

Large-Scale Simulations Using First and Second Order Adjoints with Applications in Data Assimilation

Lin Zhang

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

Adrian Sandu, Chair
Yang Cao
Calvin J. Ribbens
Jeff Borggaard

June 9th, 2007
Blacksburg, Virginia

Keywords: 4D-Var Data Assimilation, Optimization, Second
Order Adjoints, Sensitivity Analysis

Copyright 2007, Lin Zhang

Large-Scale Simulations Using First and Second Order Adjoint with Applications in Data Assimilation

Lin Zhang

(ABSTRACT)

In large-scale air quality simulations we are interested in the influence factors which cause changes of pollutants, and optimization methods which improve forecasts. The solutions to these problems can be achieved by incorporating adjoint models, which are efficient in computing the derivatives of a functional with respect to a large number of model parameters. In this research we employ first order adjoints in air quality simulations. Moreover, we explore theoretically the computation of second order adjoints for chemical transport models, and illustrate their feasibility in several aspects.

We apply first order adjoints to sensitivity analysis and data assimilation. Through sensitivity analysis, we can discover the area that has the largest influence on changes of ozone concentrations at a receptor. For data assimilation with optimization methods which use first order adjoints, we assess their performance under different scenarios. The results indicate that the L-BFGS method is the most efficient.

Compared with first order adjoints, second order adjoints have not been used to date in air quality simulation. To explore their utility, we show the construction of second order adjoints for chemical transport models and demonstrate several applications including sensitivity analysis, optimization, uncertainty quantification, and Hessian singular vectors. Since second order adjoints provide second order information in the form of Hessian-vector product instead of the entire Hessian matrix, it is possible to implement applications for large-scale models which require second order derivatives. Finally, we conclude that second order adjoints for chemical transport models are computationally feasible and effective.

Acknowledgements

I would like to take this opportunity to thank the people who have provided me with their constant support, guidance and help over the years. I would like to thank my advisor, Dr. Adrian Sandu, who has been a constant source of inspiration and help at all stages of this work. He has truly been a friend, and a guide. I would also like to thank my committee members Dr. Yang Cao, Dr. Calvin Ribbens and Dr. Jeff Borggaard for their suggestions on my research.

I would like to thank my parents who gave me lots of courage every time I feel frustrated during the work. It was their love and support that kept me going forward at all times.

I would like to recognize the contributions of my colleagues who provide me with ideas and motivation. I would especially like to thank Emil Constantinescu, who worked closely with me on the Texas H59 project and gave me bunches of advice.

I would like to thank my friends who helped me whenever in need. They made my life wonderful and meaningful.

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 The Problem	1
1.2 Research Questions	2
1.3 Outline of the Thesis	3
2 Literature Review	5
2.1 Chemical Transport Models and Sensitivity Analysis	5
2.2 Data Assimilation	6
2.2.1 4D-Var Data Assimilation	6
2.2.2 Optimization Methods to Solve Large-Scale Nonlinear Problems	7
3 The Chemical Transport Model	11
3.1 Mathematical Modeling	11
3.1.1 Forward Model	12
3.1.2 Tangent Linear Model	13
3.1.3 Continuous Adjoint Model	14
3.1.4 Discrete Adjoint Model	15
3.2 The STEM Chemical Transport Model	16
4 Sensitivity Analysis	18
4.1 Direct Sensitivity Analysis: A Source-Oriented Approach	19
4.2 Adjoint Sensitivity Analysis: A Receptor-Oriented Approach	20

4.3	Adjoint Sensitivity Results	23
4.3.1	July 1, 2004	24
4.3.2	July 25, 2004	30
5	4D-Var Data Assimilation	32
5.1	Basic Theory	32
5.2	Applications of 4D-Var Data Assimilation in STEM	35
5.2.1	1st Test Case - Texas	35
5.2.2	2nd Test Case - Northeastern United States	41
6	Optimization Using First Order Adjoints	46
6.1	Optimization Methods	46
6.1.1	L-BFGS	47
6.1.2	Nonlinear Conjugate Gradient	48
6.1.3	Hessian Free Newton	49
6.1.4	Hybrid Method	49
6.2	Experiments and Results	50
6.2.1	AR Background	51
6.2.2	NMC Background	54
6.2.3	Artificial Data	56
7	Second Order Adjoints	60
7.1	Theory of Second Order Adjoints	60
7.2	Second Order Adjoints for Stiff ODEs	61
7.2.1	Continuous SOA	61
7.2.2	Discrete SOA	66
7.2.3	Implementation of SOA for Chemistry System	67
7.2.4	Discrete SOA for Transport System	69
7.3	Validation of the 3D Second Order Adjoints	71
7.3.1	CPU time for Second Order Adjoints Calculation	71
7.3.2	Validation of Tangent Linear Model	72
7.3.3	Validation of Second Order Adjoints	73

7.3.4	Validation of Hessian Symmetry	74
8	Applications of Second Order Adjoints	77
8.1	Sensitivity Analysis	77
8.2	Optimization	80
8.2.1	Daniel’s Nonlinear Conjugate Gradient Method	80
8.2.2	Hessian Free Newton	82
8.2.3	Optimization Results	83
8.3	Uncertainty Quantification	89
8.4	Directions of Fastest Error Growth	91
9	Conclusion	98
9.1	Conclusion	98
9.2	Contributions	99
9.3	Future Work	100
	Bibliography	102
	Appendix	109
A	Second Order Adjoints	109
A.1	The ODE Model, the Jacobian, and the Hessian	109

List of Figures

4.1	(a) Direct sensitivity analysis is a source-oriented approach. (b) Ad-joint sensitivity analysis is a receptor-oriented approach.	20
4.2	Instantaneous areas of influence for the tracer at DFW at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.	24
4.3	Instantaneous areas of influence of HCHO on DFW O3 at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.	25
4.4	Instantaneous areas of influence of NO2 on DFW O3 at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.	26
4.5	Instantaneous areas of influence of O3 on DFW O3 at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.	27
4.6	Integrated areas of influence for a passive tracer; the receptor site is ground level DFW. The 36 hours integration starts at 9 am July 1st, 2004.	28
4.7	July 1, 2004. Time-integrated areas of HCHO influence on DFW O3 concentration.	28
4.8	July 1, 2004. Time-integrated areas of NO2 influence on DFW O3 concentration.	29
4.9	July 1, 2004. Time-integrated areas of O3 influence on DFW O3 concentration.	29
4.10	July 25, 2004. Areas of influence of the tracer on tracer DFW con-centration.	30
4.11	July 25, 2004. Time-integrated areas of HCHO influence on DFW O3.	30
4.12	July 25, 2004. Time-integrated areas of NO2 influence on DFW O3. .	31
4.13	July 25, 2004. Time-integrated areas of O3 influence on DFW O3. . .	31

5.1	Information feedback flows between CTMs, observations and data assimilation.	33
5.2	The location of AirNow stations used in data assimilation experiments.	36
5.3	Four selected stations where O3 time series are considered.	36
5.4	(a) and (b) The location of SCHIAMACHY total NO2 column measurements on July 16, 2004. (c) The approximation of the averaging kernel used in the construction of the observation operator in data assimilation.	38
5.5	Ground level ozone distribution in Texas at 6pm CST July 1st, 2004 (a) before data assimilation, and (b) after data assimilation.	39
5.6	Scatter plot and quantile-quantile plot of model predictions versus observations (a) for the original model predictions before data assimilation, and (b) after data assimilation.	39
5.7	Time series of ozone concentrations on July 1st, 2004.	40
5.8	(a) The location of the ground measuring stations in support of the ICARTT campaign (340 in total) (b) The location of the two ozonesondes (S1, S2) and the path of the P3-B flight that provide observations used in data assimilation. Also shown are the locations of four selected stations (A–D) that will be used to illustrate the assimilation results.	43
5.9	Ground level ozone distribution in northeastern U.S. at 1pm EDT July 20, 2004 (a) before data assimilation, and (b) after data assimilation.	43
5.10	Scatter plot and quantile-quantile plot of model-observations agreement.	44
5.11	Time series of ozone concentrations.	45
6.1	Decrease of the cost function using AR background versus each iteration and the number of model runs for two CG methods.	53
6.2	Decrease of the cost function using AR background versus each iteration and the number of model runs for CG, L-BFGS and HFN.	53

6.3	Scatter plot and quantile-quantile plot of model-observations agreement using AR background and L-BFGS method.	54
6.4	Decrease of the cost function using NMC background versus each iteration and the number of model runs for CG, L-BFGS and HFN. . .	56
6.5	Scatter plot and quantile-quantile plot of model-observations agreement using NMC background.	56
6.6	Decrease of the cost function using artificial data versus each iteration and the number of model runs for CG, L-BFGS and HFN methods. . .	58
6.7	Scatter plot and quantile-quantile plot of model-observations agreement using artificial data.	58
7.1	Validation of the tangent linear model for the three-dimensional chemistry transport model against finite difference of forward model. . . .	73
7.2	Validation of the second order adjoint for the three-dimensional chemistry transport model against finite difference of first order adjoints. . .	75
8.1	Sensitivity of final <i>PAN</i> concentration with respect to the initial concentrations of <i>NO</i> and <i>NO</i> ₂ . The changes in <i>PAN</i> concentration for different changes in the initial conditions Δc^0 are shown against the first order approximation ($\lambda^T \cdot \Delta c^0$, marked with “x”) and against the second order approximation ($\lambda^T \cdot \Delta c^0 + 1/2 \cdot \sigma^T \cdot \Delta c^0$, marked with “o”). The first order sensitivity analysis is inaccurate for this highly nonlinear system, and the second order sensitivity analysis predicts much better the change in <i>PAN</i>	79
8.2	(a) Decrease of the cost function vs. number of iterations. (b) Decrease of the cost function vs. scaled CPU time.	84
8.3	Scatter plots and quantile-quantile plots of model-observations agreement: (a) before data assimilation, and (b) after data assimilation. . .	86
8.4	Quantile-Quantile plot for Background, L-BFGS, Danile, HFN and hybrid methods.	86

8.5	Ground level ozone distribution in northeastern U.S. at 1pm EDT on July 20, 2004. (a) before data assimilation, and (b) after data assimilation.	88
8.6	Difference between optimization solutions.	88
8.7	Time series of ozone concentrations at station C for L-BFGS, Daniel, HFN and hybrid solutions.	89
8.8	First principal component of the error in the initial ozone filed. The 2 ppb error isosurface is shown in (a) 3D View, (b) Top View, and (c) East View.	92
8.9	Second principal component of the error in the initial ozone filed. The 2 ppb error isosurface is shown in (a) 3D View, (b) Top View, and (c) East View.	93
8.10	Third principal component of the error in the initial ozone filed. The 2 ppb error isosurface is shown in (a) 3D View, (b) Top View, and (c) East View.	94
8.11	The 0.02 isosurface of the dominant Hessian singular vector: (a) 3D view, (b) Top view, and (c) East view.	97

List of Tables

5.1	Correlation coefficient between model prediction and observations. . .	41
6.1	RMS and correlation coefficient for three methods using AR back-ground.	54
6.2	RMS and correlation coefficient for three methods using NMC back-ground.	57
6.3	RMS and correlation coefficient for three methods using artificial data.	59
7.1	CPU times for a 12 hours three-dimensional chemistry and transport simulation. FWD denotes the forward model, TLM the tangent linear model, FOA the first order adjoint, and SOA the second order adjoint. Shown are the wall clock times and the times relative to the forward model run.	72
7.2	Checking Hessian for Symmetry.	76
8.1	Validation of the second order adjoints against finite differences of first order adjoints. The RMS norm of the relative difference decreases for smaller perturbations.	78
8.2	The quality of different optimized solutions measured by the norm of gradient, the correlation coefficient, and root mean square distance between model predictions and observations.	85
8.3	The smallest and largest five eigenvalues of the Hessian and the corresponding eigenvalues of the a posteriori covariance matrix.	91
8.4	The largest five Hessian singular eigenvalues.	96

Chapter 1

Introduction

1.1 The Problem

With the development of modern industry, the effect on air quality has received more and more attention over the world. Thanks to computers, researchers and scientists now are able to simulate, analyze and forecast changes of air quality in both local and regional ranges. There is no denying that it is a complicated and challenging task to build and improve an air quality model in order to simulate the real environment and achieve more accurate estimations of pollutants. Air quality models mathematically describe the species transport, dispersion, emission, chemical reactions, and related processes in the atmosphere. They estimate the air pollutant concentrations at many locations, which are referred to as receptors. Usually an air quality model involves millions of states so its scale is large. Besides, transport along with wind flow and chemical reactions among hundreds of species in the air are happening from time to time, from here to there. All these components need to be examined together to model and characterize the state of the atmosphere, and then this model can be used to predict how pollutants are transported and distributed from the sources.

In air quality models, to be more specific, chemical transport models, both sensitivity analysis and data assimilation rely on the adjoint model to efficiently obtain the derivatives of a functional with respect to a large number of model parameters. Moreover, many other applications such as uncertainty quantification, Hessian

singular vectors, targeted observations, etc., also require the derivative information given by the adjoint model.

Sensitivity analysis is playing an important role in research on the influence factors of pollutants. With a sensitivity model, one is able to estimate the rate change of a model's solution when small perturbations are made to the initial conditions and/or to the model parameters. Direct sensitivity analysis is effective when we compute the effect of changing a few sources on the entire concentration field. Adjoint sensitivity analysis can be used to delineate areas of influence which provide information on the locations of major influence factors with respect to a given receptor site and time.

Data assimilation is another valuable technique in chemical transport models, especially for improving weather forecasts and for designing emission estimates. Data assimilation uses measurements to adjust the model predictions. It is helpful to obtain initial conditions, boundary conditions, emission estimates, etc. 4D-Var data assimilation is based on a 3D field and time propagation model, in which three factors are considered: a priori estimate of the state of the atmosphere, knowledge about the chemical fields and observations of some states of the model. The objective is to obtain the optimal states by minimizing the cost function to provide best fit to all observation data.

1.2 Research Questions

In this thesis, we will focus on the adjoint sensitivity analysis and adjoint models, including first order adjoints and second order adjoints. First order adjoints give the first derivatives of the cost functional with respect to the state. We explore the use of first order adjoints in chemical transport models. We use first order adjoints to show the area that has large influence on the ozone concentrations at a receptor. Besides, they are indispensable to provide optimization methods with gradients of the cost function with respect to model states. Second adjoints give second derivatives of the cost functional with respect to the state. These derivatives are useful to speed up the optimization process in data assimilation, to contribute to uncertainty analysis,

and to compute Hessian singular vectors. The first order adjoints have been widely used in many Chemical Transport Models (CTM). However, few people have done research on second order adjoints and their applications in chemical transport models up till now. Our work has provided answers to the following question: how to construct the second order adjoints in a complex chemical transport model, which consists of coupled PDEs and is discretized in both time and space. What's more, we reveal feasibility and importance of second order adjoints with several applications in the STEM chemical transport model.

The implementation of data assimilation is related to optimization methods. As chemical transport models contain millions of variables, we employ large-scale nonlinear optimization methods to obtain optimal states. A good optimization algorithm should decrease the cost function as fast as possible. The performance can be measured from two aspects: the cost function after a certain number of iterations, and the total CPU time while reaching this cost function. In our research, we have assessed the performance of L-BFGS, Fletcher-Reeves Conjugate Gradient method, Hessian Free Newton method and hybrid method, which all require gradients given by first order adjoints. We also compare the L-BFGS method with Daniel's method, modified Hessian Free Newton method and hybrid method that use Hessian-vector product, which can be provided by second order adjoints.

1.3 Outline of the Thesis

The rest of this thesis is organized as follows. In Chapter 2 we review the chemical transport models, sensitivity analysis and data assimilation with focus on large-scale nonlinear optimization methods. Chapter 3 introduces the properties of the STEM chemical transport model which is used for our experiments and also gives a general mathematical description. The theory of sensitivity analysis, especially the adjoint sensitivity analysis, followed by numerical results, is presented in Chapter 4. The process of 4D-Var data assimilation with results in two different fields is illustrated in Chapter 5. We discuss some optimization methods that only require first order adjoints in Chapter 6, and also show their performance under different scenarios.

Chapter 7 gives theoretical details on computing discrete second order adjoints in chemical transport models, as well as validation results in 3D STEM model. In Chapter 8, we demonstrate experimental results including sensitivity analysis, data assimilation, uncertainty quantification and Hessian singular vectors, to analyze the feasibility and importance of second order adjoints. Chapter 9 concludes the whole thesis, and points out future work.

Chapter 2

Literature Review

2.1 Chemical Transport Models and Sensitivity Analysis

Sensitivity analysis is a methodology that computes the change of the solution of a model with respect to the perturbation of model variables such as initial conditions, parameters, etc. Cacuci [3] presented mathematical foundations of the sensitivity for nonlinear dynamical systems and various classes of response functionals. Being a powerful tool in various applications, sensitivity analysis is receiving increasing attention in the area of air quality modeling.

Traditional direct sensitivity calculates the rate of change of model solutions to perturbation of input parameters one at a time. The Decoupled Direct Method (DDM) [17] uses this technique to obtain sensitivities of all state variables with respect to few parameters. However, it is infeasible for systems with large number of parameters. Different from direct sensitivity analysis, adjoint sensitivity analysis has the advantage of efficiently calculating the derivatives of a functional with respect to arbitrary parameters. In [52] the mathematical foundations of the adjoint sensitivity method, applied to air quality models, were discussed. Also, computational tools for performing three-dimensional adjoint sensitivity studies were presented. In addition, Henze and Seinfeld showed the development and use of adjoint sensitivity analysis on the GEOS-Chem global model in [29]. In this work, we explore the adjoint approach

and employ it to investigate the maximal area of influence on ozone concentrations at Dallas Fort Worth (DFW) area in the state of Texas using the STEM model.

2.2 Data Assimilation

Data assimilation is the process by which measurements are used to constrain the model predictions; the information from measurements can be used to obtain better initial conditions, better boundary conditions, enhanced emission estimates, etc. So it is essential in weather/climate analysis and forecast activities. Data assimilation combines information from three different sources: the physical and chemical laws of evolution (encapsulated in the model), the reality (as captured by the observations), and the current best estimate of the distribution of tracers in the atmosphere.

2.2.1 4D-Var Data Assimilation

4D-Var data assimilation allows the optimal combination of three sources of information [52]: a priori (“background”) estimate of the state of the atmosphere; knowledge about the chemical fields that are captured in the CTM; and observations of some of the state variables. The optimal state is obtained by minimizing the objective function to provide the best fit for all observational data.

The implementation of 4D-Var data assimilation for large-scale atmospheric models relies on adjoint modeling to provide the gradient of the objective function. Mathematical foundations of the adjoint sensitivity for nonlinear dynamical systems were presented by Cacuci [3,4] and Marchuk et al. [42,43].

Early applications of 4D-Var to chemical data assimilation were proposed by Fisher and Lary [23]. A similar model was used later to implement both 4D-Var and a Kalman filter method by Khattatov et al. [30]. Elbern et al. [18] use a tropospheric gas-phase box model to analyze the applicability of 4D-Var to tropospheric chemical data assimilation. In the past few years variational methods have been successfully used in data assimilation for comprehensive three-dimensional atmospheric chemistry models (Elbern and Schmidt [19], Errera and Fonteyn [22]). The work of Wang et al. [58] provided a review of adjoint methodology and data assimilation

applications to atmospheric chemistry. In [20] Elbern et al. study the skill and limits of 4D-Var techniques to analyze the emission rates of precursor constituents of ozone, with only ozone observations available. They also discuss the improvements in ozone prediction through the assimilation of observations in [20].

2.2.2 Optimization Methods to Solve Large-Scale Nonlinear Problems

To minimize the cost function, a well-performed optimization method is required. For most of the large-scale systems and practical problems, the cost function is usually nonlinear.

A Nonlinear Problem (NLP) is a problem that is formed

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to some constraints} \end{aligned}$$

where f is the objective function to minimize and x is typically a vector of variables.

