recall from Section 2.2.2, the nonlinear model used in the outer loop is linearised for use in the inner loop, causing the physical processes explained by the model to be simplified.

In practice, Algorithm 3.4.1 uses an approximate tangent linear model (TLM) to calculate the model trajectory. Although other linear models such as a perturbation forecast model (PFM) may be used and was investigated in [65]. The work of [65] compared the use of the TLM and the PFM to understand how the convergence behaviour differs when used in the inner loop of Algorithm 3.4.1. The TLM is derived from the nonlinear model equations using automatic differentiation whereas the PFM is derived by first linearising the continuous nonlinear equations and then discretising using a semi-implicit semi-Lagrangian scheme. They showed that if Algorithm 3.4.1 uses an exact TLM, the method is equivalent to the Gauss-Newton method (Algorithm 3.2.3). Furthermore, they showed that if an approximate linear model is used, or if a lower spatial resolution is used, Algorithm 3.4.1 is equivalent to an inexact GN method. They found that the effect on the quality of the analysis was greater if a reduced resolution inner loop and high resolution outer loop was used compared to when a high resolution inner and outer loop were used. The incremental 4D-Var with the TLM or PFM was converging to larger values of the nonlinear cost function and the difference between the two methods was greater compared to when the full resolution inner loop was used.

If the inner loop of Algorithm 3.4.1 is solved inexactly using an iterative solver, it is equivalent to the truncated Gauss-Newton (TGN) method [52]. In applications where the Jacobian is too computational expensive to compute (e.g. in NWP), a common simplification is to approximate the Jacobian of the residual vector for use in the inner loop, such as reducing the inner loop resolution. The convergence of Algorithm 3.4.1 with an approximate Jacobian is referred to as the perturbed Gauss-Newton (PGN) method [52]. When both an approximate Jacobian is used and the inner loop problem is solved inexactly Algorithm 3.4.1 is referred to as the truncated perturbed Gauss-Newton (TPGN) method in [52]. In the work of [52], they outline the conditions under which the TGN, PGN and TPGN converge to a stationary point. The work of [52] only looked at the use of general inexact Jacobians and made a brief investigation into the use of their proposed stopping criteria when a reduced resolution inner loop is used in [66] but the theoretical convergence was not studied.

As outlined in Section 2.2.2, the use of multiple resolutions in Algorithm 3.4.1 requires the use of restriction and prolongation operators. The technique used to map from outer loop space to inner loop space in operational DA, and in our work, is to apply a reduced resolution spatial operator to restrict the input of the inner loop method, and a spatial interpolation prolongation operator to map the output of the inner method (the increment) to the outer loop resolution. It is important to note that in practical implementations of VarDA, the restriction operator $\mathbf{S}_l$ does not necessarily only reduce the resolution, it may also be used to simplify multiple variables of the same variable type, such as temperature variables, into a single variable [109].

Practical implementations of Algorithm 3.4.1 increase the inner loop resolution as the outer loop iterates. This approach is called multi-incremental VarDA. Recall that the incremental approach uses a linearisation of the observation operator around the first guess trajectory, which uses a reduced resolution [27]. By reducing the model resolution, it makes it increasingly more difficult to resolve smaller scales. The main idea behind multi-incremental 4D-Var is to make use of the fast convergence on the large scales at a reduced resolution in

early iterations, with subsequent higher resolution iterations allowing the method to pick up smaller scales [124, 125]. Thus, the multi-incremental method can be used to resolve both small and large scales in DA by improving the large scales as the smaller scales are resolved [127].

Proposed by [127] and used operationally by ECMWF since January 2003 in Cycle 25r3, the multi-incremental procedure, also referred to as the multi-truncation incremental method, uses a series of progressively higher resolution inner loops for each outer loop update. Information from early reduced resolution iterations are used to precondition the problem in later iterations. The multi-truncation incremental method focuses on recovering the large scales in early iterations and strengthens these as the small scales are progressively introduced [127]. This gives a balance between two goals in VarDA, the need to represent small scales as well as the need for a fast minimisation procedure. In their work, [127] compared the use of multiple-truncations with the use of single truncations for both the incremental 4D-Var and the quasi-continuous data assimilation schemes, the latter of which uses observations as soon as they are available and spreads computing cost over the assimilation period [61].

In the work of [127], the authors outline a sufficient condition for the convergence of the incremental method where a reduced resolution inner loop is used. Their condition is derived by defining the incremental method as a fixed point iteration method of the form (3.3). Their sufficient condition is based on the convergence of a fixed point iteration method (see[35] Theorem 4.4.1) and is a consequence of the contraction mapping theorem (or the Banach fixed point theorem) for a unique fixed point. It guarantees that $\mathbf{G}$ in (3.3) has exactly one fixed point and that the sequence of iterates $\{\mathbf{x}^{(k)}\}$ converges to this fixed point.

In deriving the condition for the incremental method in [127], the authors first decompose $\mathbf{x}$ into

$$\mathbf{x} = \mathbf{x}_s + \mathbf{x}_s^{\perp}, \tag{3.38}$$

where $\mathbf{x}_s = \mathbf{S}_h\mathbf{S}_l\mathbf{x}$ and $\mathbf{x}_s^{\perp} = \{\mathbf{I}_n - \mathbf{S}_h\mathbf{S}_l\}\mathbf{x}$. They find that by assuming $\mathbf{S}_l\mathbf{S}_h = \mathbf{I}_r$ and applying $\mathbf{S}_h\mathbf{S}_l$ to (3.37), only $\mathbf{x}_s$ is updated in the incremental algorithm as shown in the following

$$\begin{aligned} \mathbf{S}_h\mathbf{S}_l\mathbf{x}^{(k+1)} &= \mathbf{S}_h\mathbf{S}_l\mathbf{x}^{(k)} + \mathbf{S}_h\mathbf{S}_l\mathbf{S}_h\delta\hat{\mathbf{x}}^{(k)} \\ &= \mathbf{x}_s^{(k)} + \mathbf{S}_h\delta\hat{\mathbf{x}}^{(k)} \end{aligned} \tag{3.39}$$

They subsequently find that for the incremental method, $\mathbf{G}(\mathbf{x}^{(k)})$ is given by

$$\begin{aligned} \mathbf{G}(\mathbf{x}^{(k)}) = \zeta^{(k)} + \left[\hat{\mathbf{B}}^{-1} + \hat{\mathbf{H}}^T\mathbf{R}^{-1}\hat{\mathbf{H}}\right]^{-1} \times \Big[\hat{\mathbf{B}}^{-1}\Big\{\mathbf{S}_l\mathbf{x}^b - \mathbf{S}_l(\mathbf{S}_h\zeta^{(k)} + \mathbf{x}_s^{\perp\,(k)}) \\ + \hat{\mathbf{H}}^T\mathbf{R}^{-1}\Big\{\mathbf{y} - \mathcal{H}(\mathbf{S}_h\zeta^{(k)} + \mathbf{x}_s^{\perp\,(k)})\Big\}\Big], \end{aligned} \tag{3.40}$$

where $\zeta = \mathbf{S}_l\mathbf{x}_s = \mathbf{S}_l\mathbf{x}$. Using the tangent linear hypothesis in reduced resolution space and by assuming $\mathbf{S}_l\mathbf{S}_h = \mathbf{I}_r$, the following convergence condition is obtained

$$\left\|\mathbf{I}_r - \left[\hat{\mathbf{B}}^{-1} + \hat{\mathbf{H}}^T\mathbf{R}^{-1}\hat{\mathbf{H}}\right]^{-1} \times \left[\hat{\mathbf{B}}^{-1} + \hat{\mathbf{H}}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{S}_h\right]\right\| \leq \alpha < 1, \tag{3.41}$$

where α is a constant, the size of which determines the speed of convergence of the incremental method to the unique fixed point. If $\mathbf{G}'(\ell) \neq 0$, where ℓ is the fixed point, the sequence of iterates converges to the fixed point linearly. If $g'(\ell) = 0$, the sequence of iterates converges to the fixed point at least quadratically.

The derivation of condition (3.41) for the incremental method requires the reduced resolution Hessian matrix to be positive definite. This assumption is satisfied given that $\hat{\mathbf{B}}^{-1}$ is positive definite.

Although the sufficient condition (3.41) can be used to prove there exists a solution to the variational problem, and that it is unique, it does not tell us much more than this. However, Equation (3.39) does show how the choice of the resolution used in the inner loop of the incremental method affects the update of the outer loop iteration by not modifying $\mathbf{x}_s^{\perp(k)}$. This is completely dependent on the restriction and prolongation operators chosen. The multi-incremental method approach was suggested to combat this issue while keeping computational cost low.

It is known that the accuracy with which the inner loop is solved affects the convergence of the outer loop [64, 65]. Therefore, in order to further improve the quality of the analysis, improvements to the inner loop methods must also be made. The work in [126] is key in motivating the investigation of improved inner loop minimisation algorithms that are faster than CG and unlike CG, do not overfit observations. Within our work, we focus on the convergence of the outer loop and the effect of the use of a reduced resolution inner loop on its convergence. The work of [52] found that approximately solving the linearised least-squares problem degrades the convergence rate of the GN method. Therefore, we assume that the inner loop is solved exactly so as to avoid the error associated with solving it inexactly.

In this section, we have outlined the incremental method and seen how the choice of restriction and extension operators play a role in its convergence as presented in [127]. In Chapter 6, we conduct our own theoretical investigation into the effect that the use of a reduced resolution inner loop has on the convergence on the incremental method.

3.4.2 Alternative methods

More recently, optimisation methods that do not require the use of first derivatives of the 4D-Var objective function are being investigated to avoid the development and maintenance costs associated with using the adjoint, see for example [51] and references therein. Furthermore, an extensive list of alternative assimilation schemes have been developed that do not require the use of the adjoint such as the ensemble-variational data assimilation method, 4DEnVar, developed by the Met Office for global NWP (see [3, 76]) and implemented at Environment Canada (see [19]). However, as the adjoint is already embedded in the operational infrastructure of many meteorological centres, its use is still often preferred over newer techniques as, unlike methods that require the use of an ensemble of non-linear forecast trajectories, such as the ensemble Kalman Filter (EnKF) and 4DEnVar, the adjoint is less

prone to causing sampling errors and does not require spatial and/or temporal localisation [6].

DA methods that are able to best deal with the use of more observations are becoming increasingly popular. For example, ECMWF are keen to use more observations through the use of more outer loop iterations in their IFS. The benefit of increasing the number of outer loop iterations has been studied in the work of Laroche et al. in [64]. They showed that for the incremental 4D-Var to perform reasonably well, it is necessary to allow a limited number of outer loop updates using the full-resolution model.

The use of longer assimilation time-windows in variational data assimilation allows us to use more observations over time. Fisher et al. (2005) [39] demonstrated the benefit of using long assimilation time-windows in 4D-Var to improve the analysis. In Fisher et al. (2011) [41], these results are confirmed for the ECMWF IFS - they show that a 24hr window is favourable over a 12hr window. However, when using a long assimilation time-window the linearity assumption for the tangent linear hypothesis becomes invalid and the incremental method may diverge. Furthermore, as the assimilation time-window length is increased, errors from sources other than the background and observations are introduced in to the system and are known as model errors. For this reason, the weak-constraint 4D-Var formulation, which accounts for model error has been favoured over the strong-constraint formulation for its ability to yield a more accurate analysis in long time-window assimilation [94]. To help deal with the issues that come with the use of a long time-window, Pires et al. [101] outlined the quasi-static variational assimilation (QSVA) method. The QSVA method is not a minimisation method in it self. It performs a series of minimisations using QN or CG over a gradually extended assimilation time-window length. This was further developed in [122] and [121] and applied in [37] and [48]. To enable the use of more observations in operational NWP, the ECMWF have built on the QSVA literature and integrated the use of a long assimilation time-down into their IFS [69]

In the following section, we review globally convergent methods used to solve the variational problem.

3.4.3 Globally convergent methods in VarDA

Recall from Section 3.4.1 that the incremental 4D-Var method has been shown to be equivalent to the Gauss-Newton optimisation method (GN) under certain conditions [65]. The GN method does not require the use of high order second derivatives, thus alleviating the added complexity of calculating and storing them. A drawback of the GN method, and Newton-based methods, is that they do not guarantee convergence to an estimate of the initial state given poor initialisation [33]. In NWP, the initial guess for the minimisation is generally chosen to be the predicted initial state from a previous forecast, known as the background state. However, for some applications of VarDA this choice may not be a good enough estimate of the true initial state. Therefore, GN may fail to converge. This is what motivates our focus on the investigation into the use of globally convergent methods for VarDA.

A line search is used in practical implementations of the incremental VarDA problem. Earlier

work in [106] showed that the use of a line search strategy improved the minimisation of the 4D-Var objective function. The M1QN3 method, developed in [44] and used operationally at ECMWF [126], Météo France [74] and the Meteorological Service of Canada [20], uses the Wolfe line search conditions [134] to safeguard the method. However, as discussed in Section 3.3.1, the Wolfe conditions require the use of additional evaluations of the objective function and, to rule out unacceptably short steps, its gradient [96]. This is unlike the Armijo condition [2] used in Algorithm 3.3.1 of our work, as in [50], which through the use of backtracking only requires additional evaluations of the objective function and not of the gradient [96]. We pair GN with backtracking Armijo line search and use a fixed amount of computational operations to guarantee a reduction in the outer loop objective function (assuming the inner loop is solved to a high accuracy), while considering the computational limits present in DA.

The use of the LM method has been of interest in the DA community because of its similarities with GN and its convergence guarantees. The use of the ensemble Kalman filter and ensemble Kalman smoother methods (EnKF and EnKS, respectively) as linear least-squares solvers for the inner loop problem has been proposed [82, 136]. This is the approach used in the literature when using LM in DA.

Bergou et al. [9] applied a variation of the LM method to the 4D-Var problem combined with the use of ensemble methods for the linearised subproblems where they focus on the case where only approximate gradient and Jacobian values are available and accurate within a certain probability. They provide a framework for using EnKS for solving the subproblem inexactly in their LM method and proved global convergence under an assumption of the probability of the accuracy of the gradient. In the work of Bocquet et al. [13], they applied a variation of LM using an EnKF to regularise the subproblem and obtain a faster convergence rate versus GN.

Mandel et al [83] apply the incremental 4D-Var method to the weak-constraint (where model error is accounted for) 4D-Var problem with the two-level quasi-geostrophic model, using the exact tangent and adjoint models. They find that the method diverges due to the nonlinearity of the model along a 10 day (long) time-window. They contrast the results of the incremental 4D-Var method with those obtained when they use a LM method to control the convergence of the incremental 4D-Var method, using the EnKS as the inexact solver for the 4D-Var inner-problem. They refer to this as the Ensemble Kalman smoother 4D-Var (EnKS-4DVAR) with regularisation. They assess the performance of the EnKS-4DVAR with regularisation method when varying the regularisation parameter γ. The regularisation parameter is fixed throughout the iterations, unlike Algorithm 3.3.2 in our work, which uses an adaptive method. They find that convergence is not guaranteed when the regularisation parameter is small, such as when $\gamma \leq 1$. Large values such as $\gamma \geq 10$ enable the function value to decrease as the method iterates. They conclude that investigations into an adaptive method to adjust the regularisation parameter at each iteration would be of interest.

More recently, the authors of [10] proposed a novel LM method for application to both zero and non-zero residual problems. This method is similar to Algorithm 3.3.2, except for the

use of an additional parameter that corresponds to a successful step of the method to balance local and global convergence requirements. The authors used an example to show how their local convergence is satisfied by the standard 3D-Var problem and assessed the performance of their proposed LM method using preliminary numerical experiments.

In Chapters 4 and 5, we aim to investigate whether the use of globally convergent optimisation methods, namely Algorithms 3.3.1 and 3.3.2, is beneficial in VarDA, where there is limited time and computational cost available. Such methods use safeguards to guarantee convergence to the analysis from an arbitrary background state vector by ensuring monotonic/strict and sufficient decrease of the error in the objective function.

The results outlined within this section only complement and do not replicate our research into the use of globally convergent strategies where we focus on the convergence of the 4D-Var problem on the outer loop level, where the exact gradient is used (as is the case when an adjoint is available), the inner loop is assumed to be solved to a high accuracy and the regularisation parameter in REG is updated using a simple, inexpensive strategy.

In the following section, we outline some considerations made when analysing and comparing algorithms.

3.5 Algorithmic considerations

Due to the limited time and computational cost available in VarDA practice, the incremental method is not necessarily run to convergence and a stopping criterion is used to limit the number of iterations. Each residual vector calculation requires the non-linear model to be run forward to obtain the state at each observation location in time. This can then be used to calculate the value of the objective function. Furthermore, one run of the adjoint model is required to calculate the gradient.

Due to storage limitations that exist in computers, precision is often sacrificed. As a consequence, the function may not be able to be evaluated at each iterate accurately. These issues in computer precision may cause difficulties when analysing the algorithmic output. For example, we may see that the error in the iterates is reducing as an algorithm iterates, while the function value is stagnant. Therefore, it is important to choose reasonable tolerances for the stopping criteria to ensure that iterations are not performed outside of computer precision.

In the following section, we outline the stopping criteria commonly used in the literature and within our work to terminate optimisation methods.

3.5.1 Stopping criteria

Stopping criteria are user specified criteria used within an optimisation algorithm to ensure that the algorithms do not iterate beyond when the criteria are specified. The criteria can be

used to terminate the algorithm when a reasonable approximation to the solution is found, when a specified time has passed and/or when storage limits have been met. Furthermore, stopping criteria can be used to terminate the optimisation algorithm even if a suitable solution has not been found, or if no suitable solutions exist, so as to avoid unnecessary cycling of the iterations.

One way to check whether a stationary point of a function is located is to check whether the gradient norm is close to zero. Using the following criterion, the user can specify an ϵ close enough to zero such that they can be confident that the optimisation method has located a stationary point

$$||\nabla f(\mathbf{x}^{(k)})||_2 \leq \epsilon. \tag{3.42}$$

A typical choice of ϵ would be 10^{-5}, as 10^{-6} or below would be below machine precision.

A relative form of (3.42) is one of the most popular measures used in the VarDA inner loop and is given by

$$\frac{||\nabla f(\mathbf{x}^{(k)})||_2}{||\nabla f(\mathbf{x}^{(0)})||_2} \leq \epsilon. \tag{3.43}$$

This measure compares the size of the gradient norm at the current iterate with the gradient norm at the initial guess. This comparison gauges how much progress the algorithm has made in locating a stationary point of the function. The work in [66] proposed a relative gradient stopping criterion of the form (3.43) for use in the VarDA inner loop, where f is the inner loop cost function (2.63). They showed that this is an appropriate choice when a lower spatial resolution is used in the inner loop.

As there is limited computational cost available in practice in DA, it is vital that both the inner and outer loop iterations are stopped when little progress is made. The following criterion can be used within an optimisation method to terminate the iterations evaluations once a maximum number of iterations, $k_{\max} \in \mathbb{Z}$, has been achieved,

$$k < k_{\max}. \tag{3.44}$$

Within our work, the limit on the total number of function and Jacobian evaluations is achieved by using the following criterion

$$k_J + l \leq \tau_e, \tag{3.45}$$

where k_J is the total number of Jacobian evaluations (which is equivalent to the number of outer iterations k in VarDA), l is the total number of function evaluations and τ_e is the tolerance. The tolerance τ_e can be chosen according to the maximum number of evaluations desired. We note that for GN, $k_J = l$ as the method requires as many Jacobian evaluations as function evaluations. However, for both LS and REG there could be more than one function evaluation per Jacobian evaluation since for unsuccessful steps, the Jacobian is not updated so $k_J \leq l$.

Furthermore, it is important to stop the iterations when little progress is made on either the function and iterate level. The following measure can be used to stop a method when the norm of the error in the function value between iterations is ϵ,

$$|f(\mathbf{x}^{(k-1)}) - f(\mathbf{x}^{(k)})| \leq \epsilon. \tag{3.46}$$

The norm of the error in the iterates is another useful measure and is given by

$$||\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}||_2 \leq \sqrt{\epsilon}. \tag{3.47}$$

The criterion (3.47) is useful in applications, such as VarDA, where the accuracy of the iterate, and not just the minimisation of the function, is important.

In well-behaved problems, Gill [46] has shown that the relationship between the variable $\mathbf{x}^{(k)}$ and the function $f(\mathbf{x}^{(k)})$ is well-defined at each iteration k. On the other hand, if the problem is ill-conditioned then although $f(\mathbf{x}^{(k)})$ may be close to $f(\mathbf{x}^*)$, $\mathbf{x}^{(k)}$ may not necessarily be close to $\mathbf{x}^*$. Gill [46] therefore suggested that a tolerance ϵ, which is related to the level of accuracy the user aims to obtain, must be set along with three stopping criteria for the outer loop of an optimisation algorithm. One of these criteria is the relative change in the function value, given by

$$\frac{|f(\mathbf{x}^{(k-1)}) - f(\mathbf{x}^{(k)})|}{1 + |f(\mathbf{x}^{(k)})|} \leq \epsilon \tag{3.48}$$

and is used within our work.

The tolerance should also be chosen carefully. It must be small enough so as to ensure that the iteration process converges, yet large enough so not to fit observational noise. The need for the accuracy of the inner loop is dependent on the nonlinearity of the problem [66]. Nonlinearities occur as more physics are included in the VarDA system, which is what DA is moving towards. The more nonlinear the problem is, the more accurately the inner problem must be solved in order to guarantee convergence of the outer loops, and so a smaller tolerance is required [52]. If the problem is very nonlinear, we cannot expect a linear approximation to be accurate [107]. Furthermore, if the size of the residual at the solution is large, the inner loop problem must be solved more accurately.

In the following section, we outline the considerations we must take when comparing the performance of optimisation algorithms.

3.5.2 Performance comparisons

It is important to note that a good optimisation algorithm often holds a balance between the following properties, which often conflict, as outlined in [96].

- robustness: given a reasonable initial value, they perform well for any problem within a given class.
- efficiency: the computational cost, time and storage requirements are kept to a minimum.

- accuracy: the ability to precisely identify a solution and not be sensitive to computational rounding errors or errors within the data.

When analysing the performance of an optimisation algorithm in our work, we look for the reduction in

- the error in the function value $||f(\mathbf{x}^{(k-1)}) - f(\mathbf{x}^{(k)})||_2$,
- the 2-norm of the gradient value $\|\nabla f(\mathbf{x}^{(k)})\|_2$
- and the error in the iterates $||\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}||_2$

at each iteration k. We are also interested in knowing which method obtains the most accurate algorithmic output in a fixed number of function and Jacobian evaluations, i.e. within a fixed computational cost, how far is the estimate obtained by a given method from the true solution to the minimisation problem?

In order to best present our results, we use accuracy profiling described as follows.

An *accuracy profile* shows the proportion of problems a given method can solve within a fixed amount of work (τ_e) and a given tolerance (τ_f) of the change in the function value [90]. To ensure the robustness of our results, we apply the three optimisation methods to a series of n_r randomly generated problems, where the randomness occurs through the background and observation error vectors, $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$. For each realisation, a new $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$ are generated from their respective distributions outlined in Chapter 4. The following criterion proposed in [90] is used to flag that an estimate of the initial state has been obtained by an optimisation method

$$\frac{f(\mathbf{x}_0^{(l)}) - f(\mathbf{x}_0^t)}{f(\mathbf{x}_0^{(0)}) - f(\mathbf{x}_0^t)} \leq \tau_f, \tag{3.49}$$

where $\mathbf{x}_0^t$ is a solution of (2.52) referred to as the 'truth' and τ_f is the tolerance. The measure (3.49) compares the optimality gap $f(\mathbf{x}_0^{(l)}) - f(\mathbf{x}_0^t)$ relative to the best reduction $f(\mathbf{x}_0^{(0)}) - f(\mathbf{x}_0^t)$ [90]. This ensures that the variational problem is only flagged as solved by the optimisation method once the value of the objective function is within some error (τ_f) of the truth.

For our problems, the truth is unknown. We only know that, due to the nonlinearity of the variational problem, there may exist many values of $\mathbf{x}_0$ that could minimise (2.52) locally. We are interested in the estimate $\mathbf{x}_0^t$ that gives the greatest reduction in (2.52) that any of the three methods can obtain. Therefore, we set the truth to be the $\mathbf{x}_0^{(l)}$ obtained by any of the three methods that gives the smallest function value within the given number of evaluations. Using this criterion allows us to benchmark the methods against each other using accuracy profiles.

For each experiment, we plot the proportion of the n_r realisations solved by each method against the relative accuracy obtained, τ_f. The relative accuracy obtained is varied using

$\tau_f = 10^{-i}$, where $i \in [0, 5]$.

We acknowledge in this work that the code for the accuracy profiles has been adapted from the code for the data profiles used in [90].

3.6 Conclusion

Within this section we have outlined the theory of nonlinear least-squares problems and outlined and discussed three fundamental unconstrained optimisation methods for nonlinear least-squares problems, namely, SD, Newton's method and GN. We outlined two local convergence results for the GN method before discussing two safeguards used to make GN globally convergent, namely, LS and REG.

We then reviewed relevant optimisation methods that have been applied to the variational problem and discussed what is sought from an optimisation method in VarDA. Finally, we outlined various stopping criteria used to terminate optimisation methods and discussed how to compare the performance of optimisation methods.

The global convergence proofs for optimisation methods equipped with line search and regularisation strategies do exist in the literature. However, they do not exist in the literature for the specific algorithms, Algorithms 3.3.1 and 3.3.2, that we use. Therefore, in the following chapter, we derive the global convergence for LS and REG. We then present our results when comparing the performance of the GN, LS and REG when applied to the 3D-Var problem.

Chapter 4

Globally Convergent methods for 3D-Var

Recall from Section 3.2.3 that the Gauss-Newton method (GN) can only guarantee local convergence under certain conditions and not necessarily global convergence. This is dependent on how close the initial guess is from the local minimum the algorithm locates and whether or not the residual vector $\mathbf{r}$ of (2.28) is a zero vector at a solution $\mathbf{x}^*$. Furthermore, the region of local convergence depends on problem constants not known a priori, such as Lipschitz constants of the gradient.

It is important to note here that the globally convergent methods we are concerned with, namely Gauss-Newton with line search (LS) and Gauss-Newton with regularisation (REG), can only guarantee global convergence to a local minimum under certain conditions and not necessarily to a global minimum. Both LS and REG do not require the use of higher order second derivative information to achieve global convergence guarantees (such as Newton's method), nor do they require time and computationally expensive parameter updating techniques (such as with the Woolfe conditions). Therefore, LS and REG are considered relatively inexpensive methods in the class of globally convergent methods, a classification that makes them key contenders for use in practical VarDA.

In this chapter, we prove global convergence for the Algorithms 3.3.1 and 3.3.2 when applied to nonlinear least-squares optimisation problems of the form (2.33) and discuss whether the assumptions made hold in DA. We then present our findings when applying the GN, LS and REG methods to the 3D-Var problem. In particular, we address research questions RQ1(a), RQ1(b) and partially address RQ1(c) using the 3D-Var problem.

In Section 4.1, we list the results and assumptions required to prove global convergence of LS and REG when applied to nonlinear least-squares problems (NLLSPs). We then discuss whether these are valid assumptions in VarDA. In Sections 4.2 and 4.3, we outline the theorems and proofs of global convergence of LS and REG. In Section 4.4, we begin by applying GN, LS and REG to some general NLLSPs and discuss the results in relation to the variational problem. We then apply GN, LS and REG to both linear and nonlinear 3D-Var problems with the aim of understanding how the use of safeguarding strategies within GN

affects the convergence of the outer loop in VarDA. In Section 4.5 we derive some theory to explain how the REG parameter interacts with the variational problem and test our findings by applying REG, along with GN and LS for comparison, to the 3D-Var problem outlined in Section 2.2. In Section 4.6 we set out the experimental design for the 3D-Var numerical experiments where we apply GN, LS and REG to the standard 3D-Var problem with nonlinear observation operators. In Section 4.7 we apply GN, LS and REG to two nonlinear non-zero residual standard 3D-Var problems and consider the effects of initialising GN, LS and REG with the background, where the amount of uncertainty in the background information is increased whilst the amount of uncertainty in the observations is fixed. We also consider the effects of using an alternative choice of initial REG parameter $\gamma^{(0)}$, derived earlier in Section 4.5, on the convergence rate of REG. We use accuracy profiles to show how the methods perform within a limited number of cost function and Jacobian evaluations (as in practice) and when more evaluations are allowed than in practice. We also study the effect on the accuracy of the analysis for each of the methods through the use of RMSE profiles. Finally, we conclude this chapter in Section 4.8.

4.1 Assumptions & VarDA

Before we outline the global convergence proofs for the LS and REG methods, we state some theoretical results along with the assumptions required. The following theorem is used within the global convergence proofs of the LS and REG method and shows that the Jacobian (2.29) is uniformly bounded above if the residual vector $\mathbf{r}$ is Lipschitz continuous.

Theorem 4.1.1. *[Uniformly bounded Jacobian (see [92] Lemma 1.2.2)] Let $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ be Lipschitz continuous with Lipschitz constant $L_r > 0$. Then $\mathbf{J}$, the Jacobian of $\mathbf{r}$ is uniformly bounded above, that is, for all $\mathbf{x} \in \mathbb{R}^n$,*

$$\|\mathbf{J}(\mathbf{x})\| \leq L_r. \tag{4.1}$$

Proof. See [92] Lemma 1.2.2.

The following lemma will be necessary to show that the gradient (2.30) is Lipschitz continuous given that $\mathbf{r}$ and $\mathbf{J}$ are Lipschitz continuous and that $\mathbf{r}$ is bounded above.

Lemma 4.1.2 (Lipschitz continuity of a least-squares gradient)**.** *Suppose we have a function* (2.28) *and its gradient* (2.30) *where $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ and $\mathbf{J}$ is the Jacobian of $\mathbf{r}$. Assume that $\mathbf{r}$ is uniformly bounded above by $\omega > 0$ such that for all $\mathbf{x}$ in some set $\mathcal{N} \subset \mathbb{R}^n$, we have $\|\mathbf{r}(\mathbf{x})\| \leq \omega$. Furthermore, assume $\mathbf{r}$ and $\mathbf{J}$ are Lipschitz continuous on $\mathcal{N}$ with Lipschitz constants $L_r > 0$ and $L_J > 0$, respectively. Then the gradient* (2.30) *is Lipschitz continuous with Lipschitz constant $L > 0$.*

Proof. As $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ is Lipschitz continuous with Lipschitz constant $L_r > 0$, then $\mathbf{J}$ is bounded above (see Theorem 4.1.1). That is,

$$\|\mathbf{J}^T(\mathbf{x})\| = \|\mathbf{J}(\mathbf{x})\| \leq L_r \qquad \forall \mathbf{x} \in \mathbb{R}^n. \tag{4.2}$$

From our assumptions we know that $\mathbf{r}$ and $\mathbf{J}$ are Lipschitz continuous. For all $\mathbf{x}, \mathbf{y} \in \mathcal{N}$, we have

$$\begin{aligned}\|(\mathbf{J}(\mathbf{x}))^T\mathbf{r}(\mathbf{x}) - (\mathbf{J}(\mathbf{y}))^T\mathbf{r}(\mathbf{y})\| = \|(\mathbf{J}(\mathbf{x}))^T\mathbf{r}(\mathbf{x}) - (\mathbf{J}(\mathbf{x}))^T\mathbf{r}(\mathbf{y}) \\ + (\mathbf{J}(\mathbf{x}))^T\mathbf{r}(\mathbf{y}) - (\mathbf{J}(\mathbf{y}))^T\mathbf{r}(\mathbf{y})\|.\end{aligned} \tag{4.3}$$

Using the triangle inequality (2.6), we obtain the following

$$\|(\mathbf{J}(\mathbf{x}))^T\mathbf{r}(\mathbf{x}) - (\mathbf{J}(\mathbf{y}))^T\mathbf{r}(\mathbf{y})\| \leq \|(\mathbf{J}(\mathbf{x}))^T(\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y}))\| + \|(\mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{y}))^T\mathbf{r}(\mathbf{y})\| \tag{4.4}$$

Using the submultiplicative property (2.12), we obtain

$$\|(\mathbf{J}(\mathbf{x}))^T\mathbf{r}(\mathbf{x}) - (\mathbf{J}(\mathbf{y}))^T\mathbf{r}(\mathbf{y})\| \leq \|\mathbf{J}(\mathbf{x})\|\|(\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y}))\| + \|\mathbf{r}(\mathbf{y})\|\|\mathbf{J}(\mathbf{x}) - \mathbf{J}(\mathbf{y})\|. \tag{4.5}$$

Using $\|\mathbf{r}(\mathbf{x})\| \leq \omega$ and (4.2), and setting $L = (L_r^2 + \omega L_J)$ we obtain

$$\|(\mathbf{J}(\mathbf{x}))^T\mathbf{r}(\mathbf{x}) - (\mathbf{J}(\mathbf{y}))^T\mathbf{r}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \tag{4.6}$$

so the gradient ∇f is Lipschitz continuous with Lipschitz constant $L > 0$, as required. □

In order to obtain a lower bound on $\alpha^{(k)}$ for the LS method and an upper bound on $\gamma^{(k)}$ for the REG method, we will require the use of the following auxiliary lemma, a property of Lipschitz continuous functions.

Lemma 4.1.3 (Descent lemma (see [11] Proposition A.24))**.** *Let $f \in \mathcal{C}^1(\mathbb{R}^n)$ with ∇f Lipschitz continuous with Lipschitz constant $L > 0$. Then for all $\mathbf{y} \in \mathbb{R}^n$ we have,*

$$f(\mathbf{x}^{(k)} + \mathbf{y}) \leq f(\mathbf{x}^{(k)}) + (\nabla f(\mathbf{x}^{(k)}))^T\mathbf{y} + \frac{1}{2}L\|\mathbf{y}\|^2. \tag{4.7}$$

Proof. See [11] Proposition A.24.

The following assumptions are used to prove global convergence of both the LS and REG methods.

A6. $\mathbf{r}$ *is uniformly bounded above by $\omega > 0$ such that $\|\mathbf{r}(\mathbf{x})\| \leq \omega$.*

A7. $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *is Lipschitz continuous on $\mathbb{R}^n$ with Lipschitz constant $L_r > 0$.*

A8. $\mathbf{J}$ *is Lipschitz continuous on $\mathbb{R}^n$ with Lipschitz constant $L_J > 0$.*

We remark that for the LS method, we can weaken assumptions A7 and A8 by replacing $\mathbb{R}^n$ with the open set $\mathcal{N}$ containing the level set

$$\mathcal{L} = \left\{\mathbf{x} \in \mathbb{R}^n | f(\mathbf{x}) \leq f(\mathbf{x}^{(0)})\right\}, \tag{4.8}$$

where f is a NLLS function of the form (2.28).

In order to achieve the sufficient decrease property of the LS method, the following assumption must be made.

A9. $\mathbf{J}(\mathbf{x})$ *in* (2.53) *is uniformly full rank for all* $\mathbf{x} \in \mathbb{R}^n$*, that is, the singular values of* $\mathbf{J}(\mathbf{x})$ *are uniformly bounded away from zero, so there exists a constant* ν *such that* $\|\mathbf{J}(\mathbf{x})\mathbf{z}\| \geq \nu\|\mathbf{z}\|$ *for all* $\mathbf{x}$ *in a neighbourhood* $\mathcal{N}$ *of the level set* $\mathcal{L}$ *where* $\mathbf{z} \in \mathbb{R}^n$*.*

In VarDA practice, it is reasonable to assume that the physical quantities are bounded. Therefore, we can say that both $\mathbf{x}_0 - \mathbf{x}^b$ and the innovation vector $\mathbf{y} - \mathcal{H}(\mathbf{x})$ in (2.52) are bounded in practice, thus satisfying assumption A6. In VarDA, we must assume that the nonlinear model $\mathcal{M}_{0,i}$ in (2.52) is Lipschitz continuous in order for A7 to hold. As discussed in [87], this is a common assumption made in the meteorological applications. However, we cannot say that this is necessarily the case in VarDA practice.

In order for the Jacobian $\mathbf{J}$ in (2.53) to be Lipschitz continuous, we require its derivative to be bounded above by its Lipschitz constant. Therefore, for assumption A8 to hold, we require $\mathbf{r}$ in (2.53) to be twice continuously differentiable in practice, which is a common assumption made in VarDA, and also, that these derivatives of $\mathbf{r}$ are bounded above.

As mentioned in Section 2.3, the preconditioned 4D-Var Hessian (2.70) is full rank by construction as it consists of the identity matrix and a non-negative definite term. Therefore, the Jacobian of the residual of the preconditioned problem is full rank and assumption A9 holds. This is also the case for the standard 4D-Var problem (2.52), because of the presence of $\mathbf{B}^{-1/2}$ in its Jacobian.

In the following section, we prove global convergence of the LS method, using these assumptions.

4.2 Global convergence of LS

We recall that the Gauss-Newton with bArmijo line search method minimises the least-squares function (2.28) by solving (3.10) for the new descent direction and chooses a stepsize $\alpha^{(k)}$ within each iteration k of Algorithm 3.3.1.

We want to prove global convergence for the LS method to a stationary point, that is that $\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{r}(\mathbf{x}^{(k)}) \to 0$ as $k \to \infty$ for $k = 0, 1, 2, \ldots$ where the sequence $\{\mathbf{x}^{(k)}\}$ is generated by the GN method with initial guess $\mathbf{x}^{(0)}$ and stepsizes $\alpha^{(k)}$ satisfying the Armijo condition (3.18).

Nocedal et al. outline the proof for the GN method with Wolfe line search conditions in [96], which uses the Zoutendijk condition. Alterations to this proof can be made to prove the global convergence theorem of the LS method, Algorithm 3.3.1. In this section, we present this result. We begin the proof by showing that the bArmijo chosen stepsizes $\alpha^{(k)}$ are bounded below using assumptions A6 - A8. Using this lower bound, as well as assumption A9, we are able to prove the Zoutendijk condition (as in [96]) and its variant hold. Both the Zoutendijk condition and its variant use the angle between the GN search direction and the steepest descent direction. If this angle is uniformly bounded away from zero with k,

one can show that GN with line search is a globally convergent method.

We begin by establishing a lower bound on the stepsize $\alpha^{(k)}$ for all k.

We require the following lemmas found in [22] to deduce bounds on the bArmijo chosen stepsizes $\alpha^{(k)}$.

Lemma 4.2.1 (Armijo stepsize interval (see [22] Lemma 2))**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume that A6, A7 and A8 hold. Furthermore, let* $\mathbf{x}^{(k)}$ *and* $\mathbf{s}^{(k)}$ *be the Gauss-Newton iterate and step respectively for all* $k \geq 0$*. Then the Armijo condition* (3.18) *is satisfied for all* $\alpha \in \left[0, \alpha_{max}^{(k)}\right]$*, where* $\alpha_{max}^{(k)} = \frac{(\beta-1)(\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)}}{L\|\mathbf{s}^{(k)}\|^2}$*,* $\beta \in (0,1)$ *and* L *is the Lipschitz constant of* ∇f*.*

Proof. (see [22] Lemma 2). We first note that as the assumptions of Lemma 4.1.2 are satisfied, then ∇f is Lipschitz continuous with Lipschitz constant L. Using Lemma 4.1.3 where $\mathbf{y} = \alpha \mathbf{s}^{(k)}$, we have the following upper bound on $f(\mathbf{x}^{(k)} + \alpha \mathbf{s}^{(k)})$,

$$f(\mathbf{x}^{(k)} + \alpha \mathbf{s}^{(k)}) \leq f(\mathbf{x}^{(k)}) + \alpha (\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)} + \alpha^2 L \|\mathbf{s}^{(k)}\|^2. \tag{4.9}$$

Therefore, if

$$0 \leq \alpha \leq \frac{(\beta - 1)(\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)}}{L\|\mathbf{s}^{(k)}\|^2}, \tag{4.10}$$

then

$$f(\mathbf{x}^{(k)}) + \alpha (\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)} + \alpha^2 L \|\mathbf{s}^{(k)}\|^2 \leq f(\mathbf{x}^{(k)}) + \beta \alpha (\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)} \tag{4.11}$$

and (3.18) is satisfied. Note that (4.10) is equivalent to $\alpha \in \left[0, \alpha_{max}^{(k)}\right]$, where $\alpha_{max}^{(k)}$ is the upper bound in (4.10), as required. □

Lemma 4.2.2 (Armijo stepsize lower bound (see [22] Lemma 3))**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume that A6, A7 and A8 hold. Then the Armijo stepsize* $\alpha^{(k)}$ *satisfies* $\alpha^{(k)} \geq \min\left\{\alpha_0, \tau \alpha_{max}^{(k)}\right\}$*,* $\forall k \geq 0$*, where* $\alpha_0 > 0$ *is the initial stepsize.*

Proof. (see [22] Lemma 3). Let α_i denote the value of the bArmijo parameter at the i^{th} iteration of the bArmijo loop, where $i = 0, 1, 2, \ldots$. We know that if α_0 satisfies (3.18), then $\alpha^{(k)} = \alpha_0$ and the bArmijo iterations terminate with $i = 0$. Otherwise, from Lemma 4.2.1, we know that (3.18) will be satisfied as soon as $\alpha^{(k)} \leq \alpha_{max}^{(k)}$. Let $(i-1)$ denote the subscript of the last bArmijo iteration such that $\alpha_{i-1} > \alpha_{max}^{(k)}$. Therefore, we have that in the next iteration, (3.18) will be satisfied and $\alpha_i \leq \alpha_{max}^{(k)}$ where $\alpha_i = \tau \alpha_{i-1} > \tau \alpha_{max}^{(k)}$ and $\alpha^{(k)} = \alpha_i$ so $\alpha^{(k)}$ satisfies $\alpha^{(k)} \geq \min\left\{\alpha_0, \tau \alpha_{max}^{(k)}\right\}$, $\forall k \geq 0$, as required. □

Using Lemma 4.2.1 and Lemma 4.2.2, we are able to prove the Zoutendijk condition (as in [96]) and its variant, which utilises the angle between $\mathbf{s}^{(k)}$ (the GN search direction) and $-\nabla f(\mathbf{x}^{(k)})$ (the steepest descent direction), $\theta^{(k)}$, which is given by

$$\cos(\theta^{(k)}) = \frac{(-\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)}}{\|\nabla f(\mathbf{x}^{(k)})\| \|\mathbf{s}^{(k)}\|}. \tag{4.12}$$

This enables us to eventually show that by using the GN step direction $\mathbf{s}^{(k)}$ and bArmijo stepsizes $\alpha^{(k)}$ we are able to obtain a globally convergent method. We note that the bounds on $\cos(\theta^{(k)})$ and $\|\mathbf{s}\|$ assumed in the following lemma are later proven for the LS method using A6 - A8.

Theorem 4.2.3 (Zoutendijk condition for the LS method)**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A6, A7 and A8 hold. Furthermore, let* $\cos(\theta^{(k)}) \geq \delta > 0$ *and the 2-norm of the GN step s,* $\|s\|$ *be bounded below such that* $\|s\| \geq \phi > 0$*. Then for all* $k \geq 0$ *we have that,*

Case 1: *If* $\min\left\{\alpha_0, \tau\alpha_{max}^{(k)}\right\} = \tau\alpha_{max}^{(k)}$ *then,*

$$\sum_{k\geq 0} \cos^2(\theta^{(k)}) \|\nabla f(\mathbf{x}^{(k)})\|^2 < \infty, \tag{4.13}$$

which is known as the Zoutendijk condition, or,

Case 2: *If* $\min\left\{\alpha_0, \tau\alpha_{max}^{(k)}\right\} = \alpha_0$ *then,*

$$\sum_{k\geq 0} \cos(\theta^{(k)}) \|\nabla f(\mathbf{x}^{(k)})\| \|\mathbf{s}^{(k)}\| < \infty. \tag{4.14}$$

Proof. As the assumptions of both Lemma 4.2.1 and Lemma 4.2.2 are satisfied, we can substitute the result from Lemma 4.2.1 into Lemma 4.2.2, giving us,

$$\alpha^{(k)} \geq \min\left\{\alpha_0, \tau\frac{(\beta-1)(\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)}}{L\|\mathbf{s}^{(k)}\|^2}\right\}, \qquad \forall k \geq 0, \tag{4.15}$$

which is the lower bound for the bArmijo chosen stepsize $\alpha^{(k)}$. We first consider **Case 1** where,

Case 1:

$$\min\left\{\alpha_0, \tau\frac{(\beta-1)(\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)}}{L\|\mathbf{s}^{(k)}\|^2}\right\} = \tau\frac{(\beta-1)(\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)}}{L\|\mathbf{s}^{(k)}\|^2}. \tag{4.16}$$

We can substitute (4.16) into (3.18) giving,

$$f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)}) - c_1\frac{((\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)})^2}{\|\mathbf{s}^{(k)}\|^2}. \tag{4.17}$$

where $c_1 = \frac{\beta\tau(1-\beta)}{L}$ and $c_1 > 0$. Using (4.12), we have

$$\cos^2(\theta^{(k)}) = \frac{1}{\|\nabla f(\mathbf{x}^{(k)})\|^2} \frac{((\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)})^2}{\|\mathbf{s}^{(k)}\|^2}. \tag{4.18}$$

Substituting (4.18) into (4.17) and rearranging we obtain,

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) \geq c_1 \cos^2(\theta^{(k)}) \|\nabla f(\mathbf{x}^{(k)})\|^2. \tag{4.19}$$

So for all values of k, we have the following set of inequalities

$$\begin{aligned} f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(1)}) &\geq c_1 \cos^2(\theta^{(0)}) \|\nabla f(\mathbf{x}^{(0)})\|^2. \\ f(\mathbf{x}^{(1)}) - f(\mathbf{x}^{(2)}) &\geq c_1 \cos^2(\theta^{(1)}) \|\nabla f(\mathbf{x}^{(1)})\|^2. \\ &\vdots \\ f(\mathbf{x}^{(k-1)}) - f(\mathbf{x}^{(k)}) &\geq c_1 \cos^2(\theta^{(k-1)}) \|\nabla f(\mathbf{x}^{(k-1)})\|^2. \\ f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) &\geq c_1 \cos^2(\theta^{(k)}) \|\nabla f(\mathbf{x}^{(k)})\|^2. \end{aligned} \tag{4.20}$$

We can sum the inequalities in (4.20) to give us the following,

$$\begin{aligned} \sum_{j=0}^{k} f(\mathbf{x}^{(j)}) - \sum_{j=1}^{k+1} f(\mathbf{x}^{(j)}) &\geq c_1 \sum_{j=0}^{k} \cos^2(\theta^{(j)}) \|\nabla f(\mathbf{x}^{(j)})\|^2. \\ f(\mathbf{x}^{(0)}) + \sum_{j=1}^{k} f(\mathbf{x}^{(j)}) - f(\mathbf{x}^{(k+1)}) - \sum_{j=1}^{k} f(\mathbf{x}^{(j)}) &\geq c_1 \sum_{j=0}^{k} \cos^2(\theta^{(j)}) \|\nabla f(\mathbf{x}^{(j)})\|^2. \end{aligned} \tag{4.21}$$

This simplifies to

$$f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)}) \geq c_1 \sum_{j=0}^{k} \cos^2(\theta^{(j)}) \|\nabla f(\mathbf{x}^{(j)})\|^2. \tag{4.22}$$

As f is a NLLS function of the form (2.28), for all k we have that $f(\mathbf{x}^{(k)})$ is bounded below by zero, that is, $f(\mathbf{x}^{(k)}) \geq 0$. Therefore, due to the sufficient decrease in $f(\mathbf{x}^{(k)})$ guarantee of the LS method that ensures that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$ for all k, we have that $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)})$ is bounded above for all k by $f(\mathbf{x}^{(0)})$, a positive constant. This gives

$$f(\mathbf{x}^{(0)}) \geq f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)}) \geq c_1 \sum_{j=0}^{k} \cos^2(\theta^{(j)}) \|\nabla f(\mathbf{x}^{(j)})\|^2. \tag{4.23}$$

Therefore, by rearranging and taking limits of (4.23) as $k \to \infty$, we have

$$\begin{aligned} \lim_{k\to\infty} \left(c_1 \sum_{j=0}^{k} \cos^2(\theta^{(j)}) \|\nabla f(\mathbf{x}^{(j)})\|^2 \right) &\leq \lim_{k\to\infty} \left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)}) \right) \leq f(\mathbf{x}^{(0)}) < \infty \\ \sum_{k=0}^{\infty} \cos^2(\theta^{(k)}) \|\nabla f(\mathbf{x}^{(k)})\|^2 &< \infty, \end{aligned} \tag{4.24}$$

as required. Next we consider **Case 2** where a similar result is obtained. We have,

Case 2:

$$\min\left\{\alpha_0, \tau\frac{(\beta-1)(\nabla f(\mathbf{x}^{(k)}))^T\mathbf{s}^{(k)}}{L\|\mathbf{s}^{(k)}\|^2}\right\} = \alpha_0 \tag{4.25}$$

Substituting (4.25) into (3.18) and setting $c_2 = \beta\alpha_0$, where $c_2 > 0$ gives,

$$f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)}) + c_2\frac{(\nabla f(\mathbf{x}^{(k)}))^T\mathbf{s}^{(k)}}{\|\nabla f(\mathbf{x}^{(k)})\|\|\mathbf{s}^{(k)}\|}\|\nabla f(\mathbf{x}^{(k)})\|\|\mathbf{s}^{(k)}\|. \tag{4.26}$$

Substituting (4.12) and rearranging, we have

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) \geq c_2\cos(\theta^{(k)})\|\nabla f(\mathbf{x}^{(k)})\|\|\mathbf{s}^{(k)}\|. \tag{4.27}$$

So for all values of k, we have the following set of inequalities

$$\begin{aligned}
f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(1)}) &\geq c_2\cos(\theta^{(0)})\|\nabla f(\mathbf{x}^{(0)})\|\|\mathbf{s}^{(0)}\| \\
f(\mathbf{x}^{(1)}) - f(\mathbf{x}^{(2)}) &\geq c_2\cos(\theta^{(1)})\|\nabla f(\mathbf{x}^{(1)})\|\|\mathbf{s}^{(1)}\| \\
&\vdots \\
f(\mathbf{x}^{(k-1)}) - f(\mathbf{x}^{(k)}) &\geq c_2\cos(\theta^{(k-1)})\|\nabla f(\mathbf{x}^{(k-1)})\|\|\mathbf{s}^{(k-1)}\| \\
f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) &\geq c_2\cos(\theta^{(k)})\|\nabla f(\mathbf{x}^{(k)})\|\|\mathbf{s}^{(k)}\|.
\end{aligned} \tag{4.28}$$

We can sum the inequalities in (4.28) to give us the following,

$$\begin{aligned}
\sum_{j=0}^{k} f(\mathbf{x}^{(j)}) - \sum_{j=1}^{k+1} f(\mathbf{x}^{(j)}) &\geq c_2\sum_{j=0}^{k}\cos(\theta^{(j)})\|\nabla f(\mathbf{x}^{(j)})\|\|\mathbf{s}^{(j)}\|. \\
f(\mathbf{x}^{(0)}) + \sum_{j=1}^{k} f(\mathbf{x}^{(j)}) - f(\mathbf{x}^{(k+1)}) - \sum_{j=1}^{k} f(\mathbf{x}^{(j)}) &\geq c_2\sum_{j=0}^{k}\cos(\theta^{(j)})\|\nabla f(\mathbf{x}^{(j)})\|\|\mathbf{s}^{(j)}\|.
\end{aligned} \tag{4.29}$$

This simplifies to

$$f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)}) \geq c_2\sum_{j=0}^{k}\cos(\theta^{(j)})\|\nabla f(\mathbf{x}^{(j)})\|\|\mathbf{s}^{(j)}\|. \tag{4.30}$$

We use the same argument as we did in **Case 1**. This gives

$$f(\mathbf{x}^{(0)}) \geq f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)}) \geq c_2\sum_{j=0}^{k}\cos(\theta^{(j)})\|\nabla f(\mathbf{x}^{(j)})\|\|\mathbf{s}^{(j)}\|. \tag{4.31}$$

Therefore, by rearranging and taking limits of (4.31) as $k \to \infty$, we have

$$\begin{aligned}
\lim_{k\to\infty}\left(c_2\sum_{j=0}^{k}\cos(\theta^{(j)})\|\nabla f(\mathbf{x}^{(j)})\|\|\mathbf{s}^{(j)}\|\right) &\leq \lim_{k\to\infty}\left(f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)})\right) \leq f(\mathbf{x}^{(0)}) < \infty \\
\sum_{k=0}^{\infty}\cos(\theta^{(k)})\|\nabla f(\mathbf{x}^{(k)})\|\|\mathbf{s}^{(k)}\| &< \infty,
\end{aligned} \tag{4.32}$$

as required. □

Recall from earlier in the chapter, conditions for global convergence of NLLS optimisation methods equipped with conditions on the step length exist in the literature. However, they do not exist for Algorithm 3.3.1 specifically. Recall from Section 3.3.1, the strongest convergence guarantee that we can obtain for a line-search method is that the gradient norms converge to zero, (3.21). In Theorem 10.1 of the work of [96], the authors outline a global convergence result for Gauss-Newton equipped with Wolfe line-search that satisfies the strongest convergence guarantee (3.21). Recall from Section 3.3.1, the use of the Wolfe conditions (the Armijo condition (3.18) and an additional curvature condition) to guarantee convergence is potentially more computationally costly due to the use of curvature information from the Hessian (2.31). Such information is difficult to obtain in the VarDA setting, hence why we only consider the use of the Armijo condition.

To prove global convergence to a stationary point of the Algorithm 3.3.1, we use the same framework of the proof of Theorem 10.1 in the work of [96], except we consider the use of the Armijo condition (3.18) only. This result is presented in the following theorem. We show how Zoutendijk's condition holds for the LS method and that $\|\mathbf{s}^{(k)}\|$ and $\cos(\theta^{(k)})$ are bounded away from 0, enabling us to prove global convergence of the LS method.

Theorem 4.2.4 (Global convergence for the Gauss-Newton with bArmijo line search method, Algorithm 3.3.1)**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A6 - A9 hold. Then if the iterates* $\{\mathbf{x}^{(k)}\}$ *are generated by the GN method with stepsizes* $\alpha^{(k)}$ *that satisfy the Armijo condition* (3.18)*, we have*

$$\lim_{k\to\infty} \mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}) = 0 \tag{4.33}$$

and the gradient norms converge to zero, so the Gauss-Newton method with bArmijo line search is globally convergent.

Proof. We want to satisfy the conditions of Theorem 4.2.3. We first show that $\|\mathbf{s}^{(k)}\|$ is bounded below.

Taking the norm of (3.10) and using the submultiplicative property (2.12) we have,

$$\begin{aligned} \|(\mathbf{J}(\mathbf{x}^{(k)}))^T \mathbf{J}(\mathbf{x}^{(k)}) \mathbf{s}^{(k)}\| &= \|\nabla f(\mathbf{x}^{(k)})\| \\ \|(\mathbf{J}(\mathbf{x}^{(k)}))^T \mathbf{J}(\mathbf{x}^{(k)})\| \|\mathbf{s}^{(k)}\| &\geq \|\nabla f(\mathbf{x}^{(k)})\| \end{aligned} \tag{4.34}$$

Rearranging (4.34) we have the following lower bound on $\|\mathbf{s}^{(k)}\|$,

$$\|\mathbf{s}^{(k)}\| \geq \frac{\|\nabla f(\mathbf{x}^{(k)})\|}{\|(\mathbf{J}(\mathbf{x}^{(k)}))^T \mathbf{J}(\mathbf{x}^{(k)})\|}. \tag{4.35}$$

Using (4.2) we have,

$$\begin{aligned} \|(\mathbf{J}(\mathbf{x}^{(k)}))^T \mathbf{J}(\mathbf{x}^{(k)})\| &\leq \|(\mathbf{J}(\mathbf{x}^{(k)}))^T\| \|\mathbf{J}(\mathbf{x}^{(k)})\| \\ &\leq L_r^2 \end{aligned} \tag{4.36}$$

and so $\|\mathbf{s}^{(k)}\|$ is bounded below as follows

$$\begin{aligned}\|\mathbf{s}^{(k)}\| &\geq \frac{\|\nabla f(\mathbf{x}^{(k)})\|}{\|(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\|} \\ &\geq \frac{\|\nabla f(\mathbf{x}^{(k)})\|}{L_r^2}\end{aligned} \tag{4.37}$$

We next need to show that $\cos(\theta^{(k)})$ is bounded away from 0. Multiplying Equation (3.10) through by $(\mathbf{s}^{(k)})^T$ and simplifying, we have,

$$\begin{aligned}-(\mathbf{s}^{(k)})^T\nabla f(\mathbf{x}^{(k)}) &= (\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x})^{(k)})^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} \\ &= \|\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)}\|^2.\end{aligned} \tag{4.38}$$

Substituting the results from (4.38) and the first line of (4.34) into (4.12), we have for $\mathbf{x} = \mathbf{x}^{(k)} \in \mathcal{L}$ and $\mathbf{s} = \mathbf{s}^{(k)}$,

$$\cos(\theta^{(k)}) = \frac{\|\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)}\|^2}{\|(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)}\|\|\mathbf{s}^{(k)}\|}. \tag{4.39}$$

Using the submultiplicative property (2.12), (4.2) and from A9 that $\|\mathbf{J}(\mathbf{x})\mathbf{z}\| \geq \upsilon\|\mathbf{z}\|$ where $\nu > 0$, we can show that $\cos(\theta^{(k)})$ is bounded away from 0 as follows

$$\begin{aligned}\cos(\theta^{(k)}) &= \frac{\|\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)}\|^2}{\|(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)}\|\|\mathbf{s}^{(k)}\|} \\ &\geq \frac{\|\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)}\|^2}{\|(\mathbf{J}(\mathbf{x}^{(k)}))\|^2\|\mathbf{s}^{(k)}\|^2} \\ &\geq \frac{\upsilon^2\|\mathbf{s}^{(k)}\|^2}{L_r^2\|\mathbf{s}^{(k)}\|^2} \\ &\geq \frac{\upsilon^2}{L_r^2} \\ &> 0.\end{aligned} \tag{4.40}$$

From this result, we now know that $\cos(\theta^{(k)})$ is a positive constant. Therefore, the assumptions of Theorem 4.2.3 are satisfied and we can consider the two cases.

Case 1:
The LHS of (4.13) is an infinite series of positive, real numbers. Therefore, since the limit is finite, we must have

$$\lim_{k\to\infty} \cos^2(\theta^{(k)})\|\nabla f(\mathbf{x}^{(k)})\|^2 = 0. \tag{4.41}$$

A similar result holds for **Case 2**.

Case 2:

Substituting in the result from (4.37) into Equation (4.14), we have the following lower bound on the LHS of (4.14),

$$\sum_{k\geq 0} \cos(\theta^{(k)}) \frac{\|\nabla f(\mathbf{x}^{(k)})\|^2}{L_r^2} < \infty, \tag{4.42}$$

which again is an infinite series of positive, real numbers. Therefore,

$$\lim_{k\to\infty} \cos(\theta^{(k)}) \frac{\|\nabla f(\mathbf{x}^{(k)})\|^2}{L_r^2} = 0. \tag{4.43}$$

In both cases, as $\cos(\theta^{(k)}) \geq \frac{v^2}{L_r^2} > 0$, then it must be that $\|\nabla f(\mathbf{x}^{(k)})\| \to 0$ as $k \to \infty$, so $\lim_{k\to\infty} \mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}) = 0$, as required. □

We next present the global convergence proof for the Gauss-Newton with regularisation method, Algorithm 3.3.2. The REG method has no sufficient decrease condition as in the LS method. Therefore, the use of the level set (4.8) is not required. The assumptions for convergence are similar to the LS method aside from the requirement of $\mathbf{J}(\mathbf{x})$ being full rank.

4.3 Global convergence of REG

We recall that the Gauss-Newton with regularisation method minimises the least-squares function (2.28) by solving (3.23) for the new descent direction and adapts the regularisation parameter $\gamma^{(k)}$ according to the ratio (3.26) within each iteration of Algorithm 3.3.2.

We want to prove global convergence for the REG method to a stationary point, that is that $\mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}) \to 0$ as $k \to \infty$ for $k = 0, 1, 2, \ldots$, where the sequence $\{\mathbf{x}^{(k)}\}$ is generated by the REG method with initial guess $\mathbf{x}^{(0)}$ and initial regularisation parameter $\gamma^{(0)}$.

A literature review of local and global convergence proofs of variations of the Levenberg-Marquardt (LM) method with similarities to Algorithm 3.3.2 can be found in [10]. The work of [60] considers the local convergence of the LM method, where the regularisation parameter can go to zero so as to obtain the GN step. As the requirements for fast local convergence of the REG method (a small REG parameter) contradict the requirements of global convergence (a large REG parameter), within our work, we consider only global convergence of the REG method for use in the case when GN performs poorly. This is to ensure that we can at least guarantee faster global convergence of the REG method to meet the time and computational cost limitations present in VarDA.

The work of [84], where the LM method was originally proposed (along with [70]), considers more of a geometric approach to the global convergence of the method and, as in our work, does not allow the regularisation parameter to go to zero so as to guarantee global convergence. Recall from Section 3.4, the work of [10] includes an additional parameter corresponding to a successful step of the method to balance local and global convergence requirements. They prove global convergence under Assumptions A6 - A8 and an additional

assumption on the step calculation. This additional assumption is included to consider the case where the GN subproblem may be solved inexactly through the use of an iterative solver. In our work, this latter assumption is not required as we assume that the GN subproblem is solved exactly. We consider this case as in VarDA practice, many iterations of a linear least-squares solver are performed to solve the inner loop problem, thus it is reasonable to assume that the inner loop problem is solved to a relatively high accuracy.

Some adaptations of the lemmas from the global convergence proof of the Adaptive Regularisation algorithm using Cubics (ARC method) have been used in our proof of global convergence of REG, see [23] and [24]. We begin the proof of global convergence of REG by deriving an expression for the predicted model decrease in terms of the gradient. We require the use of an upper bound on $\gamma^{(k)}$, denoted as $\gamma_{\max}$, which is derived using a property of Lipschitz continuous gradients. We show that $\gamma^{(k)} \leq \gamma_{\max}$ for all $k \geq 0$ by first showing that if $\gamma^{(k)}$ is large enough, then we have a successful step so that $\gamma^{(k)}$ can stop increasing due to unsuccessful steps in Algorithm 3.3.2. We use the expression for $\gamma_{\max}$ to prove global convergence of the REG method under assumptions A6-A8 by showing that the gradient norms converge to zero as we iterate.

We begin by deriving an expression for the predicted model decrease.

Lemma 4.3.1 (Model decrease bound)**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Furthermore, for all* $k \geq 0$*, let* $\mathbf{x}^{(k)}$ *and* $\mathbf{s}^{(k)}$ *be the REG method iterate and step respectively where* $\gamma^{(k)}$ *is bounded below, that is,* $\gamma^{(k)} \geq \gamma_{\min} > 0$ *where* $\gamma_{\min}$ *is a user chosen quantity. Then, where* $m^{(k)}$ *is defined as in* (3.25)*, we have that,*

$$f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) \geq \frac{1}{2}\gamma_{\min}\|\mathbf{s}^{(k)}\|^2, \tag{4.44}$$

for all $k \geq 0$*.*

Proof. Substituting the RHS of Equations (2.28) and (3.25) into the LHS of Equation (4.44), we obtain the following expression for the model decrease

$$\begin{aligned} f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) &= \frac{1}{2}\|\mathbf{r}(\mathbf{x}^{(k)})\|^2 - \frac{1}{2}\|\mathbf{r}(\mathbf{x}^{(k)})\|^2 - (\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{r}(\mathbf{x}^{(k)}) \\ &\qquad - \frac{1}{2}(\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} - \frac{1}{2}\gamma^{(k)}\|\mathbf{s}^{(k)}\|^2 \\ &= -(\mathbf{s}^{(k)})^T\nabla f(\mathbf{x}^{(k)}) - \frac{1}{2}(\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} \\ &\qquad - \frac{1}{2}\gamma^{(k)}\|\mathbf{s}^{(k)}\|^2. \end{aligned} \tag{4.45}$$

To simplify Equation (4.45), we use the fact that $\mathbf{s}^{(k)} = \arg\min_{\mathbf{s}} m^{(k)}(\mathbf{s})$ is obtained when $\nabla_{\mathbf{s}} m^{(k)}(\mathbf{s}^{(k)}) = 0$, which is equivalent to solving (3.23).

Substituting $(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{r}(\mathbf{x}^{(k)}) = \nabla f(\mathbf{x}^{(k)})$ to simplify (3.23), we have

$$(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} + \gamma^{(k)}\mathbf{s}^{(k)} = -\nabla f(\mathbf{x}^{(k)}). \tag{4.46}$$

Multiplying through Equation (4.46) by $(\mathbf{s}^{(k)})^T$, we obtain,

$$\begin{aligned} -(\mathbf{s}^{(k)})^T(\nabla f(\mathbf{x}^{(k)})) &= (\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} + \gamma^{(k)}(\mathbf{s}^{(k)})^T\mathbf{s}^{(k)} \\ &= (\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} + \gamma^{(k)}\|\mathbf{s}^{(k)}\|^2 \end{aligned} \tag{4.47}$$

Substituting Equation (4.47) into Equation (4.45), we have,

$$\begin{aligned} f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) &= (\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} + \gamma^{(k)}\|\mathbf{s}^{(k)}\|^2 \\ &\quad - \frac{1}{2}(\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} - \frac{1}{2}\gamma^{(k)}\|\mathbf{s}^{(k)}\|^2 \\ &= \frac{1}{2}(\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} + \frac{1}{2}\gamma^{(k)}\|\mathbf{s}^{(k)}\|^2. \end{aligned} \tag{4.48}$$

As $(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})$ is a positive semidefinite matrix, we have that

$$(\mathbf{s}^{(k)})^T(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} \geq 0, \tag{4.49}$$

for all $\mathbf{s}^{(k)}$. Therefore, we have

$$f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) \geq \frac{1}{2}\gamma^{(k)}\|\mathbf{s}^{(k)}\|^2. \tag{4.50}$$

Furthermore, using that $\gamma^{(k)} \geq \gamma_{\min} > 0$ for all $k \geq 0$, we have the following lower bound on the model decrease

$$f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) \geq \frac{1}{2}\gamma_{\min}\|\mathbf{s}^{(k)}\|^2, \forall k \geq 0, \tag{4.51}$$

as required. □

We now need to connect $\mathbf{s}^{(k)}$ to the gradient $\nabla f(\mathbf{x}^{(k)})$ so that the predicted model decrease (4.44) is in terms of the gradient. We note that the upper bound on $\gamma^{(k)}$ assumed in the following lemma is later proven for the REG method using A6, A7 and A8.

Lemma 4.3.2 (Bound on the step $\mathbf{s}^{(k)}$)**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A7 holds. Furthermore, for all* $k \geq 0$*, let* $\mathbf{x}^{(k)}$ *and* $\mathbf{s}^{(k)}$ *be the REG method iterate and step respectively and* $\gamma^{(k)}$ *be bounded above by* $\gamma_{\max} > 0$ *such that* $\gamma^{(k)} \leq \gamma_{\max}$*. Then for all* $k \geq 0$ *we have that,*

$$\|\mathbf{s}^{(k)}\| \geq \frac{\|\nabla f(\mathbf{x}^{(k)})\|}{(L_r^2 + \gamma_{\max})}. \tag{4.52}$$

Proof. By taking the norm of Equation (4.46) and using the submultiplicative property (2.12), we obtain

$$\|\nabla f(\mathbf{x}^{(k)})\| \leq \left\|\left((\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)}) + \gamma^{(k)}\mathbf{I}\right)\right\| \|\mathbf{s}^{(k)}\|. \tag{4.53}$$

Applying the triangle inequality (2.6) to Equation (4.53), we obtain

$$\|\nabla f(\mathbf{x}^{(k)})\| \leq \left(\|(\mathbf{J}(\mathbf{x}^{(k)}))^T\mathbf{J}(\mathbf{x}^{(k)})\| + \gamma^{(k)}\|\mathbf{I}\|\right) \|\mathbf{s}^{(k)}\|. \tag{4.54}$$

As a consequence of assumption A7, we know from Theorem 4.1.1 that $\mathbf{J}$ is uniformly bounded above for all $\mathbf{x} \in \mathbb{R}^n$ such that $\|\mathbf{J}(\mathbf{x})\| \leq L_r$. Therefore, we obtain

$$\begin{aligned} \|\nabla f(\mathbf{x}^{(k)})\| &\leq \left(\|(\mathbf{J}(\mathbf{x}^{(k)})\|^2 + \gamma^{(k)}\right) \|\mathbf{s}^{(k)}\| \\ &\leq (L_r^2 + \gamma^{(k)})\|\mathbf{s}^{(k)}\|. \end{aligned} \tag{4.55}$$

As $\gamma^{(k)}$ is bounded above by $\gamma_{\max}$ such that $\gamma^{(k)} \leq \gamma_{\max}$, by rearranging (4.55) we obtain

$$\|\mathbf{s}^{(k)}\| \geq \frac{\|\nabla f(\mathbf{x}^{(k)})\|}{(L_r^2 + \gamma_{\max})}, \tag{4.56}$$

as required. □

We now need to show that $\gamma^{(k)} \leq \gamma_{\max}$ for all $k \geq 0$. For this, we need to first show that if $\gamma^{(k)}$ is large enough, then we have a successful step so that $\gamma^{(k)}$ can stop increasing due to unsuccessful steps in Algorithm 3.3.2. We can show that if for any iteration k of the REG method, the regularisation parameter is bounded below by the Lipschitz constant of ∇f, then the iteration k is very successful and the regularisation parameter is decreased for the next iteration.

Lemma 4.3.3 (Bound on the regularisation parameter $\gamma^{(k)}$)**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A6, A7 and A8 hold. Furthermore, for all* $k \geq 0$*, let* $\mathbf{x}^{(k)}$ *and* $\mathbf{s}^{(k)}$ *be the REG method iterate and step respectively and* $\gamma^{(k)}$ *denote the regularisation parameter. Then if*

$$\gamma^{(k)} \geq L, \tag{4.57}$$

where L *is the Lipschitz constant of* ∇f*, then iteration* k *is very successful and* $\gamma^{(k+1)} < \gamma^{(k)}$*.*

Proof. We first note that for k to be very successful, we need that the ratio $\rho^{(k)} \geq \eta_2$ where $\eta_2 \in (0, 1)$. For this, it is sufficient for $\rho^{(k)} \geq 1$ which, by using (3.26), is equivalent to having

$$\frac{f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})}{f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)})} \geq 1. \tag{4.58}$$

From Lemma 4.3.1 we have that $f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) \geq 0$. Therefore, we can write (4.58) as

$$f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)}) \leq m^{(k)}(\mathbf{s}^{(k)}). \tag{4.59}$$

For the LHS of Equation (4.59), we can use that the assumptions of Lemma 4.1.2 are satisfied. Therefore, ∇f is Lipschitz continuous with Lipschitz constant L. Using Lemma 4.1.3 where $\mathbf{y} = \mathbf{s}^{(k)}$, we have the following upper bound on $f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)})$,

$$f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)}) \leq f(\mathbf{x}^{(k)}) + (\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)} + \frac{1}{2} L \|\mathbf{s}^{(k)}\|^2, \tag{4.60}$$

which always holds for $f \in \mathcal{C}^1(\mathbb{R}^n)$ with ∇f Lipschitz continuous.

For the RHS of Equation (4.59), we can use Equations (4.45) and (4.49) to obtain the following lower bound on $m^{(k)}(\mathbf{s}^{(k)})$,

$$m^{(k)}(\mathbf{s}^{(k)}) \geq f(\mathbf{x}^{(k)}) + (\mathbf{s}^{(k)})^T \nabla f(\mathbf{x}^{(k)}) + \frac{1}{2}\gamma^{(k)} \|\mathbf{s}^{(k)}\|^2. \tag{4.61}$$

If (4.57) holds then we have the following inequality concerning the RHS's of Equations (4.60) and (4.61),

$$\begin{aligned} f(\mathbf{x}^{(k)} + \mathbf{s}^{(k)}) &\leq f(\mathbf{x}^{(k)}) + (\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)} + \frac{1}{2} L \|\mathbf{s}^{(k)}\|^2 \\ &\leq f(\mathbf{x}^{(k)}) + (\nabla f(\mathbf{x}^{(k)}))^T \mathbf{s}^{(k)} + \frac{1}{2}\gamma^{(k)} \|\mathbf{s}^{(k)}\|^2 \\ &\leq m^{(k)}(\mathbf{s}^{(k)}), \end{aligned} \tag{4.62}$$

which gives us (4.59) and is equivalent to (4.58) and k being very successful, as required. □

In the following lemma we deduce $\gamma_{\max}$.

Lemma 4.3.4 (Upper bound on $\gamma^{(k)}$ [24])**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A6, A7 and A8 hold. Then, for all* $0 \leq k < \infty$*, we have*

$$\gamma^{(k)} \leq \gamma_{\max}, \tag{4.63}$$

where $\gamma^{(k)}$ *denotes the regularisation parameter of the REG method and*

$$\gamma_{\max} = \max\{\gamma^{(0)}, 2L\}, \tag{4.64}$$

where $L > 0$ *is the Lipschitz constant of* ∇f*.*

Proof. From Lemma 4.3.3, we have for any $0 \leq k < \infty$ that

$$\gamma^{(k)} \geq L \implies \gamma^{(k+1)} < \gamma^{(k)}, \tag{4.65}$$

holds. We need to consider two cases of (4.64); when $\gamma^{(0)} \leq 2L$ (**Case 1**) and when $\gamma^{(0)} \geq 2L$ (**Case 2**).

Case 1: When $\gamma^{(0)} \leq 2L$, from (4.65) and Lemma 4.3.3 we have that the following hold

$$\begin{aligned} L \leq \gamma^{(0)} \leq 2L \text{ and} \\ \cdots < \gamma^{(2)} < \gamma^{(1)} < \gamma^{(0)} \leq 2L, \end{aligned} \tag{4.66}$$

which implies $\gamma^{(k)} \leq 2L$ for all k where $0 \leq k < \infty$.

Case 2: When $\gamma^{(0)} \geq 2L$, the REG parameter is already sufficiently large at the start of the REG method and so again, due to Lemma 4.3.3, $\gamma^{(k)} \leq \gamma^{(0)}$ for all k. Using **Case 1** and **Case 2**, we have shown that (4.63) holds with $\gamma_{\max}$ given in (4.64), as required. □

We note that in (4.64), we scale L by 2, as we do for $\gamma^{(k)}$ in (3.29) for unsuccessful steps, to account for the case where $\gamma^{(k)}$ is just below L and k is not very successful. Furthermore, the use of $\gamma^{(0)}$ in (4.63) accounts for the initial size of $\gamma^{(k)}$. We can now prove global convergence for the REG method.

The global convergence theorem for the GN with quadratic regularisation method, Algorithm 3.3.2, is given as follows.

Theorem 4.3.5 (Global convergence for the Gauss-Newton with regularisation method, Algorithm 3.3.2)**.** *Suppose we have a NLLS function of the form* (2.28) *and its gradient* (2.30) *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A6 - A8 hold. Then if the iterates* $\{\mathbf{x}^{(k)}\}$ *are generated by the Gauss-Newton with regularisation method, we have that*

$$\lim_{k\to\infty} \mathbf{J}(\mathbf{x}^{(k)})^T \mathbf{r}(\mathbf{x}^{(k)}) = 0 \tag{4.67}$$

and the gradient norms converge to zero, so the Gauss-Newton method with regularisation is globally convergent.

Proof. From (4.44) in Lemma 4.3.1 and (4.52) in Lemma 4.3.2 we deduce that for all $k \geq 0$,

$$f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) \geq \frac{1}{2} \frac{\gamma_{\min}}{(L_r^2 + \gamma_{\max})^2} \|\nabla f(\mathbf{x}^{(k)})\|^2. \tag{4.68}$$

We will first consider when there are finitely many successful and very successful iterations. Let

$$I := \{k : k \geq k_0 + 1\}, \tag{4.69}$$

where k_0 is the last successful iterate. At $k = k_0 + 1$, we either have **Case 1**, that

$$\|\nabla f(\mathbf{x}^{(k)})\| = 0, \tag{4.70}$$

or **Case 2**, that

$$\|\nabla f(\mathbf{x}^{(k)})\| \neq 0. \tag{4.71}$$

Due to the construction of Algorithm 3.3.2, for **Case 1**, (4.70) implies that $\mathbf{x}^{k_0+1} = \mathbf{x}^{k_0+i} = \mathbf{x}^*$ for all $i \geq 1$.

For **Case 2**, we have that if (4.71) holds and we know that this can only mean that $\|\nabla f(\mathbf{x}^{(k)})\| > 0$. More specifically, we have,

$$\|\nabla f(\mathbf{x}^{(k)})\| = \|\nabla f(\mathbf{x}^{(k_0+1)})\| = \varepsilon > 0, \text{ for all } k \in I. \tag{4.72}$$

As the remaining iterates are unsuccessful, according to our updating rule in Algorithm 3.3.2, we are increasing $\gamma^{(k)}$ at each iteration. Therefore, we have as $k \to \infty$ that $\gamma^{(k)} \to \infty$ but we know from Lemma 4.3.4 that $\gamma^{(k)}$ is bounded above by $\gamma_{\max}$, which is a contradiction so only **Case 1** holds.

We next consider when there are infinitely many successful and very successful iterations. Let

$$S = \{k \geq 0 : k \text{ is a successful or very successful iteration}\}, \tag{4.73}$$

then from the definition of $\rho^{(k)}$ in (3.26), we have for all $k \in S$,

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) \geq \eta_1 \left(f(\mathbf{x}^{(k)}) - m^{(k)}(\mathbf{s}^{(k)}) \right). \tag{4.74}$$

Now, using (4.68) in (4.74), we have for all $k \in S$,

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k+1)}) \geq c_3 \|\nabla f(\mathbf{x}^{(k)})\|^2, \tag{4.75}$$

where $c_3 := \frac{\eta_1}{2} \frac{\gamma_{\min}}{(L_r^2 + \gamma_{\max})^2}$ is a constant. Summing up (4.75) for all $j \in S$ where $j \leq k$ for any $k \in S$, we have,

$$\sum_{j=0, j\in S}^{k} \left[f(\mathbf{x}^{(j)}) - f(\mathbf{x}^{(j+1)}) \right] \geq c_3 \sum_{j=0, j\in S}^{k} \|\nabla f(\mathbf{x}^{(j)})\|^2. \tag{4.76}$$

We recall that on unsuccessful iterations we have $f(\mathbf{x}^{(j)}) = f(\mathbf{x}^{(j+1)})$, and so

$$\begin{aligned} \sum_{j=0, j\in S}^{k} \left[f(\mathbf{x}^{(j)}) - f(\mathbf{x}^{(j+1)}) \right] &= \sum_{j=0}^{k} \left[f(\mathbf{x}^{(j)}) - f(\mathbf{x}^{(j+1)}) \right] \\ &= f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)}). \end{aligned} \tag{4.77}$$

Using, (4.77), Equation (4.76) simplifies to

$$f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)}) \geq c_3 \sum_{j=0, j\in S}^{k} \|\nabla f(\mathbf{x}^{(j)})\|^2. \tag{4.78}$$

As f is of the form (2.28), for all k we have that $f(\mathbf{x}^{(k)})$ is bounded below by zero, that is, $f(\mathbf{x}^{(k)}) \geq 0$. Furthermore, REG ensures that $f(\mathbf{x}^{(k+1)}) \leq f(\mathbf{x}^{(k)})$ for all k. Therefore, we have that $f(\mathbf{x}^{(0)}) - f(\mathbf{x}^{(k+1)})$ is bounded above for all k by $f(\mathbf{x}^{(0)})$, a non-negative constant. Therefore, letting $k \to \infty$ we have

$$\lim_{k\to\infty} c_3 \sum_{k=0, k\in S}^{\infty} \|\nabla f(\mathbf{x}^{(k)})\|^2 \leq f(\mathbf{x}^{(0)}) < \infty, \tag{4.79}$$

which is an infinite series of non-negative, real numbers $\|\nabla f(\mathbf{x}^{(k)})\|^2$. We note that $\nabla f(\mathbf{x}^{(k)})$ remains unchanged on unsuccessful steps. Therefore, $\nabla f(\mathbf{x}^{(k)}) \to 0$ as $k \to \infty$ for all k, as required. □

In the remainder of this chapter, we address research question RQ1(b). We apply Algorithms 3.2.3, 3.3.1 and 3.3.2 to both linear and nonlinear 3D-Var problems with the aim of understanding how the use of safeguarding strategies within GN affects the convergence of the outer loop in VarDA. We begin by applying Algorithms 3.2.3, 3.3.1 and 3.3.2 to some general NLLSPs and discussing the results in relation to the variational problem.

4.4 Behaviour of GN, LS and REG

Within this section, we apply GN, LS and REG to two NLLSPs to address research question RQ1(b). We provide an understanding of the convergence behaviour of these methods when varying different parameters.

In Section 4.4.1, we consider a problem where GN performs poorly and show how LS and REG converge within relatively few iterations compared to GN. We then consider a problem where GN performs well in Section 4.4.2 and show how the convergence rate of the LS and REG methods are affected by the initial choice of the globalisation parameters.

4.4.1 Divergence of GN & global convergence of LS and REG

We begin by considering a case where GN fails to converge to a local minimum within a reasonable number of iterations. We take a case from Example 10.2.5 in [33], which shows cases where GN does not converge locally. We show how LS and REG are able to locate the minimum within a limited number of iterations and function evaluations. The problem is outlined as follows.

We aim to find the minimum of (2.28), where $x \in \mathbb{R}$ and the residual vector $\mathbf{r}(x) \in \mathbb{R}^3$ is given by

$$\mathbf{r}(x) = \begin{pmatrix} e^x - 2 \\ e^{2x} - 4 \\ e^{3x} + 8 \end{pmatrix} \tag{4.80}$$

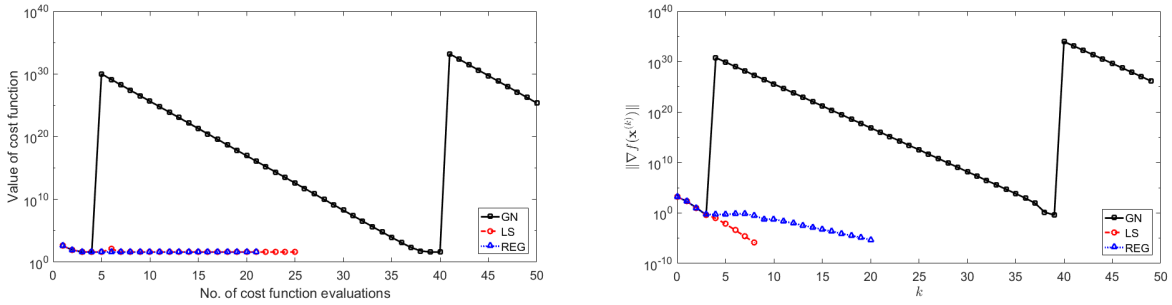
We refer to this NLLSP as DSprob. The true solution of DSprob is known to be $x^* = -0.7915$, where $f(x^*) = 41.145$, making this a nonzero residual problem.

We alter the experimental design slightly to that of Example 10.2.5 in [33]. We use the stopping criterion (3.42) with $\epsilon = 10^{-5}$ to identify a stationary point, as opposed to the stricter choice of 10^{-10} used in [33]. Furthermore, for plotting purposes, we use (3.44), where $k_{\max} = 50$ so as to avoid unnecessary iterations of GN beyond those needed for convergence of LS and REG.

For the LS method, we fix $\beta = 0.1$ and $\tau = 0.5$ in (3.18). We choose the typical choice of $\alpha_0 = 1$ so that the first step assessed by the bArmijo rule is the GN step, but if the step does not satisfy the Armijo condition, then the LS method can adjust the step. For the REG method, we select the typical choice of the initial regularisation parameter, $\gamma^{(0)} = 1$ as well as $\eta_1 = 0.1$ and $\eta_2 = 0.9$ to assess how well the model (3.25) approximates the true function value at the next iteration. Finally, we initialise GN, LS and REG using $x^{(0)} = 1$.

Figure 4.1 shows the convergence plots when applying GN, LS and REG to DSprob. The GN method does not converge to x^* within $k = 50$ iterations and in fact, diverges. From Figure 4.1(a), we see how, for the GN method, the function value abruptly increases at iteration $k = 4$, and then again at iteration $k = 40$. This is also what happens with the gradient norms, as shown in Figure 4.1(b). In fact, we find that this pattern continues until iteration

$k = 22, 1276$, where the GN method finally locates x^*. This is unlike the behaviour of the globally convergent methods LS and REG, both of which are able to converge to x^* within $k = 9$ and $k = 18$ iterations respectively.



(a) Convergence of function

(b) Convergence of gradient norms

Figure 4.1: Convergence plots showing (a) the value of the objective function at each iteration (including unsuccessful iterations) and (b) the gradient norms at each successful iteration k of the GN (black), LS (red) and REG (blue) methods when applied to DSprob.

The GN method misses the minimum of the cost function by taking steps that are too long. This can be visualised in Figure 4.2, which shows the norm of the steps taken by GN, LS and REG at each successful iteration of the methods. From this figure, we see that all three methods begin by taking steps that are similar in the first and second iteration. It is at the third iteration that the methods begin to deviate from each other. The GN step is deemed to be too large by both LS and REG, and is subsequently shortened by the globally convergent methods. This shortening of the step enables both LS and REG to continue to achieve a decrease in the function value until the minimum is achieved, as visualised in Figure 4.1(a). The GN method, however, takes an even larger step at the fourth iteration, which results in a subsequently very large increase in the function value, also visualised in Figure 4.1(a). After which, GN takes a series of cautious steps to reduce the function value, before falling into the same issue of increasing the step when nearing the minimum, demonstrating the GN method's poor local convergence. This behaviour is because, unlike LS and REG, GN is not locally convergent on problems with very large residuals at the solution: a property of DSprob.

Table 4.1 supports the results in Figure 4.1 for LS and REG. From this table we see that, although LS requires the least number of iterations to converge, it requires 25 function evaluations, 16 of which are used to adjust the line search parameter in the bArmijo iterations to guarantee a strict decrease in the function value at each successful iteration k. REG requires 18 iterations for convergence to x^*, 9 more than LS, but a total of 21 function evaluations, 4 less than LS. In these results, the REG parameter only needs to be updated two times to guarantee a monotonic decrease in the function value at each successful iteration k, much fewer updates per iteration than the LS method.

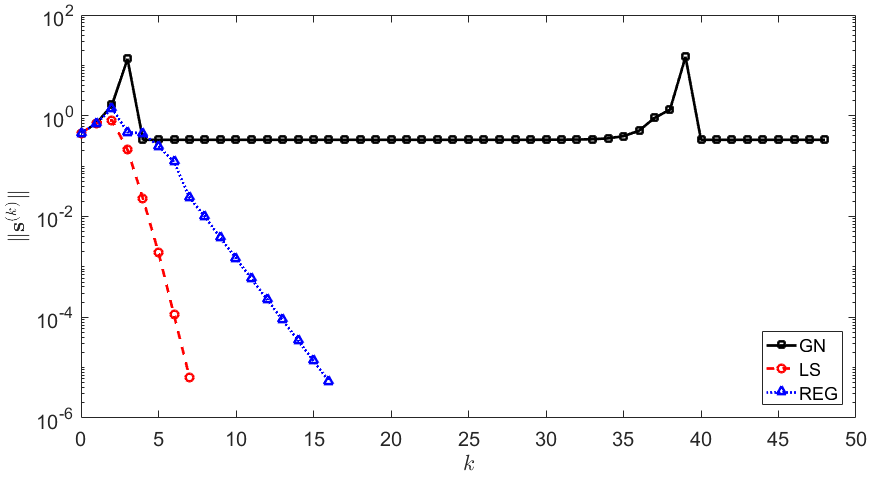


Figure 4.2: Plot of the norm of the steps $\mathbf{s}$ at each successful iteration k of the GN (black), LS (red) and REG (blue) methods when applied to DSprob.

Table 4.1: Table of DSprob implementation results

Method	Successful iterations (k)	Function evaluations	Final Value of $f(x^{(k)})$	Final Value of $x^{(k)}$
LS	9	25	41.145	-0.7915
REG	18	21	41.145	-0.7915

Relating these results to the variational problem in practice, we understand that any more than 4 outer loop iterations (k) and nonlinear function evaluations are too many to solve in the computational time and cost available in Numerical Weather Prediction. Therefore, although the globally convergent methods, LS and REG, perform better than GN in our results, we are also interested in speeding up the convergence of LS and REG so that we are able to benefit further from their global convergence properties within the computational constraints present in DA. In the following section, we use a problem where GN performs well and show how the choices of the initial choices of the globalisation parameters affect the rate of convergence of LS and REG.

4.4.2 Effect of initial choices

In this section, we choose a problem where GN performs well and show how the performance of LS and REG is linked to the initial choice of globalisation parameters, α_0 and $\gamma^{(0)}$.

We aim to find the minimum of the NLLS function $f : \mathbb{R}^2 \to \mathbb{R}$ commonly used to test optimisation methods, known as the Rosenbrock function [110],

$$f(\mathbf{x}) = 10(x_2 - x_1^2)^2 + (x_1 - 1)^2. \tag{4.81}$$

The minimum of the function (4.81) can be obtained by solving the NLLSP of the form

(2.33), where $\mathbf{x} \in \mathbb{R}^2$ and the residual vector $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^2$ is given by

$$\mathbf{r}(\mathbf{x}) = \begin{pmatrix} \sqrt{20}(x_2 - x_1^2) \\ \sqrt{2}(x_1 - 1) \end{pmatrix}. \tag{4.82}$$

The true solution to the minimisation of (4.81) is known to be $\mathbf{x}^* = [1, 1]^T$, where $\mathbf{r}(\mathbf{x}^*) = 0$, making this a zero residual problem. We refer to this minimisation problem as ROSENprob.

To ensure the robustness of our results, we apply the three optimisation methods to a series of $n_r = 1000$ randomly generated problems, where the randomness occurs through the initial guess. For each realisation, a new set of random errors ε_i, where $i = 1, 2$, is generated from the normal distribution

$$\varepsilon_i \sim \mathcal{N}(0, 1) \tag{4.83}$$

and is added to each element of the truth vector $\mathbf{x}^*$ as follows

$$x_i^{(0)} = x_i^* + \varepsilon_i, \tag{4.84}$$

for $i = 1, 2$.

The setup for LS and REG is the same as that in Section 4.4.1. However, to demonstrate the effect initial choices of the globalisation parameters have on the convergence of LS and REG, we consider other choices of the initial globalisation parameters. The choice of $\alpha_0 = 1.5$ in LS results in the first stepsize assessed by the bArmijo rule being larger than the GN step, and if the LS step is too large, the use of $\tau = 0.5$ halves the step to be smaller than the GN step, thus slowing down convergence of LS in the case when GN is performing better than LS. We also consider the choice $\alpha_0 = 2$ to allow LS to take the GN step after one update. We do not consider the choice $\alpha_0 = 1$ as when GN performs well, LS will choose the GN step and thus, have the same convergence behaviour as GN. For the REG method, we consider $\gamma^{(0)} = 1$ as in Section 4.4.1, but we also show how the use of the choice of $\gamma^{(0)} = 0.5$ in the REG method reduces the number of function evaluations required for convergence of the REG method when GN is performing better than REG.

We use the stopping criterion (3.42) with $\epsilon = 10^{-5}$ so the iterations stop once a stationary point of (4.81) is located. We also use the stopping criteria (3.46) and (3.47) to stop the iterations when little progress is being made on the function and iterate level respectively.

Figure 4.3 shows the proportion of $n_r = 1000$ ROSENprobs solved by each of the GN, LS and REG methods within a given number of function (l) and Jacobian (k_J) evaluations. From this figure, we see that GN converges to $\mathbf{x}^*$ within 6 evaluations, regardless of the choice of initial guess. We know from convergence theory that for zero residual problems, such as ROSENprob, the GN method should give quadratic convergence to a local minimum if the initial starting point is in some neighbourhood around the solution. This is demonstrated in these results where we see that, for our choices of initial guesses, there is no benefit from using a globalisation strategy as the GN method converges to a solution in the least number of function and Jacobian evaluations, with the LS and REG methods giving no improvement

on the GN method. However, we can use this example to understand how the choice of the globalisation parameters affects the convergence rate of LS and REG.

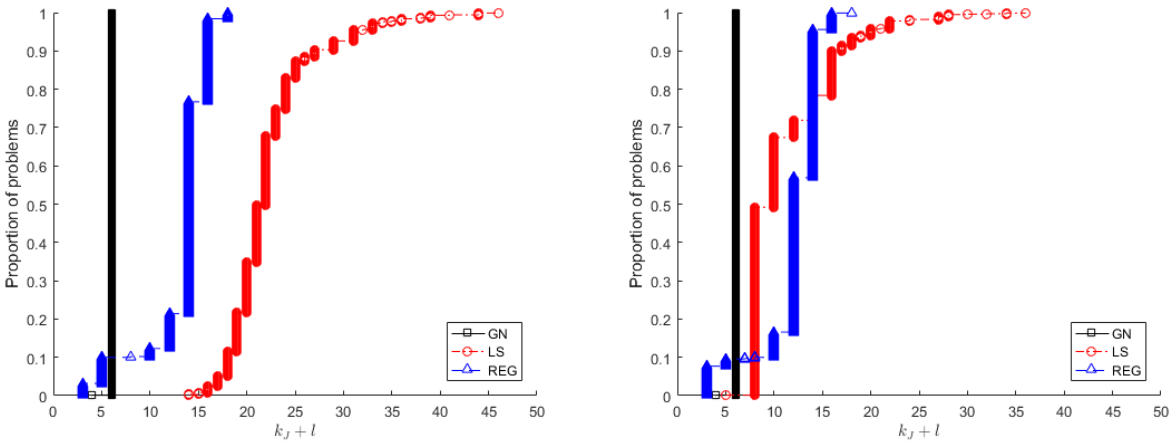
Figure 4.3(a) shows the results for the case where we choose $\alpha_0 = 1.5$ and $\gamma^{(0)} = 1$. REG appears to be the second best method with up to 18 evaluations needed for convergence, with LS being the most costly method, requiring up to 46 evaluations for convergence. Ideally, when applied to a problem where GN performs well, we would like LS and REG to perform as GN to benefit from its fast convergence property near a local minimum. LS is only able to give quadratic convergence to a local minimum if the Newton step is taken, that is when $\alpha^{(k)} = 1$ (see [96]) and this will never occur if $\alpha_0 = 1.5$ as in Figure 4.3(a). Hence, the line search strategy is obstructing the method from converging at a faster rate due to the value of the stepsize $\alpha^{(k)}$. The regularisation parameter in REG is also inhibiting the method from quadratic convergence as we know from theory that the second-order term of the Hessian at the solution of a zero residual problem is zero, and the value of $\gamma^{(k)}$ is unable to shrink to zero (as required for faster convergence) in as few iterations as Gauss-Newton alone, hence the slower convergence rate of REG. We therefore would like $\alpha^{(k)}$ to be close to 1 and $\gamma^{(k)}$ to be close to zero.

We adjust the globalisation parameters slightly to see if this improves the convergence as we would expect. In Figure 4.3(b) we set $\alpha_0 = 2$ so that the LS method can take the GN step after one adjustment in the bArmijo loop, and $\gamma^{(0)} = 0.5$ so that the REG method can reduce the number of times it needs to shrink the regularisation parameter, therefore reducing the number of function evaluations required. We notice that indeed, this does improve the performance of LS and REG. LS requires a maximum of 36 evaluations for convergence to $\mathbf{x}^*$, a reduction of 10 evaluations from when $\alpha_0 = 1.5$ is used. REG requires a maximum of 16 evaluations for convergence to $\mathbf{x}^*$ for 999 of the ROSENprobs, with 1 realisation still requiring the use of 18 evaluations. We see that almost half of the problems (492), LS only needs 8 evaluations for convergence. This is because the LS method is able to take the GN step after 1 adjustment in each bArmijo loop, unlike in Figure 4.3(a) where the choice of $\alpha_0 = 1.5$ and $\tau = 0.5$ would never allow for the GN step. For the REG method, we also see an overall reduction in the number of evaluations needed for convergence. This reduction is because the choice of $\gamma^{(0)} = 0.5$ requires less work by the REG method to reduce to close to 0 than $\gamma^{(0)} = 1$ did in Figure 4.3(a).

4.4.3 Conclusion

Within this section, we have studied the convergence behaviour of LS and REG in the case where GN performs poorly (using DSprob) and well (using ROSENprob). We found that the use of their globalisation strategies in LS and REG enables convergence to the solution of DSprob within relatively few function and Jacobian evaluations compared to GN, which appears to diverge in the same number of iterations.

When GN performs well, we show how adjusting the initial choices of the globalisation parameters improves the convergence rate of the LS and REG methods. For the LS method, we choose $\alpha_0 = 2$, so that LS can attain the GN step if needed, unlike the choice of $\alpha_0 = 1.5$.



(a) ROSENprob, $\alpha_0 = 1.5$, $\gamma^{(0)} = 1$.

(b) ROSENprob, $\alpha_0 = 2$, $\gamma^{(0)} = 0.5$.

Figure 4.3: Performance profiles showing the number of problems solved by each of the GN (black), LS (red) and REG (blue) methods within a given number of function and Jacobian evaluations when applied to ROSENprob with (a) $\alpha_0 = 1.5$ and $\gamma^{(0)} = 1$ and (b) $\alpha_0 = 2$ and $\gamma^{(0)} = 0.5$ for the $n_r = 1000$ random generations of $\mathbf{x}^{(0)}$.

For REG, we choose $\gamma^{(0)} = 0.5$ such that the REG parameter is closer to 0 than $\gamma^{(0)} = 1$, resulting in fewer adjustments by the REG updating strategy.

Considering the two cases, when GN performs well and when GN performs poorly, highlights some important points to consider when applying LS and REG a general problem. We ideally would like to benefit from GN's fast local convergence, but at the same time, we would like to benefit from LS and REG's global convergence property and faster convergence rate when GN performs poorly. To do so, we should choose an initial line search parameter that is not too small so as to benefit from global convergence when needed, but that can attain the value of the GN stepsize 1 through the use of the halving parameter $\tau = 0.5$. This is possible by choosing $\alpha_0 = 2^i$, where $i = 0, 1, 2, 3, \ldots$.

For REG, we can consider choosing the initial regularisation parameter according to the Hessian of the objective function we are aiming to minimise. Recall that for the zero residual problem, the second-order term of the Hessian is zero. We found that choosing a smaller regularisation parameter resulted in faster convergence to the minimum of the zero residual problem ROSENprob. For non-zero residual problems, such as those that occur in VarDA, using the Hessian of the variational problem (2.59) to gauge the appropriate size of $\gamma^{(0)}$ may be helpful for REG and when GN is performing poorly, choosing a larger initial regularisation parameter $\gamma^{(0)}$ may force the REG method to behave differently to the GN method. We investigate this in relation to the variational problem in Section 4.5.

In summary, we know from theory that there is no need for the use of a globalisation strategy in the zero residual case when starting near the solution as GN is quickly locally convergent

for zero residual problems. However, we have shown that LS and REG can be made to perform closer to GN when GN performs well (such as in the zero or small residual case) by using a combination of good initial choices of their globalisation parameters and suitable updating strategies to achieve the GN step. Furthermore, when GN is divergent, we are able to benefit from the global convergence advantage LS and REG have over GN.

In order to reliably measure the computational cost of each method, we compared the total number of function and Jacobian evaluations used by GN, LS and REG to see which method gives the most accurate solution in the given amount of computational cost. In VarDA practice, there is limited computational cost and time. If we limit the number of evaluations, some methods may perform better than others. Therefore, for the 3D-Var experiments in the remaining sections, we consider fixing the maximum number of function and Jacobian evaluations allowed to see which method gives the most accurate solution in the given amount of computational effort.

In the remainder of this chapter, we derive some theory to explain how the REG parameter interacts with the variational problem and test our findings by applying REG, along with GN and LS for comparison, to the standard 3D-Var problem outlined in Section 2.2.

4.5 Theoretical understanding of REG for VarDA

Recall from Section 4.3, the REG method requires its regularisation parameter to be sufficiently large to obtain global convergence. The work of Mandel et al [83] also found that the choice of regularisation parameter impacts the global convergence of the Levenberg-Marquardt regularisation method. In this section, we study the interaction between the REG term and the VarDA problem to derive a choice of the regularisation parameter that speeds up convergence of REG and thus is suitable for application to the variational problem where there is limited time and computational cost available.

The regularisation parameter $\gamma^{(k)}$ plays an important role when solving for $\mathbf{s}^{(k)}$ in the REG step equation (3.23). A different choice of $\gamma^{(k)}$ alters not only the step direction, but also the step length away from the GN step and may require the REG method to need more computational cost than available in operational DA for guaranteed convergence. In our work, we consider an appropriate choice of the REG parameter for application to the variational problem. We use the 4D-Var formulation as in Chapter 2, again noting that the 3D-Var formulation can be derived from this, and study the interaction between the REG term and the VarDA Hessian (2.59) in the REG step equation (3.23).

We begin by considering the interaction between the REG term and the inverse of the background error covariance matrix for the standard 4D-Var problem in the following section. We then consider the preconditioned 4D-Var problem and the effect the REG term has on the conditioning of the problem, before testing the choice for the standard VarDA problem using 3D-Var numerical experiments in the remainder of this chapter.

4.5.1 Standard VarDA

We first consider how the REG term interacts with the standard 4D-Var problem. By substituting the 4D-Var residual vector and its Jacobian from (2.53) into the LHS of the REG step equation (3.23), we have

$$\left((\mathbf{B}^{-1}+\gamma^{(k)})\mathbf{I}+\sum_{i=0}^{N}\mathbf{M}_{0,i}^{T}\mathbf{H}_{i}^{T}\mathbf{R}_{\mathbf{i}}^{-\mathbf{1}}\mathbf{H}_{i}\mathbf{M}_{0,i}\right)\mathbf{s}^{(k)}=-(\mathbf{J}(\mathbf{x}^{(k)}))^{T}\mathbf{r}(\mathbf{x}^{(k)}). \tag{4.85}$$

We assume $\mathbf{B}=\sigma_b^2\mathbf{C}_B$ and $\mathbf{R}=\sigma_o^2\mathbf{I}_p$, where σ_b^2 and σ_o^2 are the background and observation error variances respectively and $\mathbf{C}_B$ is a correlation matrix. We can choose the initial REG parameter according to how differently we want REG to perform from GN. If GN performs well, we want the REG parameter to be small in (4.87) so that REG takes a similar step to GN and converges quickly. If GN performs poorly, we do not want REG to take the GN step so we can use the REG parameter to alter the GN step.

From (4.85), one can see that the REG parameter is changing (inflating) the diagonal entries of $\mathbf{B}^{-1}$. Therefore, choosing $\gamma^{(0)}$ according to the entries of $\mathbf{B}^{-1}$ may be an appropriate choice in the case where GN performs poorly and we want the REG step to differ from the GN step.

We first consider the case where there are no background error correlations (i.e. $\mathbf{C}_B=\mathbf{I}_n$). In this case, we can write (4.85) as follows

$$\left((\sigma_b^{-2}+\gamma^{(k)})\mathbf{I}+\sigma_o^{-2}\sum_{i=0}^{N}\mathbf{M}_{0,i}^{T}\mathbf{H}_{i}^{T}\mathbf{H}_{i}\mathbf{M}_{0,i}\right)\mathbf{s}^{(k)}=-(\mathbf{J}(\mathbf{x}^{(k)}))^{T}\mathbf{r}(\mathbf{x}^{(k)}). \tag{4.86}$$

From (4.86), one can see how the REG parameter interacts with the inverse of the background error variance. By choosing $\gamma^{(0)}$ close to zero in (4.86), we obtain the GN step (3.10). This is a good choice in the case where GN performs well as REG can benefit from the fast local convergence properties of GN. However, in the case when GN performs poorly, we want the REG parameter to be large enough so as to guarantee global convergence within the limited number of iterations available in VarDA practice. Therefore, for the uncorrelated background error case, choosing $\gamma^{(0)}$ relative to the order of magnitude of σ_b^{-2} may be a reasonable choice as this will shorten the REG step, unlike the standard choice of $\gamma^{(0)}=1$ which does not take into account any aspect of the variational problem. This can be seen simply by considering a one dimensional problem where $\mathbf{x}\in\mathbb{R}$ such that the REG step is given as follows

$$\mathbf{s}^{(k)}=-\frac{(\mathbf{J}(\mathbf{x}^{(k)}))^{T}\mathbf{r}(\mathbf{x}^{(k)})}{(\mathbf{J}(\mathbf{x}^{(k)}))^{T}\mathbf{J}(\mathbf{x}^{(k)})+\gamma^{(k)}}. \tag{4.87}$$

Now, increasing $\gamma^{(k)}$ in (4.87) would result in a decrease in the REG stepsize and a greater deviation away from the GN step. This is the effect we are looking for in the case when GN performs poorly. When $k=0$, $\mathbf{x}^{(k)}=\mathbf{x}^{b}$ in (4.87), so σ_b^{-2} only features in the denominator of (4.87). In the case where σ_b^{-2} is much larger than 1, the standard choice of $\gamma^{(0)}=1$ would make little change to $\mathbf{s}^{(k)}$. However, if $\gamma^{(0)}$ is chosen to be σ_b^{-2}, then we know that the REG

parameter will have as much as an influence on the step as the background error variance has.

In the case when there is a lot of uncertainty in the background information relative to the observations, the inverse of the background error variance would be small and the observation term may dominate the VarDA Hessian (2.59). As the background state vector is used as the initial guess for GN in VarDA practice, having a large error in the initial guess may result in GN failing to converge to a solution in the computational time and cost available in DA. In this case, it would be important to choose an initial REG parameter that is large enough relative to the inverse background error variance. This would ensure that the REG step differs from the GN step and will limit the number of function evaluations required by REG to increase the REG parameter.

In the case where there are background error correlations, we want to choose an initial REG parameter that accounts for these. Recall from Section 2.2.2, due to its size $\mathbf{B}^{-1}$ is not calculated explicitly in VarDA [4]. The Met Office model their background error correlations using the second-order auto-regressive distribution (see [74]), as outlined for use in our numerical experiments in Section 4.6. To account for the use of background error correlations in (4.85), we propose the choice of $\gamma^{(0)} = \|\mathbf{B}^{-1}\| = \sigma_b^{-2}\|\mathbf{C}_B^{-1}\|$, where $\|\mathbf{C}_B^{-1}\|$ acts as a measure of the magnitude of the inverse background error correlations.

We propose that choosing $\gamma^{(0)} = \sigma_b^{-2}$ in REG may be a suitable choice for use on the standard VarDA problem in the case where there are no background error correlations. When background error correlations are included, an appropriate choice that accounts for the correlations may be $\gamma^{(0)} = \sigma_b^{-2}\|\mathbf{C}_b^{-1}\|$. We study the effect these choices have on the convergence of the REG method in Section 4.7.

We next study the interaction of the REG term with the preconditioned 4D-Var problem.

4.5.2 Preconditioned VarDA

By substituting the preconditioned 4D-Var Hessian (2.70) into the LHS of the REG step equation (3.23), we have

$$\left((1+\gamma^{(k)})\mathbf{I} + \sum_{i=0}^{N} \mathbf{B}^{1/2}\mathbf{M}_{0,i}^T\mathbf{H}_i^T\mathbf{R}_i^{-1}\mathbf{H}_i\mathbf{M}_{0,i}\mathbf{B}^{1/2}\right)\mathbf{s}^{(k)} = -(\mathbf{J}(\mathbf{v}^{(k)}))^T\mathbf{r}(\mathbf{v}^{(k)}), \tag{4.88}$$

where $\mathbf{v}$ is the control variable.

We omit the reduced resolution hat notation in (4.88) to consider the full resolution inner loop problem only. We again consider diagonal error covariance matrices of the form $\mathbf{B} = \sigma_b^2\mathbf{I}_n$ and $\mathbf{R}_i = \sigma_o^2\mathbf{I}_p$. Substituting these choices into (4.88), we have

$$\left((1+\gamma^{(k)})\mathbf{I} + \frac{\sigma_b^2}{\sigma_o^2}\sum_{i=0}^{N} \mathbf{M}_{0,i}^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{M}_{0,i}\right)\mathbf{s}^{(k)} = -(\mathbf{J}(\mathbf{v}^{(k)}))^T\mathbf{r}(\mathbf{v}^{(k)}). \tag{4.89}$$

Notice that if the covariance matrices $\mathbf{B}$ and $\mathbf{R}_i$ are diagonal with the same variance for all variables, these choices result in identical iteration methods for the REG method applied to the standard and preconditioned VarDA problem. That is, the REG step for the standard VarDA problem with $\gamma^{(0)} = \sigma_b^{-2}$ is equivalent to the REG step for the preconditioned VarDA problem with $\gamma^{(0)} = 1$ in the case where $\mathbf{B} = \sigma_b^2\mathbf{I}_n$ and $\mathbf{R}_i = \sigma_o^2\mathbf{I}_p$. This is because the preconditioned VarDA equations are equivalent to the standard VarDA equations multiplied through by σ_b^2 and we have that

$$\gamma_p^{(0)} = \sigma_b^2\gamma^{(0)}, \tag{4.90}$$

where $\gamma_p^{(0)}$ denotes the initial REG parameter for the preconditioned problem. Therefore, in the preconditioned case we can see that the choice of $\gamma^{(0)} = 1$ may be a suitable choice, although this does not account for the effect of the model. We discuss an alternative choice in relation to the global convergence property of REG, outlined in Theorem 4.3.5, later in this section.

Although in our work we solve the 4D-Var inner loop problem exactly, recall from Section 2.2.1 that the condition number of the Hessian $\kappa(\nabla^2\mathcal{J}(\mathbf{x}_0))$ can be used to indicate the accuracy we could be able to achieve when solving the linear minimisation problems in VarDA. In this section, we derive an expression for the condition number of the preconditioned 4D-Var Hessian (2.70) and suggest an alternative way to choose the initial REG parameter and update the REG parameter when applying REG to the preconditioned 4D-Var problem (2.69).

Bounds on the condition number of the preconditioned VarDA Hessian are derived in the work of [56], [57] and [123] for the incremental inner loop problem (GN subproblem). In the following theorem, we derive an expression for the condition number of the regularised preconditioned 4D-Var Hessian (2.70), which shows, under certain simplifying assumptions, how the REG term interacts with the preconditioned 4D-Var Hessian.

Theorem 4.5.1 (Condition number of regularised preconditioned 4D-Var Hessian)**.** *Let $\nabla^2\mathcal{J}_p$ denote the preconditioned 4D-Var Hessian as defined in* (2.70) *with $\mathbf{B} = \sigma_b^2\mathbf{I}$ and $\mathbf{R} = \sigma_o^2\mathbf{I}$, where σ_b^2 and σ_o^2 are the background and observation error variances respectively. Furthermore, assume the linearised observation operator $\mathbf{H}$ is low rank and that there is a single observation at the end of the assimilation time-window. Then the condition number of the regularised preconditioned 4D-Var Hessian is given by,*

$$\kappa\left(\nabla^2\mathcal{J}(\mathbf{x}_0) + \gamma^{(k)}\mathbf{I}\right) = \frac{1 + \gamma^{(k)} + \frac{\sigma_b^2}{\sigma_o^2}\lambda_{\max}\left(\mathbf{M}_{0,N}^T\mathbf{H}^T\mathbf{H}\mathbf{M}_{0,N}\right)}{1 + \gamma^{(k)}}, \tag{4.91}$$

where $\gamma^{(k)}$ is the REG parameter at the k^{th} iterate of Algorithm 3.3.2.

Proof. The perturbed preconditioned 4D-Var Hessian used in the REG step equation (3.23) is given by

$$\nabla^2\mathcal{J}(\mathbf{x}_0) + \gamma^{(k)}\mathbf{I}. \tag{4.92}$$

As (4.92) is a symmetric positive definite matrix, it follows that,

$$\kappa(\nabla^2\hat{\mathcal{J}}_p) = \frac{\lambda_{\max}(\nabla^2\hat{\mathcal{J}}_p + \gamma^{(k)}\mathbf{I})}{\lambda_{\min}(\nabla^2\hat{\mathcal{J}}_p + \gamma^{(k)}\mathbf{I})}. \tag{4.93}$$

As $\mathbf{H}$ is low rank, the minimum and maximum eigenvalues of (4.92) are given by

$$\lambda_{\min}\left(\nabla^2\mathcal{J}(\mathbf{x}_0)+\gamma^{(k)}\mathbf{I}\right)=1+\gamma^{(k)} \tag{4.94}$$

and

$$\lambda_{\max}\left(\nabla^2\mathcal{J}(\mathbf{x}_0)+\gamma^{(k)}\mathbf{I}\right)=1+\gamma^{(k)}+\frac{\sigma_b^2}{\sigma_o^2}\lambda_{\max}\left(\mathbf{M}_{0,N}^T\mathbf{H}^T\mathbf{H}\mathbf{M}_{0,N}\right), \tag{4.95}$$

respectively. Substituting (4.94) and (4.95) into (4.93), we have

$$\kappa\left(\nabla^2\mathcal{J}(\mathbf{x}_0)+\gamma^{(k)}\mathbf{I}\right)=\frac{1+\gamma^{(k)}+\frac{\sigma_b^2}{\sigma_o^2}\lambda_{\max}\left(\mathbf{M}_{0,N}^T\mathbf{H}^T\mathbf{H}\mathbf{M}_{0,N}\right)}{1+\gamma^{(k)}}, \tag{4.96}$$

as required. □

We have so far shown how the REG term interacts with the condition number of the preconditioned 4D-Var Hessian (2.70). Next, we discuss how the use of the maximum eigenvalue of the 4D-Var Hessian may be an appropriate choice for the REG parameter.

We saw in the proof of convergence of the REG method in Section 4.3 that if $\gamma^{(k)} \geq L$ where L is the Lipschitz constant of the gradient $\nabla\mathcal{J}$, then k is a very successful iteration. Furthermore, from Lemma 2.1.7, we know that a Lipschitz continuous gradient implies a bounded Hessian. By the equivalence of the 2-norm and the maximum eigenvalue, a bounded Hessian is equivalent to the eigenvalues of the Hessian being bounded. Therefore, assuming the second-order terms of the Hessian (2.32) that are neglected in the preconditioned 4D-Var Hessian (2.70) are close to zero, we may be able to improve the performance of the REG method by choosing our regularisation parameter to be the maximum eigenvalue of the preconditioned 4D-Var Hessian (2.70). That is, if we set $\gamma^{(k)}=\lambda_{\max}\left(\nabla^2\mathcal{J}(\mathbf{x}_0^{(k)})\right)$ then we have

$$\gamma^{(k)}=1+\frac{\sigma_b^2}{\sigma_o^2}\lambda_{\max}\left(\mathbf{M}_{0,N}^T\mathbf{H}^T\mathbf{H}\mathbf{M}_{0,N}\right) \tag{4.97}$$

and by Lemma 4.3.3, the REG iterate is very successful and we should not have any unsuccessful iterations. Note that by knowing some information about the second-order terms of the Hessian (2.32), we are able to remove the assumption that these are close to zero and choose a more suitable choice of $\gamma^{(k)}$.

In this section, we have shown how the REG term interacts with the condition number of the preconditioned 4D-Var Hessian (2.70) and suggested an alternative way to choose the initial REG parameter $\gamma^{(0)}$ and update $\gamma^{(k)}$ according to the maximum eigenvalue of the preconditioned 4D-Var Hessian (2.70).

With the aim of addressing research question RQ1(b), in Section 4.7 we conduct numerical experiments where we study the effects on the performance of GN, LS and REG when applied to the standard 3D-Var problem and when varying the background error variance (the initial guess for the minimisation). We also study the effects of using an alternative choice of initial REG parameter $\gamma^{(0)}$ on the convergence rate of REG. We first set out the experimental design for the 3D-Var numerical experiments where we apply GN, LS and REG to the standard 3D-Var problem with nonlinear observation operators.

4.6 Experimental design

Before evaluating the GN, LS and REG methods numerically, we first explain the experimental design.

From theory, we know that in the zero residual least-squares case, the rate of convergence for the GN method is quadratic. Furthermore, if the cost function is linear, the GN method will converge to a stationary point in one iteration as the descent direction points towards the minimum. Therefore, we do not consider zero residual nor linear 3D-Var problems.

Twin experiments are commonly used to test DA methods. They use synthetic observations as well as error statistics that satisfy the DA assumptions. Within this section, we define the choices made for the twin experimental design used, beginning with generating the reference state, $\mathbf{x}^{ref}$, which is used as the basis of a twin experiment in the definition of the background state (the initial guess for the optimisation algorithms) as well as to generate the observations.

4.6.1 Twin experiments

Reference state We choose the reference state $\mathbf{x}^{ref} \in \mathbb{R}^n$ to be the sine wave, with entries given by

$$\mathbf{x}_i^{ref} = \sin\left(\frac{2\pi}{n}(i-1)\right), \tag{4.98}$$

where $i = 1, 2, \ldots, n$ and we choose $n = 100$ for all our experiments. The background state vector is generated using (4.98) and is defined in the following.

Background In VarDA, the initial guess for the optimisation algorithm is taken to be the background state, $\mathbf{x}^b$, which incorporates information from previous forecasts. In our experiments, the background state vector $\mathbf{x}^b$ is generated by adding Gaussian noise

$$\varepsilon_{\mathbf{b}} \sim \mathcal{N}(0, \mathbf{B}), \tag{4.99}$$

to the reference state, $\mathbf{x}^{ref}$. In our experiments, we choose $\mathbf{B}$ to be of the form $\mathbf{B} = \sigma_b^2 \mathbf{C}_B$, where σ_b^2 is the background error variance and $\mathbf{C}_B$ is a correlation matrix. The standard deviations (SDs) of the errors from the reference solution are based on the average order of magnitude of the entries of $\mathbf{x}^{ref}$. In our work, $\sigma_b^2 = 4.0502 \times 10^{-5}, 0.0010, 0.0041, 0.0253$ and 0.1013 represent a $1\%, 5\%, 10\%$, 25% and 50% SD of the error respectively.