Generally, optimization methods use iterations of the form:

At the current point x^k

- Determine a “descent direction” s^k . i.e. a direction along which the cost function decreases.

$$(g^k)^T \cdot s^k < 0 \quad (2.1)$$

- Perform line search to minimize $f(x)$ along s^k and update the solution

$$x^{k+1} = x^k + \alpha^k s^k \quad (2.2)$$

$$\alpha^k = \text{step length determined by line search} \quad (2.3)$$

A simplest way to solve this optimization problem is to use the steepest descent method, which is formulated below:

$$s^k = -g^k = -\nabla f(x^k)(g^k)^T s^k = -\|g^k\|^2 < 0 \quad \text{if not at minimum point} \quad (2.4)$$

This method can almost always guarantee convergence, but typically it converges at slow speed. In the other words, many iterations are needed.

Newton's method refers to a large class of methodologies for solving optimization problems and is widely used. It has many variations, but the main idea is to build a quadratic model for the cost function based on Taylor series about x^k .

$$q(x) = f^k + (g^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T G^k (x - x^k) \quad (2.5)$$

where q is the quadratic function, f the cost function, g the gradient and G the Hessian with respect to f . $f^k = f(x^k)$, $g^k = g(x^k)$, $G^k = G(x^k)$.

A minimum of $q(x)$ is characterized by :

$$\nabla q(x) = 0$$

If $G(x)$ is positive definite then $q(x)$ has a unique minimum.

$$\nabla q(x) = g^k + G^k(x - x^k) = 0 \Rightarrow x^{min} - x^k = -(G^k)^{-1}g^k ,$$

where x^{min} is the minimum of the quadratic model.

In Newton's method we look for the minimum of $f(x)$ along the search direction $s^k = -(G^k)^{-1}g^k$. If G^k is positive definite, we have

$$(g^k)^T s^k = -(g^k)^T (G^k)^{-1} g^k < 0 \quad (2.6)$$

This is actually a descent direction.

The advantage of Newton's method lies in its fast convergence near the minimum. However, the search direction is obtained at each step by solving the linear system:

$$G^k \cdot s^k = -g^k \quad (2.7)$$

For large-scale problems the Hessian matrix G^k is too large to evaluate and the system is also difficult to solve.

A large-scale NLP is one in which the number of variables is large. With the development of modern technologies, and improvement of computers' performance, more and more large-scale problems in different areas are being solved. For example, the STEM Chemical Transport model has more than 10^6 variables and minimizing an objective function of these variables is considered to be a large-scale NLP. Besides this, from fluid dynamics to aero dynamics, from molecular design to chemical

kinetics, from medicine to bioinformatics, all kinds of complex applications arise. Some of them have been well solved while some others still wait to be solved by some powerful algorithms.

It is obvious that a large-scale nonlinear problem requires more effort to solve than a normal linear or nonlinear problem. Effective numerical algorithms for solving a great many coupled nonlinear equations are vital to efficient computation of solutions to such models. Many algorithms aiming at solving large-scale problems have been proposed and they have been proved to work well in certain scenarios. People usually classify these algorithms in several branches. The most widely used are: Trust Region Methods, Conjugate Gradient Methods, and Newton's Methods (including Truncated Newton, Quasi-Newton Methods, etc.) In the following, we will discuss in detail some methods in the conjugate gradient method class, Hessian Free Newton method, which is in essence a truncated Newton method, and also the L-BFGS method, which belongs to the class of quasi-Newton methods. These methods are not theoretically independent and people sometimes interlace two or more algorithms together in order to achieve higher performance. The hybrid method presented below combines Hessian Free Newton and L-BFGS together.

In this research, we focus on optimization methods applied to our STEM model and we assess the performance of L-BFGS, Hessian Free Newton, hybrid, Fletcher-Reeves and Daniel's Conjugate Gradient methods. Our goal is to find out which is the most efficient method to obtain optimal initial conditions for the STEM model.

Nocedal et al. introduced the Limited Memory BFGS Method (L-BFGS) for Large Scale Optimization in [40,46]. They studied the performance of L-BFGS and compare it with some other versions of quasi-Newton methods. They also studied the convergence properties of the L-BFGS method. This method works very well on our STEM model and is superior to the other methods such as Nonlinear Conjugate Gradient and Hessian Free Newton (HFN) methods in terms of iteration numbers required to converge.

A relatively new optimization method that interlaces iterations of L-BFGS and HFN to improve the performance of each other is presented in [44]. The curvature information about the cost function plays the dual role of preconditioning the inner

conjugate gradient iteration in the HFN method and providing an initial matrix for L-BFGS iterations. The author explains both L-BFGS and HFN methods in the paper. HFN method is so named because it uses finite differences, or automatic differentiation, to approximate the matrix vector product rather than evaluate the product directly by explicit Hessian. This process is of special interest especially when the Hessian matrix is difficult to compute or is too large to store. In practice, evaluations of the gradients twice are required by Hessian Free Newton method to perform finite differences but no second order information is needed. We can improve the HFN by utilizing a second order adjoint model, which is able to provide the Hessian vector product.

In a survey written by Hager et al. [27], they reviewed the development of different versions of conjugate gradient methods and focused on conjugate gradient methods applied to nonlinear unconstrained optimization problem. In this paper, they made a summary of different ways of updating the parameter β , in which Daniel's method is distinct because it requires second order derivatives. Daniel presented his modified Conjugate Gradient Method in [13–15] in detail to solve linear and nonlinear problems. In the following work, we will show the implementation of Daniel's Conjugate Gradient Method and how to run STEM with the second order adjoints model to minimize our cost function. Besides, we also use the CG+ package which implements the traditional Fletcher-Reeves Method and Polak-Ribiere Method described in [47] to carry out optimization. Gilbert and Nocedal made contributions on convergence studies of conjugate gradient methods and introduced a sufficient descent condition to establish global convergence results [24].

Another optimization method worthy of investigation is the Truncated Newton method. A well known package called TNPACK based on this method is designed for solving large scale problems. The usage of this package is described in [53] and [54]. A more detailed explanation of the Truncated Newton method, including the theory, the convergence discussion, the computing of second-derivative information, and an introduction of other softwares that implement Truncated Newton Method can be found in a survey written by Nash [45].

Chapter 3

The Chemical Transport Model

A chemical transport model simulates the pollutants behavior in a selected domain, taking emissions, meteorology (wind, temperature, humidity, precipitation etc.) and a set of chemical initial and boundary conditions as inputs. In the following we give the mathematical description of the chemical transport model [52].

3.1 Mathematical Modeling

In the following u is used to denote the wind field vector, K the turbulent diffusivity tensor, ρ the air density in *moles/cm³*, and c_i the mole-fraction concentration of chemical species i ($1 \leq i \leq s$). The density of this species is ρc_i *moles/cm³*. Let V_i^{dep} be the deposition velocity of species i , Q_i the rate of surface emissions, and E_i the rate of elevated emissions for this species. The rate of chemical transformations f_i depends on absolute concentration values; the rate at which mole-fraction concentrations change is then $f_i(\rho c)/\rho$.

Consider a domain Ω which covers a region of the atmosphere. Let \vec{n} be the outward normal vector on each point of the boundary $\partial\Omega$. At each time point the boundary of the domain is partitioned into $\partial\Omega = \Gamma^{\text{IN}} \cup \Gamma^{\text{OUT}} \cup \Gamma^{\text{GR}}$ where Γ^{GR} is the ground level portion of the boundary; Γ^{IN} is the set of (lateral or top) boundary points where $u \cdot \vec{n} \leq 0$ and Γ^{OUT} the set where $u \cdot \vec{n} > 0$.

3.1.1 Forward Model

The evolution of concentrations in time is described by the material balance equations

$$\frac{\partial c_i}{\partial t} = -u \cdot \nabla c_i + \frac{1}{\rho} \nabla \cdot (\rho K \nabla c_i) + \frac{1}{\rho} f_i(\rho c) + E_i, \quad t^0 \leq t \leq T \quad (3.1)$$

$$c_i(t^0, x) = c_i^0(x), \quad (3.2)$$

$$c_i(t, x) = c_i^{\text{IN}}(t, x) \quad \text{for } x \in \Gamma^{\text{IN}}, \quad (3.3)$$

$$K \frac{\partial c_i}{\partial n} = 0 \quad \text{for } x \in \Gamma^{\text{OUT}}, \quad (3.4)$$

$$K \frac{\partial c_i}{\partial n} = V_i^{\text{dep}} c_i - Q_i \quad \text{for } x \in \Gamma^{\text{GR}}, \quad \text{for all } 1 \leq i \leq s \quad (3.5)$$

We refer to the system (3.1)–(3.5) as the *forward (direct) model*. To simplify the presentation, in this work we consider the initial state c^0 of the model as parameters; it is known that this does not restrict the generality of the formulation. The solution of the forward model $c = c(t, c^0)$ is uniquely determined once the model parameters c^0 are specified.

The direct model (3.1)–(3.5) is solved by a sequence of N time steps of length Δt taken between t^0 and $t^N = T$. At each time step one calculates the numerical approximation $c^k(x) \approx c(t^k, x)$ at $t^k = t^0 + k\Delta t$ such that

$$c^{k+1} = \mathcal{N}_{[t^k, t^{k+1}]} \circ c^k, \quad c^N = \prod_{k=0}^{N-1} \mathcal{N}_{[t^k, t^{k+1}]} \circ c^0 \quad (3.6)$$

The numerical solution operator \mathcal{N} is based on an operator splitting approach, where the transport steps along each direction and the chemistry steps are taken successively. Operator splitting is standard practice in computational air pollution modeling [34]. It allows the development of the forward, tangent linear, and adjoint models with relative ease. Formally, if we denote by \mathcal{T} the numerical solution operator for directional transport, and by \mathcal{C} the solution operator for chemistry, we have

$$\mathcal{N}_{[t, t+\Delta t]} = \mathcal{T}_X^{\Delta t/2} \circ \mathcal{T}_Y^{\Delta t/2} \circ \mathcal{T}_Z^{\Delta t/2} \circ \mathcal{C}^{\Delta t} \circ \mathcal{T}_Z^{\Delta t/2} \circ \mathcal{T}_Y^{\Delta t/2} \circ \mathcal{T}_X^{\Delta t/2} \quad (3.7)$$

The numerical errors introduced by splitting are an important component of model errors (see e.g., [57]). In this work, for the purpose of 4D-Var data assimilation, we assume the model errors to be small. Indeed, in computational air pollution

models the splitting errors oscillate with the diurnal cycle and do not grow unboundedly for evolving time [34].

3.1.2 Tangent Linear Model

An infinitesimal perturbation δc^0 in the parameters will result in perturbations $\delta c_i(t)$ of the concentration fields. These perturbations are solutions of the *tangent linear model* as discussed in (3.8)–(3.12). In the direct sensitivity analysis approach one solves the model (3.1)–(3.5) together with the tangent linear model forward in time [62].

$$\frac{\partial \delta c_i}{\partial t} = -u \cdot \nabla \delta c_i + \frac{1}{\rho} \nabla \cdot (\rho K \nabla \delta c_i) + F_{i,*}(\rho c) \delta c, \quad t^0 \leq t \leq T \quad (3.8)$$

$$\delta c_i(t^0, x) = \delta c_i^0(x), \quad (3.9)$$

$$\delta c_i(t, x) = \delta c_i^{\text{IN}}(t, x) = 0 \quad \text{for } x \in \Gamma^{\text{IN}}, \quad (3.10)$$

$$K \frac{\partial \delta c_i}{\partial n} = 0 \quad \text{for } x \in \Gamma^{\text{OUT}}, \quad (3.11)$$

$$K \frac{\partial \delta c_i}{\partial n} = V_i^{\text{dep}} \delta c_i \quad \text{for } x \in \Gamma^{\text{GR}} \quad (3.12)$$

In the above F is the Jacobian of the function f , and $F_{i,*}$ denotes its i -th row.

Similar to direct discrete model, we use the numerical operator \mathcal{N}' to describe the tangent linear discrete model. A perturbation δc^0 in the parameters c^0 propagates in time according to the tangent linear discrete equation

$$\delta c^{k+1} = \mathcal{N}'_{[t^k, t^{k+1}]} \circ \delta c^k, \quad \delta c^N = \prod_{k=0}^{N-1} \mathcal{N}'_{[t^k, t^{k+1}]} \circ \delta c^0 \quad (3.13)$$

where \mathcal{N}' is the tangent linear operator associated with the solution operator \mathcal{N} . For an operator splitting approach \mathcal{N}' is built from the tangent linear transport and chemistry operators

$$\mathcal{N}'_{[t, t+\Delta t]} = \mathcal{T}'_X^{\Delta t/2} \circ \mathcal{T}'_Y^{\Delta t/2} \circ \mathcal{T}'_Z^{\Delta t/2} \circ \mathcal{C}'^{\Delta t} \circ \mathcal{T}'_Z^{\Delta t/2} \circ \mathcal{T}'_Y^{\Delta t/2} \circ \mathcal{T}'_X^{\Delta t/2} \quad (3.14)$$

For an operator splitting approach (3.7) \mathcal{N}' is built from the tangent linear transport and chemistry operators \mathcal{T}' and \mathcal{C}' .

3.1.3 Continuous Adjoint Model

In the adjoint sensitivity analysis, one distinguishes between continuous and discrete adjoint modeling (see Sirkes and Tziperman [55]). On the one hand, continuous adjoint sensitivity, in practice, is solved numerically, thus resulting in a discretization of the continuous adjoint equations. On the other hand, the discrete adjoints are computed from the adjoint of the numerical discretization. The operations of discretization and adjoint usually do not commute, i.e. the discrete and the continuous adjoint approaches lead to different results.

In continuous adjoint model, we consider a scalar response functional defined in terms of the model solution $c(t)$.

$$\mathcal{J}(c^0) = \int_{t^0}^T \int_{\Omega} g(c(t, x)) dx dt \quad (3.15)$$

The response depends implicitly on the parameters c^0 via the dependence of $c(t)$ on c^0 . The continuous adjoint model is defined as the adjoint of the tangent linear model. By imposing the Lagrange identity and after a careful integration by parts one arrives at the following equations that govern the evolution of the adjoint variables:

$$\frac{\partial \lambda_i}{\partial t} = -\nabla \cdot (u \lambda_i) - \nabla \cdot \left(\rho K \nabla \frac{\lambda_i}{\rho} \right) - (F^T (\rho c) \lambda)_i - \phi_i, T \geq t \geq t^0 \quad (3.16)$$

$$\lambda_i(T, x) = \lambda_i^F(x), \quad (3.17)$$

$$\lambda_i(t, x) = 0 \quad \text{for } x \in \Gamma^{\text{OUT}}, \quad (3.18)$$

$$\lambda_i u + \rho K \frac{\partial (\lambda_i / \rho)}{\partial n} = 0 \quad \text{for } x \in \Gamma^{\text{OUT}}, \quad (3.19)$$

$$\rho K \frac{\partial \lambda_i / \rho}{\partial n} = V_i^{\text{dep}} \lambda_i \quad \text{for } x \in \Gamma^{\text{GR}}, \quad \text{for all } 1 \leq i \leq s, \quad (3.20)$$

where

$$\phi_i(t, x) = \frac{\partial g(c_1, \dots, c_n)}{\partial c_i}(t, x), \quad \lambda_i^F(x) = 0, \quad (3.21)$$

and $\lambda_i(t, x)$ are the adjoint variables associated with the concentrations $c_i(t, x)$, $1 \leq i \leq s$. In the above $F = \partial f / \partial c$ is the Jacobian of the chemical rate function f . To obtain the ground boundary condition we use the fact that $u \cdot \vec{n} = 0$ at ground level. We refer to (3.16)-(3.20) as the (continuous) adjoint system of the

tangent linear model. In the context of optimal control where the minimization of the functional 3.15 is required, the adjoint variables can be interpreted as Lagrange multipliers.

The adjoint system (3.16)–(3.20) depends on the states of the forward model (i.e. on the concentration fields through the nonlinear chemical term $F(\rho c)$ and possibly through the forcing term ϕ for nonlinear functionals. Note that the adjoint initial condition is posed at the final time T such that the forward model must be first solved forward in time, the state $c(t, x)$ saved for all t , then the adjoint model could be integrated backwards in time from T down to t^0 .

In practice a hybrid approach is used. The forward model is solved using a numerical method, and the numerical approximations of the state are saved periodically. These checkpoints are used in the definition of the adjoint equations. The continuous adjoint equation (3.16)–(3.20) is a convection-diffusion-reaction equation (with linearized chemistry) and can be solved by any numerical method of choice. In particular an operator splitting approach could be employed using the same numerical methods as for solving the direct model

$$\lambda^k = \mathcal{N}_{[t^{k+1}, t^k]} \circ \lambda^{k+1} \quad , \quad \lambda^0 = \prod_{k=0}^{N-1} \mathcal{N}_{[t^{N-k}, t^{N-k-1}]} \circ \lambda^N \quad (3.22)$$

For different cost functionals the forcing ϕ_i and the initial values λ_i^F are chosen such that the adjoint variables are the sensitivities of the cost functional.

3.1.4 Discrete Adjoint Model

In this approach the numerical discretization (3.6) of the (3.1)–(3.5) is considered to be the forward model. This is a pragmatic view, as only the numerical model is in fact available for analysis. We denote the state of the discretized model by $c_i^k[j]$, where i is the species index, j is the space discretization index and k the time discretization index. $c^k[j]$ refers to the vector of all species at time k and grid j . The cost function is defined in terms of the discrete model state

$$\mathcal{J}(c^0) = \sum_{k=0}^N \sum_j g(c^k[j]) \quad (3.23)$$

and one wants the derivatives of this functional with respect to the discrete model parameters $c_i^0[j]$.

To each tangent linear operator (3.14) corresponds an adjoint operator (denoted here with a star superscript). The adjoint equation of (3.14) is

$$\mathcal{N}_{[t+\Delta t, t]}^{I*} = \mathcal{T}_{X}^{I* \Delta t/2} \circ \mathcal{T}_{Y}^{I* \Delta t/2} \circ \mathcal{T}_{Z}^{I* \Delta t/2} \circ \mathcal{C}^{I* \Delta t} \circ \mathcal{T}_{Z}^{I* \Delta t/2} \circ \mathcal{T}_{Y}^{I* \Delta t/2} \circ \mathcal{T}_{X}^{I* \Delta t/2} \quad (3.24)$$

such that the resulting (discrete) adjoint model is

$$\lambda^k = \mathcal{N}_{[t^{k+1}, t^k]}^{I*} \circ \lambda^{k+1} + \phi^k, \quad k = N - 1, N - 2, \dots, 0; \quad \lambda^N[j] = \lambda^F(x_j) \quad (3.25)$$

The forcing function ϕ and the initial values λ^N are chosen such that the adjoint variables are sensitivities of the functional with respect to the state variables

$$\lambda_i^k[j] = \frac{\partial \mathcal{J}(c^0)}{\partial c_i^k[j]} \quad (3.26)$$

3.2 The STEM Chemical Transport Model

The Sulfur Transport Eulerian Model (STEM) [5]

(http://www.cgrer.uiowa.edu/people/carmichael/stem2_desc.html) has been developed to provide a theoretical basis to investigate the relationships between the emissions, atmospheric transport, chemical transformation, removal processes, and the resultant distribution of air pollutants and deposition patterns. STEM model has then been used to address a wide series of policy issues in U.S., Asia and Europe, related to acidification, cloud chemistry, tropospheric ozone, and aerosols formation.

The STEM uses the SAPRC-99 (Statewide Air Pollution Research Center's chemical mechanism) [8] and KPP(the Kinetic PreProcessor) [12], to determine chemical reactions. KPP implements integrations of chemical mechanisms to approximate the model states using implicit Rosenbrock, SDIRK and Runge-Kutta methods in both forward and adjoint models. The STEM model runs multiscale simulations in both time and space. From the time point of view, it ranges from 10^{-6} seconds for fast chemical reactions to days simulation measured in hours. Fast chemical reactions are referred to as atomic level reactions, such as O , OH radical activities, while long term simulation usually accounts for atmospheric species

transportation in large range. When it comes to spatial scales, STEM is able to simulate in range measured in meters, such as emissions of pollutants like NO , NO_2 , CO_2 , Volatile Organic Compounds (VOC) and particles from vehicles. Also, continental scales as large as thousands of kilometers can be used in STEM for air quality simulation. Generally, the horizontal resolution in the STEM 3-dimensional computational model is between $4*4$ and $80*80$ Km^2 , and the vertical resolution is variable.