For the correlation structure, we choose $\mathbf{C}_B = \mathbf{I}_n$ or $\mathbf{C}_B$ to be a correlation matrix. In particular, the structure of the background correlations follow the SOAR (second-order auto-regressive) distribution with the error correlation matrix defined by

$$\mathbf{C}_B(i,j) = \left(1 + \frac{|2a\sin(\frac{\theta|i-j|}{2})|}{L}\right) \exp\left\{\frac{-|2a\sin(\frac{\theta|i-j|}{2})|}{L}\right\}, \tag{4.100}$$

where i and j correspond to the rows and the columns of the correlation matrix respectively, $\theta = \frac{2\pi}{n}$ is the angle between two points on the circle, a is the radius of the circle, L is the correlation length-scale and the chordal distance is defined as $d = 2a\sin(\frac{\theta|i-j|}{2})$. The SOAR distribution has longer tails versus the Gaussian distribution and results in better conditioning of the problem [56]. Furthermore, the UK Met Office system uses the SOAR function to model horizontal correlations [74].

We choose $a = 1/2\pi$ such that the circumference of the circle is 1. The length scale of the correlation matrix controls the spread of errors along the grid. Increasing the correlation length scale results in an increase in the number of nearby background errors that are correlated with each other [55]. The VarDA system is sensitive to the choice of length scale L. Haben et al. [57] showed that as the length scale of the background error correlations increase, the condition number of the preconditioned 4D-Var Hessian increases. For the correlation length-scales used in our work, we choose $L_1 = 0.5\Delta x$, $L_2 = \Delta x$ and $L_3 = 1.5\Delta x$, where $\Delta x = 1/n$. The background error covariance matrix with associated correlation length scale L_i is denoted by $\mathbf{B}_i = \sigma_b^2\mathbf{C}_B$, where $i = 1, 2, 3$. We denote the case where $\mathbf{C}_B = \mathbf{I}_n$ as $\mathbf{B}_0$.

As previously mentioned, we generate synthetic observations using the reference state, $\mathbf{x}^{ref}$. We next describe the choices we make when specifying these observations.

Observations We consider two 3D-Var problems, 3DVarProb1 and 3DVarProb2, with different nonlinear observation operators with $p = n/2$ observations. 3DVarProb1 has observations of the first half of the spatial domain with the observation operator given by

$$\mathcal{H}(\mathbf{x}) = \begin{pmatrix} x_1^2\sin(x_{n/2}) \\ x_2^2\sin(x_1) \\ x_3^2\sin(x_2) \\ \vdots \\ x_{n/2}^2\sin(x_{n/2-1}) \end{pmatrix}. \tag{4.101}$$

3DVarProb2 has observations at the even grid-points with the observation operator given by

$$\mathcal{H}(\mathbf{x}) = \begin{pmatrix} x_2^2\sin(x_n) \\ x_4^2\sin(x_2) \\ \vdots \\ \vdots \\ x_n^2\sin(x_{n-2}) \end{pmatrix}. \tag{4.102}$$

These choices of nonlinear observation operators of different structures allow us to test if our results hold for different spatial locations of observations, and in the presence of nonlinearities. We assume that for both problems, $\mathcal{H}$ is the exact observation operator used to map to observation space. We use imperfect observations where the observations, $\mathbf{y}$, are generated by adding Gaussian noise

$$\varepsilon_{\mathbf{o}} \sim \mathcal{N}(0, \mathbf{R}), \tag{4.103}$$

to $\mathcal{H}(\mathbf{x}^{ref})$. For the observation error covariance matrix we choose $\mathbf{R}$ to be a diagonal matrix of the form $\mathbf{R} = \sigma_o^2 \mathbf{I}_p$, where σ_o^2 is the observation error variance. For all experiments, we set the standard deviation of the observation error to be 5% of the average order of magnitude of the entries of $\mathcal{H}(\mathbf{x}^{ref})$. For 3DVarProb1, this is $\sigma_o^2 = 3.4137 \times 10^{-4}$ and for 3DVarProb2, this is $\sigma_o^2 = 3.3765 \times 10^{-4}$. We vary the background error variance σ_b^2 above and below σ_o^2. This can be thought of as having more confidence in the observations compared to background when $\sigma_b > \sigma_o$ and vice versa. Furthermore, as the initial guess is set to be the background state vector, which is dependent on the value of σ_b, by varying σ_b^2 we are essentially varying the initial guess of the algorithms, thus eliminating starting point bias from our results [7]. It is important to recall from Section 3.2.3 that under certain conditions, the GN method is known for its fast convergence properties when in close vicinity to a local minimum. By choosing a small value of σ_b^2, we expect the performance of GN to beat that of both LS and REG as it does not require the adjustment of the additional parameters $\alpha^{(k)}$ and $\gamma^{(k)}$. The effect this has on the convergence of the optimisation methods will be investigated. We next outline the algorithmic choices we have made.

4.6.2 Algorithmic choices

Stopping criteria We now outline the criteria used to terminate Algorithms 3.2.3, 3.3.1 and 3.3.2. Due to the limited time and computational cost available in VarDA practice, the GN method is not necessarily run to convergence and a stopping criterion is used to limit the number of iterations. Each residual vector calculation requires the use of the nonlinear observation operator to map from state to observation space. This can then be used to calculate the value of the objective function. Furthermore, each Jacobian (of the residual vector) matrix calculation requires the use of the tangent linear observation operator. This, along with the residual vector, can then be used to calculate the value of the gradient.

To reduce computational cost in practical implementations of VarDA, the Jacobian matrix is evaluated at a lower resolution than the objective function (2.52) when solving the inner loop problem [40]. However, as the dimension of the problems used within our work are relatively small compared to DA systems in practice, we use the full resolution residual and Jacobian given in (2.53) and solve the inner loop problem using MATLAB's backslash operator where an appropriate solver is chosen according to the properties of the Hessian matrix $\nabla^2 \mathcal{J}(\mathbf{x})$ (see [86] for more details). The limit on the total number of function and Jacobian evaluations is achieved by using the criterion (3.45), where τ_e is specified for each experiment.

To ensure that the algorithms are stopped before the function values stagnate, we use (3.48) with $\epsilon = 10^{-5}$. We also use (3.42) with $\epsilon = 10^{-5}$ to identify whether a given optimisation method has located a stationary point of the 3D-Var cost function.

Parameter choices We saw in Section 4.4 how the initial parameter choices for LS and REG affect the convergence of the methods. We want to balance the need from the fast local convergence guarantee of GN as well as the global convergence guarantees of LS and REG. For this reason, we choose $\alpha_0 = 1$ for the LS method so that the first step assessed by the bArmijo rule is the GN step. We set $\beta = 0.1$ and to adjust the step length, $\tau = 0.5$ so that

the step can be adjusted for global convergence. For the REG method, we select the typical choice of the initial regularisation parameter, $\gamma^{(0)} = 1$ unless indicated otherwise, as well as $\eta_1 = 0.1$ and $\eta_2 = 0.9$ to assess how well the model (3.25) approximates the true function value at the next iteration. These choices ensure that both the LS and REG methods are able to take steps that are close to the GN step in the case that GN performs well, but can be altered in the case where the GN step is a poor choice.

For all three optimisation methods, we set $\tau_e = 8, 100$ or 500 depending on the experiment. The choice of $\tau_e = 8$ comes from that which is used operationally in the ECMWF Integrated Forecasting System [34], whereas the choice of $\tau_e = 100$ or 500 is used to measure the performance of the optimisation methods when closer to convergence.

To ensure the robustness of our results, we apply the three optimisation methods to a series of n_r randomly generated problems, where the randomness occurs through the background and observation error vectors, $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$. For each realisation, a new $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$ are generated from their respective distributions, (4.99) and (4.103). We choose $n_r = 100$. To present our results, we use accuracy profiling described in Section 3.5.1.

4.7 Numerical results

In this section, we apply the GN, LS and REG to two nonlinear non-zero residual standard 3D-Var problems. We consider the following.

- The effects of initialising the GN, LS and REG with the background, where the amount of uncertainty in the background information is increased whilst the amount of uncertainty in the observations is fixed. In this part, we consider diagonal background error covariance matrix structures, but note that similar results are obtained using correlated errors.

- The effects of using an alternative choice of initial REG parameter $\gamma^{(0)}$ on the convergence rate of REG. More specifically, we analyse how the choice of $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ that we proposed in Section 4.5, impacts the convergence behaviour of REG.
  We consider different background error correlation structures and use accuracy profiles to show how the methods perform within a limited number of cost function and Jacobian evaluations (as in practice) and when more evaluations are allowed than in practice. We also compare the accuracy of the analysis when using each of the methods through the use of RMSE profiles.

In the following section, we consider the first point.

4.7.1 Effects of background error

Tables 4.2 and 4.3 show the algorithmic output for each of the GN, LS and REG methods for the five different choices of background error standard deviations when applied to 3DVarProb1 and 3DVarProb2, respectively. In these experiments, we consider uncorrelated

background error only, although similar results are obtained when correlated background error is used.

For the first five rows of these tables, we are interested in seeing whether the methods are able to locate a stationary point of the 3D-Var cost function, not necessarily within the number of iterations allowed in practice. We use two stopping criteria to achieve this; (3.42) with $\epsilon = 10^{-5}$ to terminate the algorithms upon finding a stationary point and (3.45) with $\tau_e = 500$ to limit unnecessary function and Jacobian evaluations.

For row 6 of each table, we are interested in filtering out the unnecessary function evaluations for the 50% case. We therefore terminate the methods when there is no longer a sufficient reduction in the value of the 3D-Var cost function at each iteration, as opposed to iterating until the gradient norms are close to zero as in the previous row. We therefore also use (3.48) with $\epsilon = 10^{-5}$ to terminate the methods when there is little change between iterations on the cost function level.

The first thing to notice from Tables 4.2 and 4.3 is that as the amount of uncertainty in the background increases, the number of function and Jacobian evaluations required to locate a stationary point of the 3D-Var cost function also increases. This is expected as in VarDA, the background state vector is used for the initial guess for the minimisation. Therefore, increasing the amount of uncertainty in the background would simultaneously worsen the accuracy of the initial guess, resulting in greater difficulty in locating a local minimum of the 3D-Var cost function.

In both Tables 4.2 and 4.3, GN, LS and REG perform almost identically for the 1%, 5%, 10% cases. The slight difference in the output figures between REG and the other two methods is to be expected given that the REG parameter alters the GN step in the REG method, whereas the GN and LS steps are the same as $\alpha^{(k)} = 1$. The presence of the globalisation parameters appears to be redundant in these cases as GN is performing well.

For the 25% case, the performance of GN, LS and REG differs significantly. For 3DVarProb1, GN does not converge to a stationary point of the 3D-Var cost function within the 500 evaluations allowed. In fact, GN appears to be diverging in this case as the gradient norms and the change in the iterates fluctuate across the iterations. GN is finally terminated when the maximum number of evaluations is reached (when criterion (3.45) is satisfied), where the gradient norm is 1.9468×10^4 and the change in the iterates is 0.9067, both of which are far from zero. The function value 36.7439 located by LS and REG is much smaller than the function value GN locates after 500 evaluations of 2.0797×10^3, showing the clear benefit of the globally convergent strategies. LS requires only 49 evaluations to converge to the same function value as REG, which requires 149 evaluations for convergence. The use of the LS parameter to adjust the length of GN step appears to be the best choice in this experiment. We see a similar pattern for 3DVarProb2.

For the 3DVarProb2 25% case, all three methods converge to the same function value. GN requires 416 function and Jacobian evaluations to converge to a stationary point of the 3D-Var

cost function. This is many more than the 40 and 106 evaluations required for convergence of LS and REG respectively. Again, for the 25% case, LS appears to outperform GN and REG. We next look at what happens when we further increase the level of background error in the 50% case.

For the 3DVarProb1 50% case, we see that GN is the worst method in terms of minimising the cost function, norm of the change in the iterates and norm of the gradient and only terminates once the maximum number of evaluations is met. For the same case, LS and REG are both satisfying the norm of gradient criterion. However, the LS method is stopping at a much larger value of the objective function than REG. Recall from Chapter 3, LS and REG are only globally convergent to a stationary point and not necessarily to a global minimum. Therefore, in the presence of multiple local minima, such as in the variational problem where the observation operator is nonlinear, LS and REG may take steps that do not allow them to converge to the same local minimum, as seen in our results. From a computational cost perspective, although the total number of evaluations required by LS and REG are comparable (185 and 180 respectively), REG requires 23 fewer function evaluations but still requires 18 more Jacobian evaluations than LS. This indicates that the LS parameter requires more updates than the REG parameter for a successful iteration. However, one should note that LS has a stricter requirement for the function decrease; LS requires that the function must strictly decrease at each iteration, whereas REG only requires a monotonic decrease. That being said, despite requiring less evaluations than LS, REG manages to converge to a much smaller function value of 84.0097, as opposed to LS, which converges to 455.5152. So REG does a better job at minimising the 3D-Var cost function within fewer evaluations.

For the 3DVarProb2 50% case, LS locates the smallest function value out of the three methods. LS requires 388 evaluations to converge to a function value of 1.0884×10^3 while REG requires 389 to converge to a larger function value of 1.1205×10^3. As with 3DVarProb1, LS again requires many more (102) function evaluations than REG to achieve convergence on the gradient norm level and GN is unable to locate a minimum within the 500 evaluations allowed. In fact, looking at the function value convergence plots (omitted here), GN appears to be diverging until beyond $k = 20,000$ iterations for both problems 3DVarProb1 and 3DVarProb2.

For the 50% case, we use the additional stopping criterion (3.48) in row 6 of Tables 4.2 and 4.3 to stop the methods when little progress is made on the function level. For both 3DVarProb1 and 3DVarProb2, GN does not terminate based on this criterion, indicating that there is significant change on the function level throughout the 250 iterations. The use of this criterion has significantly reduced the number of evaluations required for LS and REG to locate a similar function value to that when using (3.42) alone.

In this section, we saw that when there is high uncertainty in the background state vector, which is used as the initial guess for the minimisation, GN fails to converge to a stationary point of the 3D-Var cost function and in fact, may diverge, while LS and REG are able to converge. In the following section, we see if we can improve the performance of REG when applied to 3DVarProb1 and 3DVarProb2 through the use of alternative choices of $\gamma^{(0)}$.

Table 4.2: Table of algorithmic output when applying GN, LS and REG to a typical realisation of 3DVarProb1 where l and k_J are the number of cost function and Jacobian evaluations respectively required to satisfy the gradient norm stopping criterion (3.42) or the relative function value stopping criterion (3.48) for a given level of background error SD, as indicated in the first column.

Case	Method	l	k_J	$\mathcal{J}(\mathbf{x}^{(k_J)})$	$\|\mathbf{x}^{(k_J)} - \mathbf{x}^{(k_J-1)}\|$	$\|\nabla\mathcal{J}(\mathbf{x}^{(k_J)})\|$
1% (3.42)	GN	5	5	29.7517	6.1003×10^{-9}	1.6597×10^{-6}
	LS	5	5	29.7517	6.1003×10^{-9}	1.6597×10^{-6}
	REG	5	5	29.7517	6.0734×10^{-9}	1.6511×10^{-6}
5% (3.42)	GN	7	7	30.9072	2.4659×10^{-8}	2.7150×10^{-6}
	LS	7	7	30.9072	2.4659×10^{-8}	2.7150×10^{-6}
	REG	7	7	30.9072	2.3804×10^{-8}	2.5964×10^{-6}
10% (3.42)	GN	12	12	33.5679	1.1061×10^{-7}	8.1209×10^{-6}
	LS	12	12	33.5679	1.1061×10^{-7}	8.1209×10^{-6}
	REG	12	12	33.5679	1.0404×10^{-7}	7.6124×10^{-6}
25% (3.42)	GN	250	250	2.0797×10^{3}	0.9067	1.9468×10^{4}
	LS	27	22	36.7439	2.5855×10^{-7}	5.3382×10^{-6}
	REG	78	71	36.7439	1.6513×10^{-7}	9.3267×10^{-6}
50% (3.42)	GN	250	250	546.1879	0.9893	135.8018
	LS	125	60	455.5152	3.2204×10^{-7}	9.2902×10^{-6}
	REG	102	78	84.0097	7.4470×10^{-8}	9.3953×10^{-6}
50% (3.48)	GN	250	250	546.1879	0.9893	135.8018
	LS	51	22	455.5153	0.0110	1.4272
	REG	28	19	84.0112	0.0119	1.4645

4.7.2 Effects of initial REG parameter

Tables 4.4 and 4.5 show the algorithmic output when applying REG to 3DVarProb1 and 3DVarProb2 for different choices of the initial REG parameter $\gamma^{(0)}$, where the background error standard deviation is 50% such that $\|\mathbf{B}^{-1}\| = \sigma_b^{-2} = 9.8761$. In Section 4.7.1, we saw that GN failed to converge in the 50% case, so we want to choose the initial REG parameter such that the REG step differs significantly from the GN step. We therefore consider choices of $\gamma^{(0)}$ of different magnitudes from 1 to 1000. We also consider the choice $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$, as proposed in Section 4.5. We use the stopping criterion (3.42) with $\epsilon = 10^{-5}$ to terminate the algorithms upon finding a stationary point.

From Table 4.4, we see that for 3DVarProb1 $\gamma^{(0)} = 2$ is the worst choice out of the six choices of $\gamma^{(0)}$ that we consider. In this case, the REG parameter requires a total of 253 evaluations to converge to a stationary point of the 3D-Var cost function, this is an increase in the total number of function and Jacobian evaluations from the $\gamma^{(0)} = 1$ case that located the same stationary point. For 3DVarProb2, Table 4.5 shows that the $\gamma^{(0)} = 2$ case is only slightly better than the $\gamma^{(0)} = 1$ case as it requires only 1 less function evaluation.

The choice of $\gamma^{(0)} = 10$ yields a vast improvement for both 3DVarProb1 and 3DVarProb2.

Table 4.3: Table of algorithmic output when applying, GN, LS and REG to a typical realisation of 3DVarProb2 where l and k_J are the number of cost function and Jacobian evaluations respectively required to satisfy the gradient norm stopping criterion (3.42) or the relative function value stopping criterion (3.48) for a given level of background error SD, as indicated in the first column.

Case	Method	l	k_J	$\mathcal{J}(\mathbf{x}^{(k_J)})$	$\|\mathbf{x}^{(k_J)} - \mathbf{x}^{(k_J-1)}\|$	$\|\nabla\mathcal{J}(\mathbf{x}^{(k_J)})\|$
1% (3.42)	GN	5	5	33.5763	6.0733×10^{-9}	1.4958×10^{-6}
	LS	5	5	33.5763	6.0733×10^{-9}	1.4958×10^{-6}
	REG	5	5	33.5763	6.1085×10^{-9}	1.5057×10^{-6}
5% (3.42)	GN	8	8	38.1867	1.4127×10^{-8}	1.9357×10^{-6}
	LS	8	8	38.1867	1.4127×10^{-8}	1.9357×10^{-6}
	REG	8	8	38.1867	1.4352×10^{-8}	1.9666×10^{-6}
10% (3.42)	GN	11	11	37.5382	5.7913×10^{-8}	3.6038×10^{-6}
	LS	11	11	37.5382	5.7913×10^{-8}	3.6038×10^{-6}
	REG	11	11	37.5382	5.5028×10^{-8}	3.4174×10^{-6}
25% (3.42)	GN	208	208	37.4526	1.3379×10^{-7}	9.4260×10^{-6}
	LS	22	18	37.4526	4.3197×10^{-7}	6.0895×10^{-6}
	REG	56	50	37.4526	1.3245×10^{-7}	8.3094×10^{-6}
50% (3.42)	GN	250	250	2.5678×10^{3}	1.3443	4.5469×10^{3}
	LS	325	63	1.0884×10^{3}	3.9145×10^{-8}	6.5852×10^{-6}
	REG	223	166	1.1204×10^{3}	1.4441×10^{-7}	6.6316×10^{-6}
50% (3.48)	GN	250	250	2.5678×10^{3}	1.3443	4.5469×10^{3}
	LS	67	22	1.0884×10^{3}	0.0354	13.3584
	REG	48	32	1.1205×10^{3}	0.0310	2.6457

The REG method locates a smaller function value than the $\gamma^{(0)} = 1$ and $\gamma^{(0)} = 2$ cases in fewer function and Jacobian evaluations. This is because, the smaller the REG parameter, the closer the REG step (3.23) is to the GN step (3.10). Therefore, in the case where GN performs poorly, such as in the cases we consider, allowing REG to take steps similar to GN will only result in slower global convergence as the REG parameter will need to be increased to achieve a reduction in the cost function value. If the REG parameter is initially chosen to be larger, the REG method is able to take steps that require no/fewer adjustments to guarantee a reduction in the cost function value. Furthermore, as the steps taken by the REG method differ according to the choice of REG parameter and the problems we consider are nonlinear, it is possible that REG may converge to a different stationary point given a different choice of $\gamma^{(0)}$ as by Theorem 4.3.5, REG is only guaranteed to converge to a stationary point, and not necessarily a local minimum.

For 3DVarProb1, $\gamma^{(0)} = 100$ is the best choice out of the six choices we consider as the REG method requires only 104 evaluations for convergence to a stationary point. By increasing $\gamma^{(0)}$ to 1000, the REG method requires only 4 more function evaluation and 1 less Jacobian evaluation to converge to the same function value as the $\gamma^{(0)} = 100$ case. Although the choice of $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ is not the best choice for REG when applied to 3DVarProb1, it still yields a vast improvement compared to the standard choice of $\gamma^{(0)} = 1$. It requires 25 less

function evaluations and 12 less Jacobian evaluations to converge to a smaller function value than the standard initial choice. Note that the performance for $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ is similar to that of the $\gamma^{(0)} = 10$ case as $\|\mathbf{B}^{-1}\| = 9.8761 \approx 10$.

For 3DVarProb2, the choice of $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ is the best choice out of the six choices we consider as the REG method requires only 156 evaluations for convergence to a significantly smaller function value than that which REG converges to after 489 evaluations when using the standard $\gamma^{(0)} = 1$ choice. The choices of $\gamma^{(0)} = 100$ and $\gamma^{(0)} = 1000$ also yield an improvement to the standard choice of initial REG parameter. However, in VarDA practice, we will not be able to know a priori which value of initial REG parameter will yield the best convergence results for REG. Therefore, the choice of $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ is both convenient as in practical applications of the standard 3D-Var problem we have some knowledge of the entries of $\mathbf{B}$, and our results show that the choice of $\gamma^{(0)}$ according to the entries of $\mathbf{B}$ yields better convergence results than the standard choice of $\gamma^{(0)} = 1$.

Table 4.4: Table of algorithmic output when applying REG to 3DVarProb1 for the 50% case of σ_b, where $\gamma^{(0)}$ is varied.

$\gamma^{(0)}$	l	k_J	$\mathcal{J}(\mathbf{x}^{(k_J)})$	$\|\mathbf{x}^{(k_J)} - \mathbf{x}^{(k_J-1)}\|$	$\|\nabla\mathcal{J}(\mathbf{x}^{(k_J)})\|$
1	102	78	84.0097	7.4470×10^{-8}	9.3953×10^{-6}
2	139	114	84.0097	5.8388×10^{-8}	8.9431×10^{-6}
10	75	64	38.4162	2.4535×10^{-7}	9.5638×10^{-6}
100	55	49	38.4162	2.7258×10^{-7}	9.0701×10^{-6}
1000	59	48	38.4162	2.3163×10^{-7}	6.4996×10^{-6}
$\|\mathbf{B}^{-1}\|$	77	66	38.4162	2.1688×10^{-7}	8.4473×10^{-6}

Table 4.5: Table of algorithmic output when applying REG to 3DVarProb2 for the 50% case of σ_b, where $\gamma^{(0)}$ is varied.

$\gamma^{(0)}$	l	k_J	$\mathcal{J}(\mathbf{x}^{(k_J)})$	$\|\mathbf{x}^{(k_J)} - \mathbf{x}^{(k_J-1)}\|$	$\|\nabla\mathcal{J}(\mathbf{x}^{(k_J)})\|$
1	223	166	1.1204×10^{3}	1.4441×10^{-7}	6.6316×10^{-6}
2	222	166	1.1204×10^{3}	1.4441×10^{-7}	6.6316×10^{-6}
10	85	76	36.9526	1.6385×10^{-7}	7.4808×10^{-6}
100	89	73	36.9526	1.7806×10^{-7}	7.9060×10^{-6}
1000	188	143	38.7421	1.4457×10^{-7}	6.4595×10^{-6}
$\|\mathbf{B}^{-1}\|$	82	74	36.9526	2.3062×10^{-7}	9.4980×10^{-6}

Our findings from Tables 4.4 and 4.5 suggest that the choice of $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ results in better performance of the REG method compared to the standard choice of $\gamma^{(0)} = 1$. We have so far only considered a single realisation of 3DVarProb1 and 3DVarProb2 when $\mathbf{B}$ is uncorrelated. To see if our results hold for a wider set of 3D-Var problems, we next take the two cases of REG; one using the standard choice of $\gamma^{(0)} = 1$ and one using $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$. For simplicity, we denote the REG method with $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ by REGB. We compare the

performance of GN, LS, REG and REGB through the use of accuracy and RMSE profiles where we consider $n_r = 100$ realisations and different choices of $\mathbf{B}$.

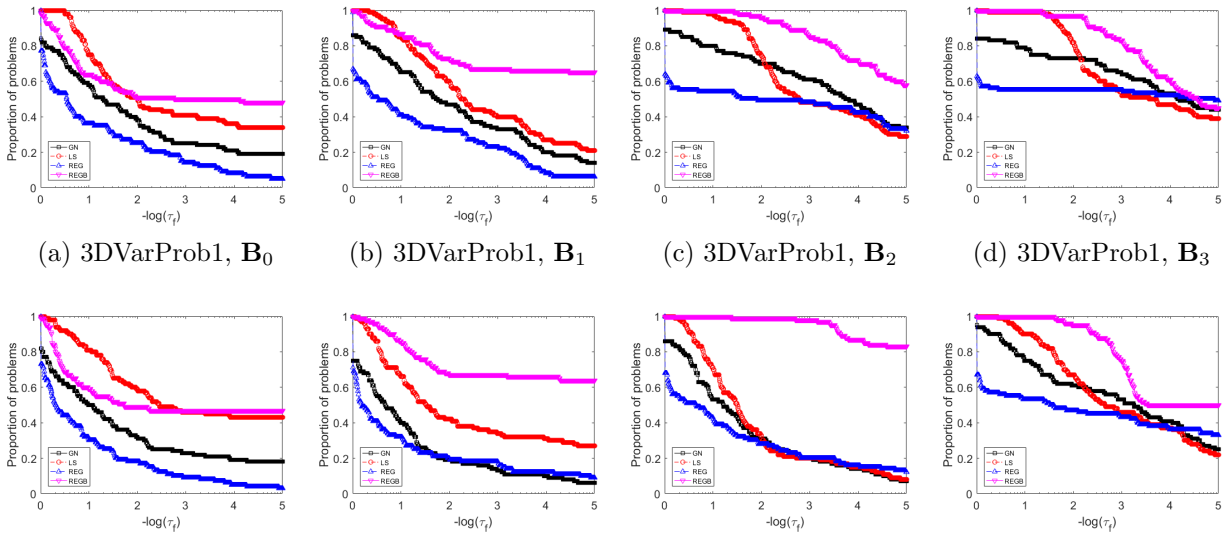
Figure 4.4 shows the accuracy profiles used to benchmark the performance of the GN, LS, REG and REGB methods as the tolerance τ_f is reduced, where $\tau_e = 8$, while Figure 4.5 allows τ_e to increase to $\tau_e = 100$ to understand how the methods perform both within a limited number of computations (similar to that which is available in VarDA practice) and where there is more evaluations than that which is available in practice. In addition to considering a diagonal background error covariance matrix $\mathbf{B}_0$, we also consider the three correlated background error covariance matrix choices, $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ to see if our findings hold for more realistic background error correlation matrix structures, where the correlation length scale is varied.

For the case where $\tau_e = 8$, Figure 4.4 shows that, for a given level of accuracy and for any choice of $\mathbf{B}$ that we consider, GN is able to solve more problems than REG. The performance of LS or REGB is, for almost all levels of accuracy, better than GN and REG. The only case where REG appears to perform marginally better than GN, LS and REGB is in Figure 4.4(d) where $\tau_f > 10^{-4.5}$. Figure 4.4 also shows that, for both 3DVarProb1 and 3DVarProb2 and for all choices of $\mathbf{B}$ that we consider, REGB is significantly better than REG in minimising the 3D-Var cost function. Through the use of a larger (compared to REG) initial REG parameter, REGB is able to locate a smaller function value that REG (and GN) struggle to locate. These results indicate that when there is limited computational cost available, the choice we proposed of $\gamma^{(0)} = \|\mathbf{B}^{-1}\|$ is better than the standard initial REG parameter choice. Overall, for the $\tau_e = 8$ case, REGB appears to be the most suitable method, with LS being the second best method.

Figure 4.5 shows a much greater difference between the globally convergent methods and GN than when fewer evaluations were allowed in the results of Figure 4.4. It appears that when more evaluations are allowed, GN performs considerably worse than the globally convergent methods. The globally convergent methods are able to locate much better estimates of the initial states for the 3D-Var problem than GN. We recall that this is what we saw in Tables 4.2 and 4.3 for the case when σ_b is large. Figure 4.5 also shows that the difference between LS and REG is marginal and that REGB, again, appears to be the favourable choice.

Recall from Section 2.2.2, it is known that the use of realistic background error statistics is important in VarDA as it has a profound impact on the analysis [4]. By increasing the correlation length scale, we are increasing the number of non-zero off diagonal entries of $\mathbf{B}$, thus allowing more observation information to spread across the spatial domain during the assimilation. Generally, Figure 4.5 shows that for both 3DVarProb1 and 3DVarProb2 where $\tau_e = 100$ is used, the performance of all four methods improves as stronger background error correlations are introduced. This improvement is seen more subtly in Figure 4.4, this is due to the limit on the number of evaluations, $\tau_e = 8$, used, which we expect to result in fewer problems being solved to a high accuracy. From Figure 4.4 we are still able to see that generally, the performance of all four methods improved as stronger background error correlations are introduced, as seen in the upward shift in the lines in each profile. Although

there are some cases, in particular, in Figures 4.4(d) and 4.4(h), where increasing the level of background error correlations results in REGB solving fewer problems within a high level of accuracy, and similarly for GN and LS in Figures 4.4(b) and 4.4(f). By looking at Figure 4.5, it appears that this is purely due to the strict limit on the number of evaluations. When more background error correlations are included in the assimilation, Figures 4.5(c), 4.5(d) and 4.5(h) show that all three globally convergent methods are solving close to all of the problems within a high level of accuracy. The GN method also performs better in the presence of more correlated background errors, but still fails to compete with the globally convergent methods.



(a) 3DVarProb1, $\mathbf{B}_0$ (b) 3DVarProb1, $\mathbf{B}_1$ (c) 3DVarProb1, $\mathbf{B}_2$ (d) 3DVarProb1, $\mathbf{B}_3$

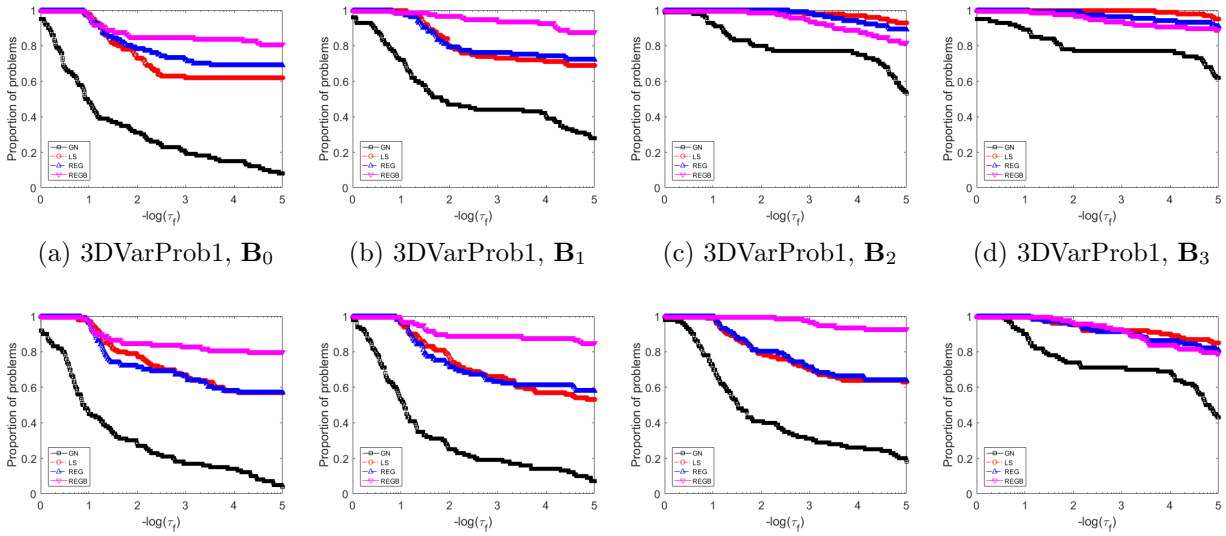
(e) 3DVarProb2, $\mathbf{B}_0$ (f) 3DVarProb2, $\mathbf{B}_1$ (g) 3DVarProb2, $\mathbf{B}_2$ (h) 3DVarProb2, $\mathbf{B}_3$

Figure 4.4: Accuracy profiles for the GN (black), LS (red), REG (blue) and REGB (magenta) methods applied to 3DVarProb1 in (a)-(d) and 3DVarProb2 in (e)-(h) where $n_r = 100$, the observation error is 5% and the background error is 50%. These show the proportion of problems solved by each of the methods against the specified accuracy $-\log(\tau_f)$ when $\tau_e = 8$.

In this section, we have studied the effects of choosing the initial REG parameter according to the background error covariance matrix. We have also compared the performance of GN, LS, REG and REGB. In DA, we are interested in knowing the accuracy of the estimate obtained as in applications such as NWP, the estimate is used as the initial conditions for a forecast and so the quality of this forecast will depend on the errors in the estimate. In the following section, we quantify and compare the errors in the estimates obtained by each method.

4.7.3 Quality of the analysis

We recall that the initial guess of the algorithms is the reference state $\mathbf{x}^{ref}$ perturbed by the background error $\varepsilon_{\mathbf{b}}$. In order to compare the quality of the estimate obtained by each method, we compare their estimate to the reference state $\mathbf{x}^{ref}$ to understand how far the



(a) 3DVarProb1, $\mathbf{B}_0$ (b) 3DVarProb1, $\mathbf{B}_1$ (c) 3DVarProb1, $\mathbf{B}_2$ (d) 3DVarProb1, $\mathbf{B}_3$

(e) 3DVarProb2, $\mathbf{B}_0$ (f) 3DVarProb2, $\mathbf{B}_1$ (g) 3DVarProb2, $\mathbf{B}_2$ (h) 3DVarProb2, $\mathbf{B}_3$

Figure 4.5: Accuracy profiles for the GN (black), LS (red), REG (blue) and REGB (magenta) methods applied to 3DVarProb1 in (a)-(d) and 3DVarProb2 in (e)-(h) where $n_r = 100$, the observation error is 5% and the background error is 50%. These show the proportion of problems solved by each of the methods against the specified accuracy $-\log(\tau_f)$ when $\tau_e = 100$.

estimates obtained by the methods have deviated from this. The analysis error for each state variable is the difference between the reference state and the estimate obtained by each method, given by $\varepsilon_i^a = x_i^a - x_i^{ref}$. For each realisation, we calculate the root mean square error (RMSE) of the analysis error given by,

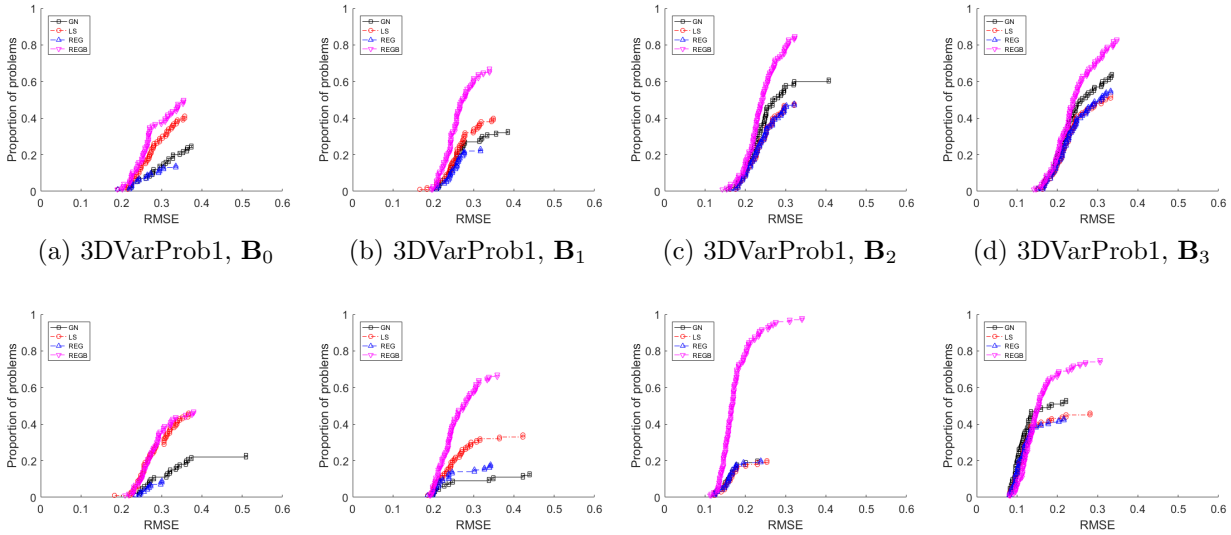
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\varepsilon_i^a)^2}{n}}. \tag{4.104}$$

For each method, we plot the percentage of problems solved (according to the criterion (3.49) where $\tau_f = 10^{-3}$) within a specified tolerance of the RMSE (4.104), with no restriction on the values that (4.104) can take. We acknowledge in this work that the code for the RMSE profiles has been adapted from the code for the data profiles used in [90].

The results for 3DVarProb1 and 3DVarProb2 for the case where $\tau_e = 8$ are in Figure 4.6, which coincides with the case shown in Figure 4.4 where $\tau_f = 10^{-3}$. From this, we see that REGB solves the most problems within the same level of RMSE accuracy as GN, LS and REG. Figures 4.6(a), 4.6(b), 4.6(e) and 4.6(f) show that GN and REG solve fewer problems within the same level of RMSE accuracy as LS and REGB when using $\mathbf{B}_0$ and $\mathbf{B}_1$. As stronger background error correlations are introduced, Figures 4.6(c), 4.6(d), 4.6(g) and 4.6(h) show that LS and REG are the methods that solve fewer problems within the same level of RMSE accuracy as GN and REGB. This is because of the improvement in the performance of GN as stronger background error correlations are introduced; a pattern that was

also seen in the accuracy profiles in Figure 4.4 and discussed in Section 4.7.2.

Recall from Section 2.2.2, the use of correlated background error in NWP, where a method equivalent to GN is used, has shown to improve the quality of the analysis [4]; a finding that is also reflected in our results. We see how the RMSE of the analyses successfully found by each GN-type method reduces as stronger background error correlations are introduced, this can be seen in the scale of the x axis in Figures 4.6(a), 4.6(b), 4.6(c) and 4.6(d) for 3DVarProb1 and Figures 4.6(e), 4.6(f), 4.6(g) and 4.6(h) for 3DVarProb2. The y axis shows that the proportion of problems solved also increases as we strengthen the background error correlations.



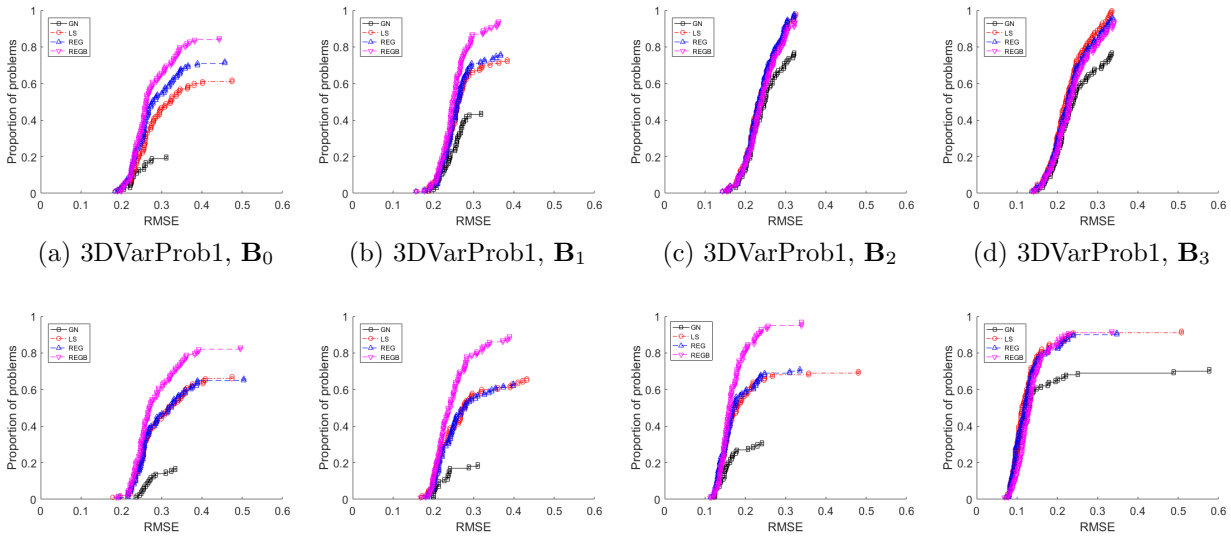
(a) 3DVarProb1, $\mathbf{B}_0$ (b) 3DVarProb1, $\mathbf{B}_1$ (c) 3DVarProb1, $\mathbf{B}_2$ (d) 3DVarProb1, $\mathbf{B}_3$

(e) 3DVarProb2, $\mathbf{B}_0$ (f) 3DVarProb2, $\mathbf{B}_1$ (g) 3DVarProb2, $\mathbf{B}_2$ (h) 3DVarProb2, $\mathbf{B}_3$

Figure 4.6: RMSE plots for the GN (black), LS (red), REG (blue) and REGB (magenta) methods applied to 3DVarProb1 in (a)-(d) and 3DVarProb2 in (e)-(h), where $n_r = 100$, $\tau_f = 10^{-3}$ and $\tau_e = 8$. The observation error is 5% and the background error is 50%.

The results for 3DVarProb1 and 3DVarProb2 for the case where $\tau_e = 100$ are in Figure 4.7, which coincides with the case shown in Figure 4.5 where $\tau_f = 10^{-3}$. As expected, we see that all methods perform better when more function and Jacobian evaluations are allowed than in practice; they are able to solve a larger proportion of problems, as indicated by the y axis. GN appears to be the worst method out of the four, solving fewer problems within a given RMSE accuracy. The majority of the plots in Figure 4.7 show that REGB solves the most problems within the same level of RMSE accuracy as GN, LS and REG. However, in Figures 4.7(c), 4.7(d) and 4.7(h) we see that the three globally convergent methods are performing similarly. The RMSE errors for each of the LS, REG and REGB analyses are very close together, while GN solves fewer problems at a lower level of RMSE accuracy.

Not only is it important for the fast convergence of the REG method to choose an appropriate initial REG parameter, in the case where REG is performing well, we also want to quickly reduce the REG parameter. We have proven in Section 4.3 that the updating strategy for



(a) 3DVarProb1, $\mathbf{B}_0$ (b) 3DVarProb1, $\mathbf{B}_1$ (c) 3DVarProb1, $\mathbf{B}_2$ (d) 3DVarProb1, $\mathbf{B}_3$

(e) 3DVarProb2, $\mathbf{B}_0$ (f) 3DVarProb2, $\mathbf{B}_1$ (g) 3DVarProb2, $\mathbf{B}_2$ (h) 3DVarProb2, $\mathbf{B}_3$

Figure 4.7: RMSE plots for the GN (black), LS (red), REG (blue) and REGB (magenta) methods applied to 3DVarProb1 in (a)-(d) and 3DVarProb2 in (e)-(h), where $n_r = 100$, $\tau_f = 10^{-3}$ and $\tau_e = 100$. The observation error is 5% and the background error is 50%.

very successful iterations of REG allows it to achieve global convergence. However, the speed of convergence may be improved by the use of alternative updating strategies for very successful REG iterations. In experiments not included in this chapter, we experimented using different updating strategies for very successful iterations of REG from the work of [49] and [62], but found that these slowed down convergence compared to the strategy used in REG.

In the following section, we conclude the results from this chapter.

4.8 Conclusion

In this chapter, we address research questions RQ1(a), RQ1(b) and part of RQ1(c). In Sections 4.2 and 4.3, we outline and prove the global convergence theorems of LS and REG when applied to a general nonlinear least-squares problem. We discuss whether the assumptions made in these convergence proofs are satisfied in DA in Section 4.1.

In Section 4.4, we use NLLSPs to understand the general behaviour of GN, LS and REG and identify the cases where GN performs well and poorly. We consider an NLLSP where GN outperforms LS and REG and study how the choices of the globalisation parameters in LS and REG affect the convergence rate. For the case where GN performs well, we want the LS and REG methods to be as close to GN as possible. We then consider an NLLSP where GN diverges and show that the LS and REG converge, highlighting the benefits of these methods.

In Section 4.5, we use the variational problem (both standard and preconditioned) to show

how to choose an initial REG parameter to speed up convergence of the REG method. We propose that the standard choice of initial REG parameter of 1 is suitable for the preconditioned VarDA problem. However, for the standard VarDA problem where there is high uncertainty in the background information, the REG parameter must be large enough to change the GN step and the choice of 1 does not achieve this. We propose that the standard VarDA problem benefits from choosing the initial REG parameter based on the background error covariance matrix, which is already available in VarDA practice. We refer to this method as REGB.

In Section 4.7, we use two nonlinear 3D-Var problems with different observation operator structures and consider the convergence behaviour of GN, LS and REG. We study which method is better in the sense of most accurate for a given number of function and Jacobian evaluations and which methods are convergent within a given RMSE accuracy. We show that when there is high uncertainty in the background state vector, which is used as the initial guess for the minimisation, the GN method fails to obtain an estimate of the initial state that is as accurate as the globally convergent methods' estimates.

We then focus on improving the REG method using our theoretical knowledge of the interaction between the REG parameter and the VarDA Hessian. For the standard 3D-Var problem, we choose the initial REG parameter according to the background error covariance matrix. We compare the quality of the estimate obtained using the RMSE of the analysis and show that even in the case where correlated background error is used, the globally convergent methods find estimates with an RMSE less than or equal to the RMSE of the estimates GN obtains. We show that for both uncorrelated and correlated background error matrices, the convergence of the REG method is improved if the initial REG parameter is chosen according to the inverse of the background error covariance matrix. Our results show that, overall, REGB appears to be the best method to solve the standard 3D-Var problem, both in terms of minimising the 3D-Var cost function and in terms of the accuracy of the analysis.

Our findings are important in DA as they show that in cases where the accuracy of the prior information is poor and when there is limited computational cost, the globally convergent methods are still able to minimise the 3D-Var objective function, unlike GN. We have also shown a practical way of choosing an appropriate initial REG parameter.

We also consider the effects of the REG parameter on the conditioning of the 4D-Var problem. For future work, an investigation into the use of second-order information in variational data assimilation may be of interest as we have seen how the regularisation parameter interacts with the 4D-Var Hessian (2.59). We have also seen how the initial choice of the regularisation parameter impacts the solution obtained by the REG method. Knowing more information about the second-order terms (2.32) will enable us to understand more about the curvature of the function and which choices of method/parameters to make.

In Section 4.5, we hypothesised that the influence of the initial REG parameter may not be an issue if we use the preconditioned problem. In the following chapter, we investigate

the performance of the globally convergent methods, LS and REG, when applied to the preconditioned 4D-Var problem where there is the added complication of a numerical model.

Chapter 5

Convergent least-squares optimisation methods for variational data assimilation

In this chapter, we present a draft paper where we study the convergence behaviour of Algorithms 3.2.3, 3.3.1 and 3.3.2 when applied to the preconditioned 4D-Var problem, with the aim of addressing the research questions RQ1(c) and RQ1(d). This paper is intended for submission by C. Cartis, M. H. Kaouri, A. S. Lawless and N. K. Nichols. In the paper, we consider uncorrelated background error only. We then present a discussion on the additional results (not included in the paper), where correlated background error is used.

For simplicity, throughout this chapter we omit both the subscript p notation (for preconditioned) and the hat notation (for reduced resolution) as we are considering the preconditioned 4D-Var problem with a full resolution inner loop only and do not need to differentiate this from the standard, reduced resolution 4D-Var problem.

5.1 Abstract

Data assimilation combines prior (or background) information with observations to estimate the initial state of a dynamical system over a given time-window. A common application is in numerical weather prediction where a previous forecast and atmospheric observations are used to obtain the initial conditions for a numerical weather forecast. In four-dimensional variational data assimilation (4D-Var), the problem is formulated as a nonlinear least-squares problem, usually solved using a variant of the classical Gauss-Newton (GN) method. However, GN may not converge if poorly initialised. This could occur when there is greater uncertainty in the background information compared to the observations, or when a long time-window is used in 4D-Var allowing the use of more observations. The difficulties GN encounters may lead to inaccurate initial state conditions for subsequent forecasts. To overcome this, we apply two convergent GN variants (line search and regularisation) to the long time-window 4D-Var problem and investigate the cases where they locate a more accurate estimate compared to GN within a given budget of computational time and cost.

Keywords: Data assimilation, Gauss-Newton, least squares, line search, optimisation, regularisation

Highlights:

- Poor initialisation of Gauss-Newton may result in the method not converging
- Safeguarded Gauss-Newton used on long window variational data assimilation problem
- Twin experiments with limited computational budget and chaotic Lorenz models
- Results in improved initial state estimate despite inaccurate prior information
- Application of least squares convergence theory to variational data assimilation

5.2 Introduction

Data assimilation (DA) is a technique used to estimate the state of a dynamical system. In DA, a prior estimate of the state from a previous forecast, known as the background state, is combined with observations using an optimisation method to obtain an estimate of the evolving state of the system. In Numerical Weather Prediction (NWP), four-dimensional variational data assimilation (4D-Var) is used to estimate the initial conditions for a weather forecast [68]. The 4D-Var scheme is able to incorporate information from a prior forecast along with observations over both temporal and spatial domains in the form of a nonlinear least-squares objective function, which is then minimised using an iterative method. From a Bayesian point of view, minimising the 4D-Var objective function is equivalent to maximising the posterior probability to obtain the maximum a posteriori estimate [95]. Within this paper, we focus on the strong-constraint 4D-Var problem where we assume the numerical model of the system perfectly represents the true dynamics of the system, or the model errors are small enough to be neglected. This formulation has been commonly used operationally in many meteorological centres [103], including the Meteorological Service of Canada [43], ECMWF [63, 80, 105] and the Met Office [109].

In practice, the 4D-Var problem is a nonlinear least-squares problem which can be viewed as a large-scale unconstrained optimisation problem [68]. The quality of the estimate and the subsequent forecast depends on how accurately the variational problem is solved within the time and computational cost available. The desire to improve the current methods and develop new methods for this class of problems has a long history, briefly summarised in the following. Ideally in large-scale unconstrained optimisation, we seek a fast rate of convergence, which can usually be achieved using a Newton-type method. However, these methods require the use of second derivatives of the 4D-Var objective function, which are too costly to compute and store in practice. Therefore, it is common practice to use methods that approximate the high order terms, such as the limited memory conjugate gradient [91, 102, 113], limited memory Quasi-Newton (LMQN) [135] e.g. L-BFGS [72] and M1QN3 [44], Truncated Newton (TN) [67, 131], hybrid LMQN and TN [29], Adjoint Newton [130] or Gauss-Newton

(GN) methods [33, 96].

Solving the 4D-Var problem requires the use of the adjoint of the numerical model (simply referred to as the 'adjoint') to efficiently compute the first derivatives of the objective function [68]. More recently, optimisation methods that do not require the use of first derivatives of the 4D-Var objective function are being investigated to avoid the development and maintenance costs associated with using the adjoint, see for example [51] and references therein. Furthermore, an extensive list of alternative assimilation schemes have been developed that do not require the use of the adjoint such as the ensemble-variational data assimilation method, 4DEnVar, developed by the Met Office for global NWP (see [3, 76]) and implemented at Environment Canada (see [19]). However, as the adjoint is already embedded in the operational infrastructure of many meteorological centres, its use is still often preferred over newer techniques as, unlike methods that require the use of an ensemble of non-linear forecast trajectories, such as the ensemble Kalman Filter (EnKF) and 4DEnVar, the adjoint is less prone to causing sampling errors and does not require spatial and/or temporal localisation [6].

In a 4D-Var scheme, the 4D-Var problem is solved as a sequence of linear least-squares problems using an incremental method, which has been shown to be equivalent to the GN method under certain conditions [65]. In the incremental method, the minimisation of the nonlinear objective function and the linearised subproblem are referred to as the 'outer loop' and the 'inner loop' respectively. It is known that the accuracy with which the inner loop is solved affects the convergence of the outer loop [64, 65]. Within our work, we focus on the convergence of the outer loop and assume that the inner loop is solved exactly. Furthermore, we use a variable transformation commonly used in operational DA to precondition the 4D-Var problem, see [5] for a detailed explanation.

The GN method does not require the use of high order second derivatives, thus alleviating the added complexity of calculating and storing them. A drawback of the GN method is that it does not guarantee convergence to an estimate of the initial state given poor initialisation [33]. In NWP, the initial guess for the minimisation is generally chosen to be the predicted initial state from a previous forecast, known as the background state. However, for some applications of a 4D-Var scheme, this choice may not be a good enough estimate of the true initial state. Therefore, GN may fail to converge. Furthermore, the use of long assimilation time-windows (in the order of days) has recently become of interest in global NWP as it enables the use of more observations, improving the quality of the estimate of the initial state of the system, known as the 'analysis' in DA [69]. However, when the NWP system is sensitive to small changes in the initial conditions, the errors in the initial conditions are amplified over time through the use of the model, and more so when a long assimilation time-window is used.

There are three main strategies that safeguard GN and make it convergent from an arbitrary initial guess: line search, trust-region and regularisation [96]. GN with quadratic regularisation is strongly related to GN with trust-region (see Lemma 10.2. of [96]), also referred to as the Levenberg-Marquardt method (LM) [26]. Within our work, we focus on GN

with quadratic regularisation (REG) and compare its performance to GN with backtracking Armijo line search (LS) and GN alone when applied to the preconditioned 4D-Var problem.

Earlier work in [106] showed that the use of a line search strategy improved the minimisation of the 4D-Var objective function. The M1QN3 method, developed in [44] and used operationally at ECMWF [126], Météo France [74] and the Meteorological Service of Canada [20], uses the Wolfe line search conditions [134] to safeguard the method. However, the Wolfe conditions require the use of additional evaluations of the objective function and, to rule out unacceptably short steps, its gradient [96]. This is unlike the Armijo condition [2] used in our work, as in [50], which through the use of backtracking only requires additional evaluations of the objective function and not of the gradient [96]. We pair GN with backtracking Armijo line search and use a fixed amount of computational operations to guarantee a reduction in the outer loop objective function (assuming the inner loop is solved to a high accuracy), while considering the computational limits present in DA.

The use of the LM method has been of interest in the DA community because of its similarities with GN and its convergence guarantees. The use of the ensemble Kalman filter and ensemble Kalman smoother methods (EnKF and EnKS, respectively) as linear least-squares solvers for the inner loop problem has been proposed [82, 136]. This is the approach used in the literature when using LM in DA. Bergou et al. [9] applied a variation of the LM method to the 4D-Var problem combined with the use of ensemble methods for the linearised subproblems where they focus on the case where only approximate gradient values are available and accurate within a certain probability. They provide a framework for using EnKS for solving the subproblem inexactly in their LM method and proved global convergence under an assumption of the probability of the accuracy of the gradient. In the work of Bocquet et al. [13], they applied a variation of LM using an EnKF to regularise the subproblem and obtain a faster convergence rate versus GN. Mandel et al [83] use a LM method to control the convergence when solving the 4D-Var weak-constraint (where model error is accounted for) problem using the EnKS as the inexact solver for the 4D-Var inner-problem. They concluded that the choice of the regularisation parameter greatly impacts the estimate obtained by their method. Within our work, we focus on the convergence of the 4D-Var problem on the outer loop level, where the exact gradient is used (as is the case when an adjoint is available), the inner loop is assumed to be solved to a high accuracy and the regularisation parameter in REG is updated using a simple, inexpensive strategy.

We aim to investigate whether the use of convergent optimisation methods is beneficial in 4D-Var, where there is limited time and computational cost available. Such methods use safeguards to guarantee convergence to the analysis from an arbitrary background state vector by ensuring monotonic/strict and sufficient decrease of the error in the objective function. We refer to these methods as 'globally convergent'. Using two test models within the 4D-Var framework, we show that when there is more uncertainty in the background information compared to the observations, the GN method may diverge in the long time-window case, yet the convergent methods, LS and REG, are able to improve the estimate of the analysis. We use accuracy profiles to show numerically that in the long time-window case and when there is higher uncertainty in the background information versus the observations, the

globally convergent methods are able to solve more problems than GN in the limited cost available. By 'solve' we mean satisfy a criterion based on the reduction in the objective function within a set number of evaluations. We also show the effect that poor background information has on the quality of the estimate obtained. We consider the case where the background information is highly inaccurate compared to the observations and find that the convergence of all three methods is improved when more observations are included along the time-window. Finally, for the case where GN performs well, we recommend further research into the parameter updating strategies used within the globally convergent methods.

This paper is organised as follows. In Section 5.3 we outline the strong-constraint 4D-Var problem as a nonlinear least-squares problem and the GN method that is frequently used to solve it. In Section 5.4 we outline the globally convergent methods used within this paper. In Section 5.5 we describe the experimental design including the dynamical models used. In Section 5.6 we present the numerical results obtained when applying GN and the globally convergent methods to the 4D-Var problem with different features. Finally, we conclude our findings in Section 5.7.