The numerical experiments in this work use STEM to solve the mass-balance equations for concentrations of trace species in order to determine the fate of pollutants in the atmosphere. STEM has first order adjoint capabilities [52] and has been used extensively in real life chemical data assimilation studies [6, 9, 28]. In this work we apply the first adjoint model in STEM to sensitivity analysis and data assimilation. Also, we endow STEM with the capability to compute second order adjoints and we illustrate several applications of this capability.

Chapter 4

Sensitivity Analysis

In this chapter we simplify the CTM model as a system of coupled partial differential equations discretized in both time and space:

$$y^k = M(t^{k-1}, y^{k-1}, p), \quad y^0 = y(t^0, p), \quad k = 1, 2, \dots, K, \dots, F \quad (4.1)$$

where $y^k \in \mathbb{R}^n$ is the state vector representing the concentration field at time t^k , M is a discrete solution operator of the advection-diffusion-reaction equation, and $p \in \mathbb{R}^m$ the vector of model parameters (e.g., initial conditions, emission rates, etc). Its solution is uniquely determined by the initial state and the model parameters $y = y(t, p)$.

Sensitivity analysis is a formal method to assess the rate of change of a model's solution (4.1) when small perturbations are made to the initial values and/or to the model parameters. The rate of change of the solution with respect to the i -th model parameter (i.e., the sensitivity of the solution with respect to the i -th parameter) is denoted by

$$S_i(t) = \frac{\partial y(t)}{\partial p_i}, \quad S_i^k = \frac{\partial y^k}{\partial p_i} \quad (4.2)$$

where k represents the time level.

When the model solution and model parameters have different magnitudes, or different units, it is advantageous to consider scaled sensitivity coefficients

$$\hat{S}_i(t) = \frac{\partial y(t)}{\partial p_i} \cdot \frac{p_i}{y(t)} \quad (4.3)$$

The scaled sensitivity coefficients can be interpreted as the percentage change in the solution when the parameter value is increased by 1%.

4.1 Direct Sensitivity Analysis: A Source-Oriented Approach

The sensitivities of the model solution evolve in time according to the linearized model dynamics:

$$S_i^k = \frac{\partial M}{\partial y}(t^{k-1}, y^{k-1}, p) S_i^{k-1} + \frac{\partial M}{\partial p_i}(t^{k-1}, y^{k-1}, p), \quad S_i^0 = \frac{\partial y^0}{\partial p_i}, \quad 1 \leq k \leq F. \quad (4.4)$$

The direct sensitivity analysis solves both the model and the sensitivity equation, and advances forward in time. Note that there are as many sensitivity equations to solve as there are parameters, $1 \leq i \leq m$. Computational savings are possible by reusing the same linear algebra factorizations in the forward model and in all the sensitivity equations (the direct decoupled method for sensitivity analysis).

To interpret direct sensitivity analysis, consider $p = y^0$ to be the vector of initial concentrations, and consider a small change δp_i in the concentration of a certain species at the initial time and at a specific ‘‘source’’ location i (e.g., more NO_2 has been released at the initial time at the source location). The changes in the concentration field at later times and at all locations $\delta y(t) = S_i(t) \delta p_i$, due to the change in the source at the initial time, are obtained at successive times in the future by solving the sensitivity equation forward in time,

$$S_i^k = \frac{\partial M}{\partial y}(y^{k-1}) \cdot S_i^{k-1}, \quad S_i^0 = \frac{\partial y^0}{\partial p_i} = e_i, \quad 0 \leq k \leq F \quad (4.5)$$

where e_i is a vector with all entries equal to zero, except for entry i , which is equal to one. The source-oriented sensitivity analysis approach is illustrated in Figure 4.1(a), where an initial perturbation at a source location i is propagated throughout the modeling domain at future times.

Consequently, the direct sensitivity analysis approach is effective when the changes in all concentration levels across all grid points with respect to changes in few model parameters are needed. In our interpretation, direct sensitivity analysis is effective when we compute the effect of changing a few sources on the entire concentration field.

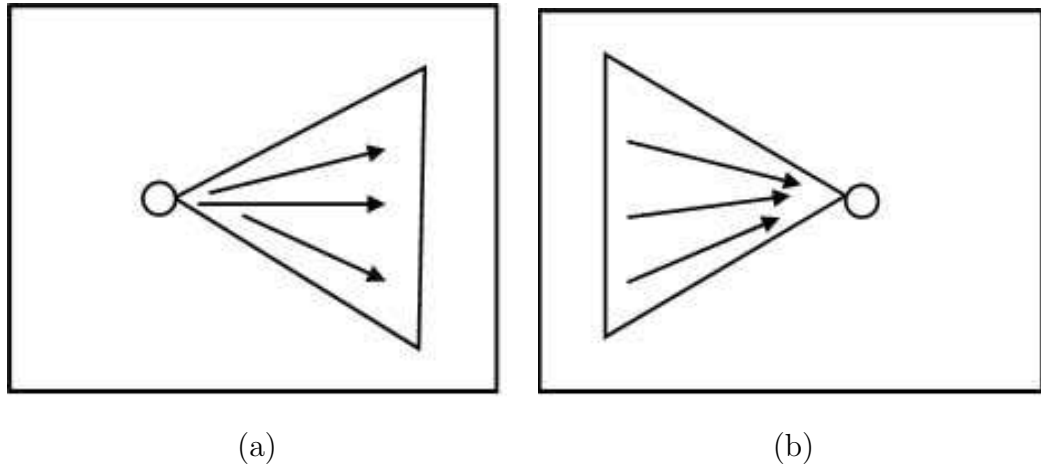


Figure 4.1: (a) Direct sensitivity analysis is a source-oriented approach. (b) Adjoint sensitivity analysis is a receptor-oriented approach.

4.2 Adjoint Sensitivity Analysis: A Receptor-Oriented Approach

In many instances one is interested in assessing the sensitivities of a cost function defined on the concentration field at the final time:

$$\text{Given } \Psi(y^F(p)) \in \mathbb{R} \text{ evaluate } \nabla_p \Psi = \left(\frac{\partial \Psi}{\partial p} \right)^T = \begin{bmatrix} \frac{\partial \Psi}{\partial p_1} \\ \vdots \\ \frac{\partial \Psi}{\partial p_m} \end{bmatrix} \in \mathbb{R}^m \quad (4.6)$$

The simplest example of a cost function is the concentration of a given species (e.g., ozone) at a given “receptor” location at the end of the simulation interval: $\Psi(y^F) = y_j(t^F)$. This is illustrated in 4.1(b): the value of the cost function at the receptor time and location is influenced by changes in concentrations, emissions, etc. at earlier times throughout the modeling domain.

Using the chain rule, the sensitivity of the cost function with respect to the parameters can be expressed as:

$$\frac{\partial \Psi(y^F)}{\partial p_i} = \frac{\partial \Psi(y^F)}{\partial y^F} S_i^F \quad (4.7)$$

Consequently, using the direct sensitivity analysis approach, the sensitivity of the concentration at the receptor location can be obtained only after computing the sensitivities of all concentrations at all grid points.

For simplicity, consider again the case where the parameters are the initial conditions $p = y^0$. The sensitivity of the cost function with respect to all parameters is given by the chain rule:

$$\begin{aligned} \frac{\partial \Psi(y^F)}{\partial p_i} &= \frac{\Psi(y^F)}{\partial y^F} \cdot \frac{\partial y^F}{\partial y^{F-1}} \cdots \frac{\partial y^1}{\partial y^0} \cdot \frac{\partial y^0}{\partial p_i}, \\ \frac{\partial y^k}{\partial y^{k-1}} &= \frac{\partial M}{\partial y}(t^{k-1}, y^{k-1}) \in \mathbb{R}^{n \times n}, \\ \frac{\partial y^0}{\partial p_i} &= e_i \in \mathbb{R}^n, \quad 1 \leq i \leq m. \end{aligned}$$

Working from right to left for each parameter $i = 1, \dots, m$ the vector e_i is multiplied by matrices $\partial y^k / \partial y^{k-1} = \partial M / \partial y(t^{k-1}, y^{k-1})$ for $k = 1, \dots, F$. Each matrix-vector multiplication corresponds to solving one step of the sensitivity equations.

A more effective computational process is obtained by the transposed chain rule:

$$\nabla_p \Psi(y^F) = \left(\frac{\partial \Psi(y^F)}{\partial p} \right)^T = \left(\frac{\partial y^1}{\partial y^0} \right)^T \cdots \left(\frac{\partial y^F}{\partial y^{F-1}} \right)^T \cdot \left(\frac{\partial \Psi(y^F)}{\partial y^F} \right)^T$$

Working from right to left again the vector $\partial \Psi / \partial y^F$ is multiplied successively by matrices $(\partial y^k / \partial y^{k-1})^T$ for $k = F, \dots, 1$. This process needs to be performed only once regardless of the number of parameters m , and is therefore very efficient. Each matrix-multiplication corresponds to one step of the adjoint model; note that the adjoint steps are taken in reverse order, from F down to 0. Formally if we denote the adjoint variables by λ^k , and impose the condition that they satisfy the following adjoint equations:

$$\begin{aligned} \lambda^F &= \left(\frac{\partial \Psi(y^F)}{\partial y^F} \right)^T, \\ \lambda^{k-1} &= \left(\frac{\partial y^F}{\partial y^{F-1}} \right)^T \lambda^k = \left(\frac{\partial M}{\partial y}(t^{k-1}, y^{k-1}) \right)^T \lambda^k \quad F \geq k \geq 1. \end{aligned}$$

then the adjoint variables are the gradients of the cost function with respect to changes in the state at earlier time:

$$\lambda^k = \left(\frac{\partial \Psi(y^F)}{\partial y^k} \right)^T = \nabla_{y^k} \Psi(y^F)$$

For the general situation the sensitivity of the cost function with respect to model parameters is obtained by a single integration of the adjoint model backwards in

time, and the relation:

$$\nabla_{p_i} \Psi = \left(\frac{\partial y^0}{\partial p_i} \right)^T \lambda^0 + \sum_{k=1}^F \left(\frac{\partial M}{\partial p_i} (t^{k-1}, y^{k-1}) \right)^T \lambda^k, \quad 1 \leq i \leq m \quad (4.8)$$

Note that the same adjoint variables are used to obtain the sensitivities with respect to all parameters; a single backward integration of the adjoint model is sufficient. Marchuk et al [43] presented the computation of adjoint equations for complicated systems.

The adjoint variables (4.8) are also called influence functions. They represent the sensitivity of the response functional with respect to the variations in the model state at time t^k and location i :

$$\lambda_i^k = \frac{\partial \Psi (y^F)}{\partial y_i^k} \quad (4.9)$$

Similar to (4.3), it is of interest to consider scaled influence functions, which represent the percentage change in the cost function when the concentration of a certain species at a certain location is changed by 1%,

$$\hat{\lambda}_i^k = \frac{\partial \Psi (y^F)}{\partial y_i^k} \cdot \frac{y_i^k}{\Psi (y^F)} \quad (4.10)$$

The distributions of the influence functions (adjoint variables) in the three-dimensional computation domain, which are available at any instant, provide the essential information for the sensitivity analysis [52]. For instance, isosurfaces of the i^{th} adjoint variable ($x : \lambda_i(t, x) = \text{constant}$) delineate “instantaneous areas of influence”, i.e., locations where perturbations in the concentration of the i^{th} species will produce significant changes in the response function, e.g., the observed Particulate Matter (PM) level at the receptor site and time. Denote time integrals of the adjoint variables over the time period of interest as

$$\sum_{k=0}^F \lambda^k \quad \text{or} \quad \sum_{k=0}^F \hat{\lambda}^k \quad (4.11)$$

These are “integrated areas of influence”, i.e. regions where the cumulative effect of concentration changes of the i^{th} species over the interval of interest will affect the target most.

In conclusion, we come to the following statement.

- Adjoint sensitivity analysis can be used to delineate areas of influence which provide information on the location of major influence factors with respect to a given receptor site and time. This offers a powerful method to characterize source-receptor relationships.

4.3 Adjoint Sensitivity Results

Adjoint sensitivity analysis was performed for a cost function that measures ground level ozone concentration in the Dallas/Fort Worth (DFW) receptor area. In order to understand the influence of the meteorological conditions (separately from the chemistry) on the adjoint, the adjoint sensitivities of a passive tracer were also computed. Simulations were carried out for two 36-hour intervals in July 2004; the first interval starts at 9 am July 1st, while the second starts at 9 am July 25th 2004. We also examined the July 16 episode and found that it had a flow pattern somewhat similar to July 25, so the corresponding results are not presented here.

We first consider the scaled adjoint sensitivities of DFW ozone with respect to earlier concentrations of O₃, NO₂, and HCHO. Specifically, we present the “integrated areas of influence”, i.e. the time integrals of scaled adjoint sensitivities of DFW ozone. The scaled adjoint variables are non-dimensional and can be interpreted as the relative contributions of each perturbation to the observed change in DFW ozone concentrations. The total relative (percent) change in DFW ozone is the sum of the scaled adjoint variable in each grid cell times the relative (percent) perturbation of the precursor concentrations in that grid cell. Since perturbations that appear at any of the earlier times can affect DFW ozone, the adjoint variables are time dependent: the contribution of each grid cell to the DFW ozone changes with the time a perturbation is introduced. To present synthetically all these influences, we “aggregate” the adjoint variables at all times into a single area of influence by performing a time integration. These results are shown next.

For the tracer calculations, the integrated areas of influence for the adjoint sensitivity (without scaling) are shown. These adjoint variables can be interpreted as the relative contributions of the perturbations in each grid cell to ozone concentration

in the DFW area. Specifically, the total change in DFW ozone concentration is the sum of the adjoint variable in each grid cell times the perturbation (of ozone or a precursor) in that grid cell.

4.3.1 July 1, 2004

We first discuss the instantaneous areas of influence for DFW ground level ozone on July 1-2, 2004. We assess which perturbations in ozone precursors at earlier times (beginning at 9 am July 1, 2004) have the largest influence on DFW ground level ozone measured at 9 pm July 2, 2004.

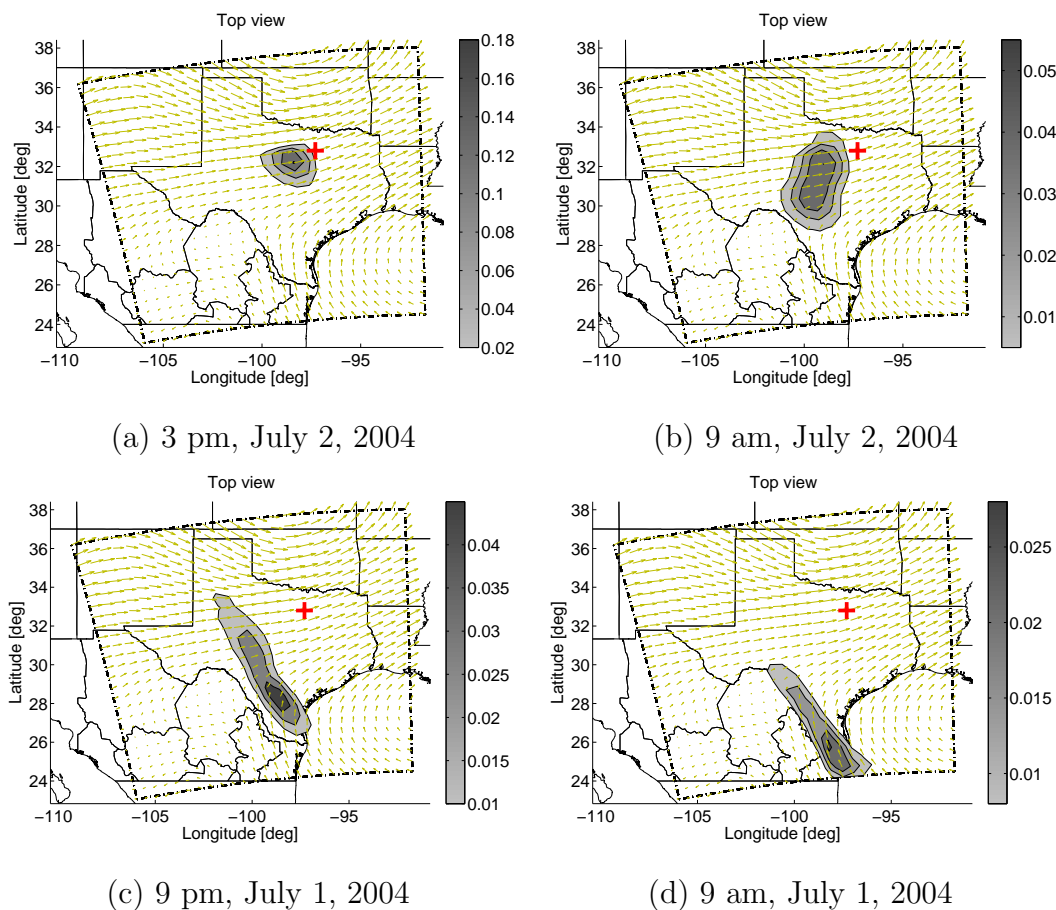


Figure 4.2: Instantaneous areas of influence for the tracer at DFW at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.

The results in Figure 4.2 indicate areas where perturbations in the tracer have the maximum impact on the observed tracer value at DFW. A perturbation of 1 ppb in the tracer at the given instant at a specific location will result in a change of the

observed tracer level equal to the magnitude of the adjoint variable at that location. Perturbations of 1ppb in the tracer level in the areas of maximum influence (dark gray) done 6h, 12h, 24h, and 36h before the receptor time lead to changes of 0.2ppb, 0.05ppb, 0.04ppb, and 0.025 ppb respectively in the tracer at the receptor. Note that the areas of maximum influence are farther away from the receptor for earlier times.

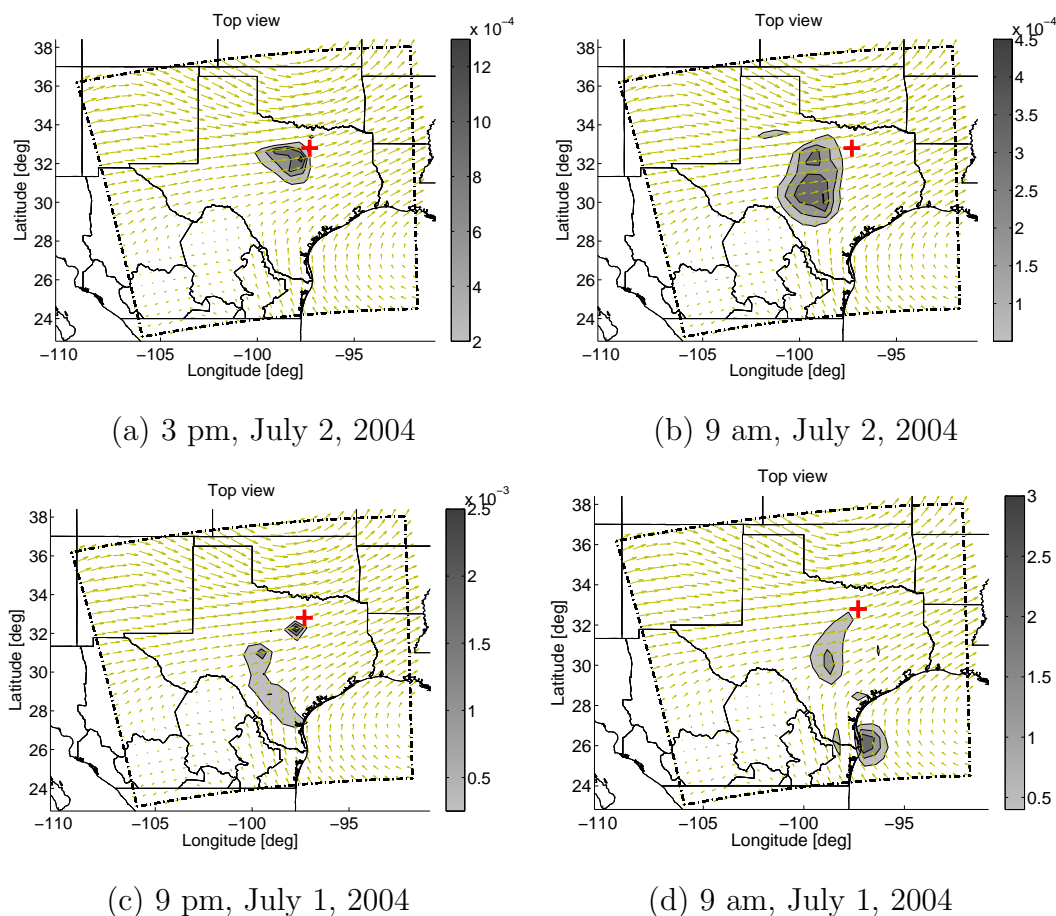


Figure 4.3: Instantaneous areas of influence of HCHO on DFW O₃ at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.

Figure 4.3 presents the instantaneous areas of influence of HCHO on DFW ozone. The adjoint variables in Figure 4.3 are scaled. A 1% change in the HCHO concentration in a specific area leads to a percent change in the DFW ozone equal to the value of the adjoint variable. The intensity of the perturbation influence increases for earlier times, presumably due to the time needed for ozone production; the ozone produced remotely is then transported to DFW. Note that a 1% change in HCHO

concentrations in the vicinity of the Texas Gulf Coast 36h before the target time yields up to 3% change in DFW ground level ozone.

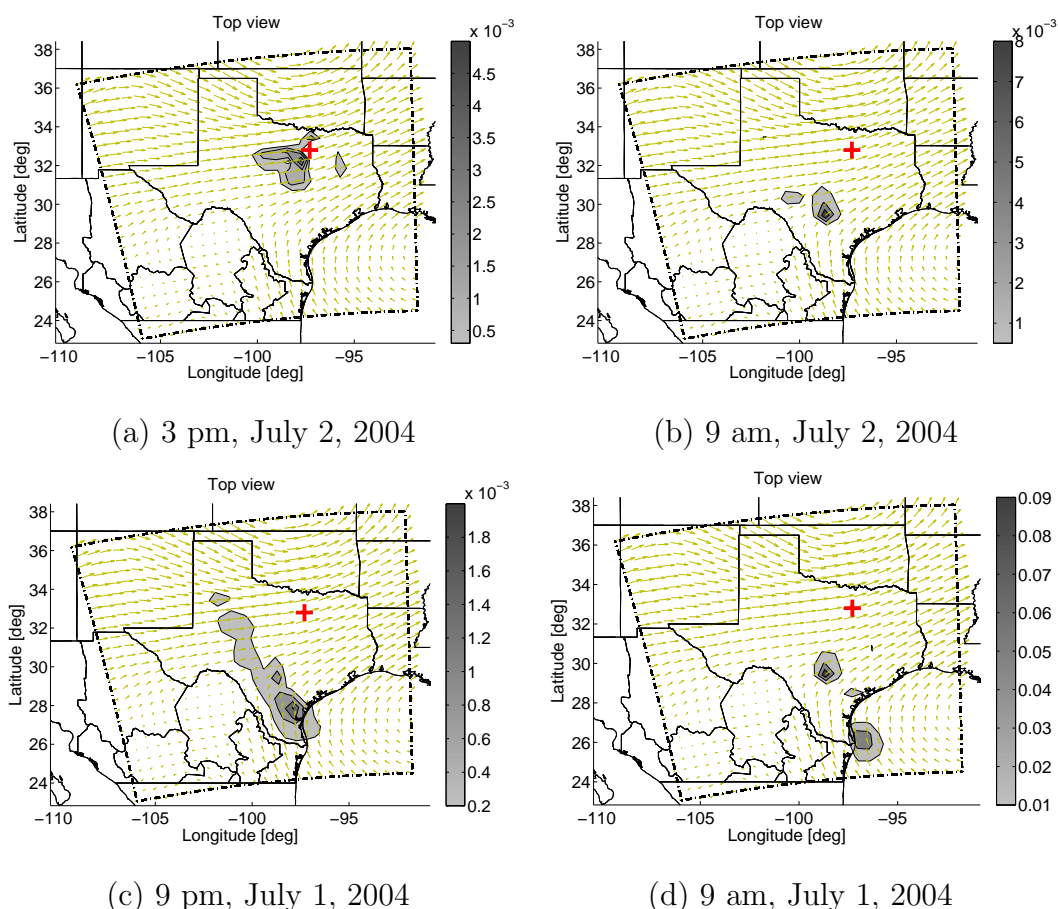


Figure 4.4: Instantaneous areas of influence of NO₂ on DFW O₃ at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.

Areas where changes in NO₂ have the maximum impact on DFW ozone are shown in Figure 4.4. A 1% change in the NO₂ concentrations in the areas of maximum influence done 6h, 12h, 24h, and 36h before the target time would result in an increase in DFW ozone by about 0.005%, 0.008%, 0.002%, and 0.09% respectively. This influence is surprisingly smaller in relative terms than that of HCHO. The areas of greatest influence at the earlier times tend to be in central Texas or on the Texas Gulf Coast.

The influence of ozone perturbations on DFW ozone at 9 pm July 2, 2004 is shown in Figure 4.5. The perturbations at earlier times tend to have a larger influence on the receptor in this scenario, presumably due to cumulative effects of

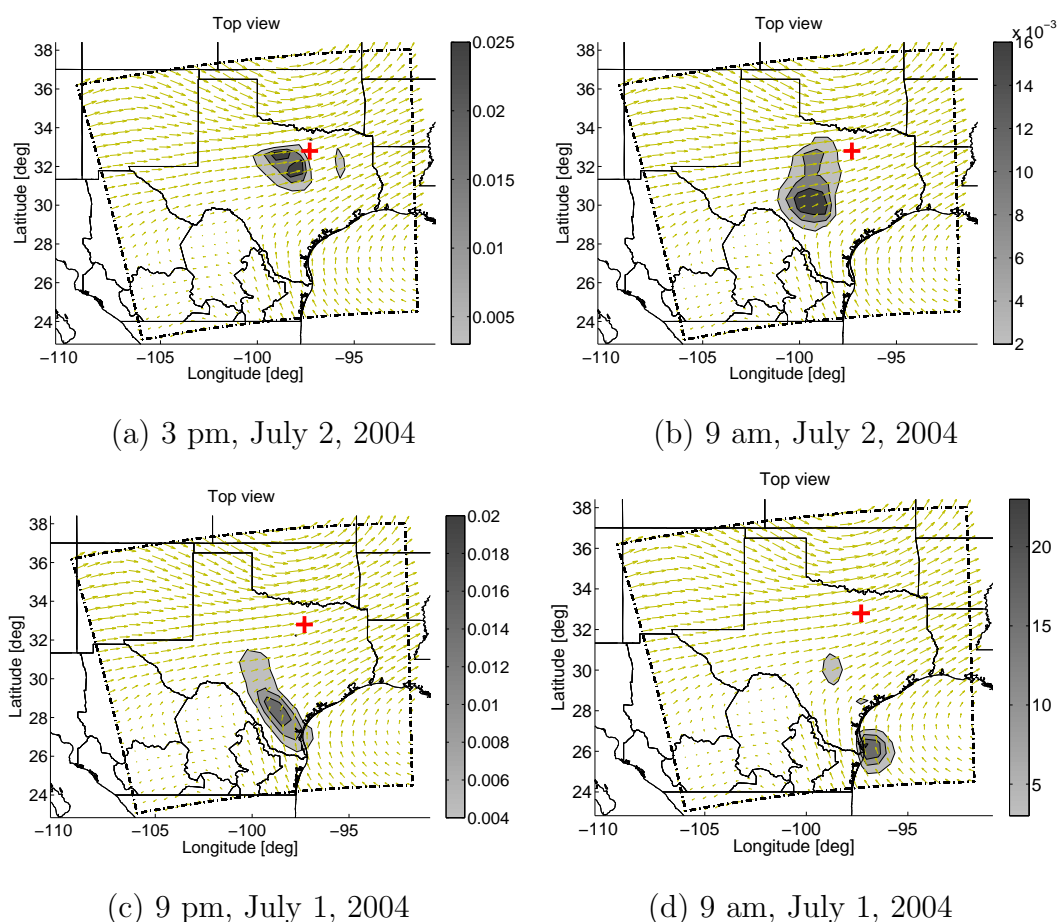


Figure 4.5: Instantaneous areas of influence of O₃ on DFW O₃ at (a) 6h, (b) 12h, (c) 24h, and (d) 36h hours before the receptor time.

subsequent changes due to chemical and transport processes.

We next present and analyze the time integrated areas of influence for each of the cases discussed above.

The integrated area of influence for the passive tracer shown in Figure 4.6 reveals the effect of meteorological conditions on July 1st (independent of the effects of chemistry). The effects trace a corridor going South over central Texas. A tracer perturbation of 1 ppb at a given location (added hourly throughout the simulation interval) would result in a perturbation of the tracer concentration at the receptor equal to the adjoint variable magnitude (gray-coded). Due to vertical mixing and transport the maximum influence region is reported above and slightly west of DFW. Note that this region is near the boundary between the free troposphere and the Planetary Boundary Layer (PBL) at around 2 km, indicating the important role of

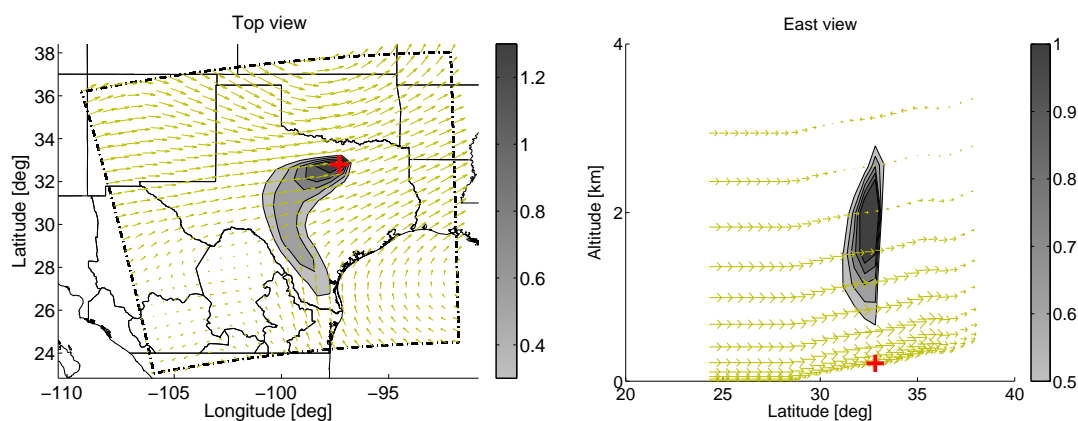


Figure 4.6: Integrated areas of influence for a passive tracer; the receptor site is ground level DFW. The 36 hours integration starts at 9 am July 1st, 2004.

subsidence.

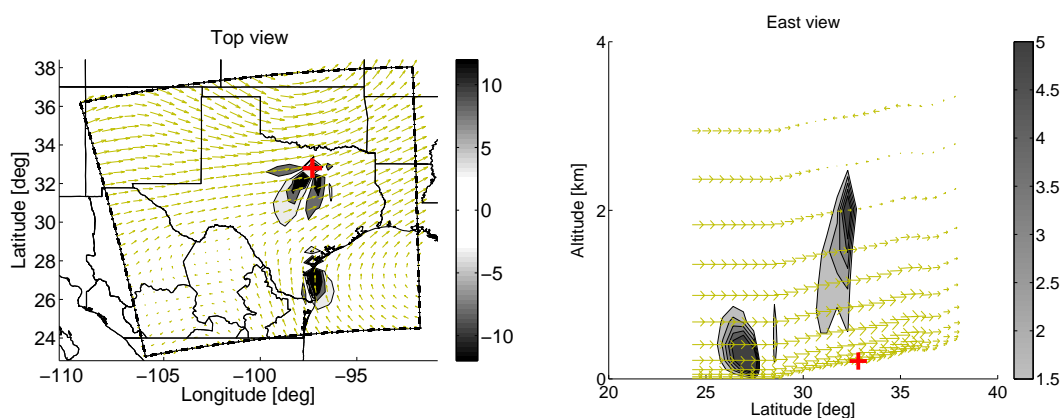


Figure 4.7: July 1, 2004. Time-integrated areas of HCHO influence on DFW O₃ concentration.

The time integrated areas of influence in Figure 4.7 represent the effect of HCHO perturbations on DFW ground level ozone. The results can be interpreted as follows. If we make a 1% change in the local HCHO concentration each hour (of the 36 hour simulation interval) the total percent change in DFW ozone at the final time equals the magnitude of the integrated adjoint variable. Thus a repeated 1% change in the HCHO near the Gulf Coast will result in an 10% change in DFW ozone (due to ozone formation the previous day followed by long-range transport); a repeated 1% change in HCHO southwest of DFW would also result in a 10% change in DFW ozone (due to the formation of ozone during the same day).

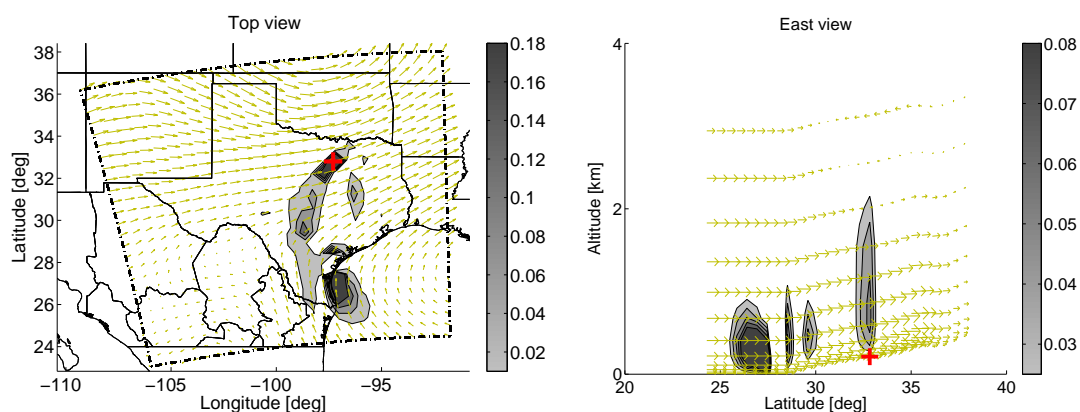


Figure 4.8: July 1, 2004. Time-integrated areas of NO₂ influence on DFW O₃ concentration.

The time integrated areas of influence in Figure 4.8 represent the effect of NO₂ perturbations on DFW ground level ozone. There are two areas of maximum influence, where 1% changes in the NO₂ concentration introduced every hour result in DFW ozone concentration changes of over 0.18%. One area is near the Gulf Coast; changes in NO₂ the previous day result in ozone formation, which is then transported to DFW. The second area is near DFW; local NO₂ changes result in O₃ changes during the same day.

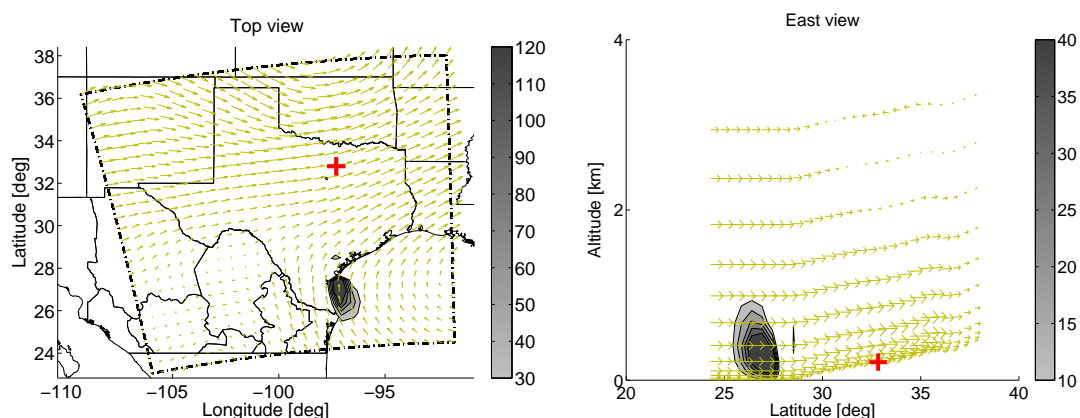


Figure 4.9: July 1, 2004. Time-integrated areas of O₃ influence on DFW O₃ concentration.

The effects of ozone perturbations on DFW ozone are illustrated in Figure 4.9. A 1% perturbation of ozone introduced every hour near the Gulf Coast would result in a 120% change in DFW ozone at the final time. This is due to chemistry in conjunction with long-range transport. The effect of changes in the Gulf area dwarfs

the ozone changes due to the local perturbations (in the DFW area) for this scenario.

4.3.2 July 25, 2004

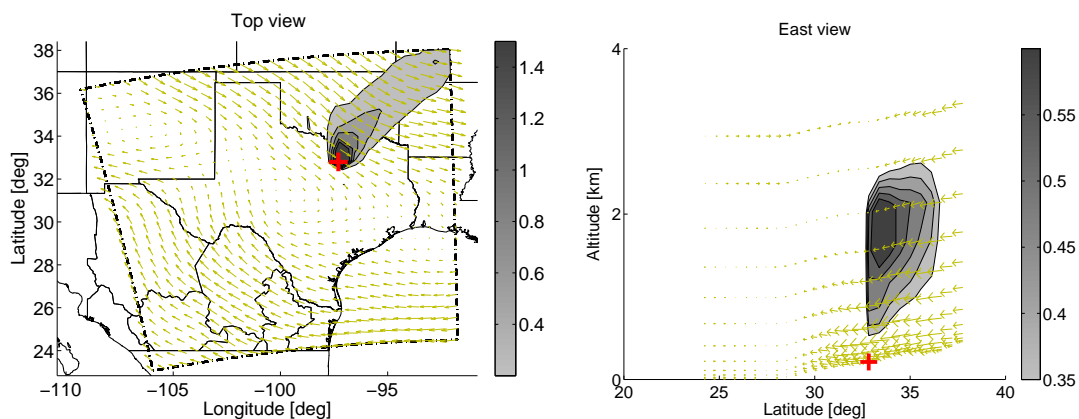


Figure 4.10: July 25, 2004. Areas of influence of the tracer on tracer DFW concentration.

The integrated area of influence for the passive tracer shown in Figure 4.10 reveals the effect of meteorological conditions on July 25 (independent of the effects of chemistry). The effects include a corridor going North-East of DFW, with a considerable local influence. The meteorological conditions are quite different from the ones on July 1, 2004.

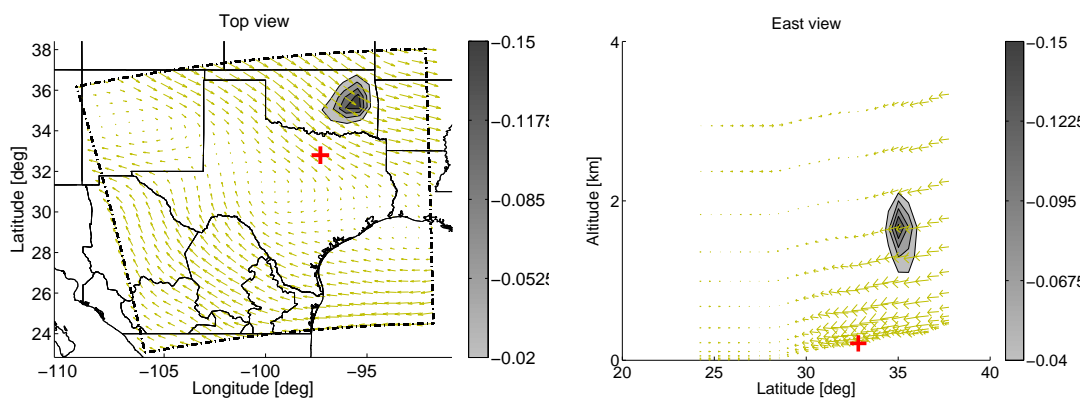


Figure 4.11: July 25, 2004. Time-integrated areas of HCHO influence on DFW O₃.