5.3 Variational data assimilation

5.3.1 4D-Var: least-squares formulation

In four-dimensional variational data assimilation (4D-Var), the analysis $\mathbf{x}_0^a \in \mathbb{R}^n$ is obtained by minimising a objective function consisting of two terms: the background term and the observation term, namely;

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}_0^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathbf{x}_i)). \tag{5.1}$$

The background term measures the difference between the initial state of the system and the background state vector $\mathbf{x}_0^b \in \mathbb{R}^n$, which contains prior information. The observation term measures the difference between information from observations at times t_i in the observation vector $\mathbf{y}_i \in \mathbb{R}^{p_i}$ and the model state vector $\mathbf{x}_i \in \mathbb{R}^n$ at the same time through use of the observation operator $\mathcal{H}_i : \mathbb{R}^n \to \mathbb{R}^{p_i}$ that maps from the model state space to the observation space. Both terms are weighted by their corresponding covariance matrices to represent the uncertainty in the respective measures, the background error covariance matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ and the observation error covariance matrices at times t_i, $\mathbf{R}_i \in \mathbb{R}^{p_i \times p_i}$, which are assumed to be symmetric positive definite. We note that observations are distributed both in time and space and there are usually fewer observations available than there are state variables so $p < n$, where $p = \sum_{i=0}^{N} p_i$. The 4D-Var objective function (5.1) is subject to the nonlinear dynamical model equations which contain the physics of the system

$$\mathbf{x}_i = \mathcal{M}_{0,i}(\mathbf{x}_0), \tag{5.2}$$

where the nonlinear model $\mathcal{M}_{0,i} : \mathbb{R}^n \to \mathbb{R}^n$ evolves the state vector from the initial time point t_0 to the time point t_i.

We precondition the 4D-Var problem using a variable transform, which has been shown to improve the conditioning of the variational optimisation problem [56, 57]. To be able to use the negative square root of $\mathbf{B}$ in our variable transformation, we first require the assumption that the matrix $\mathbf{B}$ is full rank. This assumption is satisfied for our choices of $\mathbf{B}$ in Section 5.6. We define a new variable $\mathbf{v}$ to be,

$$\mathbf{v} = \mathbf{B}^{-1/2}(\mathbf{x}_0 - \mathbf{x}_0^b). \tag{5.3}$$

The 4D-Var objective function can then be written in terms of $\mathbf{v}$, known as the control variable in DA, and minimised with respect to this instead. Furthermore, by including the model information within the objective function, we are able to write the constrained optimisation problem (5.1)-(5.2) in the form of an unconstrained optimisation problem and apply the minimisation methods described later in this paper. The preconditioned 4D-Var objective function is given by

$$\mathcal{J}(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T\mathbf{v} + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{y}_i - \mathcal{H}_i(\mathcal{M}_{0,i}(\mathbf{B}^{1/2}\mathbf{v} + \mathbf{x}_0^b)))^T\mathbf{R}_i^{-1}(\mathbf{y}_i - \mathcal{H}_i(\mathcal{M}_{0,i}(\mathbf{B}^{1/2}\mathbf{v} + \mathbf{x}_0^b))). \tag{5.4}$$

We note that the function (5.4) is continuously differentiable if the operators $\mathcal{H}_i$ and $\mathcal{M}_{0,i}$ are continuously differentiable. To save both computational cost and time in 4D-Var, the nonlinear operators in (5.4) are linearised for use in the inner loop, often using a tangent linear approximation [27]. The tangent linear model and observation operator are derived by linearising the discrete nonlinear model equations.

In a nonlinear least-squares problem, the function $\mathcal{J} : \mathbb{R}^n \to \mathbb{R}$ has a special form, as defined in the following,

$$\min_{\mathbf{v}} \mathcal{J}(\mathbf{v}) = \frac{1}{2}\|\mathbf{r}(\mathbf{v})\|_2^2, \tag{5.5}$$

where $\mathbf{r}(\mathbf{v}) = [r_1(\mathbf{v}), ..., r_{n+p}(\mathbf{v})]^T$ and each $r_j : \mathbb{R}^n \to \mathbb{R}$, for $j = 1, 2, \ldots, n+p$, is referred to as a residual. In (5.5), $\|.\|_2$ denotes the l_2-norm, which will be used throughout this paper. Equation (5.4) is equivalent to (5.5) where the residual vector $\mathbf{r}(\mathbf{v}) \in \mathbb{R}^{n+p}$ and its Jacobian $\mathbf{J}(\mathbf{v})$ are given by

$$\mathbf{r}(\mathbf{v}) = \begin{pmatrix} \mathbf{v} \\ \mathbf{R}_0^{-1/2}(\mathbf{y}_0 - \mathcal{H}_0(\mathbf{B}^{1/2}\mathbf{v} + \mathbf{x}_0^b)) \\ \mathbf{R}_1^{-1/2}(\mathbf{y}_1 - \mathcal{H}_1(\mathcal{M}_{0,1}(\mathbf{B}^{1/2}\mathbf{v} + \mathbf{x}_0^b))) \\ \vdots \\ \mathbf{R}_N^{-1/2}(\mathbf{y}_N - \mathcal{H}_N(\mathcal{M}_{0,N}(\mathbf{B}^{1/2}\mathbf{v} + \mathbf{x}_0^b))) \end{pmatrix} \text{ and } \mathbf{J}(\mathbf{v}) = \begin{pmatrix} \mathbf{I} \\ -\mathbf{R}_0^{-1/2}\mathbf{H}_0\mathbf{B}^{1/2} \\ -\mathbf{R}_1^{-1/2}\mathbf{H}_1\mathbf{M}_{0,1}\mathbf{B}^{1/2} \\ \vdots \\ -\mathbf{R}_N^{-1/2}\mathbf{H}_N\mathbf{M}_{0,N}\mathbf{B}^{1/2} \end{pmatrix}, \tag{5.6}$$

where

$$\mathbf{M}_{0,i} = \frac{\partial \mathcal{M}_{0,i}}{\partial \mathbf{v}}\Big|_{\mathcal{M}_{0,i}(\mathbf{B}^{1/2}\mathbf{v}+\mathbf{x}_0^b)} \text{ and } \mathbf{H}_i = \frac{\partial \mathcal{H}_i}{\partial \mathbf{v}}\Big|_{\mathcal{M}_{0,i}(\mathbf{B}^{1/2}\mathbf{v}+\mathbf{x}_0^b)} \tag{5.7}$$

are the Jacobian matrices of the model operator $\mathcal{M}_{0,i}$ and observation operator $\mathcal{H}_i$ respectively, where $\mathbf{M}_{0,i} \in \mathbb{R}^{n\times n}$ is the tangent linear of $\mathcal{M}_{0,i}$ and $\mathbf{H}_i \in \mathbb{R}^{p_i\times n}$ is the tangent linear

of $\mathcal{H}_i$ [95]. In practice, an adjoint method is used to calculate the gradient of (5.4), defined as

$$\nabla \mathcal{J}(\mathbf{v}) = \mathbf{J}(\mathbf{v})^T \mathbf{r}(\mathbf{v}). \tag{5.8}$$

The Hessian is the matrix of second-order partial derivatives of (5.4),

$$\nabla^2 \mathcal{J}(\mathbf{v}) = \mathbf{J}(\mathbf{v})^T \mathbf{J}(\mathbf{v}) + \sum_{j=1}^{n+p} r_j(\mathbf{v}) \nabla^2 r_j(\mathbf{v}). \tag{5.9}$$

In data assimilation, the second-order terms in (5.9) are often difficult to calculate in the time and cost available and too large to store, and so one cannot easily use Newton-type methods for 4D-Var. Therefore, a first-order approximation to the Hessian of the objective function (5.4) is used, resulting in a GN method, and is given by

$$\mathbf{S} = \mathbf{J}(\mathbf{v})^T \mathbf{J}(\mathbf{v}) = \mathbf{I} + \sum_{i=0}^{N} \mathbf{B}^{1/2} \mathbf{M}_{0,i}^T \mathbf{H}_i^T \mathbf{R}_i^{-1} \mathbf{H}_i \mathbf{M}_{0,i} \mathbf{B}^{1/2}, \tag{5.10}$$

which is, by construction, full rank and symmetric positive definite. The condition number in the l_2-norm of (5.10), $\kappa(\mathbf{S})$, is the ratio of its largest and smallest eigenvalues and is related to the number of iterations used for the linear minimisation problems in 4D-Var and how sensitive the estimate of the initial state is to perturbations of the data. We can use $\kappa(\mathbf{S})$ to indicate how quickly and accurately the optimisation problem can be solved [47].

5.3.2 4D-Var implementation

The incremental 4D-Var method, which was first proposed for practical implementation of the NWP problem in [27], has been shown to be equivalent to the GN method when an exact tangent linear model is used to linearise the discrete nonlinear model. When an approximate tangent linear model (TLM) is used, the method is equivalent to an inexact GN method [52, 65]. A summary of the GN method is given by the following.

Algorithm 5.3.1: GN algorithm applied to (5.5) **[33].**

Step 0: Initialisation. Given $\mathbf{v}^{(0)} \in \mathbb{R}^n$ and some stopping criteria. Set $k = 0$.

Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation. Compute a step $\mathbf{s}^{(k)}$ that satisfies

$$\mathbf{J}(\mathbf{v}^{(k)})^T \mathbf{J}(\mathbf{v}^{(k)}) \mathbf{s}^{(k)} = -\mathbf{J}(\mathbf{v}^{(k)})^T \mathbf{r}(\mathbf{v}^{(k)}). \tag{5.11}$$

Step 3: Iterate update. Set $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \mathbf{s}^{(k)}$, $k := k + 1$ and go to Step 1.

In Algorithm 5.3.1, the updated control variable $\mathbf{v}^{(k+1)}$ is computed by finding a step $\mathbf{s}^{(k)}$ that satisfies (5.11), which is known as the preconditioned linearised subproblem. By substituting $\mathbf{v}^{(k+1)}$ into (5.3) and rearranging, we obtain the current estimate $\mathbf{x}_0^{(k+1)}$ of the initial

state to the original nonlinear 4D-Var problem.

To reduce the computational cost in large DA systems and to solve the DA problem in real time, the series of problems (5.11) can be solved approximately as a series of linear least-squares problems in the inner loop. The inner loop can be solved using iterative optimisation methods such as Conjugate Gradient (CG) where a limited number of CG iterations are allowed and an exact or approximate $\mathbf{J}$ is used [52]. We do not focus on this here and assume that (5.11) is solved exactly.

We note that the step calculation (5.11) uniquely defines $\mathbf{s}^{(k)}$, and $\mathbf{s}^{(k)}$ is a descent direction when $\mathbf{J}(\mathbf{v})$ is full column rank. This is the case in 4D-Var as the Jacobian, $\mathbf{J}(\mathbf{v})$ in (5.6) is full column rank due to the presence of the identity matrix, thus ensuring that $\mathbf{s}^{(k)}$ is a descent direction.

The definitions of two solution types, namely; local and global minima, are stated in Appendix 5.8, along with a brief explanation of the local convergence property of GN. Although the GN method benefits from local convergence properties, convergence can only be guaranteed if the initial guess $\mathbf{v}^{(0)}$ of the algorithm is in some neighbourhood around an (unknown) local solution $\mathbf{v}^*$, that is, convergence from an arbitrary initial guess is not guaranteed. Even if the GN method does converge, it may not necessarily converge to the global minimum due to the fact that multiple local minima of a nonlinear least-squares objective function may exist.

GN has no way of adjusting the length of the step $\mathbf{s}^{(k)}$ and hence, may take steps that are too long and fail to decrease the objective function value and thus to converge, see Example 10.2.5 in [33] and later in Section 5.6 where the divergence of GN is demonstrated. As GN only guarantees local convergence, we are interested in investigating methods that converge when $\mathbf{v}^{(0)}$ is far away from a local minimiser $\mathbf{v}^*$. We refer to these methods as 'globally convergent'. Mathematical theory on global strategies can be found in [96] and [33]. Two globally convergent methods are GN with line search and GN with quadratic regularisation, which use a strategy within the GN framework to achieve convergence to a stationary point given an arbitrary initial guess by adjusting the length of the step. These methods will be presented in the next section.

5.4 Globally convergent methods

Within this section, we outline the two globally convergent algorithms that we apply in Section 5.6 to the preconditioned 4D-Var problem.

5.4.1 Gauss-Newton with line search (LS)

A line search method aims to restrict the step $\mathbf{s}^{(k)}$ in (5.11) so as to guarantee a decrease in the error for $\mathcal{J}$. Within our work, an inexact line search method known as the backtracking-Armijo (bArmijo) algorithm is used within the inner loop of GN to find a step length $\alpha > 0$

that satisfies the Armijo condition [2]. The Gauss-Newton with backtracking-Armijo line search (LS) method is presented as follows.

Algorithm 5.4.1: LS algorithm applied to (5.5) **[96].**

Step 0: Initialisation. Given $\mathbf{v}^{(0)} \in \mathbb{R}^n$, $\tau \in (0,1)$ and $\beta \in (0,1)$ and $\alpha_0 > 0$ and some stopping criteria. Set $k = 0$.

Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation. Compute a step $\mathbf{s}^{(k)}$ that satisfies

$$\mathbf{J}(\mathbf{v}^{(k)})^T\mathbf{J}(\mathbf{v}^{(k)})\mathbf{s}^{(k)} = -\mathbf{J}(\mathbf{v}^{(k)})^T\mathbf{r}(\mathbf{v}^{(k)}) \tag{5.12}$$

and set $\alpha^{(k)} = \alpha_0$.

Step 3: Check Armijo condition.
While the following (Armijo) condition is not satisfied

$$\mathcal{J}(\mathbf{v}^{(k)} + \alpha^{(k)}\mathbf{s}^{(k)}) \leq \mathcal{J}(\mathbf{v}^{(k)}) + \beta\alpha^{(k)}{\mathbf{s}^{(k)}}^T\nabla\mathcal{J}(\mathbf{v}^{(k)}), \tag{5.13}$$

do:

Step 4: Shrink stepsize. Set $\alpha^{(k)} := \tau\alpha^{(k)}$ and go to Step 3.

Step 5: Iterate update.
Set $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + \alpha^{(k)}\mathbf{s}^{(k)}$, $k := k + 1$ and go to Step 1.

In Algorithm 5.4.1, the control parameter β in (5.13) is typically chosen to be small (see [96]). The step equation (5.12) is the same as the GN step equation (5.11); thus when $\alpha^{(k)} = 1$, the GN and LS iterates coincide. The use of condition (5.13) in this method ensures that the accepted steps produce a sequence of strictly decreasing function values given $\nabla\mathcal{J}(\mathbf{v}^{(k)})^T\mathbf{s}^{(k)} < 0$. This latter condition is satisfied by $\mathbf{s}^{(k)}$ defined in (5.12) [96].

Despite its global convergence property (see Appendix 5.8.1), the LS method has some disadvantages. We remark that the use of the step length $\alpha^{(k)}$ may sometimes unnecessarily shorten the step $\mathbf{s}^{(k)}$, slowing down the convergence. Furthermore, LS may be computationally costly due to the need to calculate the value of the function $\mathcal{J}$ each time $\alpha^{(k)}$ is adjusted, although more sophisticated updating strategies for α may be used to try to reduce this effect.

Other line search strategies are possible such as Wolfe, Goldstein-Armijo and more [96], but they are more involved and potentially more computationally costly. As LS requires the re-evaluation of the outer loop objective function each time it adjusts its line search parameter, its applicability to real systems has been in doubt due to the constraint on the computational cost in 4D-Var [106]. In Section 5.6, we show that given the same cost as the GN method, the LS method can in some cases, better minimise the preconditioned 4D-Var objective function.

5.4.2 Gauss-Newton with regularisation (REG)

The GN method may also be equipped with a globalisation strategy by including a regularisation term $\gamma^{(k)}\mathbf{s}^{(k)}$ in the step calculation (5.11) of Algorithm 5.3.1. This ensures that the accepted steps produce a sequence of monotonically decreasing function values. This is a common variation of the GN method known as the Levenberg-Marquardt method, proposed in [70] and [84]. The effect of $\gamma^{(k)}$ is to implicitly control the length of the step $\mathbf{s}^{(k)}$. Increasing $\gamma^{(k)}$ shortens the steps, thus increasing the possibility that the procedure will decrease the objective function in the next iteration. The REG method is presented as follows.

Algorithm 5.4.2: REG algorithm applied to (5.5) **[89].**

Step 0: Initialisation. Given $\mathbf{x}^{(0)} \in \mathbb{R}^n$, $1 > \eta_2 \geq \eta_1 > 0$, $\gamma^{(0)} > 0$ and some stopping criteria. Set $k = 0$.

Step 1: Check stopping criteria. While the stopping criteria are not satisfied, do:

Step 2: Step computation. Compute a step $\mathbf{s}^{(k)}$ that satisfies

$$\left(\mathbf{J}(\mathbf{v}^{(k)})^T\mathbf{J}(\mathbf{v}^{(k)}) + \gamma^{(k)}\mathbf{I}\right)\mathbf{s}^{(k)} = -\mathbf{J}(\mathbf{v}^{(k)})^T\mathbf{r}(\mathbf{v}^{(k)}). \tag{5.14}$$

Step 3: Iterate update. Compute the ratio

$$\rho^{(k)} = \frac{\mathcal{J}(\mathbf{v}^{(k)}) - \mathcal{J}(\mathbf{v}^{(k)} + \mathbf{s}^{(k)})}{\mathcal{J}(\mathbf{v}^{(k)}) - m(\mathbf{s}^{(k)})}, \tag{5.15}$$

where

$$m(\mathbf{s}^{(k)}) = \frac{1}{2}\|\mathbf{J}(\mathbf{v}^{(k)})\mathbf{s}^{(k)} + \mathbf{r}(\mathbf{v}^{(k)})\|_2^2 + \frac{1}{2}\gamma^{(k)}\|\mathbf{s}^{(k)}\|_2^2. \tag{5.16}$$

Set

$$\mathbf{v}^{(k+1)} = \begin{cases} \mathbf{v}^{(k)} + \mathbf{s}^{(k)}, & \text{if } \rho^{(k)} \geq \eta_1 \\ \mathbf{v}^{(k)}, & \text{otherwise.} \end{cases} \tag{5.17}$$

Step 4: Regularisation parameters update. Set

$$\gamma^{(k+1)} = \begin{cases} \frac{1}{2}\gamma^{(k)}, & \text{if } \rho^{(k)} \geq \eta_2 \text{ (very successful iteration)} \\ \gamma^{(k)}, & \text{if } \eta_1 \leq \rho^{(k)} < \eta_2 \text{ (successful iteration)} \\ 2\gamma^{(k)}, & \text{otherwise, (unsuccessful iteration)} \end{cases} \tag{5.18}$$

Let $k := k + 1$ and go to Step 1.

We note that when $\gamma^{(k)} = 0$ in (5.14), the REG step in (5.14) is the same as the GN step in (5.11); thus the GN and REG iterates coincide at the iterate $\mathbf{x}^{(k)}$. As in Algorithms 5.3.1 and 5.4.1, the step equation (5.14) is solved directly in the numerical experiments in Section 5.6. By comparing (5.14) with (5.11), we are able to see how the REG step differs from the GN step. The diagonal entries of the Hessian of the 4D-Var objective function (5.4) are increased by the regularisation parameter $\gamma^{(k)}$ at each iteration of the REG method. The

method is able to vary its step between a GN and a gradient descent step by adjusting $\gamma^{(k)}$ (see [96]) but may be costly due to the need to calculate the value of the function $\mathcal{J}$ on each iteration. Note that other choices of the factors $\frac{1}{2}$ and 2 for updating $\gamma^{(k)}$ in (5.18) are possible and even more sophisticated variants for choosing $\gamma^{(k)}$ have been proposed. The proof of global convergence of the REG method is presented in Appendix 5.8.2.

5.5 Experimental design

Before evaluating the GN, LS and REG methods numerically, we first explain the experimental design.

Twin experiments are commonly used to test DA methods. They use error statistics that satisfy the DA assumptions as well as synthetic observations generated by running the nonlinear model forward in time to produce a reference state (not generally a local minimum of (5.5)). Within this section, we define our choices for the twin experimental design. We begin by briefly outlining two commonly used dynamical models, which are sensitive to initial conditions (chaotic nature), a property shared with NWP models.

5.5.1 Models

Lorenz 1963 model (L63) Proposed in [77], the Lorenz 63 model (L63) is a popular experimental dynamical system that represents meteorological processes using a simple model. The model consists of three nonlinear, ordinary differential equations given as

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \beta z, \end{aligned} \tag{5.19}$$

where the state vector consists of $n = 3$ time-dependent variables $\mathbf{x} = [x(t), y(t), z(t)]^T \in \mathbb{R}^3$. The scalar parameters are chosen to be $\sigma = 10$, $\rho = \frac{8}{3}$ and $\beta = 28$, making the system chaotic. A second-order Runge-Kutta method is used to discretise the model equations using a time step $\Delta t = 0.025$.

Lorenz 1996 model (L96) Another popular experimental system is the atmospheric Lorenz 96 model (L96) [78] given by the following n equations,

$$\frac{dx_j}{dt} = -x_{j-2}x_{j-1} + x_{j-1}x_{j+1} - x_j + F, \tag{5.20}$$

where $j = 1, 2, \ldots, n$ is a spatial coordinate. For a forcing term $F = 8$ and $n = 40$ state variables, the system is chaotic [78]. The variables are evenly distributed over a circle of latitude of the Earth with n points with a cyclic domain and a single time unit is equivalent

to approximately 5 atmospheric days. A fourth-order Runge-Kutta method is used to discretise the model equations using a time step $\Delta t = 0.025$ (approximately 3 hours).

For both the L63 and L96 models, the time-window length t_a is varied in the numerical experiments in Section 5.6.1. We will now outline how we formulate the twin experiments, beginning with generating the reference state.

5.5.2 Twin experiments

The reference state at time t_0, $\mathbf{x}_0^{ref}$ is used as the basis of a twin experiment in the definition of the background state (the initial guess for the optimisation algorithms) as well as to generate the observations using a nonlinear model run called the 'nature' run. We begin by explaining how we obtain $\mathbf{x}_0^{ref}$.

Reference state A vector of length n is drawn from the uniform distribution and used as the initial vector of state variables $\mathbf{x}^{rand}$. For the L63 model, $\mathbf{x}^{rand}$ is integrated forward using a second-order Runge-Kutta method, which is spun-up over 1000 time steps to obtain the reference state on the model attractor for the L63 twin experiments, $\mathbf{x}_0^{ref} \in \mathbb{R}^3$. This is the same for the L96 model except a fourth-order Runge-Kutta method is used to obtain $\mathbf{x}_0^{ref} \in \mathbb{R}^{40}$. The reference state at time t_0, $\mathbf{x}_0^{ref}$ can then be used to obtain the full nonlinear model trajectory.

We next explain how we obtain the background state vector used within our twin experiments to be used as the initial guess for the optimisation algorithms.

Background In 4D-Var, the initial guess for the optimisation algorithm is taken to be the background state at time t_0, $\mathbf{x}_0^b$, which incorporates information from previous forecasts. In our experiments, the background state vector $\mathbf{x}_0^b$ is generated by adding Gaussian noise

$$\varepsilon_{\mathbf{b}} \sim \mathcal{N}(0, \mathbf{B}), \tag{5.21}$$

to the reference state at time t_0, $\mathbf{x}_0^{ref}$. For the background error covariance matrix, we choose $\mathbf{B} = \sigma_b^2 \mathbf{I}_n$ where σ_b^2 is the background error variance. The standard deviations of the errors from the reference state for each model are based on the average order of magnitude of the entries of $\mathbf{x}_0^{ref}$. For the L63 experiments, $\sigma_b^2 = 0.25, 1, 6.25$ and 25 represent a $5\%, 10\%, 25\%$ and 50% error respectively. Similarly for the L96 experiments we set $\sigma_b^2 = 0.0625, 0.25, 1.5625$ and 6.25.

As previously mentioned, we generate synthetic observations from a nonlinear model run using the reference state at time t_0, $\mathbf{x}_0^{ref}$. We next describe the choices we made when specifying these observations.

Observations We consider both the spatial and temporal locations of the observations. We assume that for both models observations of single state variables are taken and $\mathbf{H}_i$ are the exact observation operators at times t_i used to map to observation space. For the L63

model, we consider where we have $p = 2$ observations, one of x and one of z per observation location in time. For the L96 model, we consider where we have an observation of the first half of the state variables per observation location in time. This choice mimics what we may expect in reality where we have more observations concentrated in one part of the globe. For both models, we first consider where there is only one set of observations at time N (Nobs1) and then show the effect of using more observations along the time-window in later experiments. We use imperfect observations where the observations $\mathbf{y}_i$ are generated by adding Gaussian noise

$$\varepsilon_{\mathbf{o}} \sim \mathcal{N}(0, \mathbf{R}_i), \tag{5.22}$$

to $\mathbf{H}_i \mathbf{x}_i^{ref}$ for each observation location in time. For the observation error covariance matrix we choose $\mathbf{R}_i = \sigma_o^2 \mathbf{I}_p$ where σ_o^2 is the observation error variance. We expect the problem (5.4) to be more ill-conditioned, thus difficult to solve accurately, when the ratio

$$\frac{\sigma_b}{\sigma_o} \tag{5.23}$$

is large [56, 57]. The ratio (5.23) controls the influence of the observation term in the preconditioned objective function (5.4). For all experiments, we set the standard deviation of the observation error to be 10% of the average order of magnitude of the entries of $\mathcal{H}(\mathbf{x}_i^{ref})$ for both models. For the L63 model, this is $\sigma_o^2 = 1$ and for the L96 model, this is $\sigma_o^2 = 0.25$. We vary the background error variance σ_b^2 above and below σ_o^2 such that the ratio (5.23) varies. This can be thought of as having more confidence in the observations compared to background when $\sigma_b > \sigma_o$ and vice versa. Furthermore, as the initial guess is set to be the background state vector, which is dependent on the value of σ_b, by varying σ_b^2 we are essentially varying the initial guess of the algorithms, thus eliminating starting point bias from our results [7]. It is important to recall here that under certain conditions, the GN method is known for its fast convergence properties when in close vicinity to a local minimum, see [33]. By choosing a small value of σ_b^2, we expect the performance of GN to beat that of both LS and REG as it does not require the adjustment of the additional parameters $\alpha^{(k)}$ and $\gamma^{(k)}$. Also, when assuming that the observations are more accurate than the background, the use of more observation locations in time means that we are constraining the estimate of the initial state more tightly to the reference state in the twin experiment design. The effect this has on the convergence of the optimisation methods will be investigated. We next outline the algorithmic choices we have made.

5.5.3 Algorithmic choices

Stopping criteria We now outline the criteria used to terminate Algorithms 5.3.1, 5.4.1 and 5.4.2. Due to the limited time and computational cost available in practice, the GN method is not necessarily run to convergence and a stopping criterion is used to limit the number of iterations. Each residual vector calculation requires the non-linear model to be run forward to obtain the state at each observation location in time. This can then be used to calculate the value of the objective function. Furthermore, one run of the adjoint model is required to calculate the gradient.

To reduce computational cost in practical implementations of 4D-Var, the Jacobian matrix is evaluated at a lower resolution than the objective function (5.4) when solving the inner loop problem [40]. However, as the dimension of the problems used within this paper are relatively small compared to DA systems in practice, we use the full resolution residual and Jacobian given in (5.6) and solve the inner loop problem using MATLAB's backslash operator where an appropriate solver is chosen according to the properties of the Hessian matrix $\nabla^2\mathcal{J}(\mathbf{v})$ (see [86] for more details). The limit on the total number of function and Jacobian evaluations is achieved by using the following criterion

$$k_J + l \leq \tau_e, \tag{5.24}$$

where k_J is the total number of Jacobian evaluations (which is equivalent to the number of outer iterations k in 4D-Var), l is the total number of function evaluations and τ_e is the tolerance. The tolerance τ_e can be chosen according to the maximum number of evaluations desired. We note that for GN, $k_J = l$ as the method requires as many Jacobian evaluations as function evaluations. However, for both LS and REG there could be more than one function evaluation per Jacobian evaluation since for unsuccessful steps, the Jacobian is not updated so $k_J \leq l$.

To ensure that the algorithms are stopped before the function values stagnate, the following common termination criterion based on the relative change in the function at each iteration is also used

$$\frac{|\mathcal{J}(\mathbf{v}^{(k-1)}) - \mathcal{J}(\mathbf{v}^{(k)})|}{1 + \mathcal{J}(\mathbf{v}^{(k)})} \leq \tau_s, \tag{5.25}$$

for $k \geq 1$, where τ_s is the tolerance, chosen to be 10^{-5} and (5.25) is used throughout Section 5.6 unless indicated otherwise.
We expect the norm of the gradient of the objective function, $\|\nabla\mathcal{J}(\mathbf{v}^{(k)})\|$ to be close to zero at a stationary point. The following termination criterion will be used in Section 5.6.2 to identify whether or not a given method has located a stationary point

$$\|\nabla\mathcal{J}(\mathbf{v}^{(k)})\| \leq 10^{-5}. \tag{5.26}$$

Parameter choices For the LS method, we choose $\alpha_0 = 1$ so that the first step assessed by the bArmijo rule is the GN step. We set $\beta = 0.1$ and to adjust the step length, $\tau = 0.5$.

For the REG method, we select the typical choice of the initial regularisation parameter, $\gamma^{(0)} = 1$ as well as $\eta_1 = 0.1$ and $\eta_2 = 0.9$ to assess how well the model (5.16) approximates the true function value at the next iteration.

For all three optimisation methods, we set $\tau_e = 8, 100$ or 1000 depending on the experiment. The choice of $\tau_e = 8$ comes from that which is used operationally in the ECMWF Integrated Forecasting System [34], whereas the choice of $\tau_e = 100$ or 1000 is used to measure the performance of the optimisation methods when closer to convergence.

In order to best present our results, we use accuracy profiling described as follows.

Accuracy profiles An *accuracy profile* shows the proportion of problems a given method can solve within a fixed amount of work (τ_e) and a given tolerance (τ_f) of the change in the function value [90]. To ensure the robustness of our results, we apply the three optimisation methods to a series of n_r randomly generated problems, where the randomness occurs through the background and observation error vectors, $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$. For each realisation, a new $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$ are generated from their respective distributions, (5.21) and (5.22). The following criterion proposed in [90] is used to flag that an estimate of the initial state has been obtained by an optimisation method

$$\frac{\mathcal{J}(\mathbf{v}_0^{(l)}) - \mathcal{J}(\mathbf{v}_0^t)}{\mathcal{J}(\mathbf{v}_0^{(0)}) - \mathcal{J}(\mathbf{v}_0^t)} \leq \tau_f, \tag{5.27}$$

where $\mathbf{v}_0^t$ is a solution of (5.4) referred to as the 'truth' and τ_f is the tolerance. The measure (5.27) compares the optimality gap $\mathcal{J}(\mathbf{v}_0^{(l)}) - \mathcal{J}(\mathbf{v}_0^t)$ relative to the best reduction $\mathcal{J}(\mathbf{v}_0^{(0)}) - \mathcal{J}(\mathbf{v}_0^t)$ [90]. This ensures that the 4D-Var problem is only flagged as solved by the optimisation method once the value of the objective function is within some error (τ_f) of the truth.
For our problems, the truth is unknown. We only know that, due to the nonlinearity of the 4D-Var problem, there may exist many values of $\mathbf{v}_0$ that could minimise (5.4) locally. We are interested in the estimate $\mathbf{v}_0^t$ that gives the greatest reduction in (5.4) that any of the three methods can obtain. Therefore, we set the truth to be the $\mathbf{v}_0^{(l)}$ obtained by any of the three methods that gives the smallest function value within the given number of evaluations. Using this criterion allows us to benchmark the methods against each other using accuracy profiles.

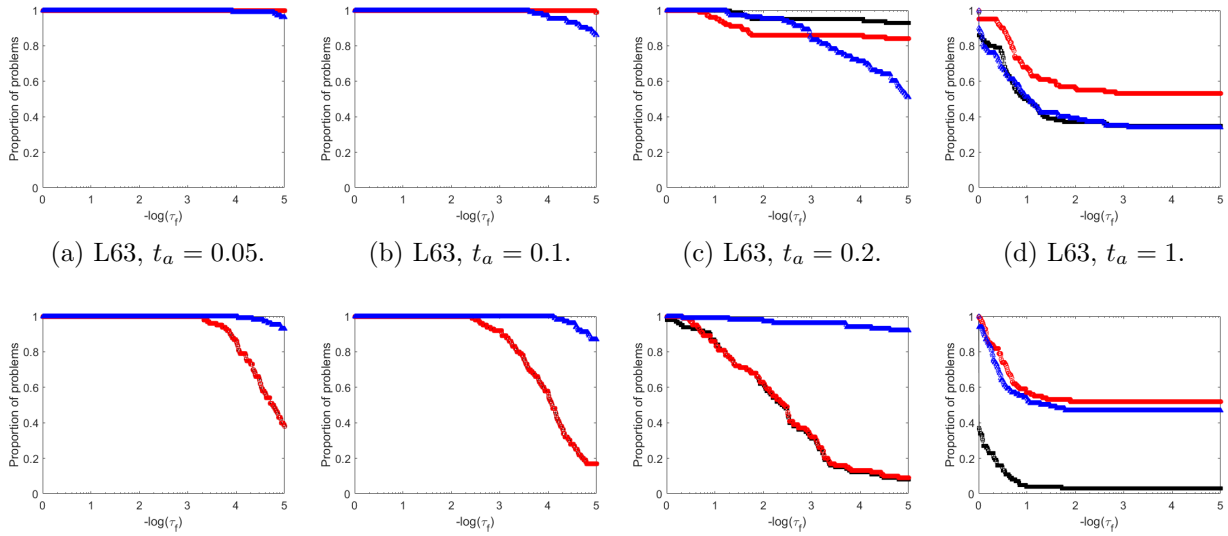
For each experiment, we plot the proportion of the n_r realisations solved by each method against the relative accuracy obtained, τ_f. The relative accuracy obtained is varied using $\tau_f = 10^{-i}$, where $i \in [0, 5]$.

5.6 Numerical results

In this section, we present the results when applying GN, LS and REG using the experimental design described in the previous section. We begin by conducting experiments showing the effect of the length of the assimilation time-window on the convergence of the three methods.

5.6.1 Effect of time-window length

We produce accuracy profiles for different time-window lengths to understand the effect this has on the convergence of each method while limiting the number of function and Jacobian evaluations to $\tau_e = 8$. We choose a background error of 50% and an observation error of 10% so that the ratio (5.23) is large relative to the other cases we consider. For both the L63 and L96 models, we consider both short and long time-window lengths of 6 hours ($t_a = 0.05$), 12 hours ($t_a = 0.1$), 1 day ($t_a = 0.2$) and 5 days ($t_a = 1$) with the results shown in Figure 5.1.



(a) L63, $t_a = 0.05$. (b) L63, $t_a = 0.1$. (c) L63, $t_a = 0.2$. (d) L63, $t_a = 1$.

(e) L96, $t_a = 0.05$. (f) L96, $t_a = 0.1$. (g) L96, $t_a = 0.2$. (h) L96, $t_a = 1$.

Figure 5.1: Accuracy profiles for the GN (black), LS (red) and REG (blue) methods applied to the L63 and L96 problems using different time-window lengths t_a. These show the proportion of $n_r = 100$ problems solved by each of the methods against the specified accuracy $-\log(\tau_f)$ when $\tau_e = 8$. The GN line is not visible in (a), (b), (e), (f) and (g) as it is printed beneath the LS line.

From Figure 5.1, we see that as the length of the time-window of both the L63 and L96 problems is increased, the performance of the GN, LS and REG methods suffers.

For the L63 problems, Figures 5.1(a) and 5.1(b) show that GN and LS perform similarly and solve more problems to the highest accuracy than REG. However, as τ_f is increased to $10^{-3.5}$, REG is solving all of the problems, so the REG estimate must be close to that of GN and LS. In Figure 5.1(c), both LS and REG solve fewer problems compared to GN, even for relatively large choices of τ_f. However, there is a choice of τ_f where all three methods are solving all problems, again indicating that the LS and REG estimates are close to the GN estimate. The initial guess for the three methods (the background) appears to be close enough to the solution and so the GN step is able to attain a sufficient decrease in the objective function as predicted by its local convergence properties. LS and REG are inadvertently shortening the GN step, which is a good step in the short time-window case. As we know, LS and REG need to adjust their respective parameters, $\alpha^{(k)}$ and $\gamma^{(k)}$ to attain GN's fast convergence, so LS and REG are requiring more evaluations than GN to achieve the same result. For the L96 short time-window results in Figures 5.1(e), 5.1(f) and 5.1(g), this is not the case. In fact, REG is outperforming GN and LS and it appears that LS is mimicking the behaviour of GN quite closely as the GN step is attaining a sufficient decrease in the objective function. However the decrease that the REG step is achieving appears to be much greater for the L96 problems. Therefore, REG is able to solve a greater number of problems within a higher level of accuracy, which explains the difference between the L63 results in Figures 5.1(a), 5.1(b) and the L96 results in 5.1(e) and 5.1(f).

The long time-window results for the L63 and L96 problems are shown in Figures 5.1(d) and 5.1(h), respectively. In both figures, LS is outperforming GN. For the L63 problems, the performance of GN does not differ much from the performance of REG. However, comparing the performance of GN in 5.1(c) with 5.1(d), we can see that performance of GN has deteriorated greatly when increasing the length of the time-window. In fact, in the results where even longer time-windows are used (not included here), LS and REG outperform the GN method for the L63 problems, as in 5.1(h).

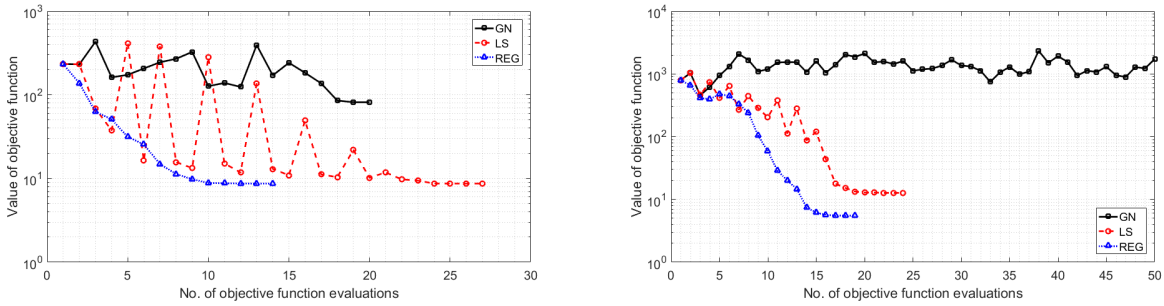
For the remainder of our experiments, we decide to use $t_a = 1$ to consider a long time-window case only.

5.6.2 Behaviour of methods and divergence of GN

Figure 5.2 shows the convergence plots for two typical cases when using the GN, LS and REG methods to obtain a solution to the preconditioned 4D-Var problem with the L63 and L96 models. In this figure, the total number of function and Jacobian evaluations allowed is set to $\tau_e = 100$ for both the L63 and the L96 problems to see if any progress is made beyond the number of evaluations allowed in practice. We recall that GN updates the gradient (5.8) when the function (5.4) is updated, so there are as many function evaluations as Jacobian evaluations. However, both LS and REG only update the Jacobian on successful iterations when there is a reduction in the objective function. Therefore, the total number of evaluations used by each of the methods could consist of a different combination of function and Jacobian evaluations. As in Section 5.6.1, we set the ratio (5.23) to be large. It is in this case that we are able to best demonstrate the benefit of the globally convergent methods, LS and REG. In Figure 5.2, we set $\tau_s = 10^{-3}$ to ensure that the methods stop before the function values stagnate. As Figure 5.2 includes function evaluations for both successful and unsuccessful step calculations, it is natural to see jumps in the function values of LS and REG while their parameters, $\alpha^{(k)}$ and $\gamma^{(k)}$ are being adjusted to guarantee a reduction in the function.

For the L63 problems (Figure 5.2(a)), all three methods stop when the relative change in the function criterion (5.25) is satisfied and before the limit on the total number of function and Jacobian evaluations (5.24) is met. Table 5.1 supports this figure by showing the algorithmic output for each of the GN, LS and REG methods when two different stopping criteria are used. From these results, we see that both LS and REG stop at the same function value, although REG requires fewer evaluations to do so, and that GN is converging towards a larger value of the objective function (5.4) than LS and REG. By instead stopping on the criterion (5.26) and setting $\tau_e = 1000$, we see in Table 5.1 that all three methods are still making progress on the gradient and iterate level, indicating that the methods are in fact locating stationary points despite a small change in the function value beyond those shown in Figure 5.2.

For the L96 problems (Figure 5.2(b)), LS and REG stop when (5.25) is satisfied and before (5.24) is satisfied, whereas GN only satisfies (5.24). Table 5.2 supports this figure by show-



(a) L63, Nobs1, $\sigma_b^2 = 25$, $\mathbf{B} = \sigma_b^2\mathbf{I}$, $t_a = 1$, $\tau_e = 100$. (b) L96, Nobs1, $\sigma_b^2 = 6.25$, $\mathbf{B} = \sigma_b^2\mathbf{I}$, $t_a = 1$ $\tau_e = 100$.

Figure 5.2: Convergence plots showing the value of the objective function at each iteration (including unsuccessful iterations) of the GN (black), LS (red) and REG (blue) methods when applied to a L63 problem (a) and a L96 problem (b).

Table 5.1: Table of algorithmic output when applying, GN, LS and REG to a L63 problem.

Criteria	Method	l	k_J	$\mathcal{J}(\mathbf{v}^{(k_J)})$	$\|\mathbf{v}^{(k_J)} - \mathbf{v}^{(k_J-1)}\|$	$\|\nabla\mathcal{J}(\mathbf{v}^{(k_J)})\|$
(5.25)	GN	20	20	81.55	0.42	86.35
	LS	27	14	8.69	0.03	5.18
	REG	14	14	8.69	0.05	1.00
(5.26)	GN	101	101	78.87	3.54^{-8}	8.47^{-6}
	LS	43	27	8.69	8.21^{-7}	8.31^{-6}
	REG	66	66	8.69	7.34^{-7}	9.24^{-6}

ing the algorithmic output for each of the GN, LS and REG methods when two different stopping criteria are used. From these results, we see that GN is in fact diverging while the LS method is stopping at a larger value of the objective function (5.4) than REG. Recall that the norm of the gradient criterion (5.26) can be used to identify whether or not a given method has located a stationary point. The values of $\|\nabla\mathcal{J}(\mathbf{v}^{(k_J)})\|$ for LS and REG when criterion (5.25) is used are much smaller than that of GN, although our results when we instead stop on the criterion (5.26) and set $\tau_e = 1000$ do not indicate that the estimates of LS and REG may indeed be stationary points of the objective function as they did for the L63 problems. However, the gradient norms of LS and REG are reducing as the methods iterate, unlike GN, which also diverges at gradient level.

Table 5.2 shows that as LS and REG iterate beyond what is shown in Figure 5.2(b), there is very little change in the value of the cost function, despite making some change on the iterate and/or gradient level. The effect of rounding error means that although we see progress made, the function value may remain stagnant because of limitations in computer precision and because of the conditioning of the problem. The condition number of the Hessian $\kappa(\mathbf{S})$ can be used to indicate the accuracy we could be able to achieve. For the L63 problems, the condition number is 58.14 and so we can expect to lose 2 figures of accuracy due to loss of arithmetic precision. For the L96 problems, the condition number is 7010.10 and so we can expect to lose 4 figures of accuracy. These values of $\kappa(\mathbf{S})$ are reasonable relative to the

Table 5.2: Table of algorithmic output when applying, GN, LS and REG to a L96 problem.

Criteria	Method	l	k_J	$\mathcal{J}(\mathbf{x}^{(k_J)})$	$\|\mathbf{v}^{(k_J)} - \mathbf{v}^{(k_J-1)}\|$	$\|\nabla\mathcal{J}(\mathbf{v}^{(k_J)})\|$
(5.25)	GN	50	50	1728.99	20.02	5758.47
	LS	24	14	12.72	0.07	10.09
	REG	19	16	5.52	0.08	1.89
(5.26)	GN	500	500	960.32	15.88	8015.13
	LS	967	32	12.71	0	10.09
	REG	967	32	5.51	0	0.03

accuracy we are finding.

The observed behaviour in this section is partly due to the fact that there is no mechanism in GN to force it to converge as there is in LS and REG. The benefit of these mechanisms is clearly shown in Figure 5.2(b) where the GN method is diverging while the LS and REG methods are converging, further motivating our investigation of these methods.

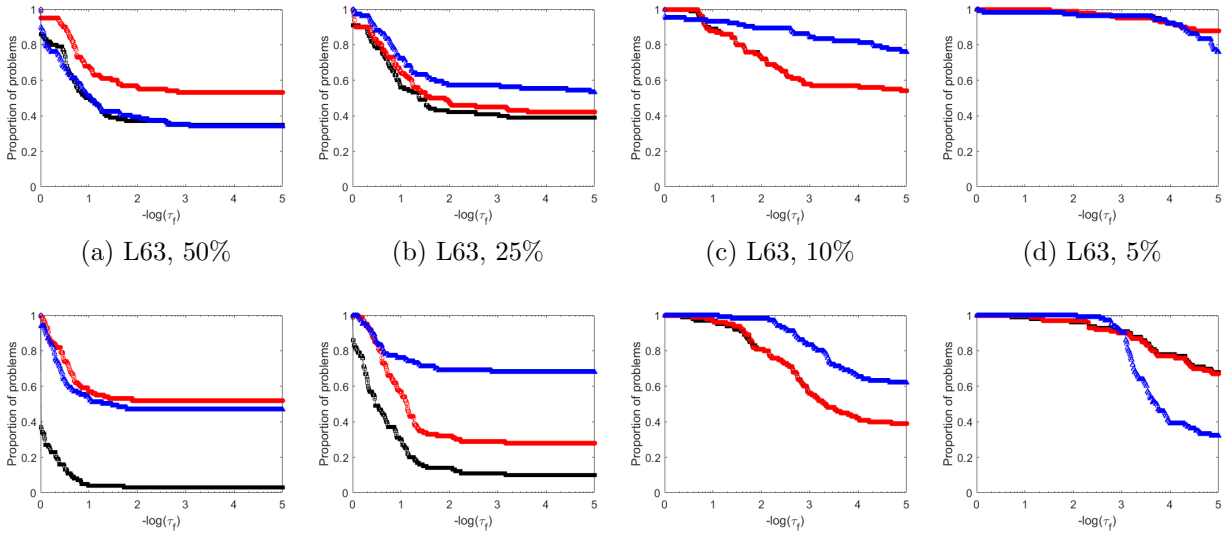
5.6.3 Effect of background error variance

In this section, we study the effect on the performance of the three methods when the amount of uncertainty in the background information is increased whilst the amount of uncertainty in the observations is fixed. Figure 5.3 shows the accuracy profiles used to benchmark the performance of the GN, LS and REG methods as the tolerance τ_f is reduced, where $\tau_e = 8$, while Figure 5.4 allows τ_e to increase for both models with the increase chosen relative to the dimension of the models, i.e. a larger increase in τ_e is allowed for the L63 problems, where $n = 3$, than the L96 problems, where $n = 40$. From both these figures, we generally see that as the error in the background is reduced, the performance of all three methods improves. The conditioning of the problem has been shown to depend on the ratio of the standard deviations of the background and observation errors (5.23) [56, 57]. Therefore, the estimate obtained by any of the optimisation methods may not be accurate enough to produce a reliable forecast if the ratio (5.23) is large. The accuracy of the estimate obtained by each method will be investigated further later on in the paper.

Figures 5.3(a) and 5.3(e) show that a globally convergent method is able to find a smaller function value than GN. As the ratio (5.23) is reduced, from Figures 5.3(b), 5.3(c), 5.3(f) and 5.3(g) we see that the REG method is able to solve the most problems at the highest level of accuracy. When there is less uncertainty in the background versus the observations, Figure 5.3(d) shows that for the L63 problems, all three methods are solving close to all of the problems within a high level of accuracy. This is because the three methods are not able to solve a large portion of the cases when the problem is ill-conditioned, which could explain this result. However, for the L96 problems Figure 5.3(h) shows that the GN and LS methods are solving the majority of the problems and REG is not performing as well at higher levels of accuracy. We can see the performance of REG improving for the L96 problems when more evaluations are allowed in Figure 5.4(h).

Figure 5.4 shows a much greater difference between the globally convergent methods and GN when the background error is larger than the observation error. In Figures 5.4(a), 5.4(b), 5.4(e) and 5.4(f), it appears that when more evaluations are allowed, GN is diverging in the case when σ_b is large while the globally convergent methods are able to locate estimates of the initial states for the preconditioned 4D-Var problem, which when compared to GN, better minimise the objective function (5.4). When the background error is the same as the observation error in Figure 5.4(c), it is GN that is performing better than LS and REG for the L63 problems. For LS, this could be because LS is unnecessarily shortening the GN step, causing slower convergence. For the REG method, the regularisation parameter must be shrunk and therefore, REG requires more iterations to benefit from GN's fast convergence property.

In Figure 5.4(d), all three methods are solving essentially the same number of problems, with a slight decrease in success for REG, that again could be due to the need to adjust the regularisation parameter. For the L96 problems, we see a slightly different result. Figures 5.4(g) and 5.4(h) show that a globally convergent method is solving more problems, more accurately than GN despite the background error being at most equal to the observation error. This is an interesting result for this higher-dimensional model as we would expect GN to locally converge at a faster rate than the globally convergent methods due to the fact that GN does not need to adjust any parameters; however, we find this not to be the case.

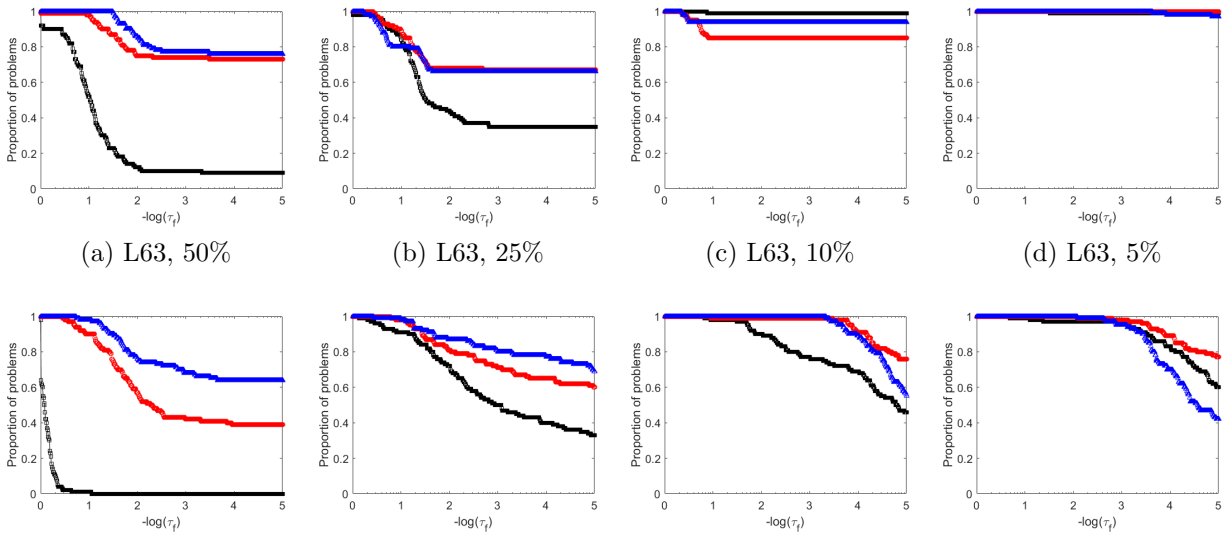


(a) L63, 50% (b) L63, 25% (c) L63, 10% (d) L63, 5%

(e) L96, 50% (f) L96, 25% (g) L96, 10% (h) L96, 5%

Figure 5.3: Accuracy profiles for the GN (black), LS (red) and REG (blue) methods applied to the L63 problems in (a)-(d) and the L96 problems in (e)-(h) where $n_r = 100$, $\tau_e = 8$ and where there is one observation at the end of the time-window. The observation error is 10% and the background error is varied above and below this, as indicated in the plot captions. The GN line is not visible in (c), (d), (g) and (h) as it is printed beneath the LS line.

In DA, we are interested in knowing the accuracy of the estimate obtained as in applications such as NWP, the estimate is used as the initial conditions for a forecast and so the quality of this forecast will depend on the errors in the estimate. In the following section, we quantify and compare the errors in the estimates obtained by each method.



(a) L63, 50% (b) L63, 25% (c) L63, 10% (d) L63, 5%

(e) L96, 50% (f) L96, 25% (g) L96, 10% (h) L96, 5%

Figure 5.4: Accuracy profiles for the GN (black), LS (red) and REG (blue) methods applied to the L63 problems where $\tau_e = 1000$ in (a)-(d) and the L96 problems where $\tau_e = 100$ in (e)-(h). We set $n_r = 100$ and there is one observation at the end of the time-window. The observation error is 10% and the background error is varied above and below this, as indicated in the plot captions.

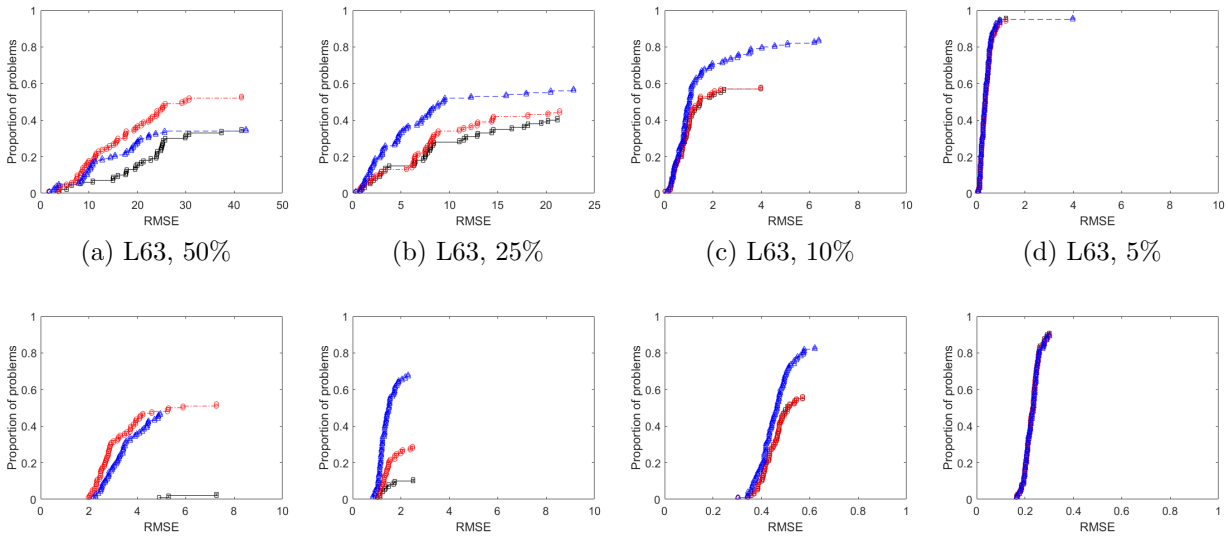
5.6.4 Quality of the analysis

We recall that the initial guess of the algorithms is the reference state $\mathbf{x}_0^{ref}$ perturbed by the background error $\varepsilon_{\mathbf{b}}$. In order to compare the quality of the estimate obtained by each method, we compare their estimate to the reference state $\mathbf{x}_0^{ref}$ to understand how far the estimates obtained by the methods have deviated from this. The analysis error for each state variable is given by $\varepsilon_i^a = x_i^a - x_i^{ref}$. For each realisation, we calculate the root mean square error (RMSE) of the analysis error, which is the difference between the reference state and the estimate obtained by each method,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\varepsilon_i^a)^2}{n}}. \tag{5.28}$$

For each method, we plot the percentage of problems solved (according to the criterion (5.27) where $\tau_f = 10^{-3}$) within a specified tolerance of the RMSE (5.28). We acknowledge in this work that the code for the RMSE profiles has been adapted from the code for the data profiles used in [90].

The results for the L63 and L96 problems are in Figure 5.5, which coincides with the case shown in Figure 5.3 where $\tau_f = 10^{-3}$. From this, we see that the GN method solves fewer problems within the same level of RMSE accuracy as LS and REG when the background error is large in Figures 5.5(a), 5.5(b), 5.5(e) and 5.5(f). Furthermore, we see how the RMSE of the analyses successfully found by each method reduces as the background error variance is reduced, this can be seen in the scale of the x axis in Figures 5.5(a), 5.5(b), 5.5(c) and 5.5(d) for the L63 problems and Figures 5.5(e), 5.5(f), 5.5(g) and 5.5(h) for the L96 problems. For both models, the concentration of points in Figures 5.5(a) and 5.5(e) shows us that the LS method is solving more problems than GN and REG within the same RMSE tolerance. A similar result can be seen for REG in Figures 5.5(b), 5.5(c), 5.5(f) and 5.5(g). In Figures 5.5(d) and 5.5(h), we see that all three methods are performing similarly, the RMSE errors for each of the analyses are very close together.



(a) L63, 50% (b) L63, 25% (c) L63, 10% (d) L63, 5%

(e) L96, 50% (f) L96, 25% (g) L96, 10% (h) L96, 5%

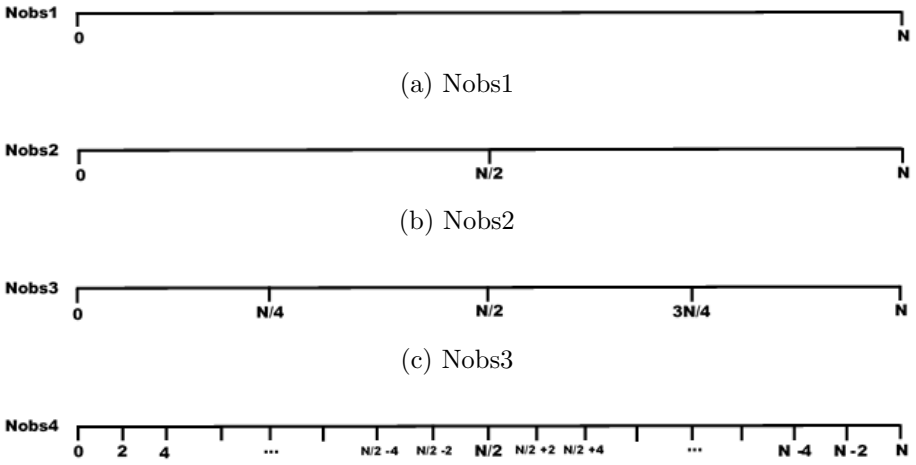
Figure 5.5: RMSE plots for the GN (black), LS (red) and REG (blue) methods applied to the L63 problems in (a)-(d) and the L96 problems in (e)-(h) where $n_r = 100$, $\tau_e = 8$, $\tau_f = 10^{-3}$ and where there is one observation at the end of the time-window. The observation error is 10% and the background error is varied above and below this, as indicated in the plot captions.

Including more observations constrains the problem closer to the reference state when the observation error is small. We next show the effect on the performance of the methods as we include more observations and see if this gives any improvement in the performance of the methods when the background error is much larger than the observation error.

5.6.5 Effect of observations

Within this section, we show how the use of more observation locations in time affects the performance of the three methods. We take the worst case for the three methods when there

is a 50% error in the background and see if including more observations in time with a 10% error affects the performance of the methods. For both models, we consider only equally spaced observations in time, one set of observations at time N (Nobs1), times $N/2$ and N (Nobs2), times $N/4, N/2, 3N/4$ and N (Nobs3) and the even time points (Nobs4). These choices can be visualised in Figure 5.6. For the Nobs1 case, observations are based on the reference state at the end of the time-window and more observations are included over time in the Nobs2, Nobs3 and Nobs4 cases. This not only increases the condition number of the problem but also constrains the estimate more tightly to the reference state.