Figure 4.11 shows the integrated area of influence of the HCHO on DFW ozone. The maximum sensitivity of DFW ozone is with respect to HCHO perturbations

above central Oklahoma. A 1% change in HCHO concentration introduced every hour would result in a 0.15% reduction in the DFW ozone at the final time.

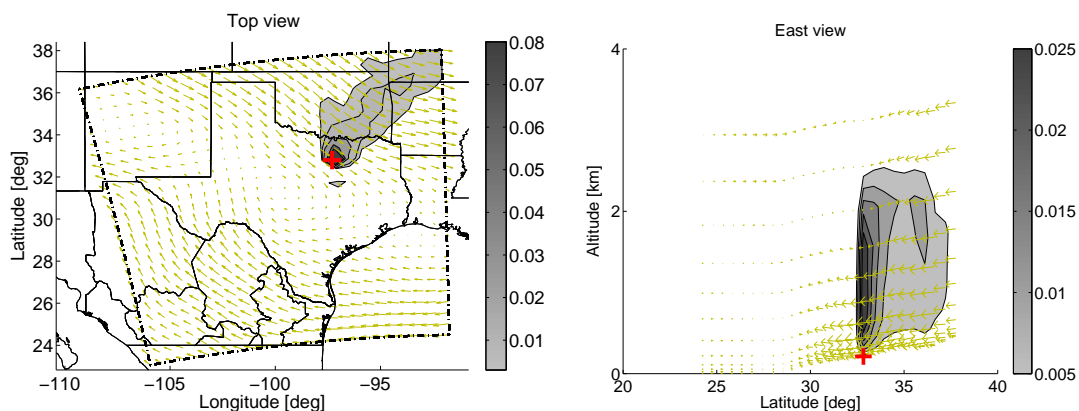


Figure 4.12: July 25, 2004. Time-integrated areas of NO₂ influence on DFW O₃.

The integrated area of influence of the NO₂ on DFW ozone reported in Figure 4.12 reveals that local NO₂ perturbations have the maximum impact. A 1% change in the NO₂ concentration introduced every hour would result in a 0.08% increase in the DFW ozone at the final time.

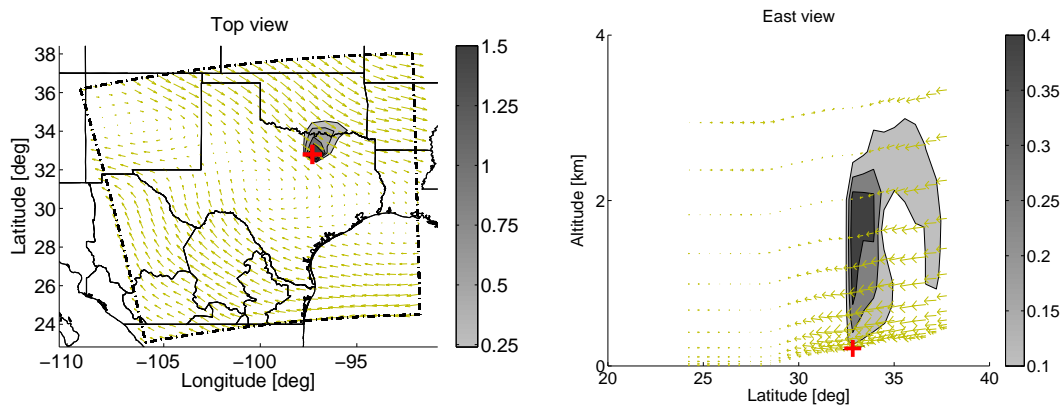


Figure 4.13: July 25, 2004. Time-integrated areas of O₃ influence on DFW O₃.

Figure 4.13 reports the influence of ozone perturbations on DFW ozone. For the July 25 episode these influences are quite localized. A 1% hourly perturbation introduced throughout the simulation interval would result in a 1% increase in the final ozone concentration.

Chapter 5

4D-Var Data Assimilation

5.1 Basic Theory

Data assimilation is the process by which measurements are used to constrain the model predictions; the information from measurements can be used to obtain better initial conditions, better boundary conditions, enhanced emission estimates, etc. Data assimilation combines information from three different sources: the physical and chemical laws of evolution (encapsulated in the model), the reality (as captured by the observations), and the current best estimate of the distribution of tracers in the atmosphere.

In this thesis we focus on obtaining optimized initial conditions. 4D-Var data assimilation can be used in STEM and is expected to improve air quality forecasting. In practice, directly measuring the parameters of the atmospheric conditions in large range is difficult because of sampling, technical and resource requirements. And due to the complexity of the chemical transport model, the number of possible state variables is extremely large. Data assimilation enables data acquisition for fields and parameter estimates, even though enough observations are still needed to fulfill the process.

Figure 5.1 (Adrian Sandu, personal communication, Feb, 2007) presents information feedback flows between CTMs, observations and data assimilation. CTMs are complex systems incorporating transport models, chemical models, aerosol models and radiation. They use information from meteorological simulations like wind

and temperature fields, turbulent diffusion parameterizations, etc. They also use information from emission inventories to produce chemical weather forecasts, i.e. forecasts of the distribution of chemical pollutants in the atmosphere. Another source of information for concentrations of pollutants in the atmosphere is observations ground stations, ozone balloons, plane flights, satellite observations. Data assimilation combines these two sources of information, plus background knowledge according to prior experience, to improve environmental policy decisions.

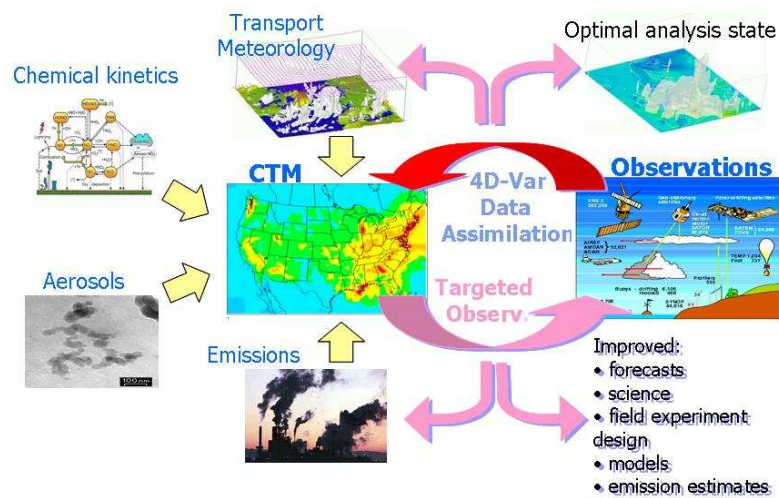


Figure 5.1: Information feedback flows between CTMs, observations and data assimilation.

We applied 4D-Var data assimilation to obtain the optimal initial conditions using STEM by minimizing a cost function that measures the misfit between model predictions and observations, as well as the deviation of the solution from the background state. The cost function is formulated as

$$\min \mathcal{J}(c^0) = \frac{1}{2} (c^0 - c^B)^T B^{-1} (c^0 - c^B) + \frac{1}{2} \sum_{k=0}^N (H_k c^k - c_{obs}^k)^T R_k^{-1} (H_k c^k - c_{obs}^k) \quad (5.1)$$

and our goal is to minimize the cost function \mathcal{J} . In the above formula, c^B represents the ‘a priori’ estimate (background) of the initial values and B is associated covariance matrix of the estimated background error. H_k is an observation operator and c_{obs}^k is the real observations depending on time k . The covariance matrix R_k^{-1} accounts for observations and representativeness errors.

Note that the gradient of the cost function 5.1 with respect to the initial values reads

$$\nabla_{c^0} \mathcal{J}(p, c^0) = B^{-1}(c^0 - c^B) + \sum_{k=0}^N \left(\frac{\partial c^k}{\partial c^0} \right)^T H_k^T R_k^{-1} (H_k c^k - c_{obs}^k) \quad (5.2)$$

If the calculation of this gradient is done via direct sensitivity analysis it requires the computation of the solution derivatives with respect to all n components of the initial state:

$$\frac{\partial c^k}{\partial c^0} = \left[\frac{\partial c^k}{\partial c_1^0}, \frac{\partial c^k}{\partial c_2^0}, \dots, \frac{\partial c^k}{\partial c_n^0} \right] = [S_1^k, S_2^k, \dots, S_n^k] \quad (5.3)$$

Each S_i^k is obtained by a different forward integration of the tangent linear model with a different initial condition. For realistic CTMs this approach would require to run the tangent linear model $n \approx 10^7$ times, an intractable task.

However, the same gradient can be computed efficiently by a single backward integration with the adjoint model as follows:

$$\begin{aligned} \lambda^N &= H_N^T R_N^{-1} (H_N c^N - c_{obs}^N) \\ \text{for } k &= N-1, \dots, 0 \text{ do} \\ \lambda^k &= \left(\frac{\partial c^{k+1}}{\partial c^k} \right)^T \lambda^{k+1} + H_k^T R_k^{-1} (H_k c^k - c_{obs}^k) \\ \text{end} \\ \nabla_{c^0} \mathcal{J}(p, c^0) &= B^{-1}(c^0 - c^B) + \lambda^0 \end{aligned}$$

The 4D-Var data assimilation is implemented by an iterative optimization procedure: each iteration requires STEM to run a forward integration to obtain the value of the cost function and an adjoint model to evaluate the gradient of the cost function. Since model states are as high as 10^7 in our air quality simulation problem, it is prohibitive to evaluate Hessian of the cost function. In the following chapters, we will start the discussion with some optimization approaches that only need the first order derivatives of the cost function, which can be given by first adjoint model, and then move on to other methods requiring second order information, which is computed by second adjoint model.

One way of measuring the performance of optimization method is through Root Mean Square (*RMS*) and R^2 correlation factor between observations and model predictions. The better model predictions fit the observations, the smaller the value

of RMS . R^2 ranges from 0 to 1. If model forecasts match with the observations very well, R^2 is close to 1. The RMS and R^2 of two data sets X and Y (X, Y are vectors of length n) are computed as follows:

$$RMS(X, Y) = \sqrt{\frac{1}{n} \sum_{i=1}^n (X - Y)^2} \quad (5.4)$$

$$R^2(X, Y) = \frac{(n \sum_{i=1}^n X_i Y_i - \sum_{i=1}^n X_i \sum_{i=1}^n Y_i)^2}{\left(n \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2\right) \left(n \sum_{i=1}^n Y_i^2 - (\sum_{i=1}^n Y_i)^2\right)} \quad (5.5)$$

In our case, X and Y represent model predictions and real observations.

5.2 Applications of 4D-Var Data Assimilation in STEM

5.2.1 1st Test Case - Texas

We have applied data assimilation using STEM to ozone predictions over Texas in July 2004.

Domain

The simulation domain covers the entire state of Texas and parts of its surrounding states of United States, ranging from $92^\circ W$ to $108^\circ W$ in longitude, and from $28^\circ N$ to $32^\circ N$ in latitude. The 3D computational domain includes 26 by 26 by 21 grid-points with 60 Km by 60 Km horizontal resolution and variable vertical resolutions. The simulation time is in July of 2004 when observation data is available for data assimilation, and dynamical time step is 15 minutes.

Data Sets

Two data sets have been used for assimilation. The first employs ground level ozone measurements from AirNow stations. These stations provide hourly observations of ground level ozone throughout the entire month of July 2004. Figure 5.2 shows the locations of the AirNow stations. Figure 5.3 gives locations of four selected stations.

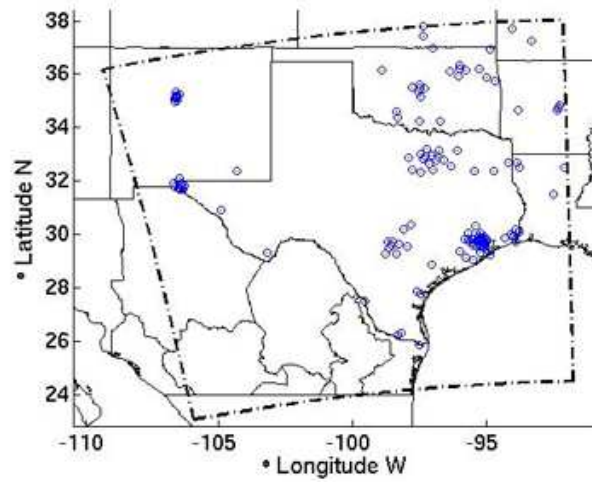


Figure 5.2: The location of AirNow stations used in data assimilation experiments.

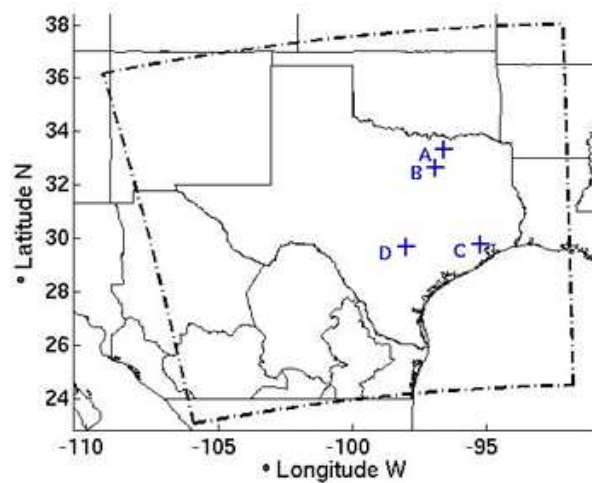


Figure 5.3: Four selected stations where O₃ time series are considered.

The second data set consists of NO₂ and HCHO total columns observed by the SCHIAMACHY instrument on board the ENVISAT satellite (<http://www.esa.int/envisat/instruments.html>). SCHIAMACHY is an imaging spectrometer designed to detect many pollutants by measuring the emitted, reflected and backscattered infrared radiation in the atmosphere. The SCHIAMACHY NO₂ and HCHO total column data are aggregated for two hours in each 24 hour interval. The locations of SCHIAMACHY observed columns are shown in Figure 5.4 (a) and (b). Circles represent SCHIAMACHY measured columns.

An issue that requires special consideration is the use of the satellite averaging kernel in the construction of the observation operator for satellite data. The SCHIAMACHY averaging kernel values decrease from 1 (near the top layer) to about 0.25-0.75 near the ground. The column-integrated NO₂ value is obtained using an approximate averaging kernel function for the SCHIAMACHY data

$$Column(c) = \int_{ground}^{top} c(z)A(z)dz \approx \sum_{k=1}^{Nlev} c(z_k)A(z_k)dz_k \quad (5.6)$$

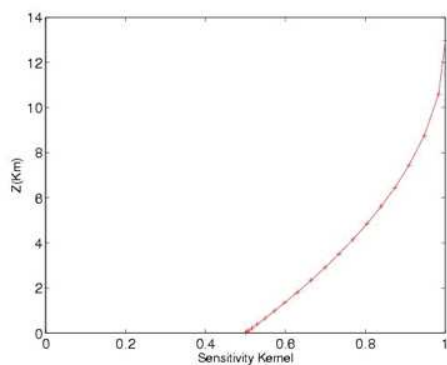
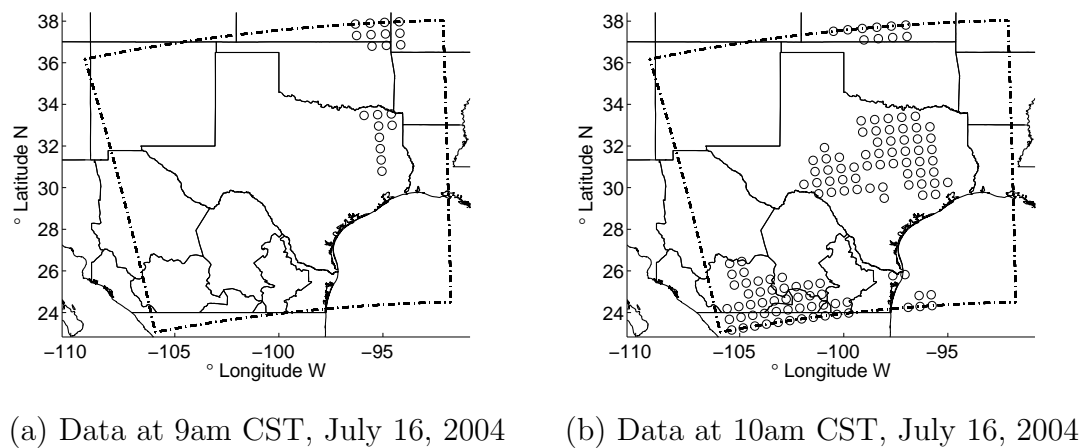
The averaging kernel is approximated by the following quadratic polynomial that takes the value 1 at the highest modeling level and the value 0.5 at the ground level, as seen in Figure 5.4(c).

$$A(z) = -0.5 \left(\frac{z - z_{ground}}{z_{top} - z_{ground}} \right)^2 + \frac{z - z_{ground}}{z_{top} - z_{ground}} + 0.5 \quad (5.7)$$

Experiment Results

1. July 01

We consider the 24-hour simulation starting at 4am CST on July 1, 2004. Data assimilation uses the AirNow ground level observations in the time window 4am July 1 to 4am July 2, 2004. The change in the ground level ozone fields at 6pm CST of July 1 are shown in Figure 5.5. The colored circles represent the AirNow stations and their measured values. Background color represents model predicted values. The unit of the value is Parts Per Billion (ppb). Visually there is a better agreement between the model and observations after assimilation. This is confirmed by the scatter and quantile-quantile plots of



(c) SCHIAMACHY averaging kernel is approximated by a quadratic polynomial

Figure 5.4: (a) and (b) The location of SCHIAMACHY total NO₂ column measurements on July 16, 2004. (c) The approximation of the averaging kernel used in the construction of the observation operator in data assimilation.

Figure 5.6, which indicate that the correlation coefficient between model and observations increased considerably from $R^2 = 0.36$ for the original model to $R^2 = 0.74$ after assimilation.

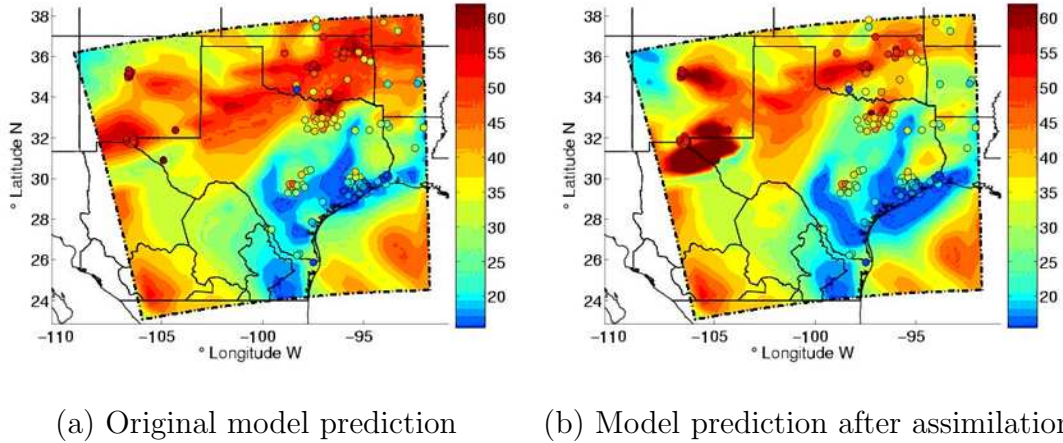


Figure 5.5: Ground level ozone distribution in Texas at 6pm CST July 1st, 2004 (a) before data assimilation, and (b) after data assimilation.