(a) Nobs1

(b) Nobs2

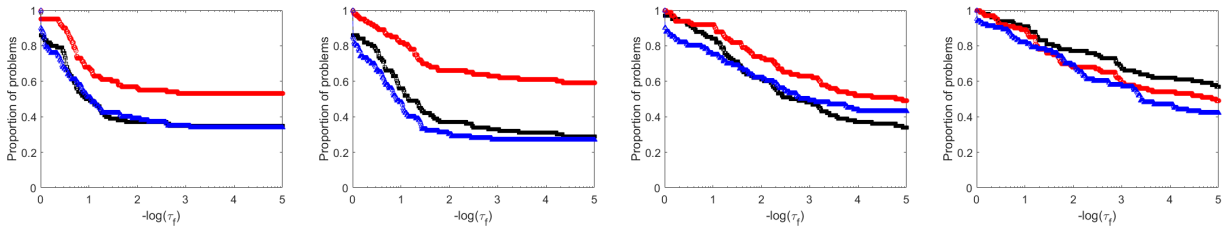
(c) Nobs3

(d) Nobs4

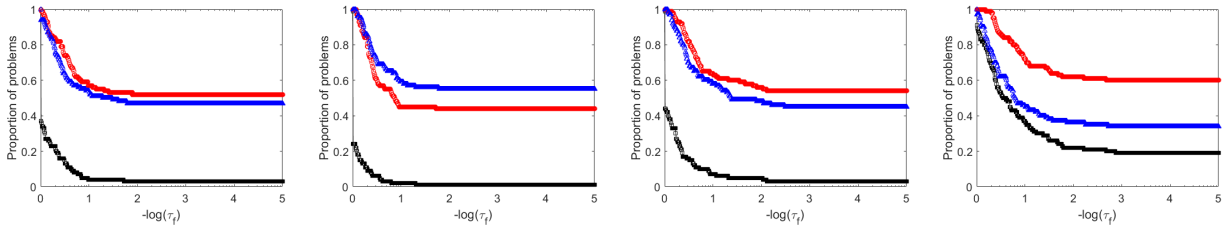
Figure 5.6: Observation locations schematic where N is the length of the time-window.

For the L63 problems from Figures 5.7(a), 5.7(b), 5.7(c) and 5.7(d), we see that as the number of observation locations in time is increased, the performance of all three methods is improved. All three methods are solving more problems at a higher level of accuracy as the number of observation locations in time is increased. This is more apparent when more evaluations are allowed as shown in Figure 5.8(a), 5.8(b), 5.8(c) and 5.8(d). Here, the performance of GN improves drastically between the Nobs1 and Nobs2 cases (Figures 5.7(a) and 5.7(b)) while there is less significant change in the behaviour of LS and REG. In Figure 5.7(d), we see that GN is able to solve more problems than LS and REG. Again, this could be because the LS and REG methods require more iterations to converge when GN is performing well due to the need to adjust their parameters. This argument coincides with Figure 5.8(d) where more evaluations are allowed and the LS and REG methods are able to perform as well as or better than GN due to the use of more evaluations. For the L96 problems, we see a different result. From Figure 5.7, we only see a significant improvement in the performance of GN in the Nobs4 case (Figure 5.7(h)). Otherwise, there is little effect. This conclusion can also be drawn from Figure 5.8(g) and 5.8(h) where more evaluations are allowed.

Similar studies were carried out on the performance of GN, LS and REG when applied to the preconditioned 4D-Var problem where we instead choose $\mathbf{B} = \sigma_b^2 \mathbf{C}_B$, where $\mathbf{C}_B$ is

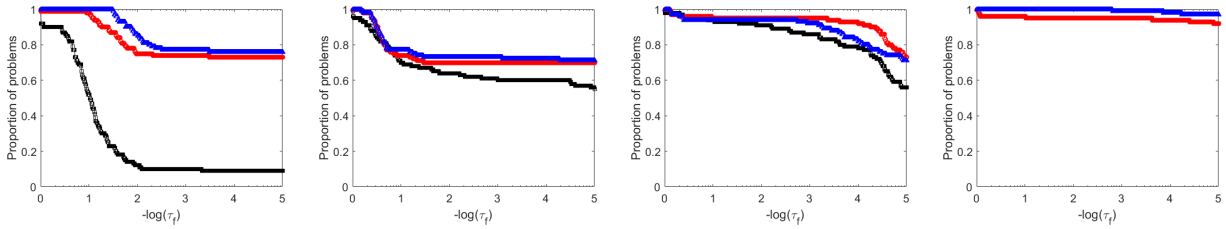


(a) L63, Nobs1 (b) L63, Nobs2 (c) L63, Nobs3 (d) L63, Nobs4

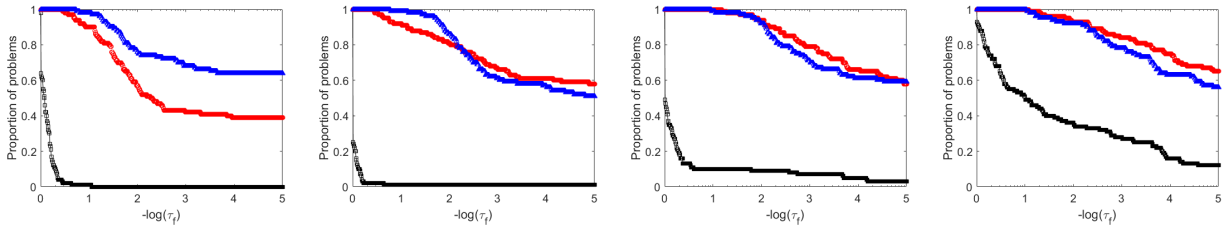


(e) L96, Nobs1 (f) L96, Nobs2 (g) L96, Nobs3 (h) L96, Nobs4

Figure 5.7: Accuracy profiles where $n_r = 100$ and $\tau_e = 8$ for the L63 problems in (a)-(d) and the L96 problems in (e)-(h) for different observation locations in time, as indicated in the plot captions, where the background error is 50% and the observation error is 10%.



(a) L63, Nobs1 (b) L63, Nobs2 (c) L63, Nobs3 (d) L63, Nobs4



(e) L96, Nobs1 (f) L96, Nobs2 (g) L96, Nobs3 (h) L96, Nobs4

Figure 5.8: Accuracy profiles where $n_r = 100$ for the L63 problems where $\tau_e = 1000$ in (a)-(d) and the L96 problems where $\tau_e = 100$ in (e)-(h) for different observation locations in time, as indicated in the plot captions, where the background error is 50% and the observation error is 10%. The GN line is not visible in (d) as it is printed beneath the LS line.

a correlation matrix; similar conclusions are drawn but due to space constraints, are not included within this paper.

5.7 Conclusion

We have shown that the globally convergent methods, LS and REG, have the capacity to improve the current estimate of the DA analysis that we have within the limited time and cost available in DA, through the use of safeguards within GN which is guaranteed to converge from initial guesses close enough to a local minimum.

Using the L63 and L96 models in the preconditioned 4D-Var framework, we have shown that when there is more uncertainty in the background information compared to the observations, the GN method may diverge in the long time-window case yet the globally convergent methods LS and REG are able to improve the estimate of the initial state. We compare the quality of the estimate obtained using the RMSE of the analysis and show that even in the case where the background is highly inaccurate compared to the observations, the globally convergent methods find estimates with an RMSE less than or equal to the RMSE of the estimates GN obtains. We take the case where the background is highly inaccurate compared to the observations and find that the convergence of all three methods is improved when more observations are included along the time-window. In addition to the numerical results, the assumptions made in the global convergence theorems of both LS and REG when applied to a general nonlinear least-squares problem and a discussion as to whether these assumptions are satisfied in DA is presented in the appendix. We note that preconditioning of the Hessian is not necessary for these results to hold, although this is the case we have focused on within our work.

Our findings are important in DA as they show that in cases where the accuracy of the prior information is poor and when there is limited computational cost, the globally convergent methods are still able to minimise the 4D-Var objective function, unlike GN. We recommend that these methods are tested on DA problems with realistic models and for different applications to understand if these conclusions hold. In particular, one should consider such problems where an accurate initial guess for the algorithms is unavailable and a long assimilation time-window is used, as we found that it is in this case that LS and REG have an advantage over GN.

Within this paper, the 4D-Var inner loop problem is solved exactly. In practice this must be solved inexactly, due to the size of the control vector, and by the use of approximations to meet the computational and time constraints. This is a common area of research in the DA community in order to improve the quality of the assimilation analysis as well as the speed of convergence of the algorithms and is something we will consider in future work. Furthermore, in the case where GN performs better than LS and REG, further research is needed on updating the globalisation parameters (stepsize $\alpha^{(k)}$ and regularisation parameter $\gamma^{(k)}$) to speed up convergence.

Acknowledgements This work has been funded in part by the UK Engineering and Physical Sciences Research Council (EPSRC) Centre for Doctoral Training in Mathematics of Planet Earth, the University of Reading EPSRC studentship (part of Grant/Award Number: EP/N509723/1) and by the NERC National Centre for Earth Observation.

Declarations of interest None.

5.8 Appendix: Convergence theorems

In this section, we outline the global convergence theorems for the LS and REG methods and discuss whether the assumptions made hold in DA. We first state the definitions of a local and global minimum of an optimisation problem $\min_{\mathbf{v}\in\mathbb{R}^n} f(\mathbf{v})$ where $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^n$.

Definition 5.8.1 (Local minimiser [96])**.** *A point $\mathbf{v}^*$ is a local minimiser of f if there is a neighbourhood $\mathcal{N}$ of $\mathbf{v}^*$ such that $f(\mathbf{v}^*) \leq f(\mathbf{v})$ for all $\mathbf{v} \in \mathcal{N}$.*

Definition 5.8.2 (Global minimiser [96])**.** *A point $\mathbf{v}^*$ is a global minimiser of $f : \mathbb{R}^n \to \mathbb{R}$ if $f(\mathbf{v}^*) \leq f(\mathbf{v})$ for all $\mathbf{v} \in \mathbb{R}^n$.*

A global solution is difficult to locate in most cases due to the nonlinearity of the problems. Therefore, a local solution is often sought by algorithms for nonlinear optimisation.

We focus on nonlinear least-squares optimisation problems of the form (5.5) for the remainder of this section. The GN method can only guarantee local convergence under certain conditions and not necessarily global convergence. This is dependent on how close the initial guess is from the local minimum the algorithm locates and whether or not the residual vector $\mathbf{r}$ of (5.5) is a zero vector at a solution $\mathbf{v}^*$. Furthermore, the region of local convergence depends on problem constants not known a priori, such as Lipschitz constants of the gradient.

A local convergence result for the GN method can be found in Theorem 10.2.1 of [33] where the performance of GN is shown to be dependent on whether or not the second-order terms in (5.9) evaluated at the solution $\mathbf{v}^*$ are close to zero. Another local convergence result can be found in Theorem 4 of [52] where GN is treated as an inexact Newton method. The theorem guarantees convergence of the GN method if for each iteration $k = 0, 1, \ldots,$ the norm of the ratio of $\mathbf{Q}(\mathbf{v}^{(k)})$ and $\mathbf{J}(\mathbf{v}^{(k)})^T\mathbf{J}(\mathbf{v}^{(k)})$, the second and first terms of (5.9) respectively, is less than or equal to some constant $\hat{\eta}$ where $0 \leq \hat{\eta} \leq 1$.

It is important to note here that the globally convergent methods we are concerned with, namely LS and REG, can only guarantee global convergence to a local minimum under certain conditions and not necessarily to a global minimum.

Before we list the assumptions for the global convergence theorems, we first state the definition of the Lipschitz continuity property of a general function g as this is widely used in the theorems.

Definition 5.8.3 (Lipschitz continuous function (see [96] A.42))**.** *Let g be a function where $g : \mathbb{R}^n \to \mathbb{R}^m$ for general n and m. The function g is said to be Lipschitz continuous on some set $\mathcal{N} \subset \mathbb{R}^n$ if there exists a constant $L > 0$ such that,*

$$\|g(\mathbf{v}) - g(\mathbf{w})\| \leq L\|\mathbf{v} - \mathbf{w}\|, \qquad \forall \mathbf{v}, \mathbf{w} \in \mathcal{N}. \tag{5.29}$$

The following assumptions are used to prove global convergence of both the LS and REG methods.

A10. $\mathbf{r}$ *is uniformly bounded above by* $\omega > 0$ *such that* $\|\mathbf{r}(\mathbf{v})\| \leq \omega$.

A11. $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *is Lipschitz continuous on* $\mathbb{R}^n$ *with Lipschitz constant* $L_r > 0$.

A12. $\mathbf{J}$ *is Lipschitz continuous on* $\mathbb{R}^n$ *with Lipschitz constant* $L_J > 0$.

We remark that for the LS method, we can weaken assumptions A11 and A12 using the open set $\mathcal{N}$ containing the level set

$$\mathcal{L} = \left\{ \mathbf{v} \in \mathbb{R}^n | \mathcal{J}(\mathbf{v}) \leq \mathcal{J}(\mathbf{v}^{(0)}) \right\}. \tag{5.30}$$

In order to achieve the sufficient decrease property of the LS method, the following assumption must be made.

A13. $\mathbf{J}(\mathbf{v})$ *in* (5.6) *is uniformly full rank for all* $\mathbf{v} \in \mathbb{R}^n$*, that is, the singular values of* $\mathbf{J}(\mathbf{v})$ *are uniformly bounded away from zero, so there exists a constant* ν *such that* $\|\mathbf{J}(\mathbf{v})\mathbf{z}\| \geq \nu\|\mathbf{z}\|$ *for all* $\mathbf{v}$ *in a neighbourhood* $\mathcal{N}$ *of the level set* $\mathcal{L}$ *where* $\mathbf{z} \in \mathbb{R}^n$.

In 4D-Var practice, it is reasonable to assume that the physical quantities are bounded. Therefore, we can say that both $\mathbf{x}_0 - \mathbf{x}^b$ and the innovation vector $\mathbf{y} - \mathcal{H}(\mathbf{x})$ are bounded in practice, thus satisfying assumption A10. In 4D-Var, we must assume that the nonlinear model $\mathcal{M}_{0,i}$ is Lipschitz continuous in order for A11 to hold. As discussed in [87], this is a common assumption made in the meteorological applications. However, we cannot say that this is necessarily the case in 4D-Var practice.

In order for the Jacobian $\mathbf{J}$ to be Lipschitz continuous, we require its derivative to be bounded above by its Lipschitz constant. Therefore, for assumption A12 to hold, we require $\mathbf{r}$ to be twice continuously differentiable in practice, which is a common assumption made in 4D-Var, and also, that these derivatives of $\mathbf{r}$ are bounded above.

As mentioned in Section 5.3, the preconditioned 4D-Var Hessian (5.10) is full rank by construction as it consists of the identity matrix and a non-negative definite term. Therefore, the Jacobian of the residual of the preconditioned problem in (5.6) is full rank and assumption A13 holds. This is also the case for the standard 4D-Var problem (5.1), because of the presence of $\mathbf{B}^{1/2}$ in its Jacobian.

We now outline the global convergence theorems for the LS and REG methods, using these assumptions.

5.8.1 Global convergence of the LS method

Nocedal et al. outline the proof for the GN method with Wolfe line search conditions in [96], which uses the Zoutendijk condition. Alterations to this proof can be made to prove the global convergence theorem of the LS method, Algorithm 5.4.1, given as follows.

Theorem 5.8.1 (Global convergence for the Gauss-Newton with bArmijo line search method, Algorithm 5.4.1)**.** *Suppose we have a function* $\mathcal{J} = \frac{1}{2}\mathbf{r}^T\mathbf{r}$ *and its gradient* $\nabla\mathcal{J} = \mathbf{J}^T\mathbf{r}$ *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A10 - A13 hold. Then if the iterates* $\{\mathbf{v}^{(k)}\}$ *are generated by the GN method with stepsizes* $\alpha^{(k)}$ *that satisfy the Armijo condition* (5.13)*, we have*

$$\lim_{k\to\infty} \mathbf{J}(\mathbf{v}^{(k)})^T\mathbf{r}(\mathbf{v}^{(k)}) = 0. \tag{5.31}$$

That is, the gradient norms converge to zero, and so the Gauss-Newton method with bArmijo line search is globally convergent.

The proof of Theorem 5.8.1 requires the bArmijo chosen stepsizes $\alpha^{(k)}$ to be bounded below, which can be derived using assumptions A10 - A12. Using this lower bound, as well as assumption A13, we are able to prove the Zoutendijk condition (as in [96]) and its variant

$$\sum_{k\geq 0} \cos(\theta^{(k)})\|\nabla\mathcal{J}(\mathbf{v}^{(k)})\|_2\|\mathbf{s}^{(k)}\|_2 < \infty \tag{5.32}$$

hold. Both the Zoutendijk condition and its variant (5.32) use the angle between $\mathbf{s}^{(k)}$ (the GN search direction) and $-\nabla\mathcal{J}(\mathbf{v}^{(k)})$ (the steepest descent direction), $\theta^{(k)}$, which is given by

$$\cos(\theta^{(k)}) = \frac{(-\nabla\mathcal{J}(\mathbf{v}^{(k)}))^T\mathbf{s}^{(k)}}{\|\nabla\mathcal{J}(\mathbf{v}^{(k)})\|_2\|\mathbf{s}^{(k)}\|_2}. \tag{5.33}$$

If this angle is uniformly bounded away from zero with k, one can show that GN with line search is a globally convergent method.

We will next present the global convergence theorem for the REG method. The REG method has no sufficient decrease condition as in the LS method. Therefore, the use of the level set (5.30) is not required. The assumptions for convergence are similar to the LS method aside from the requirement of $\mathbf{J}(\mathbf{v})$ being full rank.

5.8.2 Global convergence of the REG method

The global convergence theorem for the GN with quadratic regularisation method, Algorithm 5.4.2, is given as follows.

Theorem 5.8.2 (Global convergence for the Gauss-Newton with regularisation method, Algorithm 5.4.2)**.** *Suppose we have a function* $\mathcal{J} = \frac{1}{2}\mathbf{r}^T\mathbf{r}$ *and its gradient* $\nabla\mathcal{J} = \mathbf{J}^T\mathbf{r}$ *where* $\mathbf{r} \in \mathcal{C}^1(\mathbb{R}^n)$ *and* $\mathbf{J}$ *is the Jacobian of* $\mathbf{r}$*. Assume A10 - A12 hold. Then if the iterates* $\{\mathbf{v}^{(k)}\}$ *are generated by the Gauss-Newton with regularisation method, we have that*

$$\lim_{k\to\infty} \mathbf{J}(\mathbf{v}^{(k)})^T\mathbf{r}(\mathbf{v}^{(k)}) = 0. \tag{5.34}$$

That is, the gradient norms converge to zero, and so the Gauss-Newton method with regularisation is globally convergent.

We first note that some adaptations of the lemmas from the global convergence proof of the Adaptive Regularisation algorithm using Cubics (ARC method) are used to prove Theorem 5.8.2, see [23] and [24]. We begin the proof by deriving an expression for the predicted model decrease in terms of the gradient. We require the use of an upper bound on $\gamma^{(k)}$, denoted as $\gamma_{\max}$, which is derived using a property of Lipschitz continuous gradients. We show that $\gamma^{(k)} \leq \gamma_{\max}$ for all $k \geq 0$ by first showing that if $\gamma^{(k)}$ is large enough, then we have a successful step so that $\gamma^{(k)}$ can stop increasing due to unsuccessful steps in Algorithm 5.4.2. We use the expression for $\gamma_{\max}$ to prove global convergence of the REG method under assumptions A10-A12 by showing that the gradient norms converge to zero as we iterate.

Note that for both the LS and REG, if $\mathbf{r}(\mathbf{v}^{(k)}) \to 0$, i.e. (5.5) is a zero residual problem, then we have that (5.31) and (5.34) hold as $|\mathcal{J}(\mathbf{v}^{(k)})|$ is uniformly bounded. However, in practice the variational problem is not usually a zero residual problem.

5.9 Additional 4D-Var results

As mentioned in Section 5.6, similar studies where we vary the assimilation time-window length, the background error variance and the number of observations in time are carried out on the performance of GN, LS and REG when applied to the preconditioned 4D-Var problem where we instead choose $\mathbf{B} = \sigma_b^2 \mathbf{C}_B$, where $\mathbf{C}_B$ is a correlation matrix. In this section, we include a discussion on how these results compare to those in Section 5.6.

The experimental design is exactly as it is in Section 5.5, except we choose $\mathbf{B}$ to be of the form $\mathbf{B} = \sigma_b^2 \mathbf{C}_B$. For the L96 experiments, we choose $\mathbf{C}_B$ to be a SOAR correlation matrix defined in (4.100), where we choose $a = 1/2\pi$ such that the circumference of the circle is 1. The length scale of the correlation matrix controls the spread of errors along the grid. Increasing the correlation length scale results in an increase in the number of nearby background errors that are correlated with each other. For the correlation length-scales, we choose $L_1 = 0.5\Delta x$, $L_2 = \Delta x$ and $L_3 = 1.5\Delta x$, where $\Delta x = 1/n$ is the grid spacing in the L96 model. For the L96 problems, the background error covariance matrix with associated correlation length scale L_i is denoted by $\mathbf{B}_i = \sigma_b^2 \mathbf{C}_B$, where $i = 1, 2, 3$.

To study the effect of the presence of background error correlations on the L63 experiments, we use the following three symmetric positive definite correlation matrix structures of increasing cross correlations,

$$\mathbf{B}_1 = \begin{pmatrix} 1 & 1/8 & 1/16 \\ 1/8 & 1 & 1/8 \\ 1/16 & 1/8 & 1 \end{pmatrix}, \mathbf{B}_2 = \begin{pmatrix} 1 & 1/4 & 1/8 \\ 1/4 & 1 & 1/4 \\ 1/8 & 1/4 & 1 \end{pmatrix} \text{ or } \mathbf{B}_3 = \begin{pmatrix} 1 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1 \end{pmatrix}. \quad (5.35)$$

We begin by discussing the results of our experiments showing the effect of the length of the assimilation time-window on the convergence of GN, LS and REG for the correlated background error case.

5.9.1 Effect of time-window length

As we did in Section 5.6.1 for the uncorrelated $\mathbf{B}$, for each choice of $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$, we produce accuracy profiles for different time-window lengths to understand the effect this has on the convergence of GN, LS and REG while limiting the number of function and Jacobian evaluations to $\tau_e = 8$. We omit the plots from this section due to the strong similarities with those in Section 5.6.1 and instead comment on the differences.

For the $t_a = 0.05$, 0.1 and 0.2 cases of the L63 problems, the results for where a correlated $\mathbf{B}$ is used are almost identical to those in Figure 5.1 where an uncorrelated $\mathbf{B}$ is used. However, for the $t_a = 1$ (long time-window) case, the results differ slightly from those in Figure 5.1(d). Recall from Section 5.6.1 that it was LS that solved the most problems in the $t_a = 1$ case within the limited number of $\tau_e = 8$ evaluations, with GN and REG performing similarly. In all three cases of correlated $\mathbf{B}$, GN solves the least number of problems within a given accuracy τ_f. For the $\mathbf{B}_1$ case, we find that REG solves the most problems whereas for the $\mathbf{B}_2$ and $\mathbf{B}_3$ cases, we find that LS solves the most problems as in the uncorrelated $\mathbf{B}$ results in Figure 5.1(d). This change in results for the $\mathbf{B}_1$ long time window case is because there is no guarantee that LS and REG converge to the same local minimum of the cost function (5.4) due to the methods taking different steps. Therefore, it is possible that LS locates a smaller function value than REG, as in Figure 5.1(d), or vice versa. In either case, the conclusion still holds that a globally convergent method is performing better than GN in the long time-window case.

For the L96 problems, the results where $\mathbf{B}_1$ is used are very similar to those where uncorrelated error is used in Figure 5.1. There is a slight difference between the $t_a = 0.2$ correlated $\mathbf{B}$ cases and 5.1(g): GN performs slightly worse than LS when correlated error is used. There is also a slight difference when $t_a = 1$ between the $\mathbf{B}_1$ and $\mathbf{B}_2$ cases and 5.1(h): there is a greater difference between the performance of LS and REG when correlated error is used. We notice that in the short time-window length cases, as we strengthen the background error correlations, the performance of GN and LS improves. When $t_a = 0.05$, we find that the performance of GN and LS is better than REG for high levels of accuracy (when $\tau_f < 10^{-4}$) for the $\mathbf{B}_2$ and $\mathbf{B}_3$ cases. This is unlike what we find when $\mathbf{B}_1$ is used and in Figure 5.1(e) where an uncorrelated $\mathbf{B}$ is used.

Recall from Section 5.4 that, unlike REG, LS uses the same step calculation as GN in the case where $\alpha^{(k)} = 1$. As we choose $\alpha_0 = 1$, the GN step is the first step that is assessed by the Armijo condition (5.13). Therefore, if the GN step is a successful step according to the Armijo condition (5.13), the LS method would perform identically to GN. This is unlike REG, which requires its REG parameter $\gamma^{(k)}$ to be shrunk close to zero to achieve similar results to GN. Hence why REG is performing relatively worse than GN and LS as stronger background error correlations are introduced/as the performance of GN improves for the short time-window case. From these results, it is clear that there is no benefit in using a globalisation strategy in the short time-window length case as the GN method is able to solve the majority of problems within a high accuracy, without the added computational cost of tuning $\alpha^{(k)}$ in the LS method, or $\gamma^{(k)}$ in the REG method. However, as the

time-window length is increased in all three cases of correlated $\mathbf{B}$, the performance of REG improves and so for the long time-window case, LS and REG are again, the favourable choice.

We find that for each case of correlated $\mathbf{B}$ that as the length of the time-window increases, the performance of GN worsens, as it did in Figure 5.1, and the globally convergent methods are able to solve more problems within the same level of accuracy. These results are evidence that the LS and REG methods may be favourable for long assimilation time-windows. We know that in 4D-Var, when the assimilation window length is increased, the effects of the nonlinear dynamics of the model increase [16]. Furthermore, recall from Section 3.2.3, GN is known to perform poorly in the presence of nonlinearities [33]. The results when using a correlated $\mathbf{B}$ confirm this. We can find that when using a long time-window, the GN method fails to locate as small a function value as the globally convergent methods. In addition, due to the use of the Armijo condition (3.18) in the LS method, we know that the function values will strictly decrease at each iteration. However, the GN method, which uses the same descent direction as LS, has no control on how far it can go in this direction and so may descend too far. The REG method uses a different descent direction completely and deviates from the GN step by the given choice of $\gamma^{(k)}$, see (3.23). These different step choices result in different convergence behaviour of the methods, as demonstrated in the results of this chapter, as well as in Section 4.4.

The slight variation in the L63 and L96 results between the uncorrelated and correlated cases are to be expected as a different choice of $\mathbf{B}$ results in a different assimilation problem. However, the overall conclusion from Section 5.6.1 that there is a benefit to using a globally convergent method in the long time-window case still holds, even in the presence of background error correlations.

We have seen that the length of the assimilation time-window affects the performance of the GN, LS and REG methods and that the use of globally convergent methods is beneficial when GN performs poorly in the long time-window case. For the remainder of our experiments, we decide to use $t_a = 1$ to consider a long time-window case only. Next, we will study the impact of varying the background error variance in the long time-window case.

5.9.2 Effect of background error variance

As we did in Section 5.6.3 for the uncorrelated $\mathbf{B}$, for each choice of $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$, we generate accuracy profiles to benchmark the performance of the GN, LS and REG methods as the tolerance τ_f is reduced, where $\tau_e = 8$ for the L63 and L96 problems respectively. We omit the plots from this section due to the strong similarities with those in Section 5.6.3 and instead comment on the differences.

For both the L63 and L96 problems, we find that in the case where the background error is much larger than the observation error, the results when using a correlated $\mathbf{B}$ show that a globally convergent method is able to find a smaller function value than GN. We find a difference in the L63 results between the uncorrelated background error case in Figure 5.3(a) and the correlated background error case when $\mathbf{B}_1$ is used: REG is solving more problems

than GN and LS, unlike in the other cases of $\mathbf{B}$ where LS solves the most problems within a given accuracy. Furthermore, the results for the L63, $\mathbf{B}_3$, 25% case show LS and REG solving the most problems within a given accuracy, unlike in the other cases of $\mathbf{B}$ where REG solves the most problems within a given accuracy. As we mentioned in Section 5.9.1, these differences are to be expected as there is no guarantee that LS and REG converge to the same local minimum of the cost function (5.4) due to the methods taking different steps.

The results where we allow τ_e to increase for the correlated background error cases are very similar to those in Figure 5.4, where uncorrelated background error is considered. We find that increasing the number of evaluations allowed by each method shows a greater difference between the performance of the globally convergent methods and GN in the large background error cases for both the L63 and L96 problems. This is because LS and REG are able to guarantee a strict/monotonic decrease in the objective function value (5.4) at each successful iteration, whereas GN has no such guarantee. Therefore, the additional iterations allowed may not result in an improvement in the estimate obtained by GN, whereas the estimate of LS and REG may continue to improve, resulting in a greater difference between the performance of GN and the globally convergent methods when τ_e is increased, as found in our results.

When τ_e is increased for the L96 problems, we find a unique case in the 25% error results: the performance of GN when $\mathbf{B}_3$ is used is better than it is for the other choices of $\mathbf{B}$. In these results, all three methods perform similarly unlike in the L96 $\mathbf{B}_1$, 25% case, the $\mathbf{B}_2$, 25% case and the uncorrelated background error 25% case in Figure 5.4(f), where either LS or REG are performing best. The effect of increasing the level of background error correlations appears to have a positive effect on the performance of GN, as we found when $\tau_e = 8$ was used. However, the overall message from when τ_e is increased for both the L63 and L96 problems still holds for the majority of the cases, that is, it is evident that using a globally convergent method is beneficial when the background error variance is large relative to the observation error variance.

So far in this section and in Section 5.6.3, we have seen the effect increasing the level of background error has on the convergence of GN, LS and REG. To develop our understanding of these results, we next study what happens in the models when the level of background error is increased. We begin by showing that the linearity assumption used when deriving the TLM, $\mathbf{M}_{0,i}$, is not a suitable assumption in the case where the background error SD, σ_b, is large relative to the reference state.

Recall from Section 5.5, varying the background error variance can be seen as perturbing $\mathbf{x}_0^{ref}$ by a different initial perturbation ε_b in our twin experimental set up. The second-order Taylor series expansion of the nonlinear model $\mathcal{M}_{0,i}$ about $\mathbf{x}_0^{ref}$ with perturbation ε_b is given by

$$\mathcal{M}_{0,i}(\mathbf{x}_0^{ref} + \varepsilon_b) \approx \mathcal{M}_{0,i}(\mathbf{x}_0^{ref}) + \mathbf{M}_{0,i}\varepsilon_b + \frac{1}{2}\varepsilon_b^T \frac{\partial^2}{\partial \mathbf{x}_0^2}\mathcal{M}_{0,i}(\mathbf{x}_0^{ref})\varepsilon_b. \tag{5.36}$$

The TLM assumes the model is approximately linear and the following approximation is

made

$$\mathcal{M}_{0,i}(\mathbf{x}_0^{ref} + \varepsilon_b) - \mathcal{M}_{0,i}(\mathbf{x}_0^{ref}) \approx \mathbf{M}_{0,i}\varepsilon_b. \tag{5.37}$$

We can easily see from (5.21), (5.36) and (5.37) that the larger the background error SD σ_b is, the larger the perturbation ε_b can be, thus the larger the term

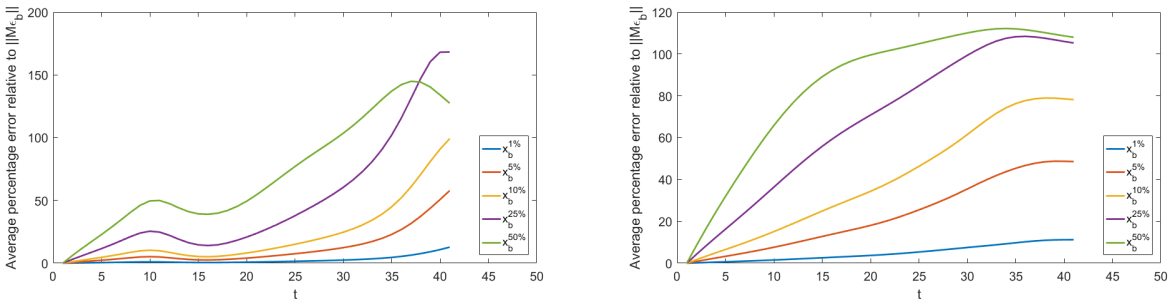
$$\frac{1}{2}\varepsilon_b^T \frac{\partial^2}{\partial \mathbf{x}_0^2}\mathcal{M}_{0,i}(\mathbf{x}_0^{ref})\varepsilon_b \tag{5.38}$$

is. Therefore the (5.37) becomes a poorer approximation as the level of background error is increased.

As we have chosen a chaotic set up of the L63 and L96 models, we expect that a small perturbation in $\mathbf{x}_0^{ref}$ to have a large effect at later time points. We conduct validity tests on the tangent linear model of the L63 and L96 models. The following measure is used to calculate the relative error between the nonlinear perturbation and the linearised perturbation at each time point i,

$$100\frac{\|\mathcal{M}_{0,i}(\mathbf{x}_0^{ref} + \varepsilon_b) - \mathcal{M}_{0,i}(\mathbf{x}_0^{ref}) - \mathbf{M}_{0,i}(\varepsilon_b)\|}{\|\mathbf{M}_{0,i}(\varepsilon_b)\|}. \tag{5.39}$$

The measure (5.39) gauges whether the TLM is a good approximation at a given time point. For both the L63 and L96 models, (5.39) is calculated for each of the $n_r = 100$ realisations for robustness, at each time point where a time-window length of $t_a = 1$ time units (5 days) is used and where we consider different perturbations ε_b. The average of these 100 values at each time point is then taken and presented in the validity test plots in Figure 5.9. This figure confirms what we expect to see for a chaotic model; as t increases, the deviation of the TLM from the nonlinear model increases, and more so when perturbation from $\mathbf{x}_0^{ref}$ is increased. Figure 5.9 also shows that for each case of background error variance that we consider, the average percentage errors are larger for L63 model than they are for the L96 model. This may explain why we see a slight difference between the L63 and L96 results in the numerical results of this chapter.



(a) L63. (b) L96.

Figure 5.9: Results of the TLM validity tests for the L63 (a) and L96 (b) models for different levels of background error, where $t_a = 1$.

Notice that it is not only the background error being large that can cause the term (5.38) to be large. If the second derivative of the nonlinear model is significant, this could also result in the TLM deviating from the nonlinear model. The second derivative of the nonlinear model also features in the higher order terms $\mathbf{Q}(\mathbf{x})$ of the Hessian of the cost function (2.31). From [33] we know that the convergence of GN depends on whether or not $\mathbf{Q}(\mathbf{x})$ in (2.31) is close to zero. If $\frac{\partial^2 \mathcal{M}_{0,i}}{\partial \mathbf{x}_0^2}$ is significant, this could cause $\mathbf{Q}(\mathbf{x})$ to deviate from zero, thus affecting the speed of convergence of GN if $\mathbf{Q}(\mathbf{x})$ is small relative to $\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})$. Furthermore, if $\mathbf{Q}(\mathbf{x})$ is large relative to $\mathbf{J}(\mathbf{x})^T\mathbf{J}(\mathbf{x})$, GN may fail to converge locally.

The difference between the L63 and the L96 numerical results could be due to the difference between the second derivative of the numerical models. The difference between the size of the error between the nonlinear and TLM models shown in Figures 5.9(a) and 5.9(b) may result in the L63 problems having a more significant $\mathbf{Q}(\mathbf{x})$ term that is being neglected by GN than the L96 problems. The LS and REG methods also neglect the $\mathbf{Q}(\mathbf{x})$ term. However, the REG term $\gamma^{(k)}\mathbf{I}_n$ in REG may be seen as a diagonal approximation to the $\mathbf{Q}(\mathbf{x})$ term and may give REG an advantage over GN and LS, both of which assume $\mathbf{Q}(\mathbf{x}) = 0$, in the case that $\mathbf{Q}(\mathbf{x}) \approx \gamma^{(k)}\mathbf{I}_n$.

In this section, we have studied how the GN, LS and REG methods perform in the presence of different levels of background error, when correlated background errors are considered. In the following section, we quantify the error in the outputs of the GN, LS and REG methods using the RMSE measurement for the correlated background error cases.

5.9.3 Quality of the analysis

As we did in Section 5.6.4 for the uncorrelated $\mathbf{B}$, for each choice of $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$, for each realisation, we calculate the root mean square error (RMSE) of the analysis error and plot the percentage of problems solved (according to the criterion (5.27) where $\tau_f = 10^{-3}$) within a specified tolerance of the RMSE (5.28). However, we omit the plots from this section due to the strong similarities with those in Section 5.6.4 and instead comment on the differences.

From the RMSE results for both the L63 and L96 problems, we find that for all cases of correlated $\mathbf{B}$, the number of problems solved within a given RMSE reduces as the level of background error increases. This is to be expected as in DA, the accuracy of the background information determines the accuracy of the analysis. Therefore, the greater the uncertainty is in our background information, the greater the uncertainty in our analysis, as found in our results. Furthermore, in all cases of background error, aside from the 5% case where GN, LS and REG perform similarly, either LS or REG are able to locate better estimates of the initial state.

We find that the correlated $\mathbf{B}$ results are very similar to those where uncorrelated background error is used. The main difference that we see is in the $\mathbf{B}_1$, 50% case of the L63 problems: REG is solving more problems than GN and LS, unlike in the other cases of $\mathbf{B}$ where LS solves the most problems within a given accuracy. We also see that in the $\mathbf{B}_3$, 50%

case of the L96 problems that LS and REG are performing comparably.

The conclusions made in Sections 5.6.3 and 5.6.4 regarding the effect of increasing the background error variance and the performance of GN compared to the globally convergent methods remain for the correlated background error experiments we have conducted. We next show the effect on the performance of the methods as we include more observations and compare the results where correlated background error is used to those in Section 5.6.5, where uncorrelated background error is used.

5.9.4 Effect of observations

As we did in Section 5.6.5 for the uncorrelated $\mathbf{B}$, for each choice of $\mathbf{B}_1$, $\mathbf{B}_2$ and $\mathbf{B}_3$ and for each realisation, we discuss how the use of more observation locations in time affects the performance of GN, LS and REG. We take the worst case for the three methods when there is a 50% error in the background and see if including more observations in time with a 10% error affects the performance of the methods. We omit the accuracy profiles from this section due to the strong similarities with those in Section 5.6.5 and instead comment on the differences.

For the L63 problems, we see very similar results when a correlated $\mathbf{B}$ is used to those where uncorrelated background error is used in Figure 5.7. As the number of observation locations in time is increased, the performance of GN, LS and REG is improved; all three methods are solving more problems at a higher level of accuracy as the number of observation locations in time is increased. This is more apparent when more evaluations are allowed. We find that LS is solving the most L63 problems in most cases of observation locations. This is also what we saw in the uncorrelated $\mathbf{B}$ case. However, if we allow for more evaluations we find that REG outperforms LS.

For the L96 problems, we find that the correlated $\mathbf{B}$ results are very similar to the uncorrelated $\mathbf{B}$ results in Figure 5.7; the globally convergent methods always outperform GN, except in the Nobs4 case where we saw a significant improvement in the performance of GN. However, from the correlated $\mathbf{B}$ results, we do see a gradual improvement in the GN method when we increase the number of observations, something that we did not see when using an uncorrelated $\mathbf{B}$.

In the Nobs4 case and within $\tau_e = 8$ evaluations, the results for all cases of correlated $\mathbf{B}$ show that GN appears to be performing very well compared to when fewer observation locations are used, and outperforming both LS and REG for the L63 problems. For the L96 problems, the Nobs4 results show that there is an improvement in GN for all cases of correlated $\mathbf{B}$, but the globally convergent methods are still able to solve more problems. When more evaluations are allowed for the L63 problems, the Nobs4 results for all cases of correlated $\mathbf{B}$ show that it is REG that outperforms GN and LS, but all three methods appear to solve the majority of the problems to a high level of accuracy. Similarly, for the L96 problems the Nobs4 results for all cases of correlated $\mathbf{B}$ show that, as expected, when many more evaluations are allowed than in practice, there is a very vast improvement in GN, LS and

REG as more accurate observations are included, although most notably in GN. In fact, for the L96 $\mathbf{B}_3$ case, GN appears to be the better choice in the Nobs3 case, although this appears to be a one off case.

The improvement in the performance of GN when more observations are included may be because we are using relatively accurate observations, which are generated by adding error to the reference state, and thus constraining the estimate more tightly to the reference state. This may cause GN to perform well as the observations are taken into account when calculating the GN step (3.10). Furthermore, recall from Section 4.4 that LS and REG require more iterations to converge when GN is performing well due to the need to adjust their parameters, hence why when $\tau_e = 8$, GN performs better than LS and REG, but when more evaluations are allowed, the performance of all three methods is comparable.

Recall from Section 2.2.2, it is known that the use of realistic background error statistics is important in 4D-Var as it has a profound impact on the analysis [4]. By increasing the correlation length scale, we are increasing the number of non-zero off diagonal entries of $\mathbf{B}$, thus allowing more observation information to spread across the spatial domain during the assimilation. For the L63 problems, we have observations of two out of the three variables. Therefore increasing the level of background error correlations is not expected to have as much of an effect on the assimilation results as in the L96 problems, where there are only observations of the first half of the spatial domain. As expected, the effect of strengthening the background error correlations is not apparent in the results of the L63 problems, nor in the results when allowing for more evaluations. However, for the L96 problems both when $\tau_e = 8$ and when allowing for more evaluations, we find that the performance of GN improves as more background error correlations are included.

In the following section, we conclude this chapter.

5.10 Conclusion

In this chapter, we present a draft paper where we study the convergence behaviour of Algorithms 3.2.3, 3.3.1 and 3.3.2 when applied to the preconditioned 4D-Var problem, with the aim of addressing the research questions RQ1(c) and RQ1(d). In the paper, we consider uncorrelated background error only. We then present the additional experiments (not included in the paper), where correlated background error is used and and discuss the effect the tangent linear approximation of the nonlinear model has on the convergence of the optimisation methods. We consider various lengths of assimilation time-window and focus on the long time-window case, a case that is popular in 4D-Var practice.

We begin by studying the effect of the assimilation time-window length on the convergence of GN, LS and REG. We find that in the short time-window case, there is no benefit in using a globally convergent method as GN is able to solve the majority of problems, even for larger values of the ratio (5.23). Furthermore, in the long time-window case, when the ratio (5.23) is small, we are constraining the solution more tightly to the truth and GN is, again,

able to solve the majority of problems; it is able to obtain the same solution as the LS and REG method and does not require the use of parameter updating strategies. This is because we are constraining the solution at the start of the time-window more tightly when using a small ratio (5.23) and therefore, the initial guess is closer to the reference state compared to when the ratio is large. We know that the GN method works best when initialised close to the truth, which ties with our findings. However, in the long time-window case, when the ratio (5.23) is large this is not the case.

Using two test models within the preconditioned 4D-Var framework, we show that when there is more uncertainty in the background information compared to the observations, the GN method may diverge in the long time-window case, yet the convergent methods, LS and REG, are able to improve the estimate of the analysis. We use accuracy profiles to show numerically that in the long time-window case and when there is higher uncertainty in the background information versus the observations, the globally convergent methods are able to solve more problems than GN in the limited cost available. By 'solve' we mean satisfy a criterion based on the reduction in the objective function within a set number of evaluations. Thus, we have identified which situations the globally convergent methods are a better option then GN in the presence of a long assimilation time-window, addressing RQ1(d).

We show the effect that poor background information has on the quality of the estimate obtained for each of GN, LS and REG when applied to the preconditioned 4D-Var problem. We consider the case where the background information is highly inaccurate compared to the observations in the 4D-Var framework and find that the convergence of all three methods is improved when more observations are included along the time-window. Thus, we have provided an understanding of how GN, LS and REG behave if the initial guess of the minimisation (the background) is highly inaccurate compared to the observations, addressing RQ1(c).

Our results from Chapters 4 and 5 have addressed the first main research question RQ1: is the use of globally convergent strategies within GN beneficial in variational data assimilation? We have compared the performance of GN, LS and REG when applied to both the 3D-Var and 4D-Var problems, where we account for the computational limits that exist in practical implementations and consider different practical scenarios, such as the use of correlated background error and a long assimilation time-window.

In Chapter 4, we focus on the theoretical properties of the LS and REG methods, providing detailed proofs of their global convergence to a stationary point along with a discussion on whether the assumptions of the global convergence theorems hold in practical DA. We only include the theorem statements and the discussion in Chapter 5, omitting the detailed proofs as this is not the focus of the paper. Furthermore, in Chapter 4, we use both theoretical and numerical results to study the interaction of the regularisation parameter on the convergence of REG. This in contrast with Chapter 5, which uses the typical choice of the REG parameter and only briefly discusses the effect of the REG parameter in relation to the numerical results. In Chapter 5 there is less of a focus on theoretical results and a greater focus on testing the performance of LS and REG numerically when applied to more realistic data

assimilation problems using the L63 and L96 numerical models.

In the following chapter, we address the second and final main research question RQ2, where we investigate the use of reduced resolution inner loop methods in incremental 3D-Var.

Chapter 6

Investigating the use of reduced resolution inner loop methods in incremental VarDA

Recall from Chapter 1, in variational data assimilation (VarDA) [68], the data assimilation (DA) problem is formulated as a nonlinear least-squares problem and solved as a sequence of linear least-squares problems using an incremental method, which has been shown to be equivalent to the Gauss-Newton (GN) method under certain conditions [65]. In VarDA, the minimisation of the nonlinear objective function and the linearised subproblem are referred to as the 'outer loop' and the 'inner loop' respectively. Simplifications are made within the inner loop to reduce the computational cost in DA systems and to solve the DA problem in real time.

Recall from Section 2.2.2, one simplification used to reduce computational cost and time is the use of a reduced resolution inner loop. This is where a coarser grid is used to represent the inner loop problem than the grid used for the outer loop problem. This results in a lower dimensional problem when solving for the outer loop update in (3.37) in the incremental method, thus saving both time and computational cost in VarDA practice. This saving is extremely important in Numerical Weather Prediction (NWP), where the dimension of the outer loop problem is of order $10^8 - 10^9$, to ensure that the forecast is produced in sufficient time such that it is useful. However, more work needs to be done to understand the effect of using a reduced grid inner loop on the convergence of the incremental method and the accuracy of the analysis. Within our work, we aim to address this by answering research question RQ2.

We conduct a theoretical investigation into the effect that the use of a reduced resolution inner loop has on the convergence of the incremental method. We define a resolution parameter g, defined as the fraction of grid-points of the full resolution grid and formulate a set of linear and cubic interpolation extension matrices used to map to the outer loop (full) resolution for each choice of g. We exploit the structure of the interpolation matrices to bound various quantities in the incremental method. We prove that the 2-norm of the linear and cubic interpolation matrices are equal to the square root of the inverse of the resolution

parameter g and discuss how this result applies to higher orders of interpolation. We also derive a theoretical bound on the norm of the square-root of the background error correlation matrix (used in the preconditioner) in terms of g.

It is known that the accuracy with which the inner loop is solved affects the convergence of the outer loop [64, 65]. Recall from Section 2.2, the condition number of the VarDA Hessian (2.70) is able to provide some information on the quality of the inner loop update. We present a novel theorem highlighting how the choice of resolution affects the conditioning of the inner loop problem. We use the bound on the 2-norm of the interpolation matrix and the bound on the 2-norm of the preconditioner to derive an upper bound on the condition number of the preconditioned 3D-Var Hessian for different inner loop resolutions. We use this bound to understand the effect that the use of reduced resolution operators have on the accuracy to which we could be able to solve the inner loop problem in practice. We show theoretically that as the correlation length scale is increased, the influence of the resolution parameter g in the bound on the condition number decreases. We test our theoretical findings using numerical experiments where we apply the incremental method to the preconditioned 3D-Var problem, considering different observation operator and background error correlation matrix structures. We identify cases where the bound on the condition number performs well. We find that the effect of the resolution change does not have as great an impact on the conditioning of the inner loop problem as the presence of background error correlations has, which agrees with our theoretical findings.

We then investigate the effects of using a reduced resolution inner loop on the accuracy of the analysis. We use assimilation experiments, where we define a reference state, to compare the analysis error generated by the incremental method using different inner loop resolutions and within a limited number of iterations, similar to what is used in practice. We find that these results suggest a similar relationship between the effect of a reduced resolution inner loop, the background error correlation matrix and the observation error structure exists for the error in the analysis as observed in the conditioning of the problem in our earlier results. We conduct a wave transformation using the discrete Fourier transform (DFT), described in detail in Section 2.1.3 and use this to understand what effect the use of a reduced resolution inner loop has on the quality of the analysis when solving the variational problem. We use the power spectra of the DFT of the 3D-Var analysis to understand how accurately the non-zero wavenumbers of the reference state can be resolved. We generate error profiles for the error in the amplitudes of the non-zero wavenumbers for different resolution choices and find that the accuracy with which the preconditioned 3D-Var problem is solved is more so affected by the level of resolution in the presence of more observations along the spatial domain. We also find that as stronger background error correlations are introduced, the difference between the analysis errors of each resolution we consider decreases. This latter finding coincides with both our theoretical and experimental work on the condition number.

In Section 6.1, we outline the grid-point framework used in our work along with the restriction and extension matrices and their properties. In Section 6.2, we derive a bound on the condition number of the preconditioned 3D-Var Hessian (2.70) using bounds derived on the reduced resolution background error correlation matrix and the interpolation matrices.

In Section 6.3, we describe the experimental design used to test the bounds derived in the previous section, as well as the accuracy of the algorithmic output. In Section 6.4, we present the numerical results obtained for the preconditioned 3D-Var problem using both linear and nonlinear observation operators for the incremental method at different resolutions. We consider the error in both the numerical output (using the root mean square error of the analysis) and the error in the frequencies resolved (using the DFT). Finally, we conclude our findings in Section 6.5.

6.1 Reduced resolution framework

Within this section, we outline the grid-point formulation used within this chapter, along with our choices of restriction and extension matrices.

6.1.1 Grid-point formulation

Before defining the resolution operators, we first set out the grid-point framework used within our work. We define a one-dimensional horizontal grid with n equally spaced grid-points labelled $i = 0, 1, .., n-1$ on a cyclic domain. We assume the variables in the state $\mathbf{x}$ are quantities of the same type, e.g. temperature variables, and that there is one variable at each grid-point. We assume that the variables in $\mathbf{x}$ are ordered according to their position along the grid, such that x_i is the variable at grid-point i, for $i = 0, 1, 2, \ldots, n-1$, where $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^T$ and Δx denotes the spatial distance between the variables/the grid length, resulting in a spatial domain of size $n\Delta x$. A schematic is presented in Figure 6.1 to visualise this grid.

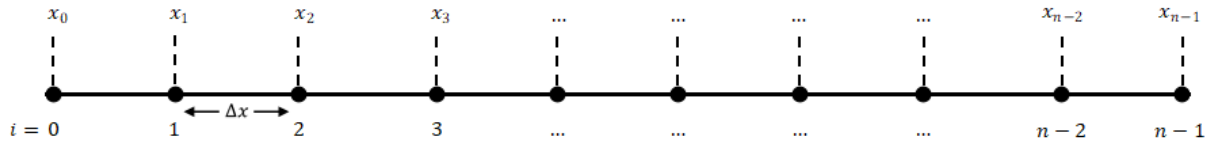


Figure 6.1: Schematic of a one-dimensional grid with variables evenly distributed along the domain.

Figure 6.1 is the full resolution grid used within our work. In order to reduce the resolution of this grid, we retain the same size of the spatial domain of the full resolution grid, $n\Delta x$, but decrease the frequency of the grid-points by removing intermediate grid-points. This results in a reduced resolution grid with grid length

$$\Delta \hat{x} = 1/g\Delta x, \tag{6.1}$$

where we define g to be the resolution parameter that satisfies $1/n \leq g \leq 1$, where g^{-1} must be a divisor of the grid size n, such that $gn \in \mathbb{N}$. For example, for a half resolution grid, we set $g = 1/2$, so the reduced resolution grid length is given by $\Delta \hat{x} = 2\Delta x$ with domain size

$(n/2)\Delta\hat{x} = n\Delta x$.

In the latest Integrated Forecast System (IFS) update (47r1) at ECMWF, four outer loop iterations of the incremental method are performed, with horizontal resolution of TCo1279 ($\sim$ 9km grid-point resolution at the equator/mid-latitudes) [34]. Their corresponding four inner loops are performed with horizontal resolution of TL255 ($\sim$ 128km), TL255, TL319 ($\sim$ 63km) and TL399 ($\sim$ 50km) respectively. These inner loops correspond to approximately $\frac{1}{5}^{\text{th}}$, $\frac{1}{5}^{\text{th}}$, $\frac{1}{4}^{\text{th}}$ and $\frac{1}{3}^{\text{rd}}$ of the resolution of the outer loop, which was found to be the optimal configuration in systematic tests to simultaneously maximise forecast accuracy while minimising computation cost [58].

Within our work, we consider minimising the 3D-Var problem (2.52) using Algorithm 3.4.1 with

- **Case 1**: a full resolution inner loop with grid length Δx and $\delta\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^n \ \forall k$.
- **Case 2**: an inner loop with half the resolution of the outer loop with grid length $2\Delta x$ and $\delta\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^{n/2} \ \forall k$.
- **Case 3**: an inner loop with a quarter of the resolution of the outer loop with grid length $4\Delta x$ and $\delta\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^{n/4} \ \forall k$.
- **Case 4**: an inner loop with an eighth of the resolution of the outer loop with grid length $8\Delta x$ and $\delta\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^{n/8} \ \forall k$.
- **Case 5**: multi-incremental 3D-Var with a varying grid length.

For Case 5, we conduct a study similar to that of [126] where we begin by using the lowest resolution, Case 4, in the first outer loop iteration $k = 1$. We then increase the resolution in the subsequent outer loop iterations to Case 3 for $k = 2$, then Case 2 for $k = 3$, then Case 1 for $k = 4$.

We note that in DA practice, it is generally too expensive to run any full resolution inner iterations. The reason behind our use of full resolution iterations is purely for comparative reasons. We are able to study the effect of the use of a reduced resolution inner loop on the convergence of the incremental method compared to when the full resolution is used. By increasing the resolution from our lowest resolution choice (eighth) to the highest (full) in Case 5, we can see if the incremental method is still able to converge to the full resolution analysis.

The operators used to map between the full resolution grid and the reduced resolution grids used in each of the cases outlined above are defined in the following section.

6.1.2 Restriction operators

We first construct the restriction operators $\mathbf{S}_{lg} : \mathbb{R}^n \to \mathbb{R}^{gn}$ for any g where $gn \in \mathbb{N}$, which are used to restrict the high resolution VarDA components to the reduced resolution grid

for use in the inner loop minimisation (Step 5 of Algorithm 3.4.1). The matrix form of the restriction operators, $\mathbf{S}_{l_g} \in \mathbb{R}^{gn \times n}$ are defined as follows

$$\mathbf{S}_{l_g}(i,j) = \begin{cases} 1, \; j = g^{-1}i & \text{for } i = 1, \ldots, gn \\ 0, & \text{otherwise,} \end{cases} \tag{6.2}$$

where $g = 1/2, 1/4$ and $1/8$ correspond to Cases 2, 3 and 4 respectively. Case 5 uses $\mathbf{S}_{l_{1/2}}$, $\mathbf{S}_{l_{1/4}}$ and $\mathbf{S}_{l_{1/8}}$.

6.1.3 Extension operators

We next construct the extension operators $\mathbf{S}_{h_g} : \mathbb{R}^{gn} \to \mathbb{R}^n$, which are used to map the reduced resolution increment $\delta\hat{\mathbf{x}} \in \mathbb{R}^{gn}$ back to full resolution space $\mathbb{R}^n$ at the end of each inner loop minimisation (Step 6 of Algorithm 3.4.1). We outline three possible choices of $\mathbf{S}_{h_g}$, namely, the pseudoinverse of $\mathbf{S}_{l_g}$, $\mathbf{S}_{l_g}^+$, a linear interpolation matrix $\mathbf{S}_{h_g}^{lin}$ and a cubic interpolation matrix $\mathbf{S}_{h_g}^{cub}$.

As $\mathbf{S}_{l_g}$ defined in (6.2) are non-square matrices, their inverses cannot be calculated. However, we can calculate the Moore-Penrose pseudoinverse, as defined in Definition 2.1.10, $\mathbf{S}_{l_g}^+ \in \mathbb{R}^{n \times gn}$ as this always exists, but it is not necessarily unique.

We want to understand how the use of the matrix $\mathbf{S}_{l_g}^+$ affects the analysis in incremental VarDA. Using (2.16), we show in the following lemma that $\mathbf{S}_{l_g}^+ = \mathbf{S}_{l_g}^T$ which, when applied to the reduced resolution vector $\delta\hat{\mathbf{x}}$, results in the entries at the full resolution grid points that were omitted on the reduced grid being zero. This can be seen simply by looking at the entries of $\mathbf{S}_{l_g}^T$.

Lemma 6.1.1 (Pseudo-inverse of $\mathbf{S}_{l_g}$)**.** *Let $\mathbf{S}_{l_g} \in \mathbb{R}^{gn \times n}$ be the restriction matrix defined in* (6.2)*. Then the pseudo-inverse of $\mathbf{S}_{l_g}$, denoted by $\mathbf{S}_{l_g}^+$, is given by*

$$\mathbf{S}_{l_g}{}^+ = \mathbf{S}_{l_g}^T. \tag{6.3}$$

Proof. The rows of matrix $\mathbf{S}_{l_g}$ are made up of the standard basis vectors denoted by $\mathbf{e}_{i/g}$. From (2.16) we have that as $r < n$, the pseudo-inverse of $\mathbf{S}_{l_g}$ is given by

$$\mathbf{S}_{l_g}^+ = \mathbf{S}_{l_g}{}^T(\mathbf{S}_{l_g}\mathbf{S}_{l_g}{}^T)^{-1}. \tag{6.4}$$

Let $\mathbf{C} = \mathbf{S}_{l_g}\mathbf{S}_{l_g}{}^T$, then $\mathbf{C}$ has entries

$$\begin{aligned} c_{ij} &= \mathbf{e}_{i/g}\mathbf{e}_{j/g}^T \\ &= \begin{cases} 1, & \text{if } i = j \\ 0, & \textit{otherwise.} \end{cases} \end{aligned} \tag{6.5}$$

Hence, $\mathbf{C} = \mathbf{I}$. Substituting this into (6.4), we have

$$\begin{aligned} \mathbf{S}_{l_g}^+ &= \mathbf{S}_{l_g}{}^T\mathbf{C}^{-1} \\ &= \mathbf{S}_{l_g}{}^T, \end{aligned} \tag{6.6}$$

as required. □

The implications of Lemma 6.1.1 is that, when mapping to the higher resolution using the pseudoinverse $\mathbf{S}_l^+$ in Step 6 of Algorithm 3.4.1, the terms that are neglected in $\delta\hat{\mathbf{x}}$ will be set to zero. Therefore, assuming the same reduced resolution matrix is used for each inner loop of incremental method, the corresponding values of $\mathbf{x}^{(k)}$ will remain fixed throughout the iteration process (for all k) as they are never updated. This result shows that the use of the pseudoinverse (2.16) as the extension matrix is not a suitable choice. We instead focus our efforts on the use of interpolation techniques in the remainder of this chapter. More specifically, linear and cubic interpolation, both of which are common choices in practice and work by approximating the higher resolution variables using nearby reduced resolution variables.

To find the linearly interpolated value of a given variable, say x_1, we find the two nearest variables that are adjacent to it and weight them by how close they are to x_1. The summation of these two weighted components is the linearly interpolated value of x_1. Note that there must be at least two known data points in the reduced resolution space in order to use linear interpolation. Therefore, we impose the following restriction on the choice of n when using a linear interpolation extension matrix

$$n \geq \frac{2}{g}. \tag{6.7}$$

We now form a linear interpolation matrix, denoted by $\mathbf{S}_{hg}^{lin} \in \mathbb{R}^{n \times gn}$, to map the reduced resolution inner loop increment to the high resolution, and subsequently update the outer loop iterate in the incremental method. The j^{th} column of the linear interpolation matrix given any valid choice of g is given by

$$\mathbf{S}_{hg}^{lin}(j) = \sum_{l=1}^{g^{-1}-1} \left\{ (1-gl)(\mathbf{e}_{g^{-1}j-l} + \mathbf{e}_{g^{-1}j+l}) \right\} + \mathbf{e}_{g^{-1}j}. \tag{6.8}$$

For Case 2, the linear interpolation matrix is defined to be

$$\mathbf{S}_{h1/2}^{lin}(i,j) = \begin{cases} 1, \; i = 2j & \text{for } j \leq n/2 \\ 1/2, \; i = \begin{cases} 2j-1, & \text{for } j \leq n/2 \\ 2j+1, & \text{for } j < n/2 \\ 1, & \text{for } j = n/2 \end{cases} & \\ 0, & \text{otherwise,} \end{cases} \tag{6.9}$$

where $j = 1, 2, \ldots, n/2$ and $n \geq 4$.

For Case 3, the extension matrix is defined to be

$$\mathbf{S}_{h1/4}^{lin}(i,j) = \begin{cases} 1,\ i = 4j & \text{for } j \le n/4 \\ 3/4,\ i = \begin{cases} 4j-1, & \text{for } j \le n/4 \\ 4j+1, & \text{for } j < n/4 \\ 1, & \text{for } j = n/4 \end{cases} & \\ 1/2,\ i = \begin{cases} 4j-2, & \text{for } j \le n/4 \\ 4j+2, & \text{for } j < n/4 \\ 2, & \text{for } j = n/4 \end{cases} & \\ 1/4,\ i = \begin{cases} 4j-3, & \text{for } j \le n/4 \\ 4j+3, & \text{for } j < n/4 \\ 3, & \text{for } j = n/4 \end{cases} & \\ 0, & \text{otherwise,} \end{cases} \tag{6.10}$$

where $j = 1, 2, \ldots, n/4$ and $n \ge 8$.

For Case 4, the extension matrix is defined to be

$$\mathbf{S}_{h1/8}^{lin}(i,j) = \begin{cases} 1,\ i = 2j & \text{for } j \leq n/8 \\ 7/8,\ i = \begin{cases} 8j-1, & \text{for } j \leq n/8 \\ 8j+1, & \text{for } j < n/8 \\ 1, & \text{for } j = n/8 \end{cases} \\ 3/4,\ i = \begin{cases} 8j-2, & \text{for } j \leq n/8 \\ 8j+2, & \text{for } j < n/8 \\ 2, & \text{for } j = n/8 \end{cases} \\ 5/8,\ i = \begin{cases} 8j-3, & \text{for } j \leq n/8 \\ 8j+3, & \text{for } j < n/8 \\ 3, & \text{for } j = n/8 \end{cases} \\ 1/2,\ i = \begin{cases} 8j-4, & \text{for } j \leq n/8 \\ 8j+4, & \text{for } j < n/8 \\ 4, & \text{for } j = n/8 \end{cases} \\ 3/8,\ i = \begin{cases} 8j-5, & \text{for } j \leq n/8 \\ 8j+5, & \text{for } j < n/8 \\ 5, & \text{for } j = n/8 \end{cases} \\ 1/4,\ i = \begin{cases} 8j-6, & \text{for } j \leq n/8 \\ 8j+6, & \text{for } j < n/8 \\ 6, & \text{for } j = n/8 \end{cases} \\ 1/8,\ i = \begin{cases} 8j-7, & \text{for } j \leq n/8 \\ 8j+7, & \text{for } j < n/8 \\ 7, & \text{for } j = n/8 \end{cases} \\ 0, & \text{otherwise,} \end{cases} \tag{6.11}$$

where $j = 1, 2, \ldots, n/8$ and $n \geq 16$.

Case 5 uses $\mathbf{S}_{h1/2}^{lin}$, $\mathbf{S}_{h1/4}^{lin}$ and $\mathbf{S}_{h1/8}^{lin}$.

Within our theoretical work in Section 6.2, we focus on the use of both the linear and cubic interpolation matrices for a general g. In our experiments in Section 6.4, we use the linear interpolation matrices for the specific cases of g. In both cases, we discuss how our results hold for higher-order interpolation techniques. Next, we outline how cubic interpolation works and define a cubic interpolation matrix for $g = 1/2$ to demonstrate the structure.

The idea of cubic interpolation is similar to that of linear interpolation, except it uses the nearest four variables to estimate the value of a variable at a chosen grid-point. This means that there must be at least four known data points at the reduced resolution in order to use cubic interpolation. Therefore, we impose the following restriction on the choice of n when

using a cubic interpolation extension operator,

$$n \geq \frac{4}{g}. \tag{6.12}$$

Again, the weightings of the four neighbouring variables are chosen according to their distance from the variable being interpolated. The j^{th} column of a cubic interpolation matrix given any valid choice of g is given as follows

$$\mathbf{S}_{hg}(j) = \sum_{l=1}^{g^{-1}-1} \left\{ w_l(\mathbf{e}_{g^{-1}j-l}+\mathbf{e}_{g^{-1}j+l})+w_{2g^{-1}-l}(\mathbf{e}_{g^{-1}j-(2g^{-1}-l)}+\mathbf{e}_{g^{-1}j+(2g^{-1}-l)}) \right\}+\mathbf{e}_{g^{-1}j}, \tag{6.13}$$

where w_l are the associated weightings that satisfy $w_l > 0$ and $\sum_l 2w_l + 1 = g^{-1}$. By setting $w_l = (1-gl)$ and $w_{2g^{-1}-l} = 0$ for all l in (6.13), we are able to obtain the linear interpolation matrix (6.8).

Using (6.13), we can form a half resolution cubic interpolation matrix, denoted by $\mathbf{S}_{h1/2}^{cub} \in \mathbb{R}^{n\times r}$, to map the half resolution inner loop increment to the high resolution, and subsequently update the outer loop iterate. For Case 2 the cubic interpolation matrix is defined to be

$$\mathbf{S}_{h1/2}^{cub}(i,j) = \begin{cases} 1, \ i = 2j & \text{for } j \leq n/2 \\ n_1/n, \ i = \begin{cases} 2j-3, & \text{for } 2 \leq j \leq n/2 \\ 2j+3, & \text{for } j \leq n/2-2 \\ 1, & \text{for } j = n/2-1 \\ 3, & \text{for } j = n/2 \\ n-1, & \text{for } j = 1 \end{cases} & \\ n_2/n, \ i = \begin{cases} 2j-1, & \text{for } j \leq n/2 \\ 2j+1, & \text{for } j \leq n/2-1 \\ 1, & \text{for } j = n/2 \end{cases} & \\ 0, & \text{otherwise,} \end{cases} \tag{6.14}$$

where $j = 1, 2, \ldots, n/2$, $n_1 = \frac{n}{4} - 1$, $n_2 = \frac{n}{4} + 1$ and $n \geq 8$.

In our work, we focus on odd orders of interpolation only, i.e. where $q = 1, 3, 5 \ldots$. This is because, the use of odd orders of interpolation would result in the same number of neighbouring low resolution grid-points from both sides of the full resolution grid-point being interpolated to, with equal weightings either side. The use of even orders of interpolation in this structure would mean that we would need to use more low resolution neighbouring grid-points from one side of the full resolution grid-point than the other, with differing weights.

For a given choice of g, the number of nonzero entries in each column of a q^{th} order interpolation matrix $\mathbf{S}_{h_g}^q$, denoted by n_q, is dependent on the resolution parameter g and the order of interpolation q. This relation is shown in the following

$$n_q = (q+1)g^{-1} - q. \tag{6.15}$$

We can form the columns of $\mathbf{S}_{hg}^{q}$ using the standard basis column vectors $\mathbf{e}_j$, where $j = 1, 2, \dots gn$. Given any valid choice of g, the j^{th} column of a q^{th} (odd) order interpolation matrix where $q = 1, 3, 5, \dots$ is given as follows

$$\begin{aligned}\mathbf{S}_{hg}^{q}(j) = \mathbf{e}_{g^{-1}j} + \sum_{l=1}^{g^{-1}-1} \Big\{ & w_l(\mathbf{e}_{g^{-1}j-l} + \mathbf{e}_{g^{-1}j+l}) + w_{2g^{-1}-l}(\mathbf{e}_{g^{-1}j-(2g^{-1}-l)} + \mathbf{e}_{g^{-1}j+(2g^{-1}-l)}) \\ & + w_{4g^{-1}-l}(\mathbf{e}_{g^{-1}j-(4g^{-1}-l)} + \mathbf{e}_{g^{-1}j+(4g^{-1}-l)}) + \dots \\ & + w_{(q-1)g^{-1}-l}(\mathbf{e}_{g^{-1}j-((q-1)g^{-1}-l)} + \mathbf{e}_{g^{-1}j+((q-1)g^{-1}-l)}) \Big\},\end{aligned} \tag{6.16}$$

where $w_{z(l)}$ denotes the weighting parameters associated with $\mathbf{e}_{g^{-1}j \pm z(l)}$ and $z : \mathbb{R} \to \mathbb{R}$.

By definition, for any interpolation $\mathbf{S}_{hg}$, the row sum is always 1. Each row of $\mathbf{S}_{hg}$ corresponds to a grid-point on the full resolution grid and the row entries correspond to the weightings used in the interpolation of the full resolution grid-point.

Each column of $\mathbf{S}_{hg}$ corresponds to a grid-point on the reduced resolution grid and the column entries are the weightings given to that grid-point. The size of the weightings w_l of each grid-point depends on its distance from the point being interpolated. These weightings satisfy $w_l > 0$ and $\sum_i 2w_l + 1 = g^{-1}$. So the column sums do not depend on the order of interpolation used, they only depend on the resolution. However, the number of nonzero entries in each column does depend on the order of interpolation and indicates how many times each reduced resolution point is used for interpolation. The higher the order of interpolation, the more each reduced resolution variable is used, so the more nonzero entries of that column. The number of nonzero entries in each column also depends on the resolution; the lower the resolution, the more each low resolution point will be used as there are more points that need interpolation, so again, the more nonzero entries of that column.

Now that we have outlined the reduced resolution framework used within this chapter, in the following section, we use the properties of the resolution operators to derive a bound on the condition number of the preconditioned 3D-Var Hessian (2.70) when using the reduced resolution incremental method.

6.2 Theoretical bounds

We aim to understand how the use of a reduced resolution inner loop affects the convergence of the incremental method. Within this section, we derive a bound on the condition number of the preconditioned 3D-Var Hessian (2.70) and show its dependence on the level of resolution used. We first derive bounds on the norm of the restriction and extension matrices, outlined in Section 6.1, and the background error correlation matrix for use in the bound on the condition number of the preconditioned 3D-Var Hessian. We later test these bounds numerically in Section 6.4.

We begin by deriving expressions for the norm of the restriction and extension matrices in the following section.

6.2.1 Bounding the resolution matrices

We use the resolution matrices from Section 6.1 to map to and from the reduced resolution. From $\mathbf{S}_{l_g}$ in (6.2), we have that $\mathbf{S}_{l_g}\mathbf{S}_{l_g}^T = \mathbf{I}_r$, so $\lambda_i(\mathbf{S}_{l_g}\mathbf{S}_{l_g}^T) = 1$ for all $l = 1, 2, \ldots, r$. Therefore, $\lambda_{\max}(\mathbf{S}_{l_g}\mathbf{S}_{l_g}^T) = \lambda_{\max}(\mathbf{S}_{l_g}^T\mathbf{S}_{l_g}) = 1$, so for all valid choices of g, the 2-norm of $\mathbf{S}_{l_g}$ is given by

$$\|\mathbf{S}_{l_g}\| = \sqrt{\lambda_{\max}(\mathbf{S}_{l_g}\mathbf{S}_{l_g}^T)} = 1. \tag{6.17}$$

For the extension operator, $\mathbf{S}_{h_g}$ is a real non-negative matrix. For any choice of n that is compatible with the resolution choice, the matrix $\mathbf{S}_{h_g}^T\mathbf{S}_{h_g}$ is a non-negative, symmetric positive semidefinite matrix, so the 2-norm of $\mathbf{S}_{h_g}$ is given by

$$\|\mathbf{S}_{h_g}\| = \sqrt{\lambda_{\max}({\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g})}. \tag{6.18}$$

To determine the expression for $\lambda_{\max}({\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g})$, we first show that ${\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g}$ is a circulant matrix of the form (2.17).

Using the following theorem, we show that for the linear and cubic interpolation matrices $\mathbf{S}_{h_g} \in \mathbb{R}^{n\times gn}$, the structure of $\mathbf{S}_{h_g}^T\mathbf{S}_{h_g} \in \mathbb{R}^{gn\times gn}$ takes the form (2.17).

Theorem 6.2.1 (${\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g}$ is circulant)**.** *Let $\mathbf{S}_{h_g} \in \mathbb{R}^{n\times gn}$ be the g resolution linear or cubic interpolation matrix whose columns are defined by Equation* (6.13)*. Then ${\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g}$ is a circulant matrix.*

Proof. Let

$$\mathbf{C} = {\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g} \in \mathbb{R}^{gn\times gn}. \tag{6.19}$$

To show that $\mathbf{C}$ is circulant, we split the proof as follows. We first show that the matrix $\mathbf{C}$ is symmetric. We then show that all entries along a given diagonal of $\mathbf{C}$ are equal.

To prove that the matrix $\mathbf{C}$ is symmetric, we need to show that $\mathbf{C}^T = \mathbf{C}$. We have that $\mathbf{C} = {\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g}$. Therefore we have

$$\begin{aligned} \mathbf{C}^T &= ({\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g})^T \\ &= {\mathbf{S}_{h_g}}^T(\mathbf{S}_{h_g}^T)^T \\ &= {\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g}, \end{aligned} \tag{6.20}$$

as required.

To prove that all entries along a given diagonal of $\mathbf{C}$ are equal, we need show that for $i = 1, 2, \ldots, gn$,

$$\mathbf{C}(i, j) = \mathbf{C}((i + 1) \pmod{gn}, (j + 1) \pmod{gn}), \tag{6.21}$$

where $i, j = 1, 2, \ldots, gn$. We note that (mod gn) is used to account for the periodic domain. This is omitted in the following for simplicity. Our expression for $\mathbf{C}(i, j)$ is

$$\begin{aligned}\mathbf{C}(i,j) = \sum_{l=1}^{g^{-1}-1} &\left[w_l(\mathbf{e}_{g^{-1}i-l} + \mathbf{e}_{g^{-1}i+l}) + w_{2g^{-1}-l}(\mathbf{e}_{g^{-1}i-(2g^{-1}-l)} + \mathbf{e}_{g^{-1}i+(2g^{-1}-l)}) + \mathbf{e}_{g^{-1}i} \right]^T \\ &\times \left[w_l(\mathbf{e}_{g^{-1}j-l} + \mathbf{e}_{g^{-1}j+l}) + w_{2g^{-1}-l}(\mathbf{e}_{g^{-1}j-(2g^{-1}-l)} + \mathbf{e}_{g^{-1}j+(2g^{-1}-l)}) + \mathbf{e}_{g^{-1}j} \right],\end{aligned} \tag{6.22}$$

where $i, j = 1, 2, \ldots, gn$. Our expression for $\mathbf{C}(i+1, j+1)$ is

$$\begin{aligned}\mathbf{C}((i+1),(j+1)) = \sum_{l=1}^{g^{-1}-1} &\left[w_l(\mathbf{e}_{g^{-1}(i+1)-l} + \mathbf{e}_{g^{-1}(i+1)+l}) + w_{2g^{-1}-l}(\mathbf{e}_{g^{-1}(i+1)-(2g^{-1}-l)} \right. \\ &\left. + \mathbf{e}_{g^{-1}(i+1)+(2g^{-1}-l)}) + \mathbf{e}_{g^{-1}(i+1)} \right]^T \times \left[w_l(\mathbf{e}_{g^{-1}(j+1)-l} + \mathbf{e}_{g^{-1}(j+1)+l}) \right. \\ &\left. + w_{2g^{-1}-l}(\mathbf{e}_{g^{-1}(j+1)-(2g^{-1}-l)} + \mathbf{e}_{g^{-1}(j+1)+(2g^{-1}-l)}) + \mathbf{e}_{g^{-1}(j+1)} \right],\end{aligned} \tag{6.23}$$

where $i, j = 1, 2, \ldots, gn$. When using the standard basis vectors, only those $\mathbf{e}^T\mathbf{e}$ terms with matching indices will be non-zero. Quite obviously, if $i = j$ then $i + 1 = j + 1$, so the nonzero terms of Equation (6.22) are the same as those of Equation (6.23). Therefore, we can conclude that $\mathbf{C}(i, j) = \mathbf{C}((i + 1), (j + 1))$, as required.

As we have shown that $\mathbf{C}$ is symmetric and that $\mathbf{C}(i, j) = \mathbf{C}((i+1), (j+1))$, we can conclude that $\mathbf{C}$ is circulant, as required. □

To determine the expression for $\lambda_{\max}({\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g})$, we use Theorem 2.1.5. Because of the positive symmetric circulant nature of ${\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g}$, its distribution of eigenvalues is symmetric. Therefore, using Corollary 2.1.6, the maximum eigenvalue of $\mathbf{C}$ is given by Equation (2.21) which is the row/column sum of the circulant matrix. All interpolation matrices $\mathbf{S}_{h_g}$ have, by definition, a row sum of 1 and a column sum of $1/g$. We can use the following lemma to show that $\mathbf{S}_{h_g}^T\mathbf{S}_{h_g}$ has a row sum of $1/g$.

Lemma 6.2.2 (Relation between row sum of $\mathbf{A}$ and $\mathbf{A}^T\mathbf{A}$)**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times r}$ be a matrix with non-negative entries, i.e. $\mathbf{A} = (a_{ij})$ where $a_{ij} \geq 0$ for $1 \leq i \leq n$ and $1 \leq j \leq r$. If for all i, $\sum_j a_{ij} = \phi$ and for all j, $\sum_i a_{ij} = 1$, then $\mathbf{A}^T\mathbf{A}$ has row sum ϕ.*

Proof. We define a column vector $\mathbf{c} \in \mathbb{R}^r$ and a row vector $\mathbf{b} \in \mathbb{R}^n$ where all the entries are ones. Using these vectors, we are able to express the row and columns sums of $\mathbf{A}$ in vector form as follows.

$$\mathbf{Ac} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n \text{ and } \mathbf{bA} = \begin{pmatrix} \phi \\ \phi \\ \vdots \\ \phi \end{pmatrix}^T \in \mathbb{R}^r. \tag{6.24}$$

From (6.24), we can see that we in fact have the following.

$$\mathbf{Ac} = \mathbf{b}^T \text{ and } \mathbf{bA} = \phi\mathbf{c}^T \tag{6.25}$$

We now use (6.25) and the vectors $\mathbf{c}$ and $\mathbf{b}$ to find the vector of row sums of $\mathbf{A}^T\mathbf{A}$ in the following.

$$\begin{aligned} \mathbf{A}^T\mathbf{Ac} &= \mathbf{A}^T\mathbf{b}^T \\ &= (\mathbf{bA})^T \\ &= \phi\mathbf{c}, \end{aligned} \tag{6.26}$$

as required. □

We next present the theorem for the 2-norm of the interpolation matrix $\mathbf{S}_{h_g}$.

Theorem 6.2.3. *Let $\mathbf{S}_{h_g} \in \mathbb{R}^{n\times gn}$ be the g resolution linear or cubic interpolation matrix whose columns are defined by Equation* (6.13)*. Then,*

$$\|\mathbf{S}_{h_g}\| = g^{-1/2}. \tag{6.27}$$

Proof. From Theorem 6.2.1, we can conclude that for both linear and cubic interpolation matrices $\mathbf{S}_{h_g}$, $\mathbf{C} = {\mathbf{S}_{h_g}}^T\mathbf{S}_{h_g}$ is circulant and thus takes the form (2.17). Now, $\mathbf{C}$ is a non-negative matrix where the row and column sums of $\mathbf{S}_{h_g}$ are 1 and $1/g$ respectively. Therefore, by Lemma 6.2.2, we have that the row sum of $\mathbf{C}$ is $1/g$ and by Theorem 2.1.5, we have that

$$\lambda_{\max}(\mathbf{C}) = 1/g. \tag{6.28}$$

Now, $\mathbf{S}_{h_g}$ is a real non-negative matrix, so for any choice of n that is compatible with the resolution choice, the matrix $\mathbf{C}$ is a non-negative, symmetric positive semidefinite matrix. Therefore, the 2-norm of $\mathbf{S}_{h_g}$ is given by

$$\|\mathbf{S}_{h_g}\| = \sqrt{\lambda_{\max}(\mathbf{C})}. \tag{6.29}$$

Substituting (6.28) into (6.29), we obtain

$$\|\mathbf{S}_{h_g}\| = g^{-1/2}, \tag{6.30}$$

as required. □

In the following section, we derive a bound on the background error correlation matrices used within our work.

6.2.2 Bounding the background error correlation matrix

Within the remainder of this chapter, we make the following assumption.

A14. *The correlation matrix $\mathbf{C}_B \in \mathbb{R}^{n\times n}$ is symmetric and circulant with entries c_{ij} that satisfy*

$$c_{ij} = \begin{cases} 1, & \text{for } i = j \\ c_{ji}, & \text{for } i \neq j, \end{cases} \tag{6.31}$$

where $i, j = 1, 2, \ldots, n$ and $c_{ji} \in [0, 1]$.

Let $\mathbf{C}_B \in \mathbb{R}^{n\times n}$ denote the full resolution background error correlation matrix. Following from the definition of the reduced resolution background error covariance matrix (2.64), the reduced resolution background error correlation matrix $\hat{\mathbf{C}}_B \in \mathbb{R}^{r\times r}$ is given as follows.

$$\hat{\mathbf{C}}_B = \mathbf{S}_l \mathbf{C}_B \mathbf{S}_l^T. \tag{6.32}$$

Looking at (6.32), one can see that to obtain $\hat{\mathbf{C}}_B$, the restriction operator $\mathbf{S}_l$ is applied to $\mathbf{C}_B$, this results in the removal of the odd rows of $\mathbf{C}_B$. The resulting matrix is then applied to $\mathbf{S}_l^T$, which results in the odd columns of $\mathbf{S}_l\mathbf{C}_B$ being removed. Therefore, $\hat{\mathbf{C}}_B$ retains a valid correlation structure that satisfies Assumption A14 at the reduced resolution as both odd rows and columns of $\mathbf{C}_B$ are being removed to form $\hat{\mathbf{C}}_B$.

Using Assumption A14, we next present a result on the norm of the square-root of the background error correlation matrix. This is used later in our work to deduce a bound on the condition number of the preconditioned 3D-Var Hessian (2.70).

Theorem 6.2.4. *Let $\mathbf{C}_B \in \mathbb{R}^{r\times r}$ be a correlation matrix that satisfies Assumption A14. Then the following bound holds,*

$$\|\mathbf{C}_B^{1/2}\| \leq \sqrt{r}. \tag{6.33}$$

Proof. We first note that as $\mathbf{C}_B$ is symmetric, its 2-norm is equal to its maximum eigenvalue. Furthermore, as it is positive symmetric circulant, we can calculate its maximum eigenvalue using (2.21).

We define a correlation matrix $\mathbf{C}_{B_{\max}}$ with entries given in (6.31), where $c_{ij} = 1$ for all i, j. So $\mathbf{C}_{B_{\max}}$ is an $r \times r$ matrix of ones. It follows from (2.21) that the 2-norm of $\mathbf{C}_B$ is given by

$$\|\mathbf{C}_B\| = \max|\lambda^{C_B}| = |\lambda_0^{C_B}| = \left|\sum_{j=1}^{r} c_{Bj}\right|. \tag{6.34}$$

It is easy to see from (6.34) that for any valid choice of $\mathbf{C}_B$, its row sum would be less than or equal to r, the row sum of $\mathbf{C}_{B_{\max}}$. Therefore, it follows that

$$\|\mathbf{C}_B\| \leq \|\mathbf{C}_{B_{\max}}\|, \tag{6.35}$$

Now, the entries of the square root of $\mathbf{C}_{B_{\max}}$ are given by

$$c_{ij}^{1/2} = 1/\sqrt{r},\ \forall i, j. \tag{6.36}$$

Using that $\mathbf{C}_B^{1/2}$ is circulant given that $\mathbf{C}_B$ is circulant, we have that

$$\begin{aligned} \|\mathbf{C}_{B_{\max}}^{1/2}\| &= r \times 1/\sqrt{r} \\ &= \sqrt{r}. \end{aligned} \tag{6.37}$$

Therefore, using (6.35) and (6.37), we have

$$\|\mathbf{C}_B^{1/2}\| \leq \sqrt{r}, \tag{6.38}$$

as required. □

We note that if $\mathbf{C}_B$ has no cross correlations, we have that $\|\mathbf{C}_B^{1/2}\| = \|\mathbf{I}_n\| = 1$, which is the lower bound on $\|\mathbf{C}_B^{1/2}\|$. We next test the bound from Theorem 6.2.4 using a background error correlation structure used in DA practice.

The UK Met Office system uses the SOAR (second-order auto-regressive) function to model horizontal correlations [74]. Within our numerical experiments, the structure of the background correlations follow the SOAR distribution with the error correlation matrix defined by

$$\mathbf{C}_B(i,j) = \left(1 + \frac{|2a\sin(\frac{\theta|i-j|}{2})|}{L}\right)\exp\left\{\frac{-|2a\sin(\frac{\theta|i-j|}{2})|}{L}\right\}, \tag{6.39}$$

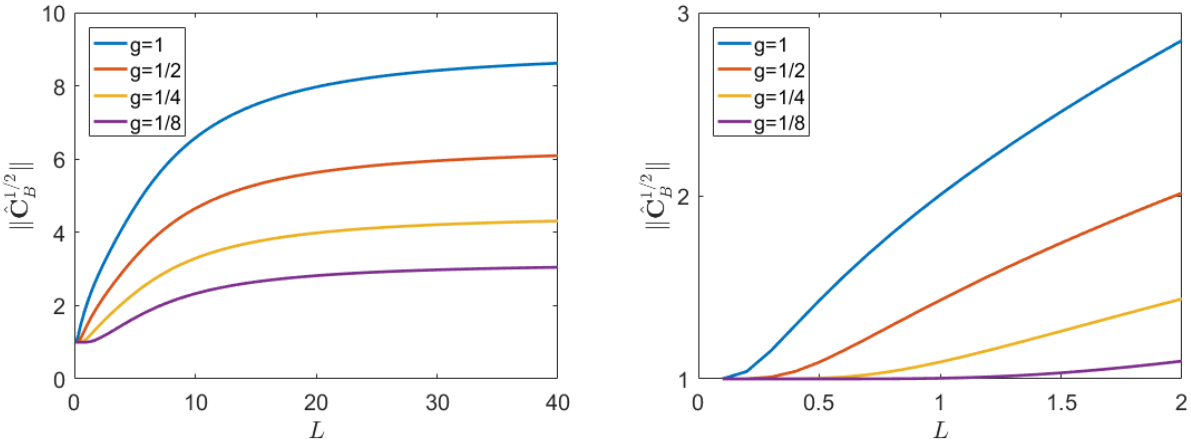
where i and j correspond to the rows and the columns of the correlation matrix respectively, $\theta = \frac{2\pi}{n}$ is the angle between two points on the circle, a is the radius of the circle, L is the correlation length-scale and the chordal distance is defined as $d = 2a\sin(\frac{\theta|i-j|}{2})$. We choose $a = 1/2\pi$ such that the circumference of the circle is 1.

The length scale of the correlation matrix controls the spread of errors along the grid. Increasing the correlation length scale results in an increase in the number of nearby background errors that are correlated with each other [55].

Now, the values of the cross correlations increase as the correlation length scale increases. This results in the off-diagonal entries of $\mathbf{C}_B$ tending to 1, thus the upper bound derived in Theorem 6.2.4 becoming tighter. This is visualised in Figure 6.2 where we choose $\mathbf{C}_B$ to be a SOAR correlation matrix and plot the results of $\|\hat{\mathbf{C}}_B^{1/2}\| = \|(\mathbf{S}_l\mathbf{C}_B\mathbf{S}_l^T)^{1/2}\|$ at different correlation length scales and resolutions.

As the entries of $\mathbf{C}_B$ are positive and monotonically decreasing away from the diagonal, it can be noted that increasing the correlation length scale results in an increase in the maximum eigenvalue [108, 112, 129], as illustrated for the SOAR matrix in Figure 2 of [42]. In our work, Figure 6.2(a) shows the norm of $\mathbf{C}_B^{1/2}$ (/its maximum eigenvalue) tending to $\sqrt{gn}$ as the length scale increases, where $r = gn$ in Theorem 6.2.4. Thus, the larger the length scale, the more correlated the background errors are, the tighter the bound in Theorem 6.2.4 would be.

Figure 6.2(b) shows the values of $\|\hat{\mathbf{C}}_B^{1/2}\|$ for more realistic choices of correlation length scale where we assume that not all background variables are highly correlated with each other. From this plot, we can see that the values of the norms for realistic choices of L are very far from the upper bound in Theorem 6.2.4, something we should consider when deriving a bound on the condition number of the preconditioned reduced resolution 3D-Var Hessian in the following section.



(a) $L_{max} = 40$ (b) $L_{max} = 2$

Figure 6.2: Plots of $\|\hat{\mathbf{C}}_B^{1/2}\|$ at different length scales L within the range $[0, L_{\max}]$ and resolutions, where $\mathbf{C}_B$ is a SOAR correlation matrix defined in (6.39) and $n = 80$.

6.2.3 Bounding the condition number of the Hessian

Recall from Section 2.3 that in VarDA practice, the variational problem is often preconditioned using a variable transformation to simplify calculations and solve the problem in real-time. Furthermore, recall from Section 2.2.1, the preconditioned 3D-Var Hessian is symmetric positive definite. Therefore, its condition number in the 2-norm, $\kappa(\nabla^2 \hat{\mathcal{J}}_p)$, is the ratio of its largest and smallest eigenvalues and is related to the number of iterations used for the linear minimisation problems in VarDA and how sensitive the estimate of the initial state is to perturbations of the data. We can use the condition number of the Hessian to indicate how quickly and accurately the optimisation problem can be solved [47].

Bounds on the condition number of the preconditioned 3D-Var Hessian are derived in the work of [56], [57] and [123]. However, these works only considered the full resolution Hessian. In this chapter, our main result is the derivation of a bound on condition number of the preconditioned 3D-Var Hessian in terms of the resolution parameter g, outlined in this section.

The condition number of the reduced resolution preconditioned Hessian, $\kappa(\nabla^2 \hat{\mathcal{J}}_p)$, is heavily dependent on the structure of the observation operator. In VarDA practice, we have fewer observations than state variables, even in reduced resolution space. Therefore, it is reasonable to make the following assumption.

A15. *The reduced resolution linearised observation operator* $\hat{\mathbf{H}}$ *is low rank.*

If A15 holds, we have that the minimum eigenvalue of the reduced resolution preconditioned Hessian is 1, so the condition number of the preconditioned inner loop Hessian is equal to its maximum eigenvalue [123]. In the following theorem, we derive a bound on the condition

number of the reduced resolution preconditioned 3D-Var Hessian (2.70) in terms of the resolution parameter g.

Theorem 6.2.5 (Upper bound on the condition number of the reduced resolution preconditioned 3D-Var Hessian)**.** *Let $\nabla^2\hat{\mathcal{J}}_p$ denote the preconditioned 3D-Var Hessian as defined in* (2.70) *with $\mathbf{B} = \sigma_b^2\mathbf{C}_B$ and $\mathbf{R} = \sigma_o^2\mathbf{I}$, where σ_b^2 and σ_o^2 are the background and observation error variances respectively and $\mathbf{C}_B$ is a correlation matrix. Furthermore, assume A14 and A15 hold. Then the condition number of the preconditioned 3D-Var Hessian satisfies,*

$$\kappa(\nabla^2\hat{\mathcal{J}}_p) \leq 1 + \frac{\sigma_b^2}{\sigma_o^2} g^{-1} \|\hat{\mathbf{C}}_B^{1/2}\|^2 \|\mathbf{H}\|^2, \tag{6.40}$$

where g is the resolution parameter and $\hat{\mathbf{C}}_b$ is the reduced resolution background error correlation matrix defined by (6.32)*.*

Proof. As (2.70) is a symmetric positive definite matrix, it follows that,

$$\kappa(\nabla^2\hat{\mathcal{J}}_p) = \frac{\lambda_{\max}(\nabla^2\hat{\mathcal{J}}_p)}{\lambda_{\min}(\nabla^2\hat{\mathcal{J}}_p)}. \tag{6.41}$$

From A15, we have that $\lambda_{\min}(\nabla^2\hat{\mathcal{J}}_p) = 1$. Therefore, the condition number of (2.70) for the 3D-Var problem is given by its 2-norm as follows

$$\kappa(\nabla^2\hat{\mathcal{J}}_p) = \|\mathbf{I} + \hat{\mathbf{B}}^{1/2}\hat{\mathbf{H}}^T\mathbf{R}^{-1}\hat{\mathbf{H}}\hat{\mathbf{B}}^{1/2}\|. \tag{6.42}$$

Substituting $\mathbf{B} = \sigma_b^2\mathbf{C}_B$ and $\mathbf{R} = \sigma_o^2\mathbf{I}$ into (6.42) and applying the triangle inequality (2.6), we have

$$\begin{aligned}
\kappa(\nabla^2\hat{\mathcal{J}}_p) &= \left\| \mathbf{I} + \frac{\sigma_b^2}{\sigma_o^2}\hat{\mathbf{C}}_B^{1/2}\mathbf{S}_h^T\mathbf{H}^T\mathbf{H}\mathbf{S}_h\hat{\mathbf{C}}_B^{1/2} \right\| \\
&\leq \|\mathbf{I}\| + \frac{\sigma_b^2}{\sigma_o^2}\|\hat{\mathbf{C}}_B^{1/2}\mathbf{S}_h^T\mathbf{H}^T\mathbf{H}\mathbf{S}_h\hat{\mathbf{C}}_B^{1/2}\| \\
&\leq 1 + \frac{\sigma_b^2}{\sigma_o^2}\|\hat{\mathbf{C}}_B^{1/2}\mathbf{S}_h^T\mathbf{H}^T\mathbf{H}\mathbf{S}_h\hat{\mathbf{C}}_B^{1/2}\|.
\end{aligned} \tag{6.43}$$

Using the submultiplicative property (2.12) and our result from Theorem 6.2.3, (6.27), we deduce a bound on (6.42) for the preconditioned 3D-Var problem in the following.

$$\begin{aligned}
\kappa(\nabla^2\hat{\mathcal{J}}_p) &\leq 1 + \frac{\sigma_b^2}{\sigma_o^2}\|\mathbf{S}_h\|^2\|\hat{\mathbf{C}}_B^{1/2}\|^2\|\mathbf{H}\|^2 \\
&\leq 1 + \frac{\sigma_b^2}{\sigma_o^2} g^{-1}\|\hat{\mathbf{C}}_B^{1/2}\|^2\|\mathbf{H}\|^2,
\end{aligned} \tag{6.44}$$

as required. □

In Theorem 6.2.5, we have theoretically derived a bound on the condition number of the preconditioned inner loop Hessian of the incremental method (6.40) that depends on the resolution parameter g. This bound indicates that as the inner loop resolution is reduced,

the bound on the condition number of the preconditioned 3D-Var Hessian increases. In Section 6.4, we use numerical experiments to understand if this relationship between the level of resolution and the actual value of the condition number of the preconditioned 3D-Var Hessian exists.

The bound in Theorem 6.2.5 holds for all $\mathbf{C}_B$ that satisfy Assumption A14. Note that by keeping the background error correlation matrix at the low resolution in (6.40), we impose a tighter bound on the condition number of the preconditioned 3D-Var Hessian in the reduced resolution. This can be shown using the submultiplicative property (2.12) and the expression for $\|\mathbf{S}_l\|$ in (6.17) as follows.

$$\begin{aligned} \|\hat{\mathbf{C}}_B\| &= \|\mathbf{S}_l \mathbf{C}_B \mathbf{S}_l^T\| \\ &\leq \|\mathbf{S}_l\|^2 \|\mathbf{C}_B\| \\ &\leq \|\mathbf{C}_B\|. \end{aligned} \tag{6.45}$$

From our result in Theorem 6.2.4, (6.33) and Figure 6.2(a), we know that as the length scale L of a SOAR correlation matrix is increased, $\|\hat{\mathbf{C}}_B^{1/2}\|$ tends to the upper bound in (6.33) and for relatively large choices of L, the upper bound is attained. Therefore, the dependence on g in (6.40) is cancelled out, as shown in the following.

Substituting (6.33) into (6.40), we have

$$\begin{aligned} \kappa(\hat{\nabla}^2 \mathcal{J}_p) &\leq 1 + \frac{\sigma_b^2}{\sigma_o^2} g^{-1} \times gn \|\mathbf{H}\|^2 \\ &\leq 1 + \frac{\sigma_b^2}{\sigma_o^2} n \|\mathbf{H}\|^2. \end{aligned} \tag{6.46}$$

The bound in (6.46) shows that the presence of large background error correlations removes the effect of g in our bound (6.40). In VarDA practice, the background variables are not necessarily very strongly correlated. In Section 6.4, we assess the tightness of the bound (6.40) when using realistic choices of correlation length scale. We then use assimilation experiments to see if the behaviour we observe in the condition number is reflected in the accuracy of the analysis obtained by the incremental method with different levels of inner loop resolution. We first explain the experimental design used for these experiments in the following section.

6.3 Experimental design

In this section, we construct six 3D-Var problems with linear and nonlinear observation operators with the aim of understanding the effects of using a reduced resolution inner loop on the convergence of the incremental 3D-Var method, Algorithm 3.4.1. The design of our numerical experiments is chosen to align with what occurs in practical implementations of the incremental method. That is, we consider

- the preconditioned problem (2.69).

- observations with errors, where the observations are not at every grid-point.
- initialising with the background state vector.
- the use of linear and nonlinear observation operators.
- the use of interpolation matrices to map from the reduced resolution grid to the outer loop resolution grid.
- the use of correlated background error by the use of a SOAR matrix.
- the frequencies resolved in the analysis.

Twin experiments are commonly used to test DA methods. They use synthetic observations as well as error statistics that satisfy the DA assumptions. We next define the choices made for the twin experimental design used, beginning with generating the reference state, $\mathbf{x}^{ref}$, which is used as the basis of a twin experiment in the definition of the background state (the initial guess for the optimisation algorithms) as well as to generate the observations.

Reference state We choose the reference state $\mathbf{x}^{ref} \in \mathbb{R}^n$ to be the sine wave, with entries given by

$$\mathbf{x}_i^{ref} = \sin\left(\frac{2\pi}{n}(i-1)\right), \tag{6.47}$$

where $i = 1, 2, \ldots, n$ and we choose $n = 80$ for all our experiments as this satisfies $gn \in \mathbb{N}$ for all the resolution choices we consider. The background state vector is generated using (6.47) and is defined in the following.

Background In VarDA, the initial guess for the optimisation algorithm is taken to be the background state, $\mathbf{x}^b$, which incorporates information from previous forecasts. In our experiments, the background state vector $\mathbf{x}^b$ is generated by adding Gaussian noise

$$\varepsilon_{\mathbf{b}} \sim \mathcal{N}(0, \mathbf{B}), \tag{6.48}$$

to the reference state, $\mathbf{x}^{ref}$. In our experiments, we choose $\mathbf{B}$ to be of the form $\mathbf{B} = \sigma_b^2 \mathbf{C}_B$, where σ_b^2 is the background error variance. The standard deviations of the errors from the reference solution are based on the average order of magnitude of the entries of $\mathbf{x}^{ref}$. In our work we choose $\sigma_b^2 = 0.004$, which represents a 10% SD of the error.

For the correlation structure, we choose $\mathbf{C}_B = \mathbf{I}_n$ or $\mathbf{C}_B$ to be a SOAR correlation matrix. For the correlation length-scales used in our work, we choose $L_1 = 0.5\Delta x$, $L_2 = \Delta x$ and $L_3 = 1.5\Delta x$, where $\Delta x = 1/n$. The background error covariance matrix with associated correlation length scale L_i is denoted by $\mathbf{B}_i = \sigma_b^2 \mathbf{C}_B$, where $i = 1, 2, 3$. We denote the case where $\mathbf{C}_B = \mathbf{I}_n$ as $\mathbf{C}_{B_0}$.

As previously mentioned, we generate synthetic observations using the reference state, $\mathbf{x}^{ref}$. We next describe the choices we make when specifying these observations.

Observations In order to test the bound (6.40), we need to choose observation operators that are low rank (satisfying Assumption A15). Recall that the definition of the reduced resolution linearised observation operator $\hat{\mathbf{H}}$ is given in (2.65). By assuming $\mathbf{S}_h$ is a linear (or cubic) interpolation operator, as outlined in Section 6.1, we are removing every $1/g^{th}$ column of $\mathbf{H}$ to obtain $\hat{\mathbf{H}}$. Therefore, if the full resolution linearised observation operator $\mathbf{H}$ has observations at even grid-points, the use of $g = 1/2$ would result in the reduced resolution linearised observation operator being full rank. A similar problem occurs when every 4^{th} point is observed with $g = 1/4$ and when every 8^{th} point is observed with $g = 1/8$. This is not realistic in DA problems as in the reduced resolution space, $p < gn$. Therefore, the observation operators used in our work are designed to avoid this problem and are outlined as follows.

We consider three linear and three nonlinear 3D-Var problems with different observation operators, denoted by LinProb$j$ and NonlinProb$j$, respectively, where $j = 1, 2, 3$. LinProb$j$ have single observations of x_i of the first $p = n/4$, $p = n/2$ and $p = 3n/4$ variables for $j = 1$, $j = 2$ and $j = 3$ respectively. NonlinProb$j$ have single observations of x_i^3 of the first $p = n/4$, $p = n/2$ and $p = 3n/4$ variables for $j = 1$, $j = 2$ and $j = 3$ respectively. These choices of observation operators of different structures satisfy Assumption A15 and allow us to test if our results hold for different spatial locations of observations, and in the presence of nonlinearities.

We assume that for all problems, $\mathcal{H}$ is the exact observation operator used to map to observation space. We use imperfect observations where the observations, $\mathbf{y}$, are generated by adding Gaussian noise

$$\varepsilon_{\mathbf{o}} \sim \mathcal{N}(0, \mathbf{R}), \tag{6.49}$$

to $\mathcal{H}(\mathbf{x}^{ref})$. For the observation error covariance matrix we choose $\mathbf{R}$ to be a diagonal matrix of the form $\mathbf{R} = \sigma_o^2 \mathbf{I}_p$, where σ_o^2 is the observation error variance. For all experiments, we set the standard deviation of the observation error to be 5% of the average order of magnitude of the entries of $\mathcal{H}(\mathbf{x}^{ref})$. For LinProb1, LinProb2 and LinProb3 this is $\sigma_o^2 = 9.342 \times 10^{-4}, 0.001$ and 9.8583×10^{-4}, respectively. For NonlinProb1, NonlinProb2 and NonlinProb3 this is $\sigma_o^2 = 3.9883 \times 10^{-4}, 4.5032 \times 10^{-4}$ and 4.3281×10^{-4}, respectively.

Algorithmic choices To ensure the robustness of our results, we consider a series of n_r randomly generated problems, where the randomness occurs through the background and observation error vectors, $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$. For each realisation, a new $\varepsilon_{\mathbf{b}}$ and $\varepsilon_{\mathbf{o}}$ are generated from their respective distributions, (6.48) and (6.49). We choose $n_r = 100$. For the linear problems, LinProb1, LinProb2 and LinProb3, both sides of (6.40) are fixed across all realisations. This is not the case for the nonlinear problems, NonlinProb1, NonlinProb2 and NonlinProb3, as there is a dependency of $\mathbf{H}$ on $\mathbf{x}$. Therefore, in the result tables, we take the average across all realisations for the nonlinear problems.

For our assimilation experiments, as the dimensions of the problems used within our work are relatively small compared to DA systems in practice, the inner loop problem is solved using MATLAB's backslash operator where an appropriate solver is chosen according to the

properties of the Hessian matrix $\nabla^2 \hat{\mathcal{J}}_p$ (see [86] for more details). The limit on the total number of function and Jacobian evaluations is achieved by using the criterion (3.45), where we set $\tau_e = 8$ to mimic the number of evaluations used operationally in the ECMWF Integrated Forecasting System [34]. Note that for the incremental method, the use of $\tau_e = 8$ is equivalent to $k_{\max} = 4$ outer loop iterations.

One cannot simply rely on the numerical values in the analysis as these can on one hand show that the analysis does not resemble the true solution whereas, if one takes the DFT (see Definition 2.1.21) of the analysis vector, the power spectra of the DFT may suggest otherwise. Within our work, we use the DFT to analyse the effect that the use of a reduced resolution inner loop in the incremental method has on the frequencies resolved in the analysis. We next outline the choices we make in relation to the DFT.

DFT For one-dimensional grids, waves can only be represented using at least three grid-points or two grid lengths. The choice of $n = 80$ is large enough so that we have enough points to accurately represent the sine wave (our choice of reference state) in full resolution space.

We use the inbuilt MATLAB function fft() to return the Fourier transform using a fast Fourier transform (FFT); an algorithm that reduces the computational cost of the DFT by reducing the number of computer operations required from $O(n^2)$ to $O(nlogn)$.

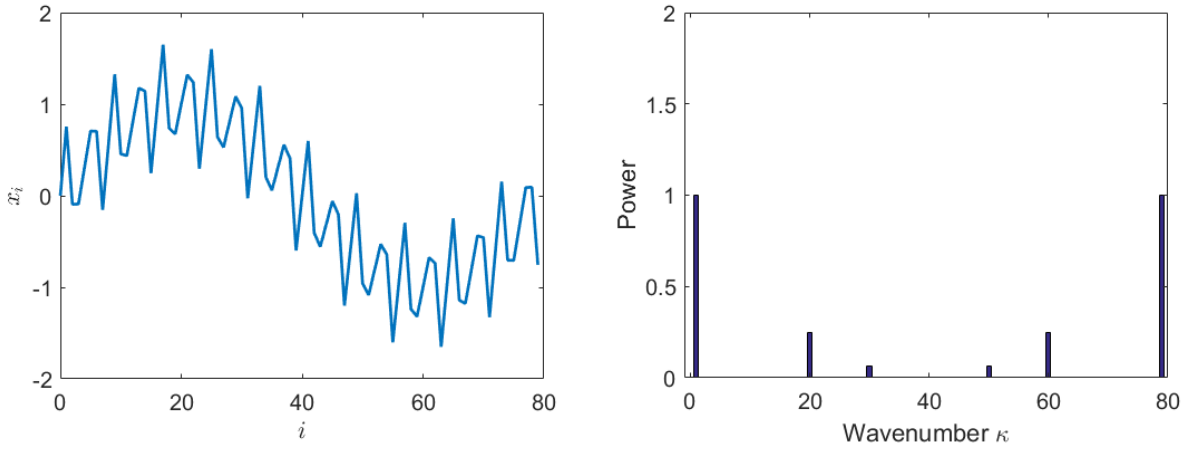
Recall from Section 2.1.3, the power spectrum of a DFT output is able to tell us which wavenumbers are present in the function being transformed and how important they are and thus allow us to effectively analyse the frequencies of the 3D-Var output. We outline the details of the power spectra in the following.

Power spectra We define the value of the function at n equally spaced points to be the reference state used in our assimilation experiments, with entries given by

$$\mathbf{x}_i^{ref} = \sin\left(\frac{2\pi}{n}(i-1)\right) + \frac{1}{2}\sin\left(20\frac{2\pi}{n}(i-1)\right) + \frac{1}{4}\sin\left(30\frac{2\pi}{n}(i-1)\right) \tag{6.50}$$

for $i = 0, \ldots, n-1$. The reference state (6.50) has higher frequencies compared to (6.47), enabling us to study the effect that the use of a reduced resolution inner loop has on the accuracy of both the large and small scales. A plot of the function (6.50) is given in 6.3(a) with its corresponding power spectrum shown in Figure 6.3(b), with the wavenumber on the x-axis and the corresponding power (2.49) on the y axis. From these plots, we can identify the wavenumbers we would like to recover in our assimilation experiments.

By calculating the power of the DFT coefficients $\mathcal{F}_\kappa$ of the 3D-Var analysis and analysing the power spectra, we are able to visualise which frequencies are resolved when using the incremental method with different spatial resolutions for the inner loop.



(a) (b)

Figure 6.3: Plot of function (6.50) in (a) and the power spectrum of its DFT in (b).

6.4 Numerical results

With the aim of developing an understanding about the effect of the use of the reduced resolution components in the incremental method have on the accuracy of the 3D-Var analysis, in this section, we test the bound on the condition number of the preconditioned 3D-Var Hessian (6.40) using the experimental design outlined in Section 6.4 and perform assimilation experiments to see the implications of using a reduced resolution inner loop problem on the accuracy of the 3D-Var analysis.

6.4.1 Condition number bound tests

Tables 6.1 and 6.2 show the results from the experiments where we test the bound on the condition number of the preconditioned Hessian for each of the linear and nonlinear observation operators outlined in Section 6.3 respectively. For each problem and choice of background error correlation matrix within these tables, we include the value of the condition number at the optimal solution, $\kappa(\nabla^2 \hat{\mathcal{J}}_p)$, which is calculated using the built-in function cond() in MATLAB. We also include the value of the bound (6.40) and the norm of the reduced resolution observation operator $\|\hat{\mathbf{H}}\|$ for each choice of the resolution parameter g. Note that, for the linear problems, $\mathbf{H}$ does not depend on the value of $\mathbf{x}$. Therefore, $\|\hat{\mathbf{H}}\|$ is fixed for all choices of background error correlation matrix. Furthermore, for the linear problems, the values in the tables are fixed for all realisations. However, this is not the case for the nonlinear problems, for which we include the averaged values in the tables. We therefore also include plots of the condition number for individual realisations of the nonlinear problems later in this section.

We begin by discussing the quality of the bound (6.40) for different choices of the background error correlation matrix. In both Tables 6.1 and 6.2, when $g = 1$ and $\mathbf{C}_{B_0}$ is used, the bound is exactly equal to the actual value of the condition number. This is because, when deriving

Table 6.1: Table of values of the condition number of the preconditioned 3D-Var Hessian (2.70), the bound (6.40) and $\|\hat{\mathbf{H}}\|$ of LinProb1, LinProb2 and LinProb3, for different choices of g and $\mathbf{C}_B$.

			$g=1$	$g=1/2$	$g=1/4$	$g=1/8$
LinProb1	$\mathbf{C}_{B_0}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	5.33	9.57	17.52	30.62
		(6.40)	5.33	9.67	18.34	35.67
	$\mathbf{C}_{B_1}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	9.74	11.13	17.61	30.62
		(6.40)	9.84	18.67	36.34	71.68
	$\mathbf{C}_{B_2}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	17.75	17.91	20.27	30.75
		(6.40)	18.41	35.83	70.65	140.30
	$\mathbf{C}_{B_3}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	25.21	25.13	25.66	31.94
		(6.40)	27.20	53.40	105.80	210.60
		$\|\hat{\mathbf{H}}\|$	1.00	1.41	1.95	2.61
LinProb2	$\mathbf{C}_{B_0}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	5.00	8.98	16.78	31.42
		(6.40)	5.00	9.00	17.00	33.00
	$\mathbf{C}_{B_1}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	9.13	10.48	16.88	31.42
		(6.40)	9.15	17.31	33.62	66.24
	$\mathbf{C}_{B_2}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	16.90	17.13	19.69	31.60
		(6.40)	17.07	33.14	65.29	129.57
	$\mathbf{C}_{B_3}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	24.61	24.65	25.57	33.19
		(6.40)	25.18	49.36	97.73	194.46
		$\|\hat{\mathbf{H}}\|$	1.00	1.41	1.99	2.76
LinProb3	$\mathbf{C}_{B_0}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	5.11	9.20	17.32	33.07
		(6.40)	5.11	9.21	17.43	33.85
	$\mathbf{C}_{B_1}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	9.36	10.75	17.42	33.07
		(6.40)	9.37	17.74	34.49	67.98
	$\mathbf{C}_{B_2}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	17.42	17.68	20.40	33.28
		(6.40)	17.50	34.00	67.00	133.01
	$\mathbf{C}_{B_3}$	$\kappa(\nabla^2\hat{\mathcal{J}}_p)$	25.55	25.61	26.68	35.07
		(6.40)	25.83	50.66	100.31	199.62
		$\|\hat{\mathbf{H}}\|$	1.00	1.41	1.99	2.79

the bound on the condition number of the preconditioned 3D-Var Hessian, the following approximation is made

$$\|\hat{\mathbf{C}}_{B_0}^{1/2}\hat{\mathbf{H}}^T\mathbf{H}\hat{\mathbf{C}}_{B_0}^{1/2}\| \leq g^{-1}\|\hat{\mathbf{C}}_{B_0}^{1/2}\|^2\|\mathbf{H}\|^2. \tag{6.51}$$

Now, when $g=1$, $g^{-1}=1$ and $\hat{\mathbf{C}}_{B_0}=\mathbf{C}_{B_0}=\mathbf{I}_n$. By substituting these quantities into (6.51), it is clear to see that the equality holds, thus the equality holds for (6.40) when using $g=1$ and $\mathbf{C}_{B_0}$.

When using $\mathbf{C}_{B_0}$ for any choice of g, the bounds are considerably close to the actual values compared to when correlated background error is included. This is because $\|\mathbf{C}_{B_0}\|=1$, thus, the influence of the term $\|\mathbf{C}_{B_0}\|$ is non-existent in the bound and the only influence comes from the observation operator. Looking at the values of $\|\hat{\mathbf{H}}\|$ in each of the tables,

Table 6.2: Table of averaged values of the condition number of the preconditioned 3D-Var Hessian (2.70), the bound (6.40) and $\|\hat{\mathbf{H}}\|$ of NonlinProb1, NonlinProb2 and NonlinProb3 for different choices of g and $\mathbf{C}_B$, where $n_r = 100$.

			$g = 1$	$g = 1/2$	$g = 1/4$	$g = 1/8$
NonlinProb1	$\mathbf{C}_{B_0}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	91.49	145.91	229.14	346.22
		(6.40)	91.49	181.99	362.98	724.95
		$\|\hat{\mathbf{H}}\|$	2.99	3.78	4.74	5.83
	$\mathbf{C}_{B_1}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	151.60	165.15	229.97	346.17
		(6.40)	185.45	369.90	738.80	1476.60
		$\|\hat{\mathbf{H}}\|$	2.99	3.78	4.74	5.83
	$\mathbf{C}_{B_2}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	250.08	244.16	255.10	346.93
		(6.40)	364.39	727.78	1454.56	2908.12
		$\|\hat{\mathbf{H}}\|$	2.98	3.78	4.74	5.83
	$\mathbf{C}_{B_3}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	323.67	315.82	303.94	353.87
		(6.40)	547.49	1093.97	2186.95	4372.90
		$\|\hat{\mathbf{H}}\|$	2.98	3.78	4.74	5.83
NonlinProb2	$\mathbf{C}_{B_0}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	83.50	151.01	272.82	449.51
		(6.40)	83.50	166.01	331.01	661.02
		$\|\hat{\mathbf{H}}\|$	3.03	4.08	5.50	7.06
	$\mathbf{C}_{B_1}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	154.14	175.35	274.08	449.50
		(6.40)	169.19	337.38	673.76	1346.51
		$\|\hat{\mathbf{H}}\|$	3.03	4.08	5.50	7.06
	$\mathbf{C}_{B_2}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	279.97	279.82	311.97	451.31
		(6.40)	332.23	663.45	1325.90	2650.81
		$\|\hat{\mathbf{H}}\|$	3.03	4.08	5.50	7.06
	$\mathbf{C}_{B_3}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	389.69	386.10	388.82	467.30
		(6.40)	498.57	996.14	1991.29	3981.57
		$\|\hat{\mathbf{H}}\|$	3.03	4.08	5.50	7.06
NonlinProb3	$\mathbf{C}_{B_0}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	86.97	157.13	283.87	467.48
		(6.40)	86.97	172.94	344.88	688.75
		$\|\hat{\mathbf{H}}\|$	3.03	4.09	5.50	7.06
	$\mathbf{C}_{B_1}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	160.39	182.45	285.18	467.46
		(6.40)	176.21	351.42	701.84	1402.68
		$\|\hat{\mathbf{H}}\|$	3.03	4.09	5.50	7.06
	$\mathbf{C}_{B_2}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	291.28	291.12	324.57	469.34
		(6.40)	345.87	690.74	1380.48	2759.96
		$\|\hat{\mathbf{H}}\|$	3.03	4.09	5.50	7.06
	$\mathbf{C}_{B_3}$	$\kappa(\nabla^2 \hat{\mathcal{J}}_p)$	405.40	401.65	404.48	485.97
		(6.40)	519.02	1037.05	2073.09	4145.19
		$\|\hat{\mathbf{H}}\|$	3.03	4.09	5.50	7.06

one can see that the value grows as g increases. This is because, when deriving the bound on the condition number of the preconditioned 3D-Var Hessian, the following approximation

is made

$$\|\hat{\mathbf{H}}\| \leq g^{-1/2}\|\mathbf{H}\|. \tag{6.52}$$

Looking at (6.52), one can see that reducing the resolution increases the upper bound on $\|\hat{\mathbf{H}}\|$. In addition, our numerical results in both Tables 6.1 and 6.2 show the actual increase in $\|\hat{\mathbf{H}}\|$ (the LHS of (6.52)). The RHS of (6.52) can be easily derived by multiplying $\|\hat{\mathbf{H}}\|$ in the tables where $g = 1$ (full resolution) by $g^{-1/2} = 1.41$, $g^{-1/2} = 2$ and $g^{-1/2} = 2.83$ for the bound on $\|\hat{\mathbf{H}}\|$ where a 1/2, 1/4 and 1/8 resolution is used, respectively.