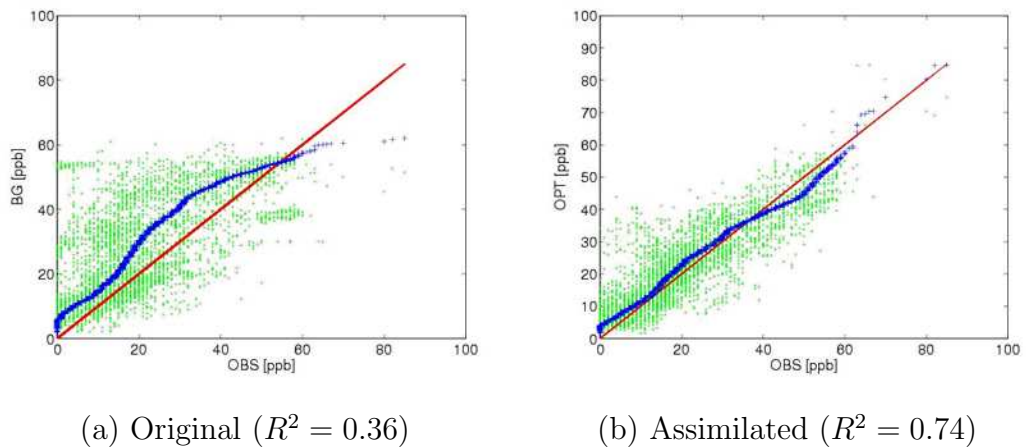
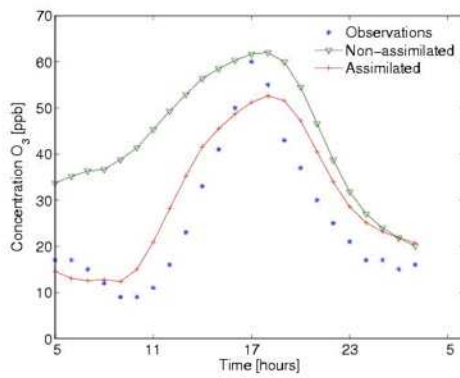
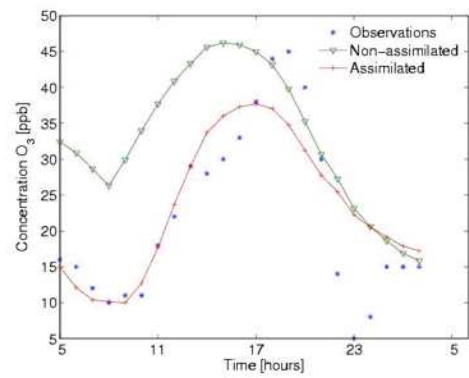


Figure 5.6: Scatter plot and quantile-quantile plot of model predictions versus observations (a) for the original model predictions before data assimilation, and (b) after data assimilation.

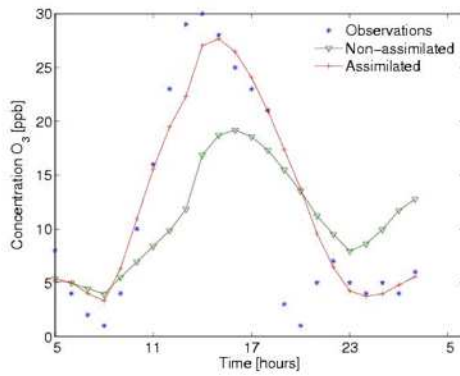
To show time series predictions we choose four AirNow stations: two in DFW area, one in Houston area and one in central Texas. Their locations are shown in Figure 5.3. The ozone time series at these four stations are illustrated in Figure 5.7. It is obvious that the assimilated data is closer to the observations than was initially predicted, which also indicate the improvements in model predictions after assimilation.



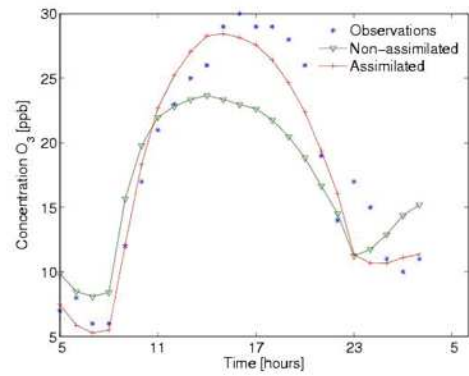
(a) O3 at station A (DFW area)



(b) O3 at station B (DFW area)



(c) O3 at station C (Houston area)



(d) O3 at station D (central Texas)

Figure 5.7: Time series of ozone concentrations on July 1st, 2004.

2. July 16

Next we consider the 24-hour simulation starting at 4 am CST on July 16, 2004. For assimilation we use both the AirNow ground level observations between 4 am July 16 and 4 am July 17, 2004 and the SCHIAMACHY NO₂ and HCHO observations at 9 am and 10 am July 16, 2004. These data sets were assimilated sequentially: first we found the optimum using AirNow data only, then we assimilated the SCHIAMACHY data using the optimized initial conditions as the new background field.

The correlation coefficient between model-predicted ozone and AirNow ozone observations increased from $R^2 = 0.37$ for the original model to $R^2 = 0.69$ after assimilation. The correlation between model-predicted NO₂ and the SCHIAMACHY NO₂ observations increased from $R^2 = 0.19$ for the original model to $R^2 = 0.36$ after assimilation. Finally, the correlation between model-predicted HCHO and the SCHIAMACHY HCHO observations decreased from $R^2 = 0.19$ for the original model to $R^2 = 0.11$ after assimilation. The results can be found in Table 5.1.

The quality of the SCHIAMACHY HCHO columns may need to be re-evaluated, as a possible cause for the disagreement between the model and this data set.

	R^2 Before Assimilation	R^2 After Assimilation
AirNow O ₃ obs	0.37	0.69
SCHIAMACHY NO ₂ obs	0.19	0.36
SCHIAMACHY HCHO obs	0.19	0.11

Table 5.1: Correlation coefficient between model prediction and observations.

5.2.2 2nd Test Case - Northeastern United States

Domain

For all the data assimilation experiments in the following, we will use the same domain as shown in this case. The test case is a real-life simulation of air pollution

in North–Eastern United States in July 2004 as shown in Figure 5.8 (the dash-dotted line delimits the computational domain). The computational domain covers $1500 \times 1320 \times 20$ Km with a horizontal resolution of 60×60 Km and a variable vertical resolution (resulting in a 3-dimensional computational grid of $25 \times 22 \times 21$ points). Real data is used for the initial concentrations, meteorological fields, boundary values, and emission rates starting at 0 GMT of July 20th, 2004. This data corresponds to the ICARTT (International Consortium for Atmospheric Research on Transport and Transformation) [32] campaign in July 2004.

Data Sets

The observations used in this paper for data assimilation are real ozone (O_3) measurements taken during the ICARTT [32] campaign in summer 2004. Ground level ozone measurements are provided hourly by the EPA’s AirNow network of ground stations (340 in total) whose locations are shown in Figure 5.8(a). Elevated ozone measurements are taken by two ozonesondes and a P3-B flight, all shown in Figure 5.8(b). More ozone observations are available from two Mozaic flights. The rich data set of observations make this period well suited for data assimilation task and ideal to provide better analysis using both measurements and model results, so as to investigate the potential to improve air quality forecast in the future.

Experiment Results

We performed data assimilation using LBFSGS method to obtain optimized initial conditions, the same as what we did for the first test case. The simulation interval is from 8am EDT to 8pm EDT on July 20, 2004. The changes in the ground level ozone fields at 1pm EDT of July 20 are shown in Figure 5.9. Visually there is a better agreement between model and observations after assimilation. This is also confirmed by the scatter plots of Figure 5.10, which indicates that the correlation coefficient between model and observations increases considerably from $R^2 = 0.15$ for the original model to $R^2 = 0.68$ after assimilation.

To show the change of ozone in this time interval at one location, we choose four out of all AirNow stations (A-D) as shown in Figure 5.8(b). The ozone time series

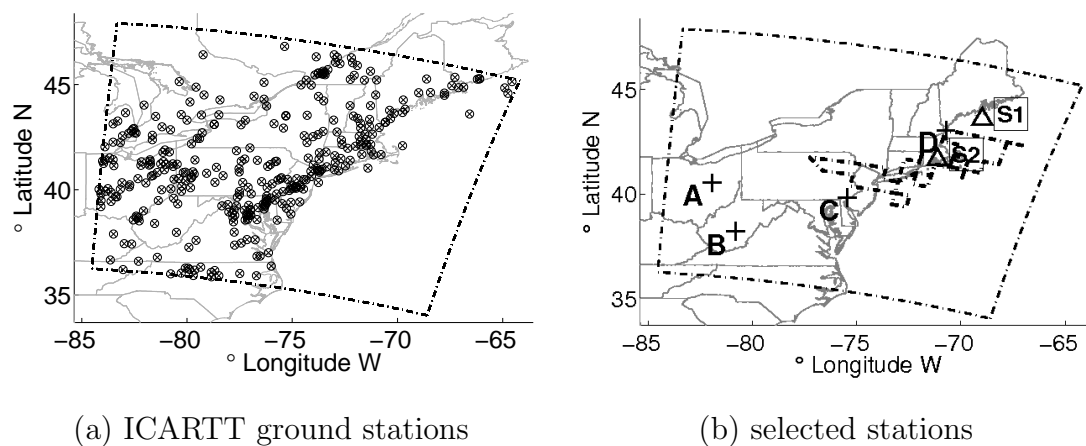


Figure 5.8: (a) The location of the ground measuring stations in support of the ICARTT campaign (340 in total) (b) The location of the two ozonesondes (S1, S2) and the path of the P3-B flight that provide observations used in data assimilation. Also shown are the locations of four selected stations (A–D) that will be used to illustrate the assimilation results.

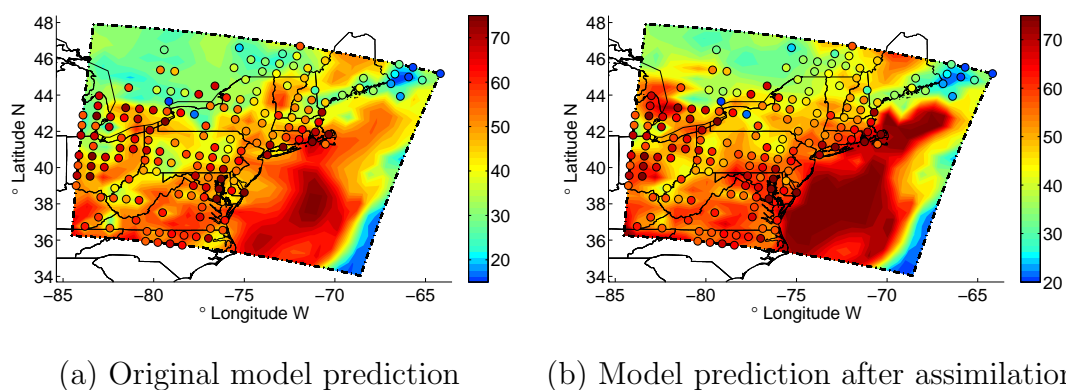


Figure 5.9: Ground level ozone distribution in northeastern U.S. at 1pm EDT July 20, 2004 (a) before data assimilation, and (b) after data assimilation.

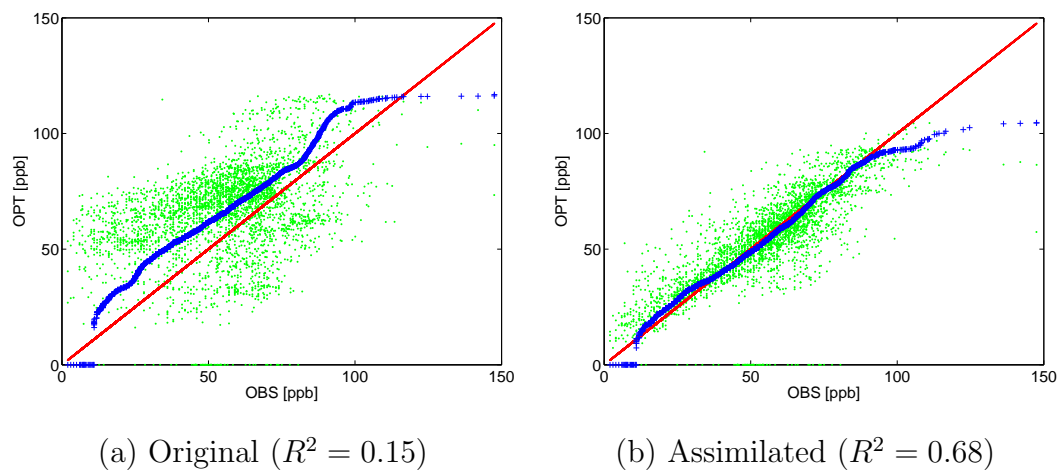


Figure 5.10: Scatter plot and quantile-quantile plot of model-observations agreement.

(in EDT) at these four stations are illustrated in Figure 5.11. From the figure, we can find that the assimilated lines are closer to observations than non-assimilated lines, which indicates the improvement in model predictions after data assimilation.

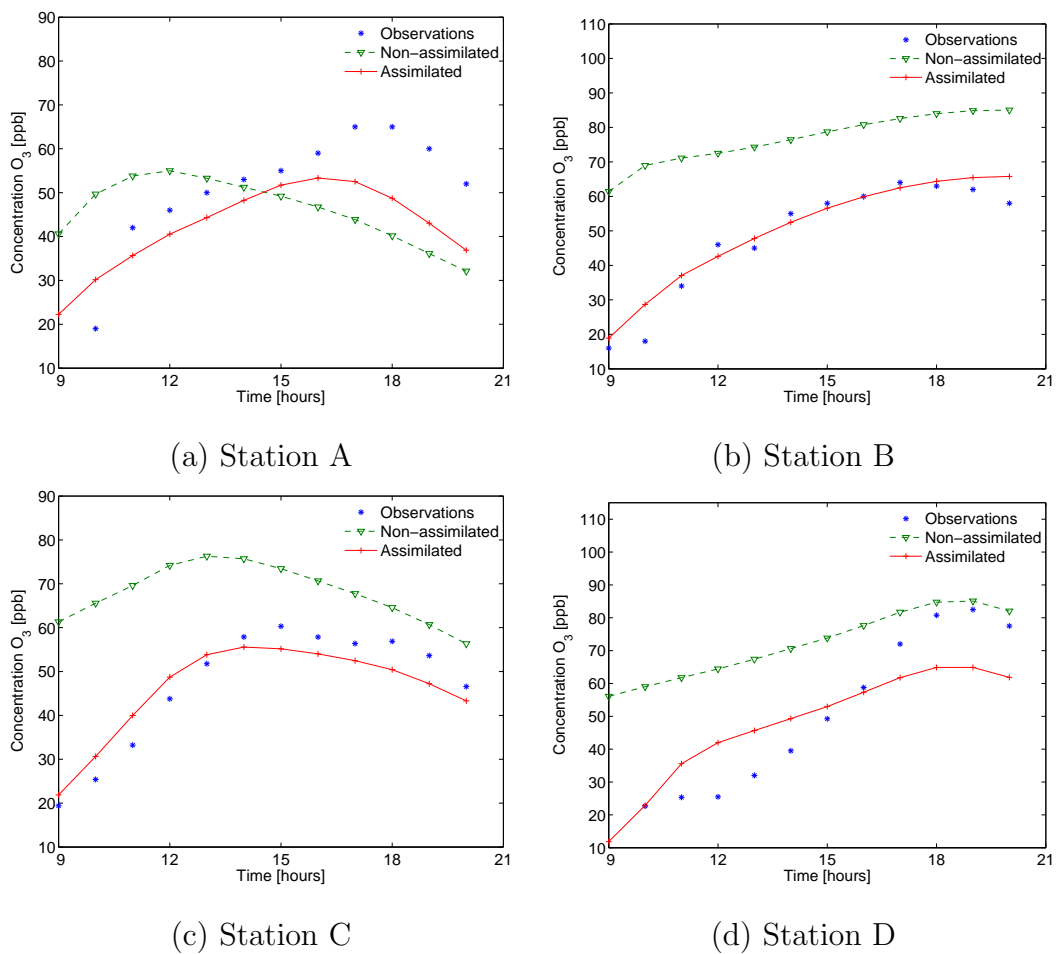


Figure 5.11: Time series of ozone concentrations.

Chapter 6

Optimization Using First Order Adjoint

6.1 Optimization Methods

Data assimilation relies on an efficient optimization method to solve the problem. Some optimization algorithms only require the first derivatives of a cost function, which can be provided by first adjoint model. We have applied three such optimization methods to STEM chemical transport model to acquire optimized initial conditions. Our goal is to assess the performance of these methods in different situations, to find out the most efficient one for the STEM system. These methods are L-BFGS, Fletcher-Reeves Conjugate Gradient, and Hessian Free Newton method. L-BFGS is a limited memory quasi-Newton method for large-scale optimization problems, which approximates the Hessian of second derivatives of an objective function by analyzing gradient vectors. Nonlinear conjugate gradient makes up another popular class of large-scale optimization. The basic idea is to avoid matrix operations and simply compute the search directions recursively. Hessian Free Newton is an inexact Newton method in which the product of Hessian times a vector is expressed by automatic differentiation or approximated by finite difference.

The performances of these methods are problem dependent. For our STEM model, L-BFGS shows its advantage in fast convergence in all of the three scenarios we designed. Therefore, we conclude that among these methods L-BFGS best fits

the STEM.

6.1.1 L-BFGS

Limited-memory Broyden Fletcher Goldfarb Shanno method (L-BFGS) ([46] and [40]) is a limited-memory technique for large scale optimization, capable of solving problems with simple bounds on the variables. L-BFGS is based on Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, which is an application of quasi-Newton approach. The principle idea for quasi-Newton is to approximate the Hessian matrix of the cost function G by a symmetric positive definite matrix H , and update H at each step. The BFGS method gives an expression to update H , and the algorithm goes as follows:

- 1) Compute Search directions p_k by solving: $H_k p_k = -\nabla f_k$
- 2) Perform line search to find the optimal α_k along the direction p_k and update the solution:

$$x_{k+1} = x_k + \alpha_k p_k$$

- 3) Compute increments of variables and gradients:

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f_{k+1} - \nabla f_k$$

- 4) Update H_{k+1}

$$H_{k+1} = H_k + \frac{(y_k y_k^T)}{(y_k^T s_k)} - \frac{(H_k s_k s_k^T H_k)}{(s_k^T H_k s_k)}$$

Quasi-Newton is a class of methods containing different ways to approximate the Hessian H . The fourth step shows how BFGS method updates H and this satisfies the quasi-Newton condition:

$$H_{k+1}(x_{k+1} - x_k) = -(\nabla f(x_{k+1}) - \nabla f(x_k))$$

The advantage of BFGS are: (1) the user of the algorithm does not need to provide $\nabla^2 f(x)$. (2) this algorithm ensures positive definiteness of Hessian approximation.

BFGS stores H_0 (initial approximation) and all pairs of $\{(s_k, y_k)\}$ to recover H_{k+1} . However, for CTMs with millions of model states such as the STEM, the

storage is prohibitive. Instead of keeping all the s and y from the past iterations, L-BFGS updates the Hessian using the information from the m most recent iterations, where m is given by the end-user.

6.1.2 Nonlinear Conjugate Gradient

Linear conjugate gradient method is an iterative method for solving a linear system $Ax = b$, where A is an n by n symmetric positive definite matrix. It is equivalent to minimizing convex quadratic function:

$$\phi(x) = \frac{1}{2}x^T Ax - b^T x ,$$

and the gradient of ϕ equals the residual of the linear system, $\nabla\phi(x) = Ax - b = r$. This gives us hints on using conjugate gradient method to solve nonlinear problems.

The conjugate gradient method generates a set of search directions $\{p_0, p_1, \dots, p_m\}$ conjugating with respect to the symmetric positive definite matrix A in the sense that $p_i^T A p_j = 0$ for $i \neq j$. With this property, we can minimize the residual $r = Ax - b$ in n steps by successively decreasing it along the directions in the conjugate set.