For the linear problems, $\|\mathbf{H}\| = 1$, so the RHS of (6.52) is $g^{-1/2}$. For the results of LinProb1 in Table 6.1, the upper bound in (6.52), $g^{-1/2}$, is very close to the actual value $\|\hat{\mathbf{H}}\|$ for all choices of g. As the number of observations are increased for LinProb2 and LinProb3, Table 6.1 shows that the actual values of $\|\hat{\mathbf{H}}\|$ for these problems are even closer to the upper bound, $g^{-1/2}$. This is because, when there are more observations distributed across the spatial domain, there are more nonzero entries in $\mathbf{H}$ on the full resolution grid that need to be interpolated to neighbouring grid-points on the reduced resolution grid. Therefore, the role of the interpolation operator $\mathbf{S}_{h_g}$ is more prevalent as instead of interpolating columns of zeros in $\mathbf{H}$ for the majority (three quarters) of the spatial domain to construct $\hat{\mathbf{H}}$ for LinProb1, more nonzero values are being interpolated for LinProb2 and LinProb3 that have observations across 1/2 and 3/4 of the domain. Thus, in the presence of more observations in space, our numerical results indicate that the quality of the approximation (6.52) improves, thus the bound (6.40) improves. We see a similar result for the nonlinear observation operator results in Table 6.2.

Looking again at the values of $\|\hat{\mathbf{H}}\|$ for each problem in both Tables 6.1 and 6.2, we see that the quality of the approximation (6.52) deteriorates with g for all choices of observation operator. This is because there is a greater mismatch between $\|\mathbf{H}\mathbf{S}_{h_g}\|$ and $\|\mathbf{H}\|\|\mathbf{S}_{h_g}\|$ as the spatial resolution decreases. The reason for this behaviour can be seen by considering the structure of the interpolation matrices $\mathbf{S}_{h_g}$. As g is reduced, the number of nonzero column entries in the interpolation matrix increases. Thus, the number of nonzero cross multiplications with the nonzero elements in $\mathbf{H}$ increases in $\mathbf{H}\mathbf{S}_{h_g}$. Therefore, by using the submultiplicative property of the norm to derive the bound on $\|\hat{\mathbf{H}}\|$, we are splitting the calculation $\mathbf{H}\mathbf{S}_{h_g}$ and essentially neglecting more and more of these cross multiplications as the resolution is reduced. Furthermore, as the resolution is reduced, the correction to the innovation vector using $\hat{\mathbf{H}}$, $\hat{\mathbf{H}}\delta\hat{\mathbf{x}}$, is further away from the correction to the innovation vector obtained by using the full resolution observation operator $\mathbf{H}$, $\mathbf{H}\delta\mathbf{x}$. This latter point also corresponds with the findings of [127]. This behaviour is only natural as the ability for the reduced grid to accurately represent the full resolution grid is limited as interpolation is not exact. Therefore, if the grid-points on the low resolution grid are far apart, the inexactness of the interpolation procedure can produce large errors. Interpolation techniques work by giving a greater weighting to the nearby variables, but this does not avoid the issue of having a large grid length.

Similar to what we see in the linear observation operator results, for the nonlinear observation operator results in Table 6.2, we see that the error between the bound (6.40) and the

actual value of the condition number increases as g decreases. However, unlike in the linear cases with uncorrelated background error, the increase in the bound as g decreases is not of the same proportion as the actual value of the condition number, i.e. halving the resolution does not result in doubling the condition number. So there is a greater difference between the bound (6.52) and the actual value of $\|\hat{\mathbf{H}}\|$ than there was for the linear problems. Looking at the values of $\|\hat{\mathbf{H}}\|$ in the tables, we can see why this is. The approximation $g^{-1/2}\|\mathbf{H}\|$ of $\|\hat{\mathbf{H}}\|$ in (6.52) is much worse in the nonlinear observation operator case, which is consequently causing the bound (6.40) to perform worse than it was in the linear observation operator case.

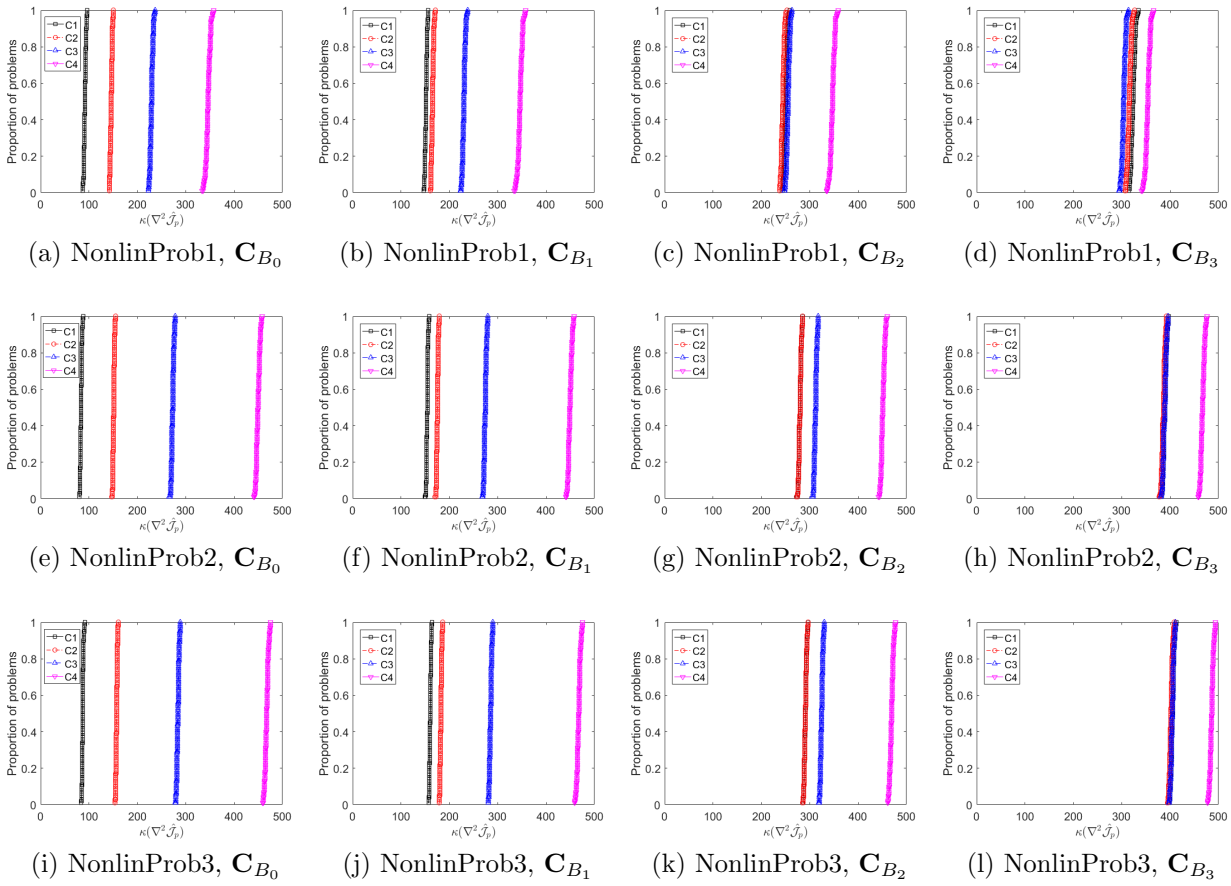
In both Tables 6.1 and 6.2, we see that as stronger background error correlations are introduced, the tightness of the bound on the condition number deteriorates. Table 6.2 for the nonlinear observation operator case shows the same relationship as in the linear problems, except there is a much more drastic difference between the true values and the bound. This again can be attributed to the approximation of $\|\hat{\mathbf{H}}\|$.

Recall from Section 6.3, strengthening the background error correlations is achieved by increasing the correlation length scale. From the work of [57] we know that the VarDA system is sensitive to the choice of length scale L. Haben et al. showed that as the length scale of the background error correlations increase, the condition number of the preconditioned VarDA Hessian increases. This conclusion can be seen in our results in both Tables 6.1 and 6.2. Looking at these numerical results, we find that the correlation length scale has a greater effect on the condition number of the preconditioned 3D-Var Hessian than the reduced resolution does. In fact, we see the behaviour that we expect to see considering the larger bound on the condition number (6.46). Recall that this bound showed that the presence of large background error correlations removes the effect of g in our bound (6.40). We have seen numerically that the difference between the values of the condition number reduces as stronger background error correlations are introduced. So the resolution reduction does not affect the condition number as much as the correlation length scale does.

This relationship between g, $\mathbf{C}_B$ and the condition number of the preconditioned 3D-Var Hessian (6.40) can be clearly visualised for the nonlinear problems in Figure 6.4. Within each plot and for each Case 1-4, we calculate $n_r = 100$ realisations of the actual value of the condition number of the preconditioned 3D-Var Hessian and plot the proportion of the n_r realisations (y-axis) that lie below a given value of the condition number (x-axis).

From Figure 6.4, one can see that for the uncorrelated background error case, as the resolution is reduced, the condition number increases. Furthermore, as we include more background error correlations, the differences between the four cases decrease, as we would expect from (6.46), and the condition number worsens. A similar result can be seen from similar plots for the linear problems. However, these are not included as for the linear problems, the values are fixed across all realisations, thus the Table 6.1 suffices.

So far within these experiments we have tested the bound (6.40) in Theorem 6.2.5 on both linear and nonlinear observation error operators, where we considered both correlated and uncorrelated background error structures. We found that when background error correla-



(a) NonlinProb1, $\mathbf{C}_{B_0}$ (b) NonlinProb1, $\mathbf{C}_{B_1}$ (c) NonlinProb1, $\mathbf{C}_{B_2}$ (d) NonlinProb1, $\mathbf{C}_{B_3}$

(e) NonlinProb2, $\mathbf{C}_{B_0}$ (f) NonlinProb2, $\mathbf{C}_{B_1}$ (g) NonlinProb2, $\mathbf{C}_{B_2}$ (h) NonlinProb2, $\mathbf{C}_{B_3}$

(i) NonlinProb3, $\mathbf{C}_{B_0}$ (j) NonlinProb3, $\mathbf{C}_{B_1}$ (k) NonlinProb3, $\mathbf{C}_{B_2}$ (l) NonlinProb3, $\mathbf{C}_{B_3}$

Figure 6.4: Plots of the condition number for the Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for NonlinProb1 in (a)-(d) and NonlinProb2 in (e)-(h), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

tions are included, the use of a reduced resolution grid has less of an effect on the condition number. Therefore, the bound on the condition number (6.40) is closest to the actual value of the condition number when there are no background error correlations. Furthermore, we found that the bound performs best in the linear observation operator case as this resulted in a more accurate approximation of the norm of the reduced resolution linearised observation operator.

Although in our work we solve the 3D-Var inner loop problem exactly, recall from Section 2.2.1 that the condition number of the preconditioned VarDA Hessian (2.70) can be used to indicate the accuracy we could be able to achieve when solving the linear minimisation problems in VarDA. The values of (2.70) in both Tables 6.1 and 6.2 are reasonably small indicating that for the linear problems, we can expect to lose 2 figures of accuracy and for the nonlinear problems, we can expect to lose 3 figures of accuracy due to loss of arithmetic precision. These figures indicate that a high accuracy is achievable for both the linear and nonlinear problems. We next consider the effect of using a reduced resolution inner loop for

the six problems on the accuracy of the analysis using assimilation experiments.

6.4.2 Accuracy of the analysis and frequencies resolved

We recall that the initial guess of the algorithms is the reference state $\mathbf{x}^{ref}$ perturbed by the background error $\varepsilon_{\mathbf{b}}$. In order to compare the quality of the estimate obtained by the incremental method at different inner problem resolutions, we compare the estimate to the reference state $\mathbf{x}^{ref}$ to understand how far the estimates obtained at different inner problem resolutions have deviated from this. The analysis error for each state variable is given by $\varepsilon_i^a = x_i^a - x_i^{ref}$. For each realisation, we calculate the root mean square error (RMSE) of the analysis error, which is the difference between the reference state and the estimate obtained by each Case 1-5 of the incremental method,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\varepsilon_i^a)^2}{n}}. \tag{6.53}$$

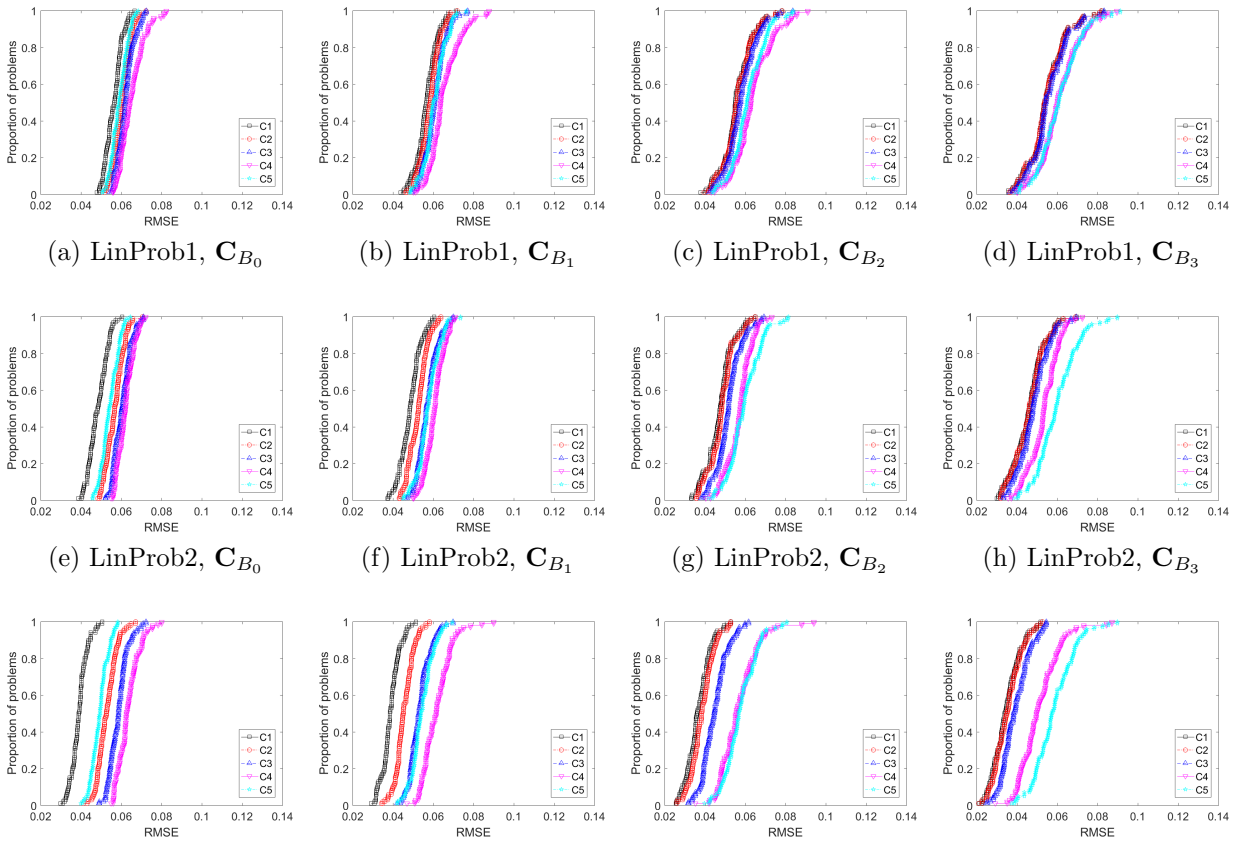
We plot the percentage of problems solved by the incremental method within a specified tolerance of the RMSE (6.53). We acknowledge in this work that the code for the RMSE profiles has been adapted from the code for the data profiles used in [90].

The RMSE profiles for LinProb1, LinProb2 and LinProb3 and NonlinProb1, NonlinProb2 and NonlinProb3 for the case where $\tau_e = 8$ are in Figures 6.5 and 6.6 respectively. These results consider the four different background error correlation structures and inner loop resolution Cases 1-5.

Recall from Section 6.1, Case 5 considers multi-incremental 3D-Var where $g = 1/8$ in the first outer loop iteration, $g = 1/4$ in the second iteration, $g = 1/2$ for the third iteration and $g = 1$ (full resolution) in the fourth and final iteration. We first focus on the RMSE results for Cases 1-4, where the resolution is fixed throughout the iterations, before discussing the results for Case 5.

From both Figures 6.5 and 6.6, we see that as stronger background error correlations are introduced, the difference between the RMSEs of each of Cases 1-3 decreases. For example, for LinProb1, Figure 6.5(d) shows that the lines for Case 1, 2 and 3 almost entirely overlap, unlike in Figure 6.5(a), where there is a clear difference between the performance of Case 1 and the other two cases. This result links to what we saw in Tables 6.1 and 6.2 and the theoretical results in Section 6.2, where we found that the stronger the background error correlations are, the less of an effect the resolution change has on the conditioning of the problem.

For Case 4, we do not see such an improvement. It appears that for any choice of $\mathbf{C}_B$ that we consider, the performance of Case 4 does not improve relative to the higher resolution cases. As explained in our discussion on numerical results of Tables 6.1 and 6.2, the difference between the LHS and the RHS of (6.52) appears to increase as g is reduced and, from our results in Figures 6.5 and 6.6, the effect of this overwrites the positive effect of using stronger background error correlations. Thus, the results from the RMSE profiles show that



(a) LinProb1, $\mathbf{C}_{B_0}$ (b) LinProb1, $\mathbf{C}_{B_1}$ (c) LinProb1, $\mathbf{C}_{B_2}$ (d) LinProb1, $\mathbf{C}_{B_3}$

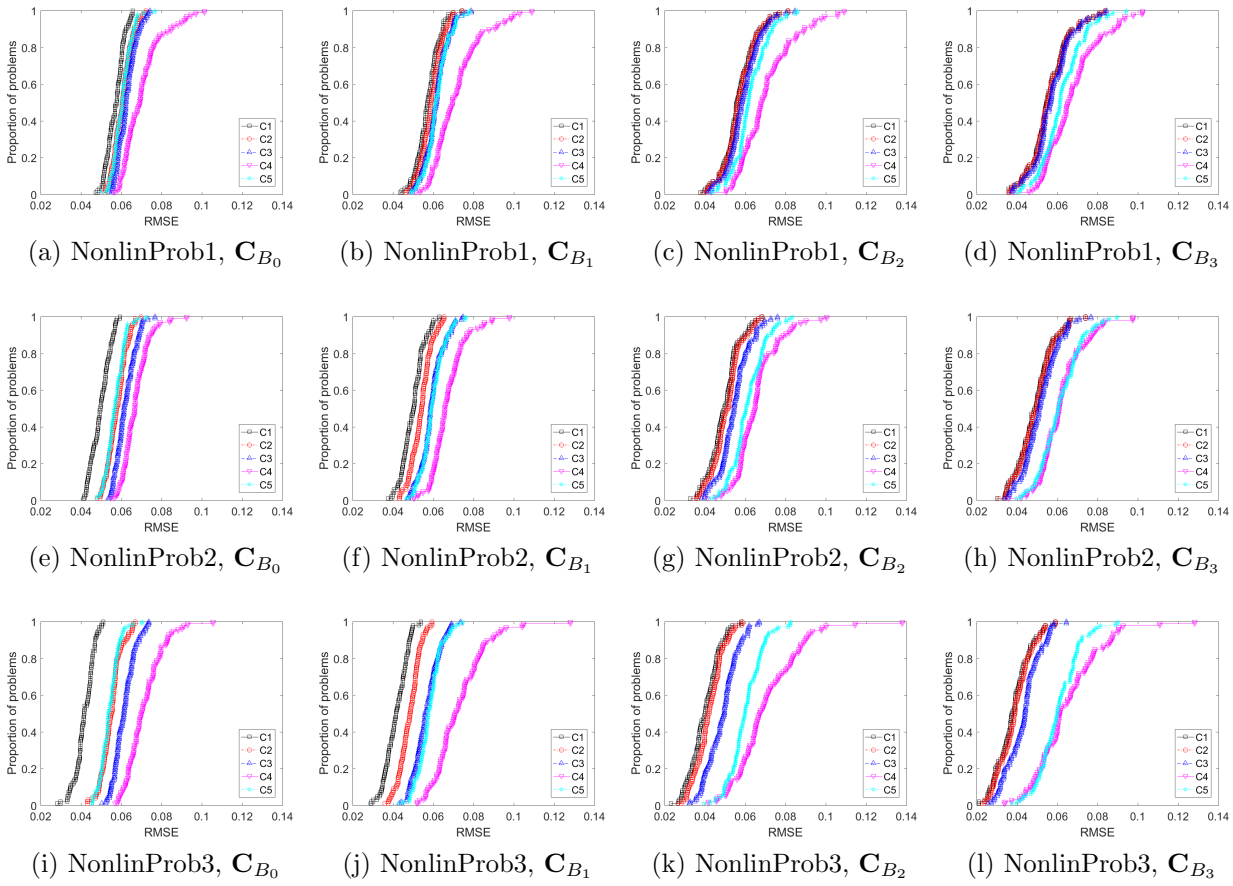
(e) LinProb2, $\mathbf{C}_{B_0}$ (f) LinProb2, $\mathbf{C}_{B_1}$ (g) LinProb2, $\mathbf{C}_{B_2}$ (h) LinProb2, $\mathbf{C}_{B_3}$

(i) LinProb3, $\mathbf{C}_{B_0}$ (j) LinProb3, $\mathbf{C}_{B_1}$ (k) LinProb3, $\mathbf{C}_{B_2}$ (l) LinProb3, $\mathbf{C}_{B_3}$

Figure 6.5: RMSE profiles for the Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for LinProb1 in (a)-(d), LinProb2 in (e)-(h) and LinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

the stronger the background error correlations are, the less of an effect the resolution change has on the convergence of the incremental method so long as g is above a certain threshold.

Our RMSE results for the full resolution case (Case 1) agree with the findings of [73], which showed that in the case that the observation errors are uncorrelated, the use of more observations will always reduce the analysis error. For both the linear and nonlinear problems, the difference between the RMSE of Cases 1-4 appears to increase as more observations are included along the spatial domain. That is, the results for LinProb3 and NonlinProb3, where there are observations at the first $p = 3n/4$ spatial grid-points appear to show that there is a greater difference between the performance of Cases 1-4 compared to the other cases of observation operators, with LinProb1 and NonlinProb1 appearing to show the least difference between the cases. This indicates that the effect of using a reduced resolution on the accuracy of the analysis is more prevalent in the presence of more observations. As discussed in relation to the results of Tables 6.1 and 6.2, this is because when there are more observations along the full resolution grid, more interpolation is performed for the reduced grid, which



(a) NonlinProb1, $\mathbf{C}_{B_0}$ (b) NonlinProb1, $\mathbf{C}_{B_1}$ (c) NonlinProb1, $\mathbf{C}_{B_2}$ (d) NonlinProb1, $\mathbf{C}_{B_3}$

(e) NonlinProb2, $\mathbf{C}_{B_0}$ (f) NonlinProb2, $\mathbf{C}_{B_1}$ (g) NonlinProb2, $\mathbf{C}_{B_2}$ (h) NonlinProb2, $\mathbf{C}_{B_3}$

(i) NonlinProb3, $\mathbf{C}_{B_0}$ (j) NonlinProb3, $\mathbf{C}_{B_1}$ (k) NonlinProb3, $\mathbf{C}_{B_2}$ (l) NonlinProb3, $\mathbf{C}_{B_3}$

Figure 6.6: RMSE profiles for the Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for NonlinProb1 in (a)-(d), NonlinProb2 in (e)-(h) and NonlinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

is an inexact procedure that can produce large errors and more so as the resolution is reduced.

In both Figures 6.5 and 6.6 we see that as stronger background error correlations are introduced, the RMSEs of the Case 5 realisations increase. For LinProb2 and LinProb3, there are even cases where Case 5 is the worst performing case with respect to RMSE of the analysis when using $\mathbf{C}_{B_3}$. As suggested by the plots in Figure 6.5, this may be because the performance of Cases 1-4 differs more so as stronger background error correlations are introduced. Therefore, as the multi-incremental method uses all four of these resolution choices, the accumulation of the differences between their performances as stronger background error correlations are introduced may result in the multi-incremental method performing worse than when a fixed resolution is used across all inner loop minimisations. In such a case, it appears to be preferential to use the lowest resolution case we consider, where $g = 1/8$ throughout all outer loop iterations, as opposed to increasing the spatial resolution at each iteration as in Case 5, thus, saving both computational storage and time.

So far in this chapter, we have seen both theoretically and numerically the effect that the use of a reduced resolution inner loop may have on the conditioning of the inner loop problem. Furthermore, the results from the error profiles suggested that the accuracy of preconditioned 3D-Var analysis is also affected by the level of resolution reduction used in the inner loop. We next focus on studying the effect on the frequencies resolved in the algorithmic output of the incremental method when using a reduced resolution inner loop to solve the preconditioned 3D-Var problem (2.70). We use the error in the power spectra of the DFT output to determine this effect.

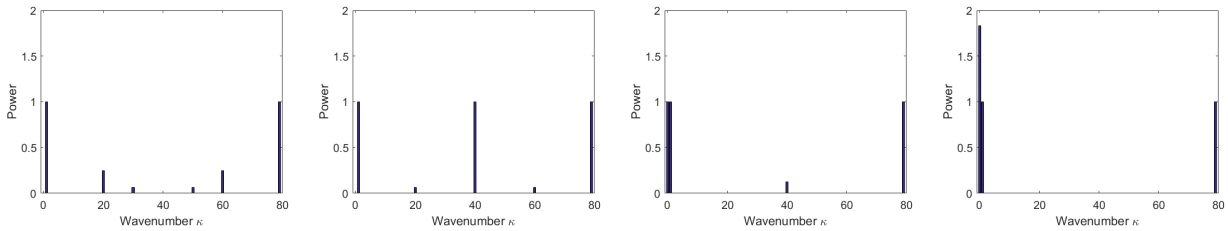
We aim to understand what effect the use of a reduced resolution inner loop has on the frequencies resolved in the analysis of the preconditioned 3D-Var problem by the incremental method at different spatial resolutions. We present our results when applying Algorithm 3.4.1 to linear problems LinProb1, LinProb2 and LinProb3 and nonlinear problems NonlinProb1, NonlinProb2, and NonlinProb3, noting that in the remainder of this section, the reference state vector is instead given by (6.50) so that there are more non-zero frequencies than in (6.47).

To gain an understanding of how the restriction and interpolation matrices defined in Section 6.1 behave at different resolutions, we first consider the power spectra of the DFT of the reduced resolution reference state $\mathbf{S}_{l_g}\mathbf{x}^{ref}$ and the interpolated reduced resolution reference state $\mathbf{S}_{h_g}\mathbf{S}_{l_g}\mathbf{x}^{ref}$. We can use these basic results about our problem to explain the behaviour later in this section, where we consider the effect this has on the wavenumbers recovered after $\tau_e = 8$ evaluations ($k_{\max} = 4$ iterations) of the incremental method by examining the absolute analysis error in the nonzero wavenumbers.

Using power spectra may help us to understand how using a reduced resolution inner loop in the incremental method affects the analysis. In each outer loop iteration, we are restricting the resolution of the increment to be either a half, a quarter or an eighth of the resolution of the state vector, and then using an extension operator to map back to the original resolution using either linear or cubic interpolation at the end of the inner loop minimisation. Therefore, we expect there to be an error in the way the waves are represented. However, it is not clear how much of an effect this may have on which waves are recovered. We can demonstrate the effects of the use of the restriction and extension matrices on the waves resolved of the reference state vector $\mathbf{x}^{ref} \in \mathbb{R}^n$. The n elements of $\mathbf{x}$ are equally spaced along the one-dimensional grid (see Figure 6.1) and are defined by (6.50).

As outlined in Section 2.2, we use the restriction operator defined to be (6.2) to map the 3D-Var increments to the reduced resolution and solve a reduced resolution inner problem. In doing so, we lose some information about the problem as reducing the spatial resolution of a grid results in a limit on the number of full resolution grid frequencies that can be resolved. By using the DFT scaling factor $2/n$ on the reduced resolution DFT output, we are able to compare the amplitude of the powers for the reduced resolution increments to that of the full resolution increment. Figure 6.7 shows the power spectra of the reduced resolution reference state $\mathbf{S}_{l_g}\mathbf{x}^{ref}$ at different spatial resolutions. From this figure, we can see the change in the frequencies resolved as the frequency of samples of the function (6.50)

along the grid is reduced, as expected. Furthermore, the change in the power of wavenumber $\kappa = 0$ (2.45) away from zero in Figures 6.7(c) and 6.7(d) indicates that there has been a shift in the wave along the y-axis of the plot of the reduced resolution reference state $\mathbf{S}_{l_g}\mathbf{x}^{ref}$, for $g = 1/4$ and $g = 1/8$. Therefore, not only has smaller scale information been lost, but the wave is no longer centred around zero.



(a) $g = 1$ (b) $g = 1/2$ (c) $g = 1/4$ (d) $g = 1/8$

Figure 6.7: Power spectra of the DFT of $\mathbf{S}_{l_g}\mathbf{x}^{ref}$, where $\mathbf{S}_{l_g}$ and $\mathbf{x}^{ref}$ are defined in (6.2) and (6.50), respectively, for different choices of g.

As expected, we see that as the inner loop resolution is reduced, the more difficult it is to accurately represent (6.50), thus the number of wavenumbers with non-zero amplitudes correctly resolved in the power spectra reduces. The implication this has on the reduced resolution increments in incremental VarDA is that the lower the resolution, the more difficult it is to recover the larger wavenumbers (smaller scales). This is a known result and in VarDA practice, where we rely on the use of the relatively few nonlinear outer loop iterations to help recover some of the information lost in the reduced resolution inner loop.

We have seen the effect that the use of the restriction operators has on the ability to represent a wave on the reduced resolution grid for our specific problems. By looking at the power spectra after the linear and cubic interpolation matrices have been applied to the reduced resolution reference state $\mathbf{S}_{l_g}\mathbf{x}^{ref}$ at different spatial resolutions, we may be able to understand the effect these extension matrices have on the quality of the update (3.37) and subsequently, the quality of the 3D-Var analysis when using the incremental method with a reduced resolution inner loop.

From the DFT reciprocity relations property (2.47), we know that the low resolution grid is only able to recover the first and last $r/2$ wavenumbers. By interpolating from the low resolution to the full resolution, we are unable to recover the information lost when restricting the resolution. This is the implication of the use of a reduced grid and subsequently interpolating, which causes a loss of information.

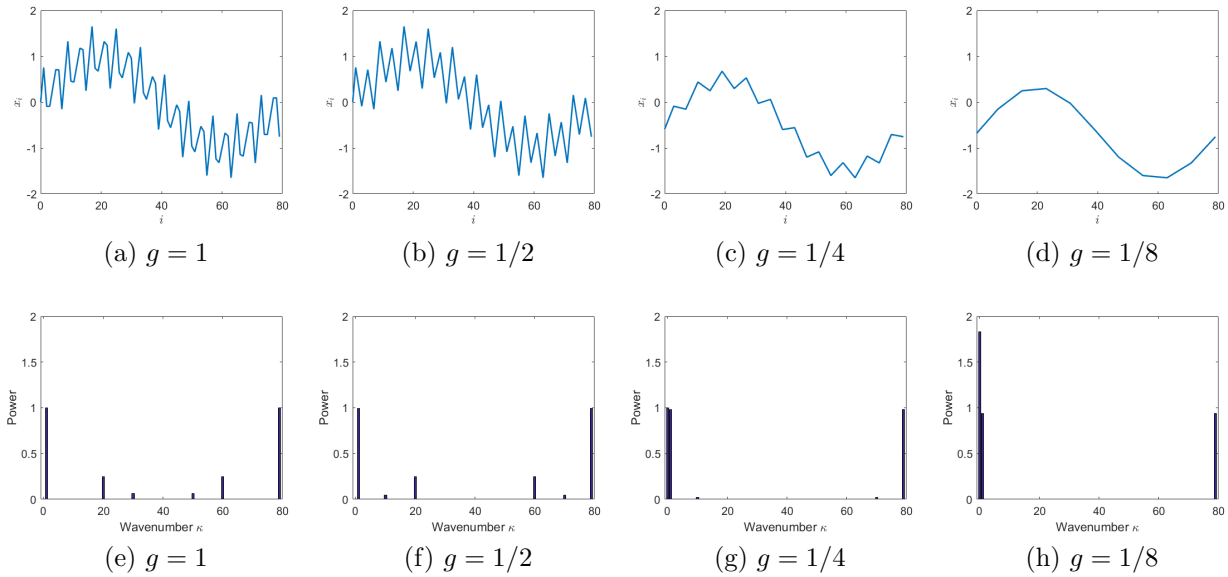
We use the reference state (6.50) to see whether all three waves are resolved after restricting $\mathbf{x}^{ref}$ by applying $\mathbf{S}_l$ and then applying the interpolation operator $\mathbf{S}_h$. Figure 6.8 contains plots of $\mathbf{S}_{h_g}^{lin}\mathbf{S}_l\mathbf{x}^{ref}$ and the power spectra of their DFTs at different spatial resolutions.

Figure 6.8(a) is a plot of the original full resolution discrete sample (6.50). We see that as the resolution is reduced in subsequent plots in Figures 6.8(b), 6.8(c) and 6.8(d), there occurs some form of smoothing due to the averaging that occurs when using linear interpolation. Linear interpolation works with the information from the reduced resolution grid only. When interpolating to the full resolution grid, it is not possible to recover the finer details of the full resolution grid that were not present in the reduced resolution, it is only possible to use the information known at the reduced resolution grid to estimate the values at the full resolution grid-points. When interpolating, the intermediate points between the reduced resolution grid-points are generated using the neighbouring grid-points at the reduced resolution. Therefore, the general shape of the wave at the reduced resolution grid is retained.

The corresponding power spectra in Figures 6.8(e), 6.8(f), 6.8(g) and 6.8(h) show the effect of applying the restriction and extension matrices to $\mathbf{x}^{ref}$ has on resolving the original waves of $\mathbf{x}^{ref}$. The wavenumber with the smallest non-zero power (wavenumber 30) is not resolved when using any of the three linear interpolation matrices. Some power is incorrectly allocated to other wavenumbers in Figures 6.8(f), 6.8(g) and 6.8(h). Although all three of these cases recover wavenumber 1, the case where the eighth resolution linear interpolation matrix is used incorrectly allocates more power to it. This is because, when reducing the number of grid-points to an eighth of the original grid, there may be too little information available to correctly represent the waves, even after interpolation. This is an example of aliasing, as discussed in Section 2.1.3. The high frequency waves are aliased as lower frequency waves as there are not enough grid-points to represent the structure of a wave. The ability to solve the 3D-Var problem in the limited time and computational cost available comes at a cost of aliasing.

From Figures 6.8(g) and 6.8(h), we see that unlike in Figures 6.8(e) and 6.8(f), wavenumber 0 has a non-zero magnitude. This is because of the special case we saw in (2.45). From Figures 6.8(a) and 6.8(b), we see that the mean of the values is 0, corresponding to a magnitude of 0 in the power spectra. From Figures 6.8(c) and 6.8(d), we see that the mean has shifted, resulting in a non-zero magnitude at wavenumber 0 in the power spectra. This is an interesting result as it shows that by reducing the resolution of the grid used in the inner loop of the incremental method, the extension matrices are not able to retrieve the original wave pattern of the full resolution grid, resulting in an incorrect representation of the waves on the power spectra. This result stresses the importance of three elements of incremental VarDA in practice. The first is to avoid this issue happening in the first place by using an appropriate sampling rate. The ECMWF moved from using a linear grid, where 2 grid points are used to model the smallest wavelength, to a cubic grid, where 4 grid points are used in their IFS update cycle 41r2 and have since moved to using a more efficient octahedral grid [81]. The second element is the use of more high resolution outer loop iterations to incorporate more high resolution information and thus improve the analysis. The ECMWF have continued to increase the number of outer loop iterations of the incremental method performed over the years, with the latest IFS update (47r1) using four outer loop iterations. The third and final element is the use of an accurate background state vector such that the initial guess for the minimisation is already at a high accuracy, resulting in less reliance on the inner loop updates. Improvements to the background are made in the latest IFS update

(47r1) through the use of continuous DA, which delays the cut-off period for observations to be included in the assimilation [34].



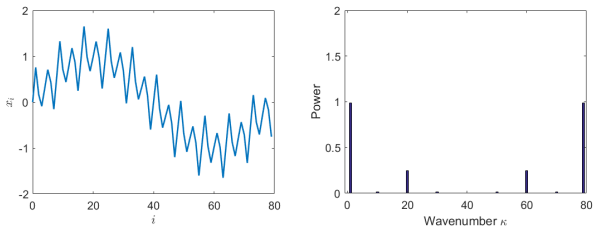
(a) $g = 1$ (b) $g = 1/2$ (c) $g = 1/4$ (d) $g = 1/8$

(e) $g = 1$ (f) $g = 1/2$ (g) $g = 1/4$ (h) $g = 1/8$

Figure 6.8: Plot of $\mathbf{S}_{h_g}^{lin}\mathbf{S}_{l_g}\mathbf{x}^{ref}$ in (a)-(d) and the power spectra of its DFT in (e)-(h), where $\mathbf{S}_{h_g}^{lin}$, $\mathbf{S}_{l_g}$ and $\mathbf{x}^{ref}$ are defined in (6.8), (6.2) and (6.50), respectively, for different choices of g.

Figure 6.9 contains similar plots where the cubic interpolation matrix for the half resolution $\mathbf{S}_{h1/2}^{cub}$ is used. We see that Figure 6.9(a) is closer to Figure 6.8(a) than Figure 6.8(b) was. This result means that the cubic interpolation matrix in this case is better at reducing the error between that original vector (6.50) and the interpolated vector $\mathbf{S}_{h1/2}\mathbf{S}_{l_g}\mathbf{x}^{ref}$ than the linear interpolation matrix was. Furthermore, from Figure 6.9(b), we can see that when cubic interpolation is used, wavenumber 30 is recovered. However, the power is smaller than that of the original wave shown in Figure 6.8(e) and some power is still incorrectly allocated to wavenumber 10. This is an improvement on when linear interpolation was used in Figure 6.8(f) where wavenumber 30 was not recovered and more power was incorrectly attributed to wavenumber 10.

We now ask ourselves, after reducing the resolution, is there enough information (the right kind of sample) to interpolate back to the higher resolution and resolve the small scales? This depends on whether the values kept after reducing the resolution give us enough information about the shape of the waves. We saw in Figures 6.8 and 6.9 that by reducing the resolution and then interpolating back to the higher resolution, we were not able to accurately recover the larger wavenumbers present in Figure 6.8(a). This is because by reducing the resolution in grid-point space using $\mathbf{S}_l$ defined in (6.2), we are essentially taking a less frequent sample. By interpolating back to the higher resolution, we are only using information from this reduced sample and if the reduced sampling rate falls below the Nyquist



(a) $g = 1/2$ (b) $g = 1/2$

Figure 6.9: Plot of $\mathbf{S}_{h_{1/2}}^{cub}\mathbf{S}_{l_{1/2}}\mathbf{x}^{ref}$ in (a) and the power spectrum of its DFT in (b) at different resolutions using cubic interpolation matrices.

rate (2.48), it would not be possible to recover the highest frequencies (largest wavenumbers).

However, in VarDA, we are using high resolution outer loops and high resolution observation information when solving the inner loop. This may mean that during the inner loop minimisation, the reduced resolution update takes into account some small scale (high frequency) information, which is then interpolated to the high resolution fields on the outer loop. In order to test if this is the case, in the remainder of this chapter, we use our understanding from this section so far to study the error in the frequencies of the algorithmic output of the incremental method when applied to the preconditioned 3D-Var problem (2.69).

For each of the linear and nonlinear problems, we apply the incremental method, Algorithm 3.4.1, where the reference state is generated as in (6.50) until the maximum number of outer loop iterations $k_{\max} = 4$ is achieved. Recall from Section 6.3, $\varepsilon_{\mathbf{b}}$ is chosen to be small relative to the average magnitude of the entries of $\mathbf{x}^{ref}$, and as the background state vector is used as the initial guess for Algorithm 3.4.1, we are starting the algorithm close to the reference solution. Therefore, the inner loop updates do not need to be very significant in order to achieve convergence on the outer loop level and we expect the standard deviation of the analysis error to be bounded above by the background and observation error standard deviations. For this reason, we expect all non-zero frequencies to be present in the analysis found using Algorithm 3.4.1 in Cases 1-5 with relatively smaller error in the wave amplitudes of the analysis. However, due to the use of a reduced spatial grid to represent the inner loop problem, we do expect there to be some error between Cases 1-5 as there is a limit on which wavenumbers can be represented by each grid. We use error profiles to quantify these errors, as explained in the following.

From Sections 6.2 and 6.4, we saw how the condition number of the preconditioned 3D-Var Hessian and the accuracy of the analysis are affected not only by the resolution of the inner loop, but also the strength of the background error correlations and the number of observations in space. Therefore, within the results of this section, we also consider the effect of the use of different inner loop resolutions, background correlation structures and increasing the number of observations along the spatial grid to see how these choices affect the quality of the frequencies resolved.

In order to compare the accuracy of the non-zero wavenumbers of the estimate obtained by the incremental method at different inner problem resolutions, for each Case 1-5 and for a given wavenumber, we take the difference between the power of the DFT output of the estimate and the power of the DFT output of $\mathbf{x}^{ref}$. The error in the power of wavenumber κ after $k_{\max}$ iterations is given by

$$\left|\rho_\kappa\Big(\frac{2}{n}DFT(x^{(k_{\max})})\Big) - \rho_\kappa\Big(\frac{2}{n}DFT(x^{ref})\Big)\right|. \tag{6.54}$$

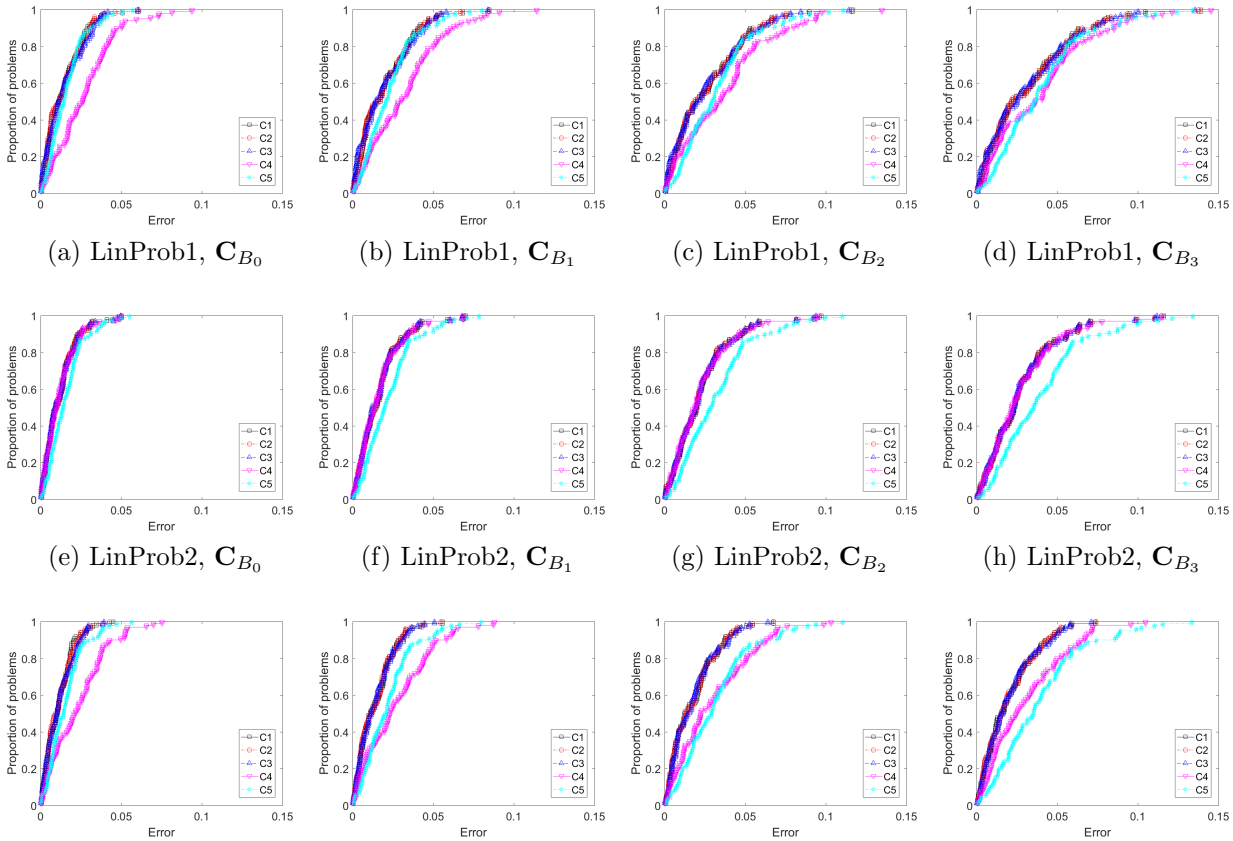
To understand what is happening with the wavenumbers as the resolution is reduced, we plot the error in the power of the DFT of each wavenumber κ at the end of the minimisation, where $k_{\max} = 4$.

We plot the percentage of problems solved by the incremental method within a specified tolerance of the error (6.54). For the linear problems LinProb1, LinProb2 and LinProb3, Figures 6.10, 6.12 and 6.14 show the profiles for the error in the power of the wavenumbers recovered using the incremental method for Cases 1-5 and wavenumbers 1, 20 and 30 respectively. Similar figures for the nonlinear problems NonlinProb1, NonlinProb2 and NonlinProb3 are shown in Figures 6.11, 6.13 and 6.15.

As we set $n = 80$, for Cases 1 and 5 we expect to have updates for the first 40 wavenumbers of $\mathbf{x}^b$, for Case 2 the first 20, for Case 3 the first 10 and for Case 4 the first 5 wavenumbers. We define $\mathbf{x}^{ref}$ using (6.50), so the wavenumbers with a non-zero amplitude for their power are $\kappa = 1$, 20 and 30. Therefore, wavenumber 1 should be recovered to a high level of accuracy by all Cases 1-5, wavenumber 20 should be recovered to a high level of accuracy by Cases 1, 2 and 5 and wavenumber 30 should be recovered to a high level of accuracy by Cases 1 and 5 only.

For wavenumber $\kappa = 1$, the results for the linear and nonlinear problems are given in Figures 6.10 and 6.11 respectively. For the linear problems, Figure 6.10 shows that the error in the power for Cases 1, 2 and 3 are very similar in all choices of observation operator choices. However, for LinProb1, LinProb3, NonlinProb1 and NonlinProb3, the error in Case 4 is considerably worse. From Figure 6.8, we can see why this is the case. When $g = 1/8$ in Figures 6.8(d) and 6.8(h) show a stark difference in numerical values and frequencies resolved between the original full resolution wave in Figures 6.8(a) and 6.8(e). In particular, wavenumber $\kappa = 1$ has a smaller amplitude than in the other wavenumber cases. The structure of the observation operators of LinProb2 and NonlinProb2 appear to allow for an improvement in resolving the first wavenumber in the analysis. This is due to the role of $\hat{\mathbf{H}}$ in (3.37). Having too few observations in Case 4 such as in LinProb1 and NonlinProb1 where $p = n/4$ results in there not being enough high resolution information to correctly represent the problem in the reduced resolution inner loop. Furthermore, having too many observations in Case 4 such as in LinProb3 and NonlinProb3 where $p = 3n/4$ results in more interpolation from the full resolution grid to the reduced resolution grid, resulting in increased errors from interpolation. The choice of $p = n/2$ in LinProb2 and NonlinProb2 appears to create a balance between the amount of high resolution information in the inner loop and the number interpolated increments required, resulting in better convergence of

wavenumber $\kappa = 1$.



(a) LinProb1, $\mathbf{C}_{B_0}$ (b) LinProb1, $\mathbf{C}_{B_1}$ (c) LinProb1, $\mathbf{C}_{B_2}$ (d) LinProb1, $\mathbf{C}_{B_3}$

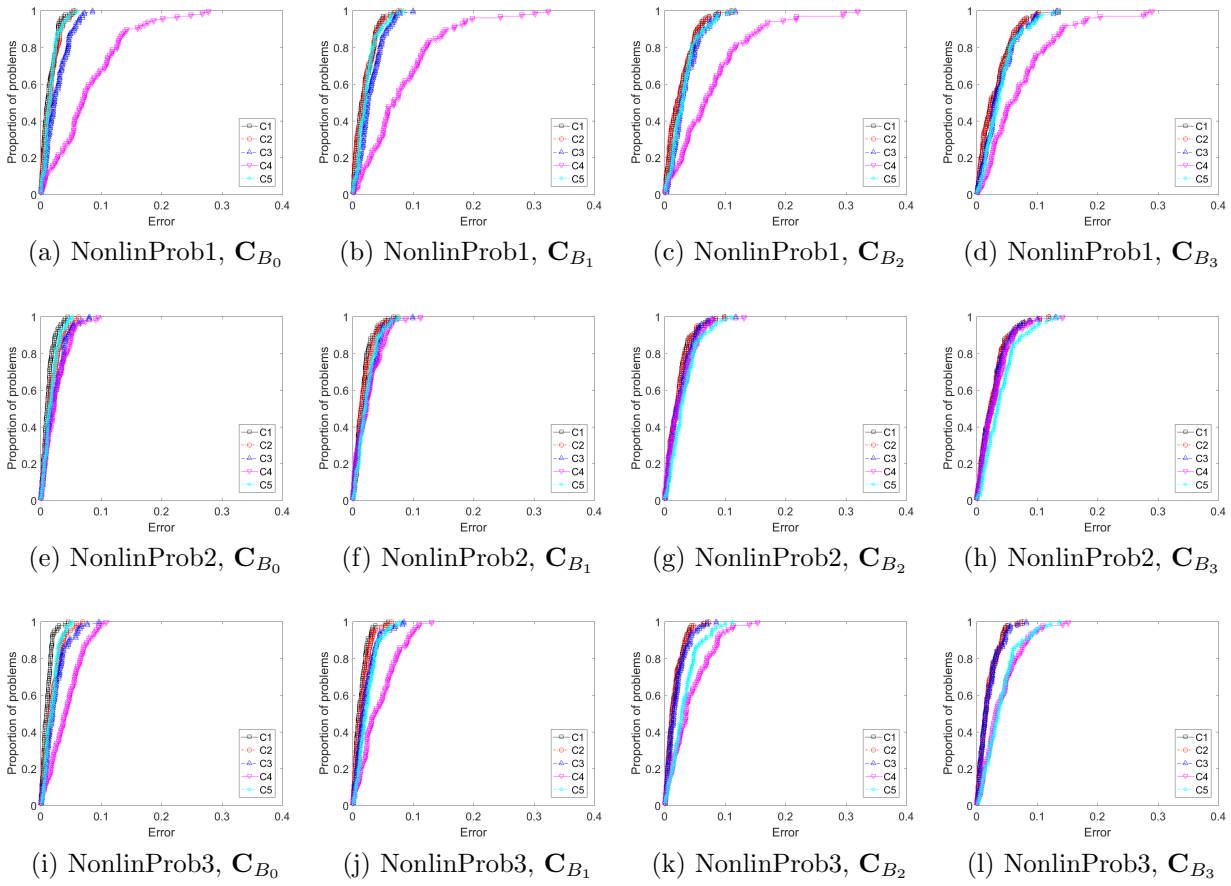
(e) LinProb2, $\mathbf{C}_{B_0}$ (f) LinProb2, $\mathbf{C}_{B_1}$ (g) LinProb2, $\mathbf{C}_{B_2}$ (h) LinProb2, $\mathbf{C}_{B_3}$

(i) LinProb3, $\mathbf{C}_{B_0}$ (j) LinProb3, $\mathbf{C}_{B_1}$ (k) LinProb3, $\mathbf{C}_{B_2}$ (l) LinProb3, $\mathbf{C}_{B_3}$

Figure 6.10: Error profiles of wavenumber $\kappa = 1$ for Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for LinProb1 in (a)-(d), LinProb2 in (e)-(h) and LinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

For wavenumber $\kappa = 20$, the results for the linear and nonlinear problems are given in Figures 6.12 and 6.13 respectively. From these, we see the two behaviours that we saw theoretically in Section 6.2 and numerically in Section 6.4.1 for the condition number of the preconditioned Hessian. That is, as we strengthen the background error correlations, Cases 1-4 behave more similarly. Furthermore, as the number of observations increase along the grid, the greater the difference between Cases 1-4. We see similar results for wavenumber $\kappa = 30$, where the results for the linear and nonlinear problems are given in Figures 6.14 and 6.15 respectively.

Due to the use of an accurate initial guess for the minimisations, we expect that wavenumber $\kappa = 20$ is present in all the analyses of each of Cases 1-5, but only in Cases 1 and 2 do we expect to recover wavenumber $\kappa = 20$ to a high accuracy. We see that for the linear problems in Figure 6.12, Cases 1 and 2 recover wavenumber $\kappa = 20$ with the highest accuracy



(a) NonlinProb1, $\mathbf{C}_{B_0}$ (b) NonlinProb1, $\mathbf{C}_{B_1}$ (c) NonlinProb1, $\mathbf{C}_{B_2}$ (d) NonlinProb1, $\mathbf{C}_{B_3}$

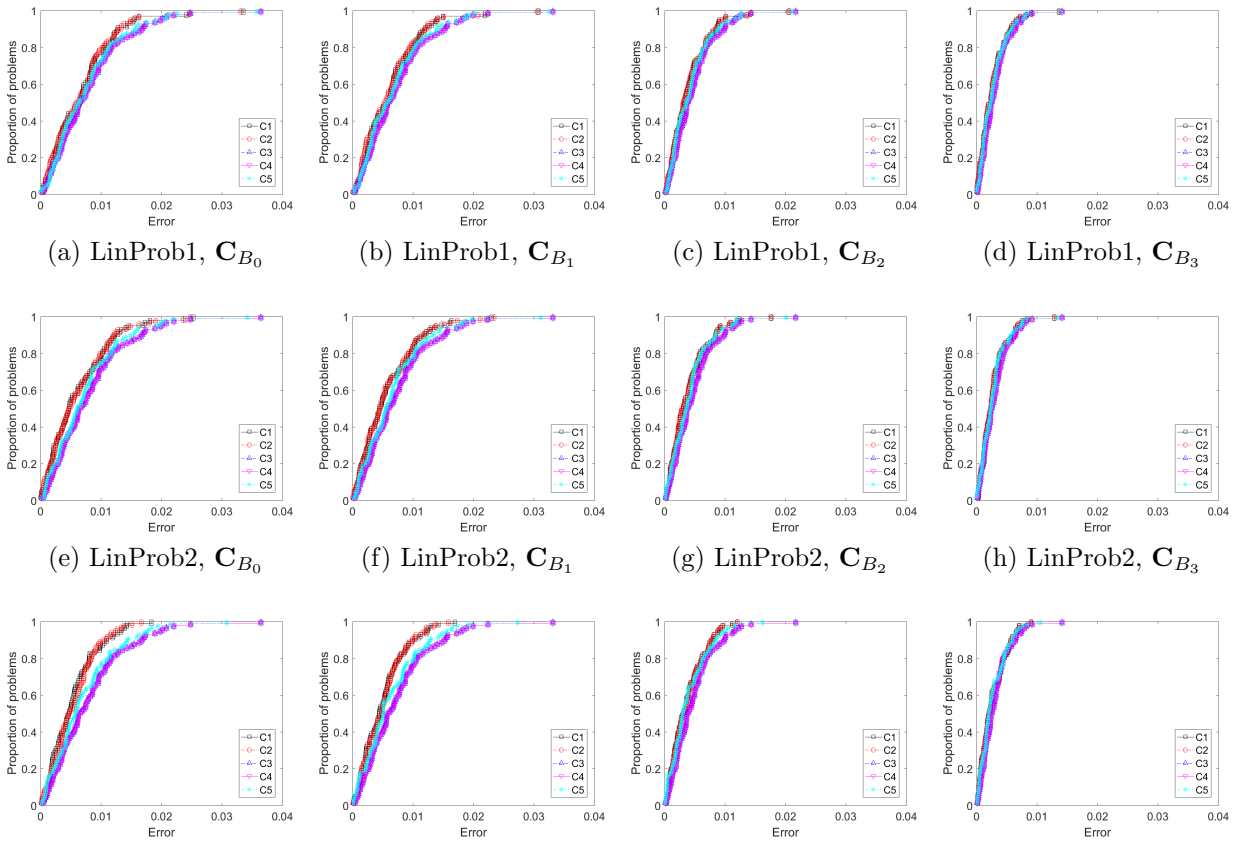
(e) NonlinProb2, $\mathbf{C}_{B_0}$ (f) NonlinProb2, $\mathbf{C}_{B_1}$ (g) NonlinProb2, $\mathbf{C}_{B_2}$ (h) NonlinProb2, $\mathbf{C}_{B_3}$

(i) NonlinProb3, $\mathbf{C}_{B_0}$ (j) NonlinProb3, $\mathbf{C}_{B_1}$ (k) NonlinProb3, $\mathbf{C}_{B_2}$ (l) NonlinProb3, $\mathbf{C}_{B_3}$

Figure 6.11: Error profiles of wavenumber $\kappa = 1$ for Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for NonlinProb1 in (a)-(d), NonlinProb2 in (e)-(h) and NonlinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

and perform almost identically. Cases 3 and 4 also perform almost identically to each other and are the worst cases in terms of accuracy, but are only marginally worse than Cases 1 and 2, especially when stronger background error correlations are included. The overall high accuracy of all 5 cases is attributed to the accurate initial guess used, which results in wavenumber $\kappa = 20$ being present in the background. The results differ for the nonlinear problems in Figure 6.13. In Figures 6.13(e) 6.13(i) and 6.13(j) we see that Case 2 is the worst choice. The updates generated by Case 2 of the incremental method are resulting in worse convergence results on the outer loop level than if a lower resolution case is used. These results show that the level of resolution is not necessarily the dominating factor in the accuracy of the analysis, the nonlinearity of the observation operator also plays a role.