Linear conjugate gradient is easy to be extended to solve nonlinear problems. We need to perform a line search that identifies an approximate minimum of the nonlinear function f along search direction p_k , and the residual r is replaced with the gradient of nonlinear objective function f . The Fletcher-Reeves Conjugate Gradient (FR-CG) method is shown as follows [47]:

1) Initialization:

given x_0 , evaluate $f_0 = f(x_0)$, $\nabla f_0 = \nabla f(x_0)$, set $p_0 = -\nabla f_0$ and $k = 0$.

2) If $\nabla f_k \neq 0$, Compute α_k and set $x_{k+1} = x_k + \alpha_k p_k$

3) Evaluate ∇f_{k+1}

4) Define $\beta_{k+1}^{FR} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$

5) Update $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k$

6) $k = k + 1$, go to 2).

The Polak-Ribiere (PR) method and the Positive Polak-Ribiere (PR+) are two variants of FR-CG method. The difference lies in the update of the parameter β_k .

For PR-CG,

$$\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}$$

Positive Polak-Ribiere method is proposed to guarantee that p_k is always a descent direction by defining $\beta_{k+1}^+ = \max(\beta_{k+1}^{PR}, 0)$.

6.1.3 Hessian Free Newton

Like L-BFGS, Hessian Free Newton (HFN) method is said to be one of the most effective algorithms for unconstrained large-scale minimization problems by Morales and Nocedal [44]. Hessian Free Newton is an inexact Newton method and only requires first derivatives of the objective cost function. The inexact Newton method doesn't require explicit knowledge of the Hessian matrix, but needs matrix-vector product for any given vector. This fact is very useful for users who cannot easily calculate second derivatives, or where the Hessian matrix takes too much storage. We can use automatic differentiation or finite difference to approximate the Hessian times a vector. This method is known as Hessian Free Newton (HFN) method and is often used when the Hessian is not available.

The finite difference is carried out by evaluating the gradient of the objective function with respect to solution twice. Let $g = \nabla f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be the gradient, G the Hessian and ε a small perturbation factor, we have

$$G \cdot v = \nabla^2 f(y) \cdot v \approx \frac{g(y + \varepsilon v) - g(y)}{\varepsilon} \quad (6.1)$$

6.1.4 Hybrid Method

In this research we also consider a hybrid optimization method [44] which interlaces L-BFGS and HFN iterations and uses the information collected by one type of iteration to improve the performance of the other. In the entire process it calls l steps of L-BFGS followed by t steps of the HFN method and then repeat. During L-BFGS, the matrix obtained at the end of the cycle is used for preconditioning the first loop of HFN iterations. During the remaining $t-1$ HFN iterations, the H is updated using information generated by the inner preconditioned CG iterations and is used to precondition the next HFN iteration. After HFN finishes, the most

current matrix H is used as initial approximated Hessian matrix in the new cycle of L-BFGS.

$$l * (L - BFGS) \xrightarrow{H} t * (HFN(PCG)) \xrightarrow{H} repeat \quad (6.2)$$

6.2 Experiments and Results

The cost function is defined as the sum of background term and misfit between observations and model predictions. The above numerical methods are used to minimize this cost function to obtain optimized initial concentrations. The STEM is utilized to evaluate the cost function by forward model and gradients of the cost function by adjoint model. We compared the performance of each method in different scenarios:

1. When cost function converges fast (using AR background).
2. When cost function converges slow (using NMC background).
3. When cost function is already small (using Artificial Data).

For these scenarios, we apply the same simulation time, i.e., 12 hours from 8 EDT of July 20 to 20 EDT of July 20, 2004 and the same domain over Northeastern United States.

Efficiency of an optimization method can be assessed from two aspects:

a) the decrease of the cost function in each optimization iteration (i.e., how much the cost function can reduce after the program reaches a new feasible point x). We call such new point new x . If every time when there is a new x , the cost function evaluated at this point decreases much, then this approach is what we expect.

However, this cannot guarantee that the method is efficient. In order to generate a new x , the program might call the model more than once due to some sub-technique like line search. For example, consider two optimization methods: one can generate a new x that decreases the cost function a lot, but the model has to run 10 times to reach this point. The other one also finds a new x , although the cost function does not go down as much as the first method, one model run can find the new

x. Therefore, we must also consider the number of model runs, especially when the model simulation takes more time than optimization subroutine itself. Usually in large-scale optimization problems, time for model simulations, rather than the optimization process itself, is dominant of the total time.

b) number of model runs for cost function to converge. For our experiment, each model run (forward and backward) takes about one and half an hours. Therefore, an optimizing method is considered to be efficient if it requires small number of model runs when it converges.

6.2.1 AR Background

The background term counts for an important part of cost function (see in 5.1). It is well known that a good representative background covariance matrix is essential for improving the fit of results in data assimilation. Constantinescu et al. proposed an autoregressive(AR) model for building background errors [10]. In this work we choose this AR background, as well as the National Meteorological Center's (NMC) model as background [50], and run the above optimization methods. Our experiments reveal that NMC background usually costs the optimization process twice or even more time to converge than AR background. We would like to assess the performance of optimization methods under these two different backgrounds.

Compared with the traditional diagonal background and NMC background model, the advantage of AR background is fast convergence. The comparison of the convergence speed of these three backgrounds using L-BFGS can be found in Constantinescu's paper [10]. In my thesis, I also use AR background and NMC background, but I focus on the performance of optimization methods when solving a cost function with different background rather than evaluating background models.

In 4D-Var data assimilation, the cost function is formulated as

$$\min \mathcal{J}(c^0) = \frac{1}{2} (c^0 - c^B)^T B^{-1} (c^0 - c^B) + \frac{1}{2} \sum_{k=0}^N (H_k c^k - c_{obs}^k)^T R_k^{-1} (H_k c^k - c_{obs}^k)$$

It is shown in [10] the $n \times n$ background error covariance matrix is

$$B = A^{-1} \sum^2 A^{-T}$$

so $B^{-1} = A^T \Sigma^{-2} A$. The 4D-Var cost function can be computed as

$$z = \Sigma^{-1} A ((c - c^B)) ,$$

$$\mathcal{J}(c^0) = \frac{1}{2} z^T z + \frac{1}{2} \sum_{k=0}^N (H_k c^k - c_{obs}^k)^T R_k^{-1} (H_k c^k - c_{obs}^k)$$

The AR model is particularly advantageous in the 4D-Var context where the evaluation of the background term in the cost function only requires one matrix-vector multiplication by the AR coefficient matrix A , and one component-wise scaling (multiplication by the diagonal matrix Σ^{-1}) [10].

Results

1. Compare FR-CG and PR-CG

Generally speaking, L-BFGS is superior to nonlinear CG method for most large-scale optimization problems. We first compare two Conjugate Gradient (CG) methods: Fletcher-Reeves CG (FR-CG), Polak-Ribiere CG (PR-CG) and would like to choose the better one for further comparison.

The two CG algorithms behave nearly the same in terms of obtaining new x in each iteration as seen in Figure 6.1.(a). Difference lies in that FR-CG needs less model runs than PR-CG in number of model runs. Therefore, in following we choose FR-CG to take part comparison with L-BFGS and HFN in different scenarios. By default CG will be referred as FR-CG method in this chapter.

2. Compare L-BFGS and hybrid

It is reported that a hybrid method (combination of L-BFGS and HFN) is more efficient than L-BFGS and HFN themselves [44]. We tried to run the hybrid method first and compared the results with L-BFGS. However, in our STEM model, it didn't show much advantage. Therefore, we now focus on comparing performance of L-BFGS, CG and HFN.

3. Compare L-BFGS, CG and HFN

These methods are tested respectively to optimize initial concentrations. They all start at the same cost function of around 54800 and converge at about

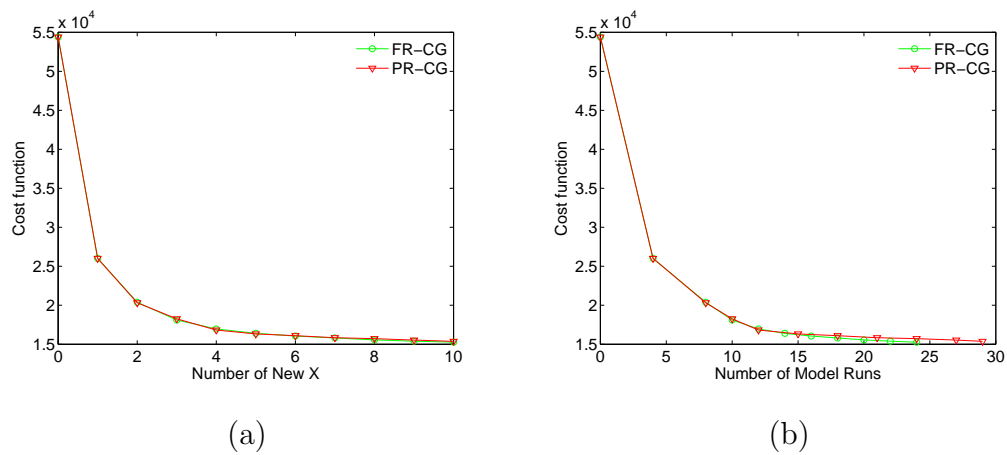


Figure 6.1: Decrease of the cost function using AR background versus each iteration and the number of model runs for two CG methods.

16000 within 10 iterations. The difference lies in the number of model runs when they converge. For one model run, the STEM calls forward model and adjoint model to evaluate value and gradient of the cost function, required by optimization subroutine. The more model runs, the more time is needed in optimization. Figure 6.2 shows performance of these methods in terms of model runs they required. It is obvious that L-BFGS converges the fastest. We can conclude that of the three optimization methods L-BFGS is the best for data assimilation in STEM.

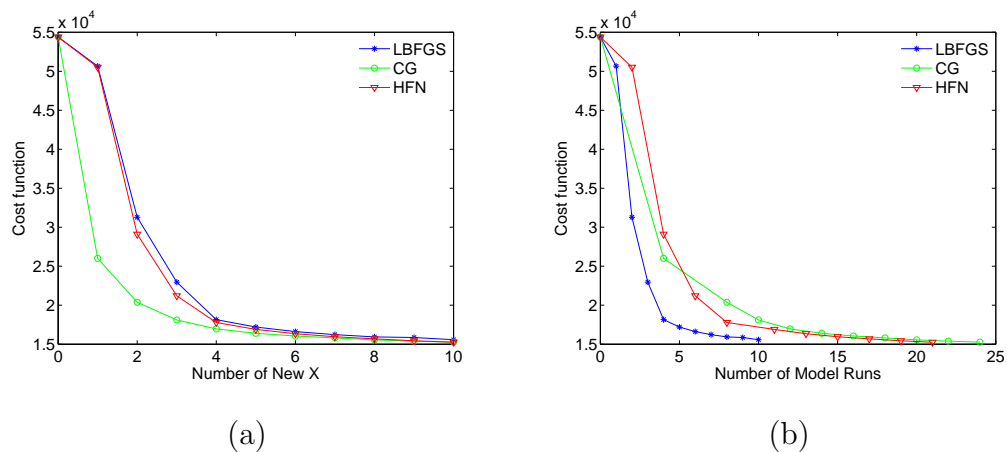


Figure 6.2: Decrease of the cost function using AR background versus each iteration and the number of model runs for CG, L-BFGS and HFN.

The scatter and quantile-quantile plot 6.3 illustrates original best guess versus

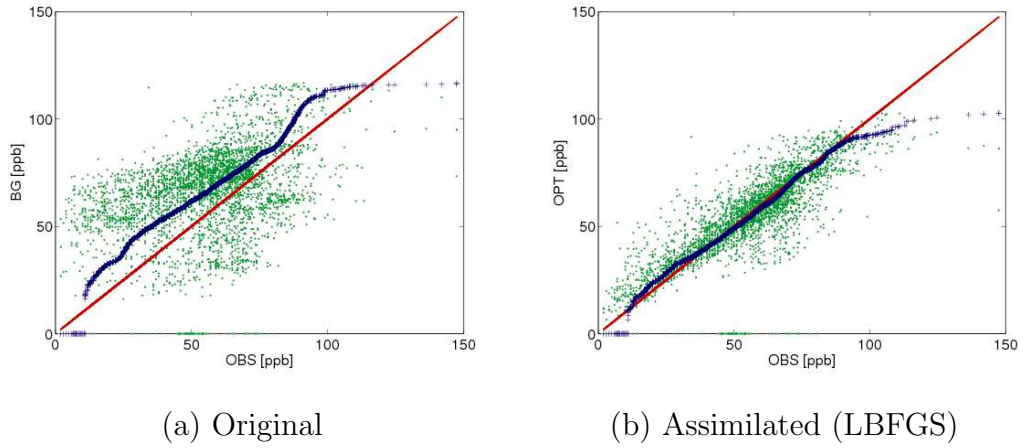


Figure 6.3: Scatter plot and quantile-quantile plot of model-observations agreement using AR background and L-BFGS method.

optimized predicts using L-BFGS method. This proves that data assimilation helps to improve forecasts.

For the AR background experiment, The R^2 of initial guess is 0.1477. RMS and R^2 can be found in Table 6.1. CG is the best within 10 iterations, then HFN and L-BFGS. Note that RMS and R^2 don't reflect the efficiency of an optimization method, buy only the result of minimization. If we consider optimization time together with optimization effect, L-BFGS should be the best of all in this AR background case, because it costs less time than the other two methods and can also approaches the minimal point.

	LBFGS	CG	HFN
RMS	11.93	11.64	11.86
R^2	0.6793	0.6968	0.6847

Table 6.1: RMS and correlation coefficient for three methods using AR background.

6.2.2 NMC Background

As was addressed before, the cost function consists two parts: misfit between observations and model predictions, as well as deviation of the solution from the background state. The covariance matrix of background error can be approximated by the method proposed by Parrish et al [50] and is denoted the NMC background in

this work. The inverse of the B matrix for the NMC model used in our experiment was obtained using a truncated SVD [25]. This approach inverts only the contributions corresponding to the largest singular values, and thus circumvents the errors coming from inverting the NMC matrix which can be ill conditioned and reduces the cost function computational effort [10].

Still trying to minimize the same cost function, we replace AR background term with NMC background and run data assimilation again. More than 100 iterations are needed for L-BFGS to converge at around 16000, which is much slower than using AR background. Considering the slow speed, we only test every method for 20 iterations, and at which the cost function reduces to about 25000.

Results

The NMC background provides us another case to assess performance of the three methods, i.e., when the cost function converge slowly. From the figure we can see that the cost function goes down slowly after several iterations. Figure 6.4.(a) shows the decrease of the cost function versus the number of new x for three methods. CG displays large advantage in the first several iterations and keeps the advantage to the end. When 10 iterations finish, CG exits with the least cost function so CG wins in terms of decreasing the value of cost function. However, from the aspect of running time, in terms of number of model runs, L-BFGS seems to be best and this is demonstrated in 6.4.(b). Of the 20 new x iterations, L-BFGS only calls for 22 model runs, while HFN needs 39 model runs and CG 41 to exit optimization. Therefore, L-BFGS is still the best in the NMC background experiment, which implies that L-BFGS might be good choice when cost function converges slowly. In reality, to choose CG or L-BFGS depends on requirement of the experiment. If efficiency is the most important and rough minimal point is acceptable, L-BFGS is a welcome candidate. If the accuracy of minimal point is a must, then CG might be better.

For the NMC background experiment, the improvement of data assimilation is revealed in Table 6.2 and Figure 6.5. The Table 6.2 shows that CG is the best within 20 iterations, and L-BFGS is better than HFN in both RMS and R^2 . The scatter and quantile-quantile plot 6.5 illustrates original best guess versus optimized

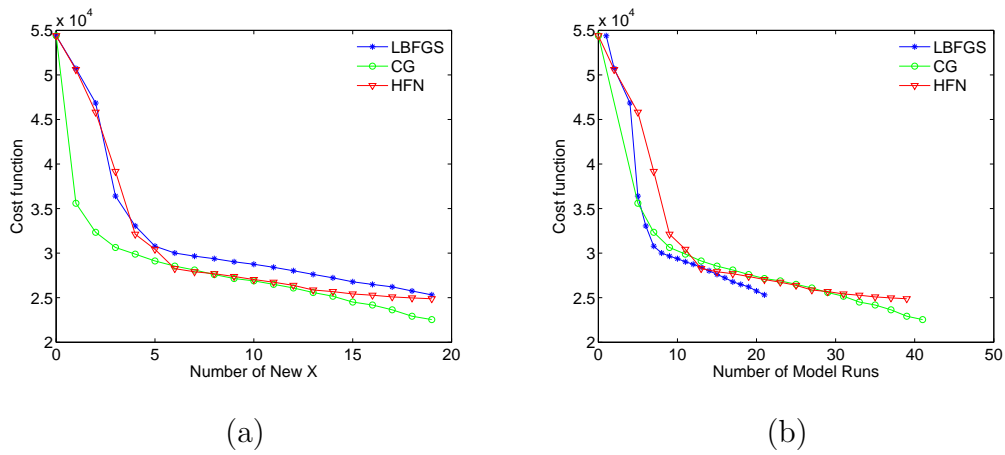


Figure 6.4: Decrease of the cost function using NMC background versus each iteration and the number of model runs for CG, L-BFGS and HFN.

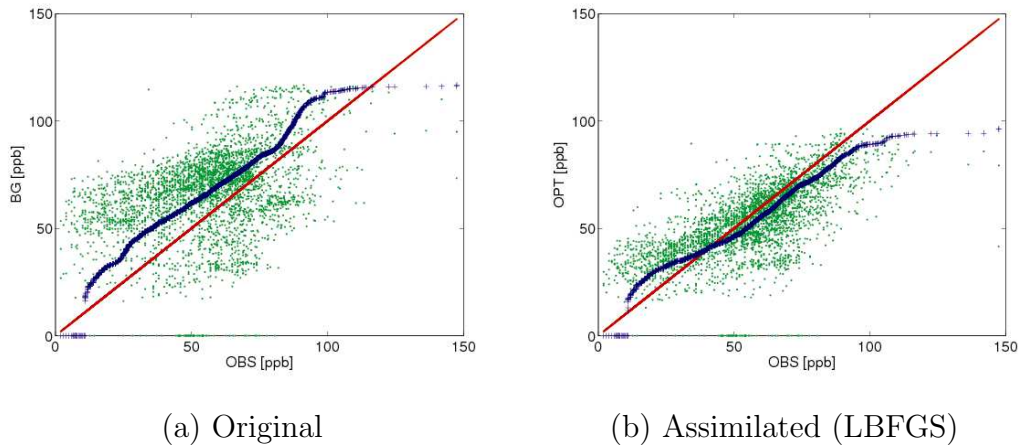


Figure 6.5: Scatter plot and quantile-quantile plot of model-observations agreement using NMC background.

predictions using L-BFGS method. This also proves that data assimilation helps to improve forecasting.

6.2.3 Artificial Data

Optimization methods may show quite different performance according to the value of a cost function, i.e., the position of the starting point. In the AR background and NMC background experiments, we use the real observations to form the cost function, and the starting point given by the initial guess is far away from the optimal point. To compare the performance of the optimization methods when the cost function

	LBFGS	CG	HFN
RMS	15.91	14.11	16.31
R^2	0.4440	0.5616	0.4066

Table 6.2: RMS and correlation coefficient for three methods using NMC background.

is small, we design the artificial data as observations instead of the previous real observations. The artificial data is chosen so that the starting point is already very close to the optimal point (using the L-BFGS solution). For this scenario, we use the AR background in the cost function. The process is implemented as follows:

1. We take L-BFGS optimized initial conditions as input, and run the STEM model for 12 hours from 8 EDT to 20 EDT of July 20, 2004. During this period, concentrations of ozone at pre-chosen grid points (totally we choose 180 points in the $25*22*11$ computational domain) are recorded at the end of each hour. These records are considered as artificial data because they are actually assimilated model predictions rather than real observations, and they will be used as “reference” later.
2. Then we run STEM for 24 hours from 8EDT of July 20, 2004 to 8 EDT the next day, and take down the concentrations at the final time. Usually there are not many changes in concentrations at the same time of two consecutive days, so we assume the recorded final concentrations at 8EDT of July 21 to be the concentrations at 8EDT of July 20, and use this as the initial conditions of simulation.
3. Performing data assimilation to optimize initial conditions for 12 hours from 8EDT to 20 EDT of July 20, 2004 with three methods respectively. Here both initial conditions and observations are artificial, so as to make the cost function much smaller than the original cost function. In fact, the artificial cost function is about $1/6$ of the original value and after assimilation the artificial cost function is less than 10 percent of the original value.