We expect that wavenumber $\kappa = 30$ is present in all the analyses of each of Cases 1-5 as an accurate initial guess for the minimisations is used, but that only Case 1 is able to recover wavenumber $\kappa = 30$ to a high accuracy. This is reflected in the results for both the linear and nonlinear problems in Figures 6.14 and 6.15. We see that Case 1 is the best choice across all



(a) LinProb1, $\mathbf{C}_{B_0}$ (b) LinProb1, $\mathbf{C}_{B_1}$ (c) LinProb1, $\mathbf{C}_{B_2}$ (d) LinProb1, $\mathbf{C}_{B_3}$

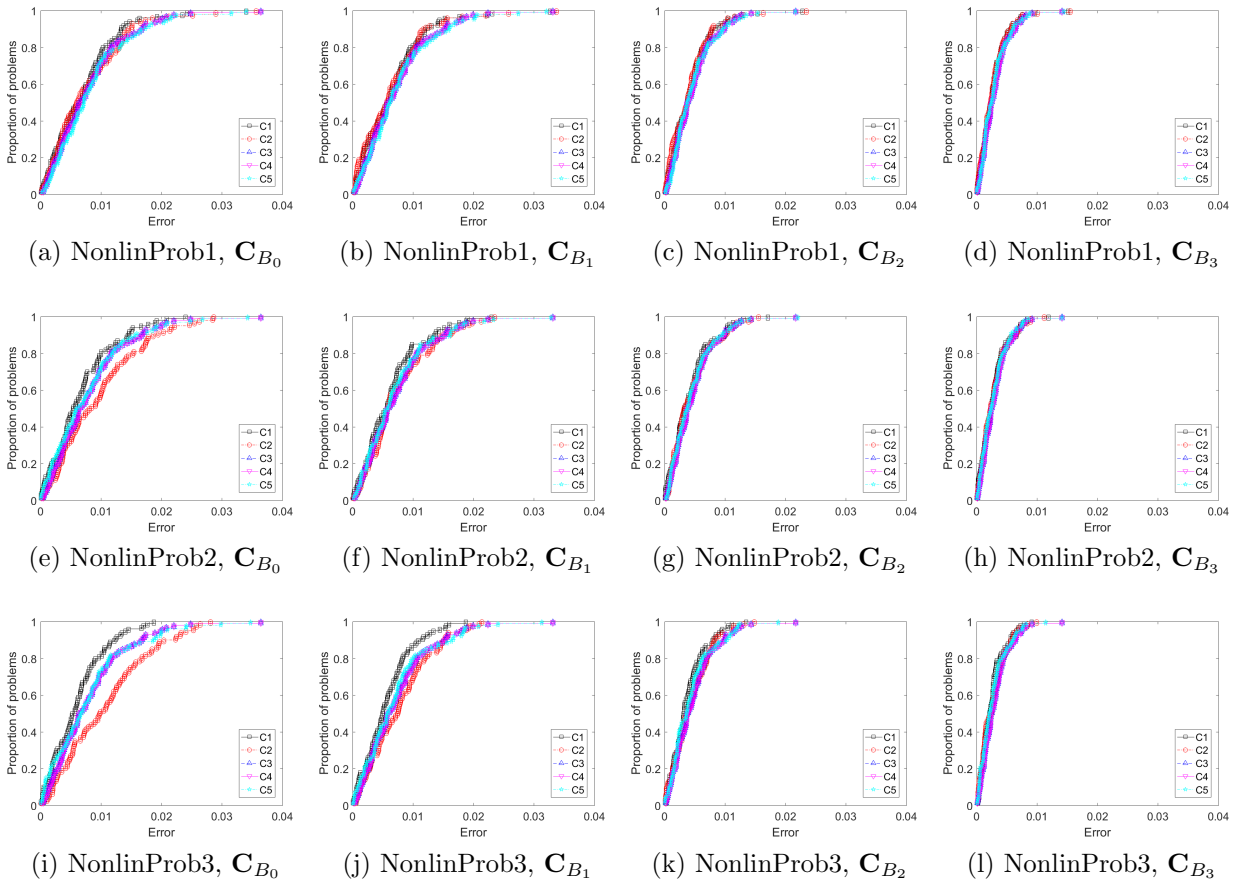
(e) LinProb2, $\mathbf{C}_{B_0}$ (f) LinProb2, $\mathbf{C}_{B_1}$ (g) LinProb2, $\mathbf{C}_{B_2}$ (h) LinProb2, $\mathbf{C}_{B_3}$

(i) LinProb3, $\mathbf{C}_{B_0}$ (j) LinProb3, $\mathbf{C}_{B_1}$ (k) LinProb3, $\mathbf{C}_{B_2}$ (l) LinProb3, $\mathbf{C}_{B_3}$

Figure 6.12: Error profiles of wavenumber $\kappa = 20$ for Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for LinProb1 in (a)-(d), LinProb2 in (e)-(h) and LinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

observation structures and background error structures, with Cases 2-4 performing almost identically in all cases. We notice that Case 5 does particularly well in recovering the small scales despite only using one full resolution outer loop iteration out of the four outer loop iterations. This aligns with the findings of ECMWF when using multi-incremental 4D-Var - that the increased resolution of the analysis increments enables updates at the smaller scales [58].

Regarding the effect of the strength of the background error correlations on the convergence of the incremental method, in Figures 6.10-6.15 we see that as stronger background error correlations are introduced, the difference between the error in the power for Cases 1-4 reduces. This is similar to what we saw theoretically in Section 6.2 and numerically in Section 6.4.1 for the condition number of the preconditioned Hessian. By increasing the correlation length scale, we are increasing the strength of the background error correlations and thus the entries of the background error covariance matrix. Recall from Section 6.3 that the background (initial guess for the minimisation) is generated by adding noise (6.48) to



(a) NonlinProb1, $\mathbf{C}_{B_0}$ (b) NonlinProb1, $\mathbf{C}_{B_1}$ (c) NonlinProb1, $\mathbf{C}_{B_2}$ (d) NonlinProb1, $\mathbf{C}_{B_3}$

(e) NonlinProb2, $\mathbf{C}_{B_0}$ (f) NonlinProb2, $\mathbf{C}_{B_1}$ (g) NonlinProb2, $\mathbf{C}_{B_2}$ (h) NonlinProb2, $\mathbf{C}_{B_3}$

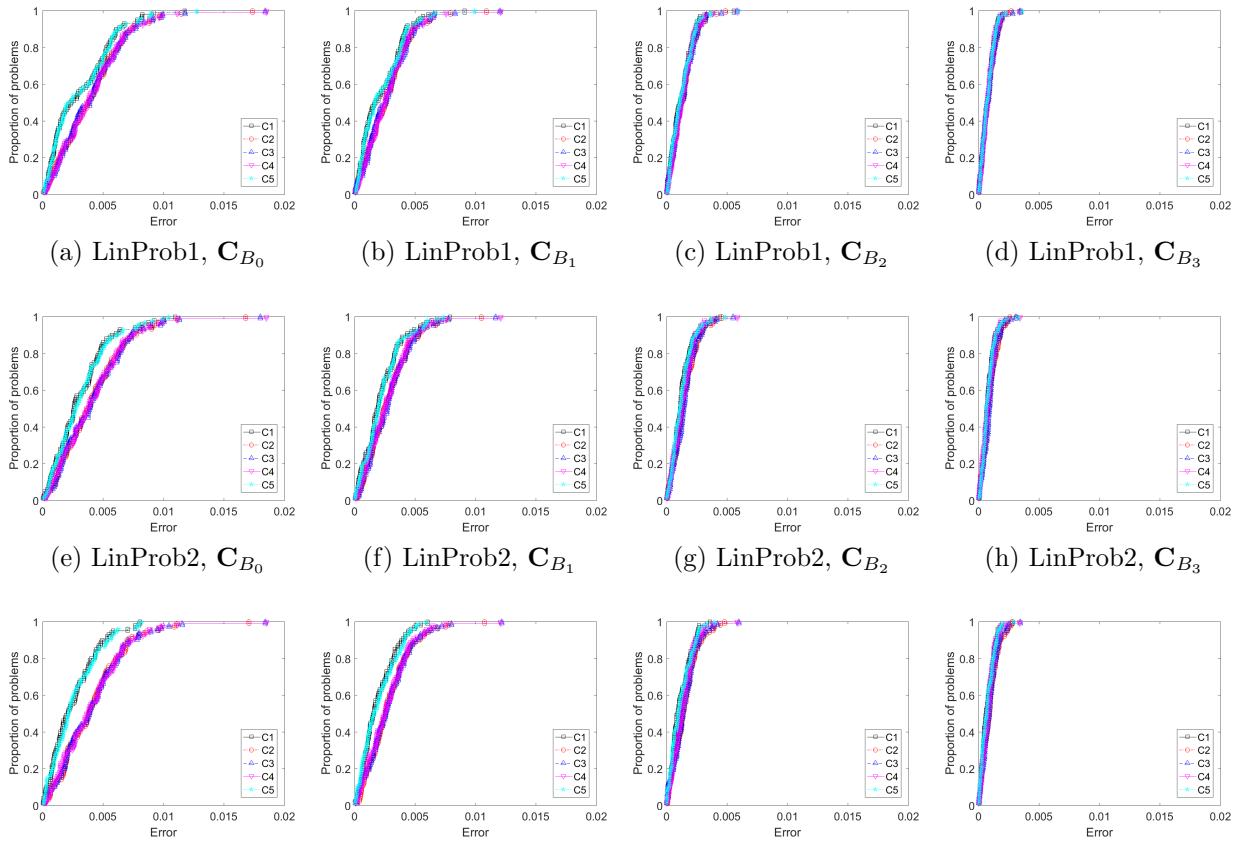
(i) NonlinProb3, $\mathbf{C}_{B_0}$ (j) NonlinProb3, $\mathbf{C}_{B_1}$ (k) NonlinProb3, $\mathbf{C}_{B_2}$ (l) NonlinProb3, $\mathbf{C}_{B_3}$

Figure 6.13: Error profiles of wavenumber $\kappa = 20$ for Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for NonlinProb1 in (a)-(d), NonlinProb2 in (e)-(h) and NonlinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

the reference state as follows

$$\mathbf{x}^b = \mathbf{x}^{ref} + \sigma_b^2 \mathbf{C}_B^{1/2} \varepsilon_b. \tag{6.55}$$

Now, increasing more of the off diagonal entries in $\mathbf{C}_B$ results in more off diagonal entries of $\mathbf{C}_B^{1/2}$, resulting in the term $\sigma_b^2 \mathbf{C}_B^{1/2} \varepsilon_b$ in (6.55) having larger entries. Considering what this means in spectral space, the length scale of the background error correlation matrix controls the spread of the background errors along the grid and subsequently, the wavenumbers. This can be visualised in Figure 6.16, which shows the power spectra of the background errors of a typical realisation for each case of $\mathbf{C}_B$ that we consider. From this figure, we see that increasing the strength of the background error correlations results in a shift from having amplitudes in the powers spread across all wavenumbers, to having considerably larger amplitudes at the smaller wavenumbers. The sensitivity of the scales of the analysis to observations has been noted by both [30] and [42]. When the correlation length scale for the observation error correlations is smaller (larger) than that of the background error correlations, the analysis will be less (more) sensitive to observations at the small scales compared



(a) LinProb1, $\mathbf{C}_{B_0}$ (b) LinProb1, $\mathbf{C}_{B_1}$ (c) LinProb1, $\mathbf{C}_{B_2}$ (d) LinProb1, $\mathbf{C}_{B_3}$

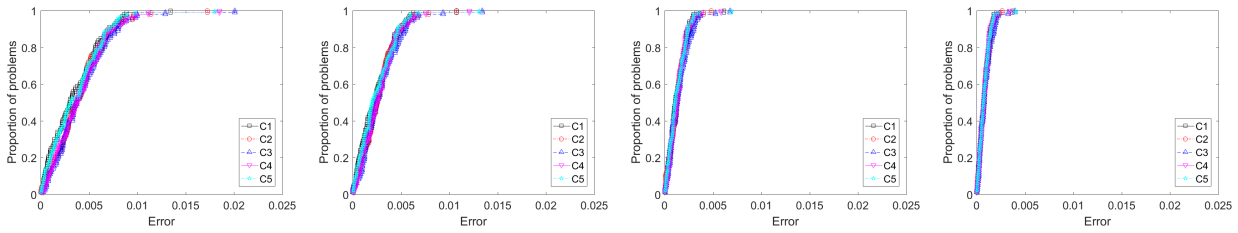
(e) LinProb2, $\mathbf{C}_{B_0}$ (f) LinProb2, $\mathbf{C}_{B_1}$ (g) LinProb2, $\mathbf{C}_{B_2}$ (h) LinProb2, $\mathbf{C}_{B_3}$

(i) LinProb3, $\mathbf{C}_{B_0}$ (j) LinProb3, $\mathbf{C}_{B_1}$ (k) LinProb3, $\mathbf{C}_{B_2}$ (l) LinProb3, $\mathbf{C}_{B_3}$

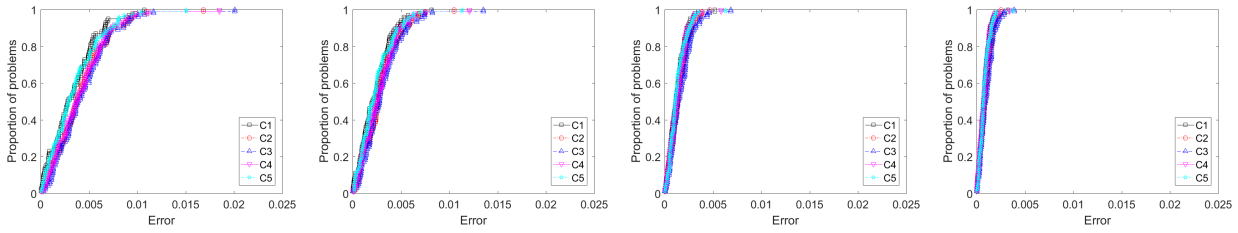
Figure 6.14: Error profiles of wavenumber $\kappa = 30$ for Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for LinProb1 in (a)-(d), LinProb2 in (e)-(h) and LinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.

to the large scales and when the length scales are equal, the sensitivity is fixed across all scales. In our work, we consider the use of uncorrelated observation errors only, where the background error correlation length scale is varied. Our findings coincide exactly with the findings of [42] in the case of positive monotonic decreasing correlation functions: increasing the correlation length scale results in an increase in uncertainty for the large scales of the analysis and a decrease in uncertainty for the small scales.

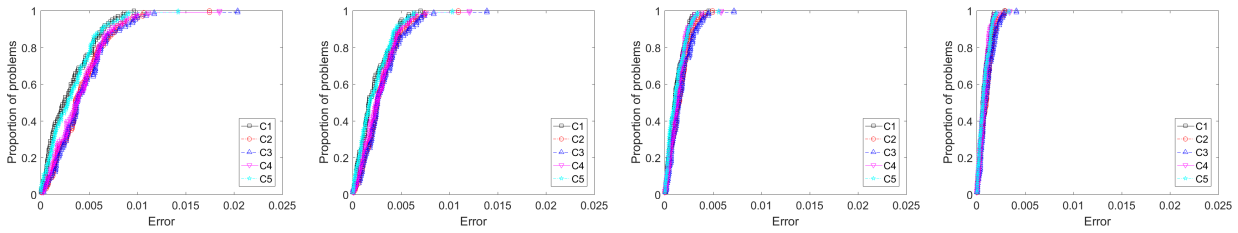
In our results, we expect all Cases 1-5 are able to update the smallest wavenumber $\kappa = 1$. But it is the larger wavenumbers $\kappa = 20$ and $\kappa = 30$ that we expect some of the cases not to update. Therefore, when there are weaker background error correlations, the background errors are distributed across all wavenumbers, so the lower resolution cases will be unable to improve the accuracy of the smaller wavenumbers. When the background error correlations are increased, the errors are focused more so on the smaller wavenumbers, which all the resolutions we consider are able to update, and the larger wavenumbers are less erroneous. Therefore, the performance of all the cases is expected to improve.



(a) NonlinProb1, $\mathbf{C}_{B_0}$ (b) NonlinProb1, $\mathbf{C}_{B_1}$ (c) NonlinProb1, $\mathbf{C}_{B_2}$ (d) NonlinProb1, $\mathbf{C}_{B_3}$

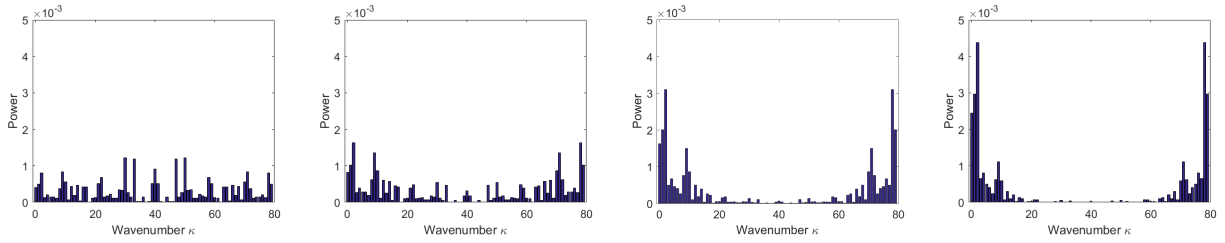


(e) NonlinProb2, $\mathbf{C}_{B_0}$ (f) NonlinProb2, $\mathbf{C}_{B_1}$ (g) NonlinProb2, $\mathbf{C}_{B_2}$ (h) NonlinProb2, $\mathbf{C}_{B_3}$



(i) NonlinProb3, $\mathbf{C}_{B_0}$ (j) NonlinProb3, $\mathbf{C}_{B_1}$ (k) NonlinProb3, $\mathbf{C}_{B_2}$ (l) NonlinProb3, $\mathbf{C}_{B_3}$

Figure 6.15: Error profiles of wavenumber $\kappa = 30$ for Case 1 (black), Case 2 (red), Case 3 (blue) and Case 4 (magenta) of the incremental method for NonlinProb1 in (a)-(d), NonlinProb2 in (e)-(h) and NonlinProb3 in (i)-(l), where $n_r = 100$ for different choices of $\mathbf{C}_B$ indicated in the plot captions. The observation error is 5% and the background error is 10%.



(a) $\mathbf{C}_{B_0}$ (b) $\mathbf{C}_{B_1}$ (c) $\mathbf{C}_{B_2}$ (d) $\mathbf{C}_{B_3}$

Figure 6.16: Power spectra of the DFT of $\sigma_b^2 \mathbf{C}_B^{1/2} \varepsilon_b$ in (6.55) for different choices of $\mathbf{C}_B$ as indicated by the plot captions.

We conclude from these results that a reduction in the inner loop resolution using the linear interpolation extension matrices negatively affects the convergence of the outer loop in the

3D-Var incremental method, both in terms of the numerical accuracy of the analysis and the accuracy of the frequencies of the analysis resolved. However, we know that using a reduced resolution inner loop has a clear computational time and cost benefit, especially considering the dimension of the variational problem in practice (of order $10^8 - 10^9$). Therefore, it is important to consider this saving in practical implementations to achieve a balance.

6.5 Conclusion

Within this chapter, we address research question RQ2 where we study the effect that the use of a reduced resolution inner loop has on the convergence of the incremental method.

We begin by outlining the reduced resolution framework in Section 6.1 where we define a simple restriction operator to map to the reduced grids we consider. We then show that the pseudo-inverse of the restriction matrix is not a good choice and introduce the linear and cubic interpolation extension matrices used to map to the original grid from the reduced resolution grids.

To understand the effect of using a linear and cubic interpolation matrix as the extension operator on different components of the incremental method and its convergence (addressing RQ2(b)), in Section 6.2 we analyse the structure of the interpolation matrices and show that we can exploit their structure in order to bound various quantities in the incremental method. We prove that the 2-norm of the linear and cubic interpolation matrices are equal to the square root of the inverse of the resolution parameter g and discuss how this result applies to higher orders of interpolation. We also derive a theoretical bound on the norm of the square-root of the background error correlation matrix (used in the preconditioner) in terms of g and use this, along with the bound on the 2-norm of the interpolation matrix to derive an upper bound on the condition number of the preconditioned 3D-Var Hessian for different inner loop resolutions. In doing so, we are also addressing RQ2(c). We use the latter bound to understand the effect that the use of reduced resolution operators have on the accuracy to which we could be able to solve the inner loop problem in practice. We find that this bound indicates that as the inner loop resolution is reduced, the bound on the condition number of the preconditioned 3D-Var Hessian increases. Furthermore, we find the use of the bound on the norm of the square-root of the background error correlation matrix suggests that as the correlation length scale increases, the influence of the resolution parameter g in the bound on the condition number (6.40) decreases. These findings are supported by our numerical experiments in Section 6.4.1 where we use the incremental method to solve the preconditioned 3D-Var problem using different inner loop resolutions. We find that the effect of the resolution change does not have as great an impact on the conditioning of the inner loop problem as the presence of background error correlations has.

To understand how the reduced resolution algorithmic output compares to the full resolution algorithmic output (addressing RQ2(d)), in Section 6.4.2 we consider the analysis error, which is of interest to the DA community. We use assimilation experiments using six preconditioned 3D-Var problems with linear and nonlinear observation operators to compare

the analysis error generated by the incremental method using different inner loop resolutions and within a limited number of iterations, similar to what is used in practice. We find that the accuracy with which the preconditioned 3D-Var problem is solved is more so affected by the level of resolution in the presence of more observations along the spatial domain. Furthermore, we find that the stronger the background error correlations are, the less of an effect the resolution change has on the convergence of the incremental method. This latter finding coincided with both our theoretical and experimental work on the condition number in Sections 6.2 and 6.4.1 respectively. The implication that these results have on operational VarDA settings is that they highlight the importance of correctly prescribing background errors, especially when using a reduced resolution inner loop problem, as we have found that the strength of background error correlations has a greater impact on the conditioning of the inner loop than the effect of the use of a reduced resolution inner problem.

To understand how the use of restriction/extension matrices affects the convergence of the wavenumbers in the analysis (addressing RQ2(e)), we use the power spectra of the discrete Fourier transform of the 3D-Var analysis and analyse the error in the amplitudes of the non-zero wavenumbers of the reference state for different resolution choices. We find that the influence of the nonlinear observation operator and the background error covariance matrix, again, play a key role in the effect that the use of a reduced resolution inner loop has on the accuracy of the frequencies resolved in the analysis and that the level of resolution is not necessarily the dominating factor in the accuracy of the analysis. In particular, we find that the weaker the background error correlations are, the stronger the errors across the higher frequencies are, thus the more difficult it is for the reduced resolution incremental method to accurately resolve the higher frequencies of the reference state. Furthermore, we identified cases where the nonlinearity of the observation operator had a negative effect on the accuracy of the wavenumbers resolved by the reduced resolution incremental method.

As mentioned in Section 6.4.2, our RMSE results for the full resolution case agree with the findings of [73], which showed that in the case that the observation errors are uncorrelated, the use of more observations will always improve the analysis error. However, the authors show that this is not necessarily the case if correlated observation errors are included. The work of [42] studied the relationship between the use of background and observation error statistics and the observation operator, and showed that spatially dense observations can be made the most out of if they are able to resolve more small scale information than the background. This is an extension to the work of [73], [108] and [118] that focused on the effect of observation error correlations on the scales of the analysis alone, and not on the interaction with the background error statistics and the observation operator. In particular, [42] found that if both the background and observation error correlations are modelled using the SOAR matrix and the state variables are directly observed, the use of dense observations is most beneficial when the observation error correlation's length scale is larger than that of the background error correlation's. Within our work, we only consider the use of uncorrelated observation error covariance matrices. The first operational implementation of horizontal correlation observation errors in a DA system for NWP was documented in [116]. This work used the Met Office system to show that within the computational limits of DA in NWP, the introduction of the correlated observation error allows the assimilation to make

better use of more observations. Thus, repeating our experiments where observation error correlations are accounted for may be of interest in future work.

Our main result of this chapter, the bound on the condition number of the reduced resolution preconditioned 3D-Var Hessian, gives the DA community an understanding of how the inner loop interacts with the background error covariance matrix and the observation operator and influences the convergence of an inner loop method. In our work, we solve the inner loop exactly. In future work, this bound can be used to give an indication of how an iterative inner loop method (such as CG) may perform given a level of resolution.

In the following chapter, we discuss the main conclusions of this thesis.

Chapter 7

Conclusion

Within this thesis, we aimed to address the following research questions.

RQ1. **Is the use of globally convergent strategies within GN beneficial in variational data assimilation?**

(a) How can the GN method benefit from the use of globally convergent strategies?

(b) How do the globally convergent method parameters interact with the variational problem?

(c) How do GN, LS and REG behave if the initial guess of the minimisation (the background) is highly inaccurate compared to the observations?

(d) In what situations are the globally convergent methods a better option then GN in the presence of a long assimilation time-window?

RQ2. **What is the effect of using a reduced resolution inner loop on the convergence of the incremental method?**

(a) How does the level of resolution reduction affect the convergence of the incremental method?

(b) What is the effect of using a linear and cubic interpolation matrix as the extension operator on the different components of the incremental method?

(c) What effect does the use of reduced resolution operators have on the accuracy to which we could be able to solve the inner loop problem in practice?

(d) How does the reduced resolution algorithmic output compare to the full resolution algorithmic output?

(e) How does the use of restriction/extension matrices affect the convergence of the wavenumbers in the analysis?

In Chapters 2 and 3, we outlined the background behind our research questions. More specifically, in Chapter 2, we introduced the mathematical preliminaries used throughout this thesis. We then outlined both the standard and incremental formulations of the VarDA method. We introduced the preconditioned formulation using the square-root of the background error covariance matrix; a preconditioner commonly used in practice for the VarDA inner loop. Finally, we outlined the numerical models used in the 4D-Var experimental work of this thesis.

In Chapter 3, we outlined the theory of nonlinear least-squares problems and outlined and discussed three fundamental unconstrained optimisation methods for nonlinear least-squares problems, namely, Steepest Descent, Newton's method and Gauss-Newton (GN). We outlined two local convergence results for the GN method before discussing two methods that use safeguards within GN to make GN globally convergent, namely, Gauss-Newton with backtracking Armijo line search (LS) and Gauss-Newton with quadratic regularisation (REG). We then reviewed relevant optimisation methods that have been applied to the variational problem and discussed what is sought from an optimisation method in VarDA. Finally, we outlined various stopping criteria used to terminate optimisation methods and discussed how to compare the performance of optimisation methods.

In the following section we consider how our research questions have been addressed within this thesis. We then highlight our main contributions in Section 7.2. Finally, in Section 7.3 we reflect on our research findings and discuss some ideas for future research.

7.1 Conclusion of Research

In this section, we outline the contributions of our research.

In Chapter 4, we address research questions RQ1(a), RQ1(b) and part of RQ1(c).

- (Addressing RQ1(a)) In Sections 4.2 and 4.3, we outline and prove the global convergence theorems of LS and REG when applied to a general nonlinear least-squares problem. We discuss how the assumptions made in these convergence proofs are satisfied in DA in Section 4.1, concluding that LS and REG are theoretically practical methods for application to VarDA.

- (Addressing RQ1(b)) In Section 4.5, we use the variational problem to propose a way to choose an initial REG parameter to speed up convergence of the REG method. We propose that the standard choice of initial REG parameter of 1 is suitable for the preconditioned VarDA problem, but the standard VarDA problem benefits from choosing the initial REG parameter based on the background error covariance matrix. Using numerical experiments on nonlinear 3D-Var test problems, we show that for both uncorrelated and correlated background error matrices, the convergence of the REG method is improved if the initial REG parameter is chosen according to the background error covariance matrix and the method appears to be better than GN, LS and REG

for solving the standard 3D-Var problem, both in terms of minimising the 3D-Var cost function and in terms of the accuracy of the analysis.

- (Addressing RQ1(c)) In Section 4.7, we use numerical experiments where we apply GN and the three globally convergent methods to nonlinear 3D-Var problems to show that in the limited computational cost available in NWP and when there is more uncertainty in the background information (initial guess for the minimisation) compared to the observations, the GN method fails to obtain an estimate of the initial state that is as accurate as the globally convergent methods' estimates.

In Chapter 5, we address research questions RQ1(c) and RQ1(d).

- (Addressing RQ1(c)) Using two test models within the preconditioned 4D-Var framework, we show that when there is more uncertainty in the background information compared to the observations, the GN method may diverge in the long time-window case, yet the globally convergent methods, LS and REG, are able to improve the estimate of the analysis. We consider the case where the background information is highly inaccurate compared to the observations in the 4D-Var framework and find that the convergence of all three methods is improved when more observations are included along the time-window.

- (Addressing RQ1(d)) We study the effect of the assimilation time-window length on the convergence of GN, LS and REG. We find that in the short time-window case, there is no benefit in using a globally convergent method as GN is able to solve the majority of problems, even for larger values of the ratio of the background and observation error standard deviations. In the long time-window case, when the ratio of the background and observation error standard deviations is small, we find that GN is able to solve the majority of problems; it is able to obtain the same solution as the LS and REG method and does not require the use of parameter updating strategies. We use accuracy profiles to show numerically that in the long time-window case and when there is higher uncertainty in the background information versus the observations, the globally convergent methods are able to solve more problems than GN in the limited cost available.

In Chapter 6, we address research question RQ2.

- (Addressing RQ2(a)) In Section 6.2, we consider the use of linear and cubic interpolation extension matrices at different resolutions and use these matrices to derive a theoretical bound on the condition number of 3D-Var Hessian. This bound suggests that as the correlation length scale increases, the influence of the resolution parameter g in the bound on the condition number decreases. Our numerical experiments in Section 6.4.1 showed that this relationship between g and the background error covariance matrix existed in the actual condition number of the 3D-Var Hessian. We found that the effect of the resolution change does not have as great an impact on the conditioning of the inner loop problem as the presence of background error correlations has.

- (Addressing RQ2(b)) In Section 6.2, we analyse the structure of the interpolation matrices and show that we can exploit their structure in order to bound various quantities in the incremental method. We prove that the 2-norm of the linear and cubic interpolation matrices are equal to the square root of the inverse of the resolution parameter g and discuss how this result applies to higher orders of interpolation. We also derive a theoretical bound on the norm of the square-root of the background error correlation matrix (used in the preconditioner) in terms of g and use this, along with the bound on the 2-norm of the interpolation matrix to derive an upper bound on the condition number of the preconditioned 3D-Var Hessian for different inner loop resolutions. We find that the latter bound indicates that as the inner loop resolution is reduced, the bound on the condition number of the preconditioned 3D-Var Hessian increases. Furthermore, we find the use of the bound on the norm of the square-root of the background error correlation matrix suggests that as the correlation length scale increases, the influence of the resolution parameter in the bound on the condition number decreases. These findings are supported by our numerical experiments where we find that the effect of the resolution change does not have as great an impact on the conditioning of the inner loop problem as the presence of background error correlations has.

- (Addressing RQ2(c)) As discussed in relation to RQ2(a), in Section 6.2, we derive a theoretical bound on the condition number of 3D-Var Hessian. From our assimilation experiments in Section 6.4.2, we find that the accuracy with which the preconditioned 3D-Var problem is solved is more so affected by the level of resolution in the presence of more observations along the spatial domain. We also find that as stronger background error correlations are introduced, the difference between the analysis errors of each resolution Case 1-4 decreases. This latter finding coincided with both our theoretical and experimental work on the condition number in Sections 6.2 and 6.4.1 respectively.

- (Addressing RQ2(d)) In Section 6.4.2, we use assimilation experiments to compare the analysis error generated by the incremental method using different inner loop resolutions and within a limited number of iterations, similar to what is used in practice. We focus on the convergence of the outer loop of the preconditioned incremental 3D-Var problem and consider when the inner loop is solved directly. We conduct experiments using six 3D-Var problems with linear and nonlinear observation operators and compare the error in the analysis when using a full, half, quarter and eighth resolution inner loop problem. We also consider the use of both (SOAR) correlated background error, as in practice, and uncorrelated background error. We produce error profiles to represent our results and find that the stronger the background error correlations are, the less of an effect the resolution change has on the convergence of the incremental method. We find that the effect of reducing the resolution of the inner loop problem on the accuracy of the analysis is more prevalent in the presence of more observations.

- (Addressing RQ2(e)) In Section 6.4.2, we use the power spectra of the discrete Fourier transform of the 3D-Var analysis to understand how accurately the non-zero wavenumbers of the reference state can be resolved. We generate error profiles for the error in the amplitudes of the non-zero wavenumbers for different resolution choices and find that, as we saw in our results on the condition number of the Hessian, the level of

resolution is not necessarily the dominating factor in the accuracy of the analysis. In particular, we find that the weaker the background error correlations are, the stronger the errors across the higher frequencies are, thus the more difficult it is for the reduced resolution incremental method to accurately resolve the higher frequencies of the reference state. Furthermore, we identified cases where the nonlinearity of the observation operator had a negative effect on the accuracy of the wavenumbers resolved by the reduced resolution incremental method.

In the following section, we highlight the main contributions of our research.

7.2 Main contributions

- We show how the convergence of the method used operationally in VarDA (GN) can be improved through the use of globally convergent strategies while considering the limited time and cost available in DA. We prove global convergence of these methods (LS and REG) and show that they are able to improve the current estimate of the DA analysis even if the initial estimate of the solution is poor and a long assimilation time-window is used.

- We show ways that the convergence of REG can be improved by choosing the initial REG parameter according to components of the VarDA problem. We find that choosing the initial REG parameter according to the background error covariance matrix results in REG locating a more accurate DA analysis than that obtained by GN, LS or the standard REG method.

- We derive an upper bound on the condition number of the preconditioned 3D-Var Hessian that accounts for different inner loop resolutions of the incremental method. This bound provides an insight into how the level of resolution interacts with the correlated background errors and the observation operator to influence the convergence of the incremental method.

We next reflect on our research findings and suggest future research areas.

7.3 Reflections and future research

Our findings for the first major research question RQ1 provide an insight into the performance of globally convergent methods (LS and REG) when applied to both 3D-Var and 4D-Var problems under the computational constraints available in NWP practice. We consider characteristics of the variational problems that occur in practice, such as the use of a long assimilation time-window, inaccurate background information and nonlinear observation operators. We show cases where the method used operationally (GN) fails to solve the variational problem as accurately as the globally convergent methods. To improve the performance of the globally convergent methods, we consider alternative choices of their globalisation parameters. We focus on the REG parameter and derive a way of choosing an initial REG parameter for the variational problem that solves the variational problem to a

higher accuracy compared to GN, LS and the standard REG in the limited computational cost available in NWP.

We have seen how the initial choice of the regularisation parameter impacts the solution obtained by the REG method. Research into further ways of choosing the REG parameter and updating strategies would be key in improving the performance of the REG method for use in NWP where there is limited time and computational cost available. The alternative choices of the initial REG parameter that we consider show that if we know enough about the structure of the background error covariance matrix, we are able to improve the REG method. We have also discussed how the REG parameter interacts with the 4D-Var Hessian. For future work, an investigation into the use of second-order information in VarDA may be of interest. Knowing more information about the second-order terms of the Hessian of the VarDA cost function will enable us to understand more about the curvature of the function and which choices of method/parameters to make.

Our findings for the second main research question RQ2 provide an insight into the use of reduced resolution inner loop methods in incremental VarDA. The bound that we derived on the condition number of the reduced resolution preconditioned 3D-Var Hessian provides the DA community with an understanding of how the inner loop interacts with the background error covariance matrix and the observation operator and influences the convergence of an inner loop method. In operational VarDA settings, correctly prescribing background errors is of utmost importance, especially when using a reduced resolution inner loop problem, as we have found that the strength of background error correlations has a greater impact on the conditioning of the inner loop than the effect of the use of a reduced resolution inner problem.

We conclude from these results that a reduction in the inner loop resolution using the linear interpolation extension matrices negatively affects the convergence of the outer loop in the 3D-Var incremental method, both in terms of the numerical accuracy of the analysis and the accuracy of the frequencies of the analysis resolved. However, we know that using a reduced resolution inner loop has a clear computational time and cost benefit, especially considering the dimension of the variational problem in practice (of order $10^8 - 10^9$) [27]. Therefore, it is important to consider this saving in practical implementations to achieve a balance.

In future work, we may consider VarDA settings that align more closely with what occurs in practical implementations of VarDA. When addressing RQ2, we only considered the 3D-Var problem. In future work, it would be appropriate to check whether our conclusions hold for the 4D-Var problem.

To strengthen the results of this thesis, three more general directions for future work would be to consider

1. the use of an inexact solver. Within our work, we only consider when the VarDA inner loop is solved exactly. In practice, the inner loop problem is solved inexactly through the use of an iterative linear solver [52]. Future work into the effect of the use of an inexact solver on the performance of LS and REG and how an iterative inner loop

method may perform at different resolutions would be of interest.

2. the use of correlated observation error. Within our work, we only consider the use of uncorrelated observation error covariance matrices. The use of correlated observation error covariance matrices has been shown to improve the accuracy of numerical weather forecasting [21]. If we instead consider using a correlation matrix when defining the observation error covariance matrix, we expect the structure of this matrix to have an impact on both the bound derived on the condition number of the preconditioned Hessian, as well as on the convergence of LS and REG.

3. the use of operational models. Within our work, we use two Lorenz models that replicate atmospheric processes. Testing to see if our results hold when applied to the 4D-Var problem with a realistic model used in NWP practice would provide further insight into whether our conclusions hold in VarDA practice.

Data assimilation of the future is being built to make use of the newest computer architectures, where parallel processes are key in reducing computational time and costs. Incremental VarDA as it is cannot be parallelised as it requires the time integration of the linear and adjoint models. Hybrid methods that combine VarDA and ensemble techniques are becoming increasingly popular in light of the latest computer architectures (see Section 5 of [3] for a detailed review). Studying the convergence of such methods is a key idea for future research.

Bibliography

[1] G. Arfken, H. Weber, and F. Harris. *Mathematical Methods for Physicists: A Comprehensive Guide*. Elsevier Science, 2011.

[2] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[3] R. Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):607–633, 2017.

[4] R. N. Bannister. 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, 2008.

[5] R. N. Bannister. 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, 2008.

[6] P. Bauer, T. Quintino, N. Wedi, A. Bonanni, M. Chrust, W. Deconinck, M. Diamantakis, P. Düben, S. English, J. Flemming, P. Gillies, I. Hadade, J. Hawkes, M. Hawkins, O. Iffrig, C. Kühnlein, M. Lange, P. Lean, O. Marsden, A. Müller, S. Saarinen, D. Sarmany, M. Sleigh, S. Smart, P. Smolarkiewicz, D. Thiemert, G. Tumolo, C. Weihrauch, C. Zanna, and P. Maciel. The ECMWF Scalability Programme: Progress and Plans. (857), 02 2020.

[7] V. Beiranvand, W. Hare, and Y. Lucet. Best practices for comparing optimization algorithms. *Optimization and Engineering*, 18(4):815–848, 2017.

[8] S. Bellavia, C. Cartis, N. I. Gould, B. Morini, and P. L. Toint. Convergence of a regularized Euclidean residual algorithm for nonlinear least-squares. *SIAM Journal on Numerical Analysis*, 48(1):1–29, 2010.

[9] E. Bergou, S. Gratton, and L. Vicente. Levenberg–Marquardt Methods Based on Probabilistic Gradient Models and Inexact Subproblem Solution, with Application to Data Assimilation. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):924–951, 2016.

[10] E. H. Bergou, Y. Diouane, and V. Kungurtsev. Convergence and complexity analysis of a Levenberg–Marquardt algorithm for inverse problems. *Journal of Optimization Theory and Applications*, 185:927–944, 2020.

[11] D. P. Bertsekas. Nonlinear Programming. Athena Scientific Belmont. *Massachusetts, USA*, 1999.

[12] A. Bjerhammar. *Application of calculus of matrices to method of least squares: with special reference to geodetic calculations*. Elander Göteborg, 1951.

[13] M. Bocquet and P. Sakov. Combining inflation-free and iterative ensemble Kalman filters for strongly nonlinear systems. *Nonlinear Processes in Geophysics*, 19(3):383–399, 2012.

[14] M. Bonavita, E. Hólm, L. Isaksen, and M. Fisher. The evolution of the ECMWF hybrid data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 142(694):287–303, 2016.

[15] M. Bonavita, L. Isaksen, and E. Hólm. On the use of EDA background error variances in the ECMWF 4D-Var. *Quarterly Journal of the Royal Meteorological Society*, 138(667):1540–1559, 2012.

[16] M. Bonavita, P. Lean, and E. Holm. Nonlinear effects in 4D-Var. *Nonlinear Processes in Geophysics*, 25(3):713–729, 2018.

[17] M. Bonavita, Y. Trémolet, E. Holm, S. T. Lang, M. Chrust, M. Janisková, P. Lopez, P. Laloyaux, P. De Rosnay, M. Fisher, et al. *A strategy for data assimilation*. ECMWF, 2017.

[18] W. Briggs and V. Henson. *The DFT: An Owners' Manual for the Discrete Fourier Transform*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995.

[19] M. Buehner, R. McTaggart-Cowan, A. Beaulne, C. Charette, L. Garand, S. Heilliette, E. Lapalme, S. Laroche, S. R. Macpherson, J. Morneau, et al. Implementation of deterministic weather forecasting systems based on ensemble–variational data assimilation at Environment Canada. Part I: The global system. *Monthly Weather Review*, 143(7):2532–2559, 2015.

[20] M. Buehner, J. Morneau, and C. Charette. Four-dimensional ensemble-variational data assimilation for global deterministic weather prediction. *Nonlinear Processes in Geophysics*, 20(5):669–682, 2013.

[21] W. F. Campbell, E. A. Satterfield, B. Ruston, and N. L. Baker. Accounting for correlated observation error in a dual-formulation 4D variational data assimilation system. *Monthly Weather Review*, 145(3):1019–1032, 2017.

[22] C. Cartis. C6.2/B2: Continuous Optimization.

[23] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

[24] C. Cartis, N. I. Gould, and P. L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function-and derivative-evaluation complexity. *Mathematical Programming*, 130(2):295–319, 2011.

[25] A. M. Clayton, A. C. Lorenc, and D. M. Barker. 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, 2013.

[26] A. R. Conn, N. I. Gould, and P. L. Toint. *Trust Region Methods*. SIAM, 2000.

[27] P. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994.

[28] M. Cullen, M. A. Freitag, S. Kindermann, and R. Scheichl. *Large scale inverse problems: computational methods and applications in the earth sciences*, volume 13. Walter de Gruyter, 2013.

[29] D. N. Daescu and I. M. Navon. An analysis of a hybrid optimization method for variational data assimilation. *International Journal of Computational Fluid Dynamics*, 17(4):299–306, 2003.

[30] R. Daley. *Atmospheric data analysis*. Number 2. Cambridge University Press, 1993.

[31] P. J. Davis. *Circulant matrices*. New York: Wiley, 1979.

[32] J. E. Dennis, Jr and J. J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, 19(1):46–89, 1977.

[33] J. E. Dennis Jr and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM, 1996.

[34] ECMWF. ECMWF Newsletter No.158 Winter 2018/19. (158):21–26, 2019.

[35] L. Elden, L. Wittmeyer-Koch, and H. B. Nielsen. Introduction to Numerical Computation-analysis and MATLAB illustrations. 2004.

[36] S. Endre and D. Mayers. An introduction to numerical analysis. *Cambridge, UK*, 2003.

[37] A. Fillion, M. Bocquet, and S. Gratton. Quasi-static ensemble variational data assimilation: a theoretical and numerical study with the iterative ensemble Kalman smoother. *Nonlinear Processes in Geophysics*, 25:315–334, 2018.

[38] M. Fisher. Background error covariance modelling. In *Seminar on Recent Development in Data Assimilation for Atmosphere and Ocean*, pages 45–63. Shinfield Park, Reading, 2003.

[39] M. Fisher, M. Leutbecher, and G. Kelly. On the equivalence between Kalman smoothing and weak-constraint four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3235–3246, 2005.

[40] M. Fisher, J. Nocedal, Y. Trémolet, and S. J. Wright. Data assimilation in weather forecasting: a case study in PDE-constrained optimization. *Optimization and Engineering*, 10(3):409–426, 2009.

[41] M. Fisher, Y. Trémolet, H. Auvinen, D. G. H. Tan, and P. Poli. Weak-constraint and long window 4DVAR. *ECMWF Technical Memorandum*, (655):47, 11/2011 2011.

[42] A. Fowler, S. Dance, and J. Waller. On the interaction of observation and prior error correlations in data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(710):48–62, 2018.

[43] P. Gauthier, M. Tanguay, S. Laroche, S. Pellerin, and J. Morneau. Extension of 3DVAR to 4DVAR: Implementation of 4DVAR at the Meteorological Service of Canada. *Monthly Weather Review*, 135(6):2339–2354, 2007.

[44] J. C. Gilbert and C. Lemaréchal. Some numerical experiments with variable-storage quasi-Newton algorithms. *Mathematical Programming*, 45(1-3):407–435, 1989.

[45] P. E. Gill and W. Murray. Algorithms for the solution of the nonlinear least-squares problem. *SIAM Journal on Numerical Analysis*, 15(5):977–992, 1978.

[46] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.

[47] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins University Press, 2012.

[48] M. Goodliff, J. Amezcua, and P. J. Van Leeuwen. Comparing hybrid data assimilation methods on the Lorenz 1963 model with increasing non-linearity. *Tellus A: Dynamic Meteorology and Oceanography*, 67(1):26928, 2015.

[49] N. I. Gould, M. Porcelli, and P. L. Toint. Updating the regularization parameter in the adaptive cubic regularization algorithm. *Computational Optimization and Applications*, 53(1):1–22, 2012.

[50] S. Gratton, S. Gürol, E. Simon, and P. L. Toint. 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, 2018.

[51] S. Gratton, P. Laloyaux, and A. Sartenaer. Derivative-free optimization for large-scale nonlinear data assimilation problems. *Quarterly Journal of the Royal Meteorological Society*, 140(680):943–957, 2014.

[52] S. Gratton, A. S. Lawless, and N. K. Nichols. Approximate Gauss–Newton methods for nonlinear least squares problems. *SIAM Journal on Optimization*, 18(1):106–132, 2007.

[53] R. M. Gray. *Toeplitz and circulant matrices: A review*. now publishers inc, 2006.

[54] A. Griewank. The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical report, Technical report NA/12, 1981.

[55] S. A. Haben, A. S. Lawless, and N. K. Nichols. Conditioning of the 3DVar data assimilation problem. *University of Reading, Dept. of Mathematics, Math Report Series*, 3:2009, 2009.

[56] S. A. Haben, A. S. Lawless, and N. K. Nichols. Conditioning and preconditioning of the variational data assimilation problem. *Computers & Fluids*, 46(1):252–256, 2011.

[57] S. A. Haben, A. S. Lawless, and N. K. Nichols. Conditioning of incremental variational data assimilation, with application to the Met Office system. *Tellus A: Dynamic Meteorology and Oceanography*, 63(4):782–792, 2011.

[58] E. Hólm, R. Forbes, S. Lang, L. Magnusson, and S. Malardel. New model cycle brings higher resolution. *ECMWF Newsletter*, 147:14–19, 2016.

[59] N. B. Ingleby. The statistical structure of forecast errors and its representation in The Met. Office Global 3-D Variational Data Assimilation Scheme. *Quarterly Journal of the Royal Meteorological Society*, 127(571):209–231, 2001.

[60] I. C. Ipsen, C. Kelley, and S. Pope. Rank-deficient nonlinear least squares problems and subset selection. *SIAM Journal on Numerical Analysis*, 49(3):1244–1266, 2011.

[61] H. Järvinen, J.-N. Thépaut, and P. Courtier. Quasi-continuous variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 122(530):515–534, 1996.

[62] E. W. Karas, S. A. Santos, and B. F. Svaiter. Algebraic rules for quadratic regularization of Newton's method. *Computational Optimization and Applications*, 60(2):343–376, 2015.

[63] E. Klinker, F. Rabier, G. Kelly, and J.-F. Mahfouf. The ECMWF operational implementation of four-dimensional variational assimilation. III: Experimental results and diagnostics with operational configuration. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1191–1215, 2000.

[64] S. Laroche and P. Gauthier. A validation of the incremental formulation of 4D variational data assimilation in a nonlinear barotropic flow. *Tellus A: Dynamic Meteorology and Oceanography*, 50(5):557–572, 1998.

[65] A. S. Lawless, S. Gratton, and N. K. Nichols. An investigation of incremental 4D-Var using non-tangent linear models. *Quarterly Journal of the Royal Meteorological Society*, 131(606):459–476, 2005.

[66] A. S. Lawless and N. K. Nichols. Inner-loop stopping criteria for incremental four-dimensional variational data assimilation. *Monthly Weather Review*, 134(11):3425–3435, 2006.

[67] F.-X. Le Dimet, I. M. Navon, and D. N. Daescu. Second-order information in data assimilation. *Monthly Weather Review*, 130(3):629–648, 2002.

[68] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A: Dynamic Meteorology and Oceanography*, 38(2):97–110, 1986.

[69] P. Lean, E. Hólm, M. Bonavita, N. Bormann, A. McNally, and H. Järvinen. Continuous data assimilation for global numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 147(734):273–288, 2021.

[70] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[71] J. L. Lions. Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. 1968.

[72] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528, 1989.

[73] Z.-Q. Liu and F. Rabier. The interaction between model resolution, observation resolution and observation density in data assimilation: A one-dimensional study. *Quarterly Journal of the Royal Meteorological Society*, 128(582):1367–1386, 2002.

[74] A. Lorenc, S. Ballard, R. Bell, N. Ingleby, P. Andrews, D. Barker, J. Bray, A. Clayton, T. Dalby, D. Li, et al. The Met. Office global three-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 126(570):2991–3012, 2000.

[75] A. C. Lorenc. Development of an Operational Variational Assimilation Scheme (gtSpecial IssueltData Assimilation in Meteorology and Oceanography: Theory and Practice). *Journal of the Meteorological Society of Japan. Ser. II*, 75(1B):339–346, 1997.

[76] A. C. Lorenc, N. E. Bowler, A. M. Clayton, S. R. Pring, and D. Fairbairn. Comparison of hybrid-4DEnVar and hybrid-4DVar data assimilation methods for global NWP. *Monthly Weather Review*, 143(1):212–229, 2015.

[77] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[78] E. N. Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

[79] R. Lyons. *Understanding Digital Signal Processing: Unders Digita Signal Proces_3*. Pearson Education, 2010.

[80] J.-F. Mahfouf and F. Rabier. The ECMWF operational implementation of four-dimensional variational assimilation. II: Experimental results with improved physics. *Quarterly Journal of the Royal Meteorological Society*, 126(564):1171–1190, 2000.

[81] S. Malardel, N. Wedi, W. Deconinck, M. Diamantakis, C. Kühnlein, G. Mozdzynski, M. Hamrud, and P. Smolarkiewicz. A new grid for the IFS. *ECMWF Newsletter*, 146:23–28, 2016.

[82] J. Mandel, E. Bergou, and S. Gratton. 4DVAR by ensemble Kalman smoother. *arXiv preprint arXiv:1304.5271*, 2013.

[83] J. Mandel, E. Bergou, S. Gürol, S. Gratton, and I. Kasanickỳ. Hybrid Levenberg-Marquardt and weak-constraint ensemble Kalman smoother method. *Nonlinear Processes in Geophysics*, 23(2):59, 2016.

[84] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[85] S. Massart and M. Fisher. ECMWF training course. Assimilation Algorithms Lecture 2: 3D-Var, 2017.

[86] MathWorks. mldivide documentation. `https://www.mathworks.com/help/matlab/ref/mldivide.html`, 2021. [Online; accessed 30-May-2021].

[87] A. J. Moodey, A. S. Lawless, R. W. Potthast, and P. J. Van Leeuwen. Nonlinear error dynamics for cycled data assimilation methods. *Inverse Problems*, 29(2):025002, 2013.

[88] E. H. Moore. On the reciprocal of the general algebraic matrix. *Bull. Am. Math. Soc.*, 26:394–395, 1920.

[89] J. J. Moré. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[90] J. J. Moré and S. M. Wild. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization*, 20(1):172–191, 2009.

[91] I. Navon and D. M. Legler. Conjugate-gradient methods for large-scale minimization in meteorology. *Monthly Weather Review*, 115(8):1479–1502, 1987.

[92] Y. Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

[93] Y. Nesterov and B. T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[94] H. Ngodock, M. Carrier, S. Smith, and I. Souopgui. Weak and Strong Constraints Variational Data Assimilation with the NCOM-4DVAR in the Agulhas Region Using the Representer Method. *Monthly Weather Review*, 145(5):1755–1764, 2017.

[95] N. K. Nichols. Mathematical concepts of data assimilation. In W. Lahoz, R. Swinbank, and B. Khattatov, editors, *Data Assimilation: Making Sense of Observations*, pages 13–39. Springer, 2010.

[96] J. Nocedal and S. J. Wright. *Numerical Optimization 2nd Edition*. Springer, 2006.

[97] B. J. Olson, S. W. Shaw, C. Shi, C. Pierre, and R. G. Parker. Circulant matrices and their application to vibration analysis. *Applied Mechanics Reviews*, 66(4), 2014.

[98] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. SIAM, 1970.

[99] M. Osborne. Nonlinear least squares—the Levenberg algorithm revisited. *The ANZIAM Journal*, 19(3):343–357, 1976.

[100] R. Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge University Press, 1955.

[101] C. Pires, R. Vautard, and O. Talagrand. On extending the limits of variational assimilation in nonlinear chaotic systems. *Tellus A: Dynamic Meteorology and Oceanography*, 48(1):96–121, 1996.

[102] M. J. D. Powell. Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12(1):241–254, 1977.

[103] F. Rabier. Overview of global data assimilation developments in numerical weather-prediction centres. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3215–3233, 2005.

[104] F. Rabier and P. Courtier. Four-dimensional assimilation in the presence of baroclinic instability. *Quarterly Journal of the Royal Meteorological Society*, 118(506):649–672, 1992.

[105] F. Rabier, H. Järvinen, E. Klinker, J.-F. Mahfouf, and A. Simmons. 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, 2000.

[106] F. Rabier, J.-N. Thépaut, and P. Courtier. Extended assimilation and forecast experiments with a four-dimensional variational assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1861–1887, 1998.

[107] G. Radnóti, Y. Trémolet, E. Andersson, L. Isaksen, E. Hólm, and M. Janisková. *Diagnostics of linear and incremental approximations in 4D-Var revisited for higher resolution analysis*. ECMWF, 2005.

[108] S. Rainwater, C. H. Bishop, and W. F. Campbell. The benefits of correlated observation errors for small scales. *Quarterly Journal of the Royal Meteorological Society*, 141(693):3439–3445, 2015.

[109] F. Rawlins, S. Ballard, K. Bovis, A. Clayton, D. Li, G. Inverarity, A. Lorenc, and T. Payne. The Met Office global four-dimensional variational data assimilation scheme. *Quarterly Journal of the Royal Meteorological Society*, 133(623):347–362, 2007.

[110] H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

[111] L. S H. *Numerical analysis of partial differential equations*, volume 102. John Wiley & Sons, 2012.

[112] R. Seaman. Absolute and differential accuracy of analyses achievable with specified observational network characteristics. *Monthly Weather Review*, 105(10):1211–1222, 1977.

[113] D. F. Shanno and K.-H. Phua. Remark on "Algorithm 500: Minimization of unconstrained multivariate functions [e4]". *ACM Transactions on Mathematical Software (TOMS)*, 6(4):618–622, 1980.

[114] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[115] D. Simonin, S. Ballard, and Z. Li. Doppler radar radial wind assimilation using an hourly cycling 3D-Var with a 1.5 km resolution version of the Met Office Unified Model for nowcasting. *Quarterly Journal of the Royal Meteorological Society*, 140(684):2298–2314, 2014.

[116] D. Simonin, J. A. Waller, S. P. Ballard, S. L. Dance, and N. K. Nichols. A pragmatic strategy for implementing spatially correlated observation errors in an operational system: An application to Doppler radial winds. *Quarterly Journal of the Royal Meteorological Society*, 145(723):2772–2790, 2019.

[117] J. O. Smith. *Mathematics of the discrete Fourier transform (DFT): with audio applications*. Julius Smith, 2007.

[118] L. M. Stewart, S. Dance, and N. Nichols. Correlated observation errors in data assimilation. *International Journal for Numerical Methods in Fluids*, 56(8):1521–1527, 2008.

[119] A. Stuart and A. R. Humphries. *Dynamical systems and numerical analysis*, volume 2. Cambridge University Press, 1998.

[120] D. Sundararajan. *The Discrete Fourier Transform: Theory, Algorithms and Applications*. World Scientific, 2001.

[121] K. Swanson, T. Palmer, and R. Vautard. Observational error structures and the value of advanced assimilation techniques. *Journal of the Atmospheric Sciences*, 57(9):1327–1340, 2000.

[122] K. Swanson, R. Vautard, and C. Pires. Four-dimensional variational assimilation and predictability in a quasi-geostrophic model. *Tellus A: Dynamic Meteorology and Oceanography*, 50(4):369–390, 1998.

[123] J. M. Tabeart, S. L. Dance, S. A. Haben, A. S. Lawless, N. K. Nichols, and J. A. Waller. The conditioning of least-squares problems in variational data assimilation. *Numerical Linear Algebra with Applications*, page e2165, 2018.

[124] M. Tanguay, P. Bartello, and P. Gauthier. Four-dimensional data assimilation with a wide range of scales. *Tellus A: Dynamic Meteorology and Oceanography*, 47(5):974–997, 1995.

[125] J.-N. Thepaut and P. Courtier. Four-dimensional variational data assimilation using the adjoint of a multilevel primitive-equation model. *Quarterly Journal of the Royal Meteorological Society*, 117(502):1225–1254, 1991.

[126] Y. Trémolet. Incremental 4D-Var convergence study. *Tellus A: Dynamic Meteorology and Oceanography*, 59(5):706–718, 2007.

[127] F. Veersé and J.-N. Thépaut. Multiple-truncation incremental approach for four-dimensional variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 124(550):1889–1908, 1998.

[128] J. Vialard, A. Weaver, D. Anderson, and P. Delecluse. Three-and four-dimensional variational assimilation with a general circulation model of the tropical Pacific Ocean. Part II: Physical validation. *Monthly Weather Review*, 131(7):1379–1395, 2003.

[129] J. A. Waller, S. L. Dance, and N. K. Nichols. Theoretical insight into diagnosing observation error correlations using observation-minus-background and observation-minus-analysis statistics. *Quarterly Journal of the Royal Meteorological Society*, 142(694):418–431, 2016.

[130] Z. Wang, K. Droegemeier, and L. White. The adjoint Newton algorithm for large-scale unconstrained optimization in meteorology applications. *Computational Optimization and Applications*, 10(3):283–320, 1998.

[131] Z. Wang, I. Navon, X. Zou, and F. Le Dimet. A truncated Newton optimization algorithm in meteorology applications with analytic Hessian/vector products. *Computational Optimization and Applications*, 4(3):241–262, 1995.

[132] A. Weaver, J. Vialard, and D. Anderson. Three-and four-dimensional variational assimilation with a general circulation model of the tropical Pacific Ocean. Part I: Formulation, internal diagnostics, and consistency checks. *Monthly Weather Review*, 131(7):1360–1378, 2003.

[133] M. Weiser, P. Deuflhard, and B. Erdmann. Affine conjugate adaptive Newton methods for nonlinear elastomechanics. *Optimisation Methods and Software*, 22(3):413–431, 2007.

[134] P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.

[135] X. Zou, I. M. Navon, M. Berger, K. H. Phua, T. Schlick, and F.-X. Le Dimet. Numerical experience with limited-memory quasi-Newton and truncated Newton methods. *SIAM Journal on Optimization*, 3(3):582–608, 1993.

[136] M. Zupanski. Maximum likelihood ensemble filter: Theoretical aspects. *Monthly Weather Review*, 133(6):1710–1726, 2005.