Results

The Figure 6.6.(a) shows the decrease of the cost function using artificial data versus the number of new x for the three methods. In the first five iterations, they behave differently in the ability of decreasing the cost function, while after that their performances are similar. The main difference can be found in 6.6.(b) in the number of model runs they need. Each point represents the number of model runs at this new x . In the 10 new x iterations, L-BFGS only calls for 10 model runs to converge, while both CG and HFN need 22 model runs to finish optimization. Therefore, L-BFGS outweighs the other two methods again in the artificial data experiment.

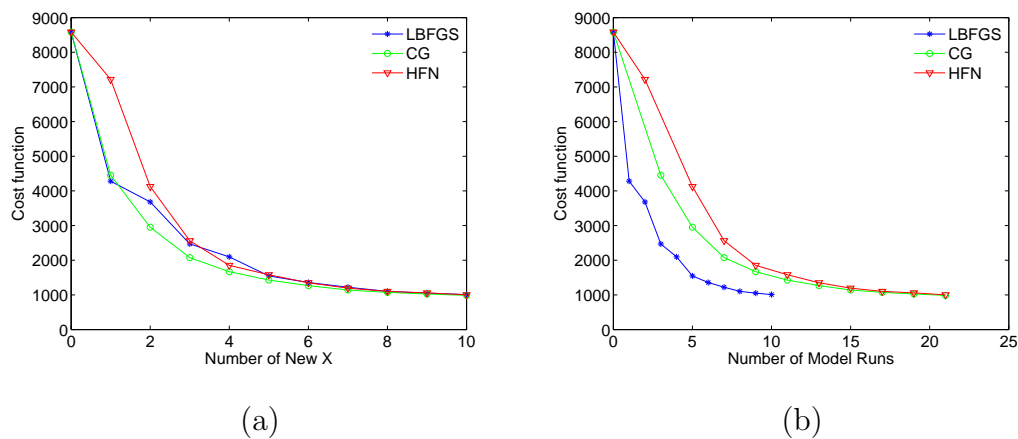


Figure 6.6: Decrease of the cost function using artificial data versus each iteration and the number of model runs for CG, L-BFGS and HFN methods.

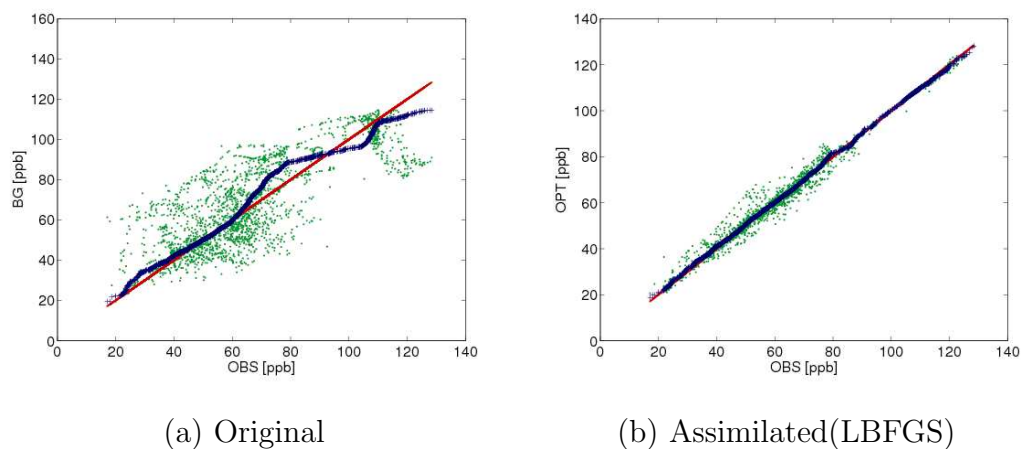


Figure 6.7: Scatter plot and quantile-quantile plot of model-observations agreement using artificial data.

The scatter and quantile-quantile plot 6.7 illustrates original best guess versus optimized predictions using L-BFGS method. It is obvious there is much improvement after data assimilation.

For this artificial data experiment, data assimilation improves R^2 from 0.7048 to more than 0.98 for all of the optimization methods. The Table 6.3 shows that L-BFGS is the best of all because RMS is the smallest and R^2 is closest to 1, which means that model predictions approach to observations. CG is slightly better than HFN in both RMS and R^2 .

	LBFGS	CG	HFN
RMS	2.51	2.78	2.79
R^2	0.9899	0.9875	0.9874

Table 6.3: RMS and correlation coefficient for three methods using artificial data.

Comments on These Methods

L-BFGS, CG and HFN share something in common: (1) they are all able to solve nonlinear optimization problem in large-scale chemical transport models. (2) they all require the first derivatives of the cost function, which is given by first order adjoints. In the above experiments we have shown that L-BFGS converges fastest regardless of background and observation data. Therefore, we conclude that for the STEM model, L-BFGS is the best candidate to complete data assimilation task, compared with CG and HFN.

Besides the methods we have discussed, there are some other optimization methods, which rely on second order information about the cost function, which can be provided by second order adjoints. In the next chapter we will give the theory on computation of second order adjoints in chemical transport models.

Chapter 7

Second Order Adjoint

7.1 Theory of Second Order Adjoint

Ozyurt and Barton [48, 49] discuss the evaluation of second order adjoints for embedded functionals of stiff systems. Their approach is to derive the second order adjoint ODE and then solve it efficiently together with the forward, the tangent linear, and the first order adjoint models. Different Jacobian matrices appearing in the definition of the second order adjoint ODE are derived from the original ODE using automatic differentiation. The computational method is based on backward differentiation formulas (DASSL). The LU factorizations of the forward model solution are reused in the tangent linear and the first and second order adjoint solutions, which leads to a computationally efficient process. The current work also discusses the calculation of second order adjoints for stiff ODEs, but focuses on the ODEs arising in chemistry. We give the theoretical formulation of first and second order adjoints, especially discrete adjoints.

Wang et al. [59] discuss the theory of second order adjoints and its applications in numerical weather prediction. Second order adjoints have been used in data assimilation within the numerical optimization algorithms (Wang et al., [60]; Le Dimet et al., [35, 36]; Ozyurt et al., [48]). Hessian vector products have been used in the calculation of Hessian singular vectors in the context of data assimilation ([33, 60]). In this work, we show the construction of second order adjoints for a three-dimensional chemical transport model and illustrate several applications including

sensitivity analysis, optimization, uncertainty quantification, and the calculation of directions of maximal error growth.

7.2 Second Order Adjoint for Stiff ODEs

Like with first order adjoints (FOA) one distinguishes between discrete and continuous second order adjoints (SOA). We now discuss these two approaches in a general framework.

7.2.1 Continuous SOA

Consider a general (stiff) ODE

$$c' = f(t, c, p), \quad c(t^0) = c^0(p), \quad t^0 \leq t \leq t^F$$

For our application the vector $c(t) \in \mathbb{R}^{n_s}$ represents the time evolving concentrations of the chemical species starting from the initial configuration c^0 . $p \in \mathbb{R}^{n_p}$ is a vector of model parameters. The rate of change in the concentrations c is determined by the nonlinear production/loss function $f = [f_1, \dots, f_{n_s}]^T$.

Consider a cost functional

$$\Psi = \int_{t^0}^{t^F} g(c(t), p) dt$$

defined on the time evolving concentrations. We want to efficiently obtain the first and second order derivatives of the cost function with respect to model parameters,

$$\frac{\partial \Psi}{\partial p_i}, \quad \frac{\partial^2 \Psi}{\partial p_i \partial p_j}, \quad 1 \leq i, j \leq n_p$$

Note that the parameters can be transformed into variables by appending additional formal evolution equations for parameters $p' = 0$. This allows to always reduce the sensitivity of the cost functional with respect to parameters to the sensitivity of the cost functional with respect to initial conditions. Moreover, the general cost functional defined as an integral of a function of the state along the trajectory can be reformulated as a cost functional defined on the state at the final time by appending

an additional variable θ and an equation that performs the time integration. The equivalent system becomes:

$$\begin{bmatrix} c \\ p \\ \theta \end{bmatrix}' = \begin{bmatrix} f(t, c, p) \\ 0 \\ g(c, p) \end{bmatrix}, \quad \begin{bmatrix} c(t^0) \\ u(t^0) \\ \theta(t^0) \end{bmatrix} = \begin{bmatrix} c^0(p) \\ p \\ 0 \end{bmatrix}, \quad \Psi = \theta(t^F)$$

Without loss of generality the mathematical formulation of the stiff nonlinear differential equations which constitute the *forward model* is

$$\frac{dy}{dt} = f(t, y), \quad y(t^0) = y^0, \quad t^0 \leq t \leq t^F \quad (7.1)$$

The solution is $y(t) = [c^T, p^T, \theta]^T \in \mathbb{R}^n$, $n = n_s + n_p + 1$, and the model parameters are the initial conditions y^0 . Throughout this work vectors will be represented in column format and an upper script $(\cdot)^T$ will denote the transposition operator. Again without loss of generality the cost functional is defined as a function of the state at the final time

$$\Psi(y^0) = g(y(t^F)) \quad (7.2)$$

We are interested to efficiently evaluate the first and second order sensitivities of the cost functional (7.2) with respect to changes in initial conditions

$$\frac{\partial \Psi}{\partial y_i^0}, \quad \text{and} \quad \frac{\partial^2 \Psi}{\partial y_i^0 \partial y_j^0}, \quad 1 \leq i, j \leq n$$

The gradient of a scalar function (e.g., $\partial \Psi / \partial y^0$) is a row vector. We denote the Jacobian of the time derivative function in (7.1) by

$$J_{i,j}(t, y) = \frac{\partial f_i(t, y)}{\partial y_j}, \quad 1 \leq i, j \leq n \quad (7.3)$$

The Hessian of the time derivative function in (7.1) is a 3-tensor of second order derivatives

$$H_{i,j,k}(t, y) = \frac{\partial J_{i,j}(t, y)}{\partial y_k} = \frac{\partial^2 f_i(t, y)}{\partial y_j \partial y_k} = \frac{\partial^2 f_i(t, y)}{\partial y_k \partial y_j} = H_{i,k,j}(t, y), \quad 1 \leq i, j, k \leq n \quad (7.4)$$

The Hessian allows to express the derivatives of the Jacobian times a user vector. As shown in Appendix A.1 for any vectors u and v we have that

$$\begin{aligned} \frac{\partial}{\partial y} [J(t, y) \cdot u] \cdot v &= (H(t, y) \cdot u) \cdot v = (H(t, y) \cdot v) \cdot u, \\ \frac{\partial}{\partial y} [J^T(t, y) \cdot u] \cdot v &= (u^T \cdot H(t, y)) \cdot v = (H(t, y) \cdot v)^T \cdot u, \end{aligned}$$

where the dot operator (\cdot) denotes the regular tensor-vector product.

Small perturbations of the solution (due to infinitesimally small changes δy^0 in the initial conditions)

$$\delta y(t) = \frac{\partial y(t)}{\partial y^0} \cdot \delta y^0 \quad (7.5)$$

propagate forward in time according to the *tangent linear model*

$$\frac{d\delta y}{dt} = J(t, y) \cdot \delta y, \quad \delta y(t^0) = \delta y^0, \quad t^0 \leq t \leq t^F \quad (7.6)$$

The change in the cost functional (7.2) due to the small change δy^0 in the initial conditions is

$$\delta \Psi = \frac{\partial g}{\partial y}(t^F) \cdot \delta y(t^F) = \frac{\partial \Psi}{\partial y^0} \cdot \delta y^0$$

In the forward sensitivity analysis each integration of the tangent linear model (7.6) allows to compute the dot product of the gradient $\partial \Psi / \partial y^0$ with the vector of initial perturbations δy^0 . The gradient is recovered after n tangent linear model (7.6) integrations initialized with linearly independent perturbation vectors.

A more efficient way of calculating the gradient $\partial \Psi / \partial y^0$ is provided by the *first order adjoint model* [11, 51, 52]

$$\frac{d\lambda}{dt} = -J^T(t, y) \cdot \lambda, \quad \lambda(t^F) = \frac{\partial g}{\partial y}(t^F), \quad t^F \geq t \geq t^0 \quad (7.7)$$

The adjoint variables $\lambda(t) \in \mathbb{R}^n$ represent the sensitivities of the cost functional with respect to (changes in) the model solution

$$\lambda(t) = \left(\frac{\partial \Psi}{\partial y(t)} \right)^T,$$

and in particular we have that the adjoints at the initial time are the transposed gradient of the cost functional

$$\lambda(t^0) = \left(\frac{\partial \Psi}{\partial y^0} \right)^T$$

We are now interested in obtaining the second order derivatives of the cost functional with respect to initial conditions. The Hessian of the cost functional is

$$H_{i,j} = \frac{\partial^2 \Psi}{\partial y_i^0 \partial y_j^0} = \frac{\partial}{\partial y_j^0} \left(\frac{\partial \Psi}{\partial y_i^0} \right) = \frac{\partial \lambda_i(t^0)}{\partial y_j^0} \quad 1 \leq i, j \leq n$$

The Hessian has n^2 elements. In many problems (including our target application, atmospheric chemical transport problems) n is very large and computing the entire Hessian is not practical. We will therefore look to compute Hessian times vector products $\sigma = H \cdot u$ for any user-defined vector u

$$(H \cdot u)_i = \sum_{j=1}^n H_{i,j} u_j = \sum_{j=1}^n \frac{\partial \lambda_i(t^0)}{\partial y_j^0} u_j = \frac{\partial \lambda_i(t^0)}{\partial y^0} \cdot u \quad (7.8)$$

To compute such products we consider the variation of the cost functional (7.2) with respect to changes in initial conditions as a new cost functional that depends on both the initial state and on the initial perturbation

$$\delta\Psi(y^0, \delta y^0) = \frac{\partial \Psi}{\partial y^0} \cdot \delta y^0 = \lambda^T(t^0) \cdot \delta y^0 = \sum_{i=1}^n \frac{\partial \Psi}{\partial y_i^0} \delta y_i^0 \quad (7.9)$$

The gradient of $\delta\Psi$ with respect to changes in y^0 can be computed by the adjoint method. This gradient represents the product of the Hessian of Ψ times the initial perturbation vector,

$$\begin{aligned} \left(\frac{\partial \delta\Psi}{\partial y^0} \right)_j &= \frac{\partial \delta\Psi}{\partial y_j^0} = \frac{\partial}{\partial y_j^0} \left(\sum_{i=1}^n \frac{\partial \Psi}{\partial y_i^0} \delta y_i^0 \right) \\ &= \sum_{i=1}^n \frac{\partial^2 \Psi}{\partial y_i^0 \partial y_j^0} \delta y_i^0 = \sum_{i=1}^n \frac{\partial^2 \Psi}{\partial y_j^0 \partial y_i^0} \delta y_i^0 \\ &= \sum_{i=1}^n H_{j,i} \delta y_i^0 \\ &= (H \cdot \delta y^0)_j \end{aligned}$$

From (7.8) we see that the Hessian-vector products can be computed as the first order adjoint gradients of the cost functional $\delta\Psi$, if the tangent linear model is initialized with the user-defined vector $\delta y^0 = u$.

The cost functional (7.9) depends on both y and δy . Consequently, to evaluate $\delta\Psi$ we consider both the forward (7.1) and the tangent linear model (7.6) evolving together:

$$\frac{d}{dt} \begin{bmatrix} y \\ \delta y \end{bmatrix} = \begin{bmatrix} f(t, y) \\ J(t, y) \cdot \delta y \end{bmatrix}, \quad \begin{bmatrix} y \\ \delta y \end{bmatrix} (t^0) = \begin{bmatrix} y^0 \\ u \end{bmatrix}, \quad t^0 \leq t \leq t^F \quad (7.10)$$

The Jacobian of the extended system (7.10) is

$$\begin{bmatrix} J(t, y) & 0 \\ \frac{\partial}{\partial y} (J(t, y) \cdot \delta y) & J(t, y) \end{bmatrix} = \begin{bmatrix} J(t, y) & 0 \\ H(t, y) \cdot \delta y & J(t, y) \end{bmatrix}$$

The adjoint of the extended system (7.10) for the cost function (7.9) reads

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \sigma \\ \lambda \end{bmatrix} &= - \begin{bmatrix} J(t, y) & 0 \\ H(t, y) \cdot \delta y & J(t, y) \end{bmatrix}^T \cdot \begin{bmatrix} \sigma \\ \lambda \end{bmatrix} \\ &= \begin{bmatrix} -J^T(t, y) \cdot \sigma - \left(H(t, y) \cdot \delta y \right)^T \cdot \lambda \\ -J^T(t, y) \cdot \lambda \end{bmatrix} \\ \begin{bmatrix} \sigma \\ \lambda \end{bmatrix} (t^F) &= \begin{bmatrix} \frac{d^2 g}{dy^2} (t^F) \cdot \delta y (t^F) \\ \frac{dg}{dy} (t^F) \end{bmatrix}, \quad t^F \geq t \geq t^0 \end{aligned} \quad (7.11)$$

We see that the second equation in (7.11) is the first order adjoint equation (7.7), and λ is the first order adjoint variable.

The first equation in (7.11) is the *second order adjoint equation* defines the time evolution of the *second order adjoint variable* σ ,

$$\frac{d\sigma}{dt} = -J^T(t, y) \cdot \sigma - \left(H(t, y) \cdot \delta y \right)^T \cdot \lambda, \quad \sigma(t^F) = \frac{d^2 g}{dy^2} (t^F) \cdot \delta y (t^F) \quad (7.12)$$

We notice that

$$\sigma(t) = \delta\lambda(t) = \frac{\partial\lambda(t)}{\partial y^0} \cdot \delta y^0 = \frac{\partial\lambda(t)}{\partial y(t)} \cdot \frac{\partial y(t)}{\partial y^0} \cdot \delta y^0 = \frac{\partial\lambda(t)}{\partial y(t)} \cdot \delta y(t)$$

Consequently the second order adjoint equation (7.12) can be obtained by formally taking the variation of the first order adjoint equation (7.7) with respect to changes δy^0 in the initial conditions y^0

$$\begin{aligned} \frac{\partial}{\partial y^0} \left(\frac{d\lambda}{dt} \right) \cdot \delta y^0 &= \frac{\partial}{\partial y^0} \left(-J^T(t, y) \cdot \lambda \right) \cdot \delta y^0 \\ \frac{d}{dt} \left(\frac{\partial\lambda}{\partial y^0} \cdot \delta y^0 \right) &= \frac{\partial}{\partial y} \left(-J^T(t, y) \cdot \lambda \right) \cdot \frac{\partial y(t)}{\partial y^0} \cdot \delta y^0 \\ \frac{d\sigma}{dt} &= \frac{\partial}{\partial y} \left(-J^T(t, y) \cdot \lambda \right) \cdot \delta y(t) \\ &= -J^T(t, y) \cdot \sigma - \left(H(t, y) \cdot \delta y \right)^T \cdot \lambda \\ \frac{\partial}{\partial y^0} \left(\frac{dg}{dy} (t^F) \right) \cdot \delta y^0 &= \frac{dg}{dy^2} (t^F) \cdot \frac{\partial y (t^F)}{\partial y^0} \cdot \delta y^0 \\ &= \frac{d^2 g}{dy^2} (t^F) \cdot \delta y (t^F) \end{aligned}$$

7.2.2 Discrete SOA

Similar considerations hold for the discrete system

$$y^k = \mathcal{N}_k(y^{k-1}) \ , \quad k = 1, \dots, N \ , \quad y^0 \text{ given} \quad (7.13)$$

and the discrete cost function is of the form

$$\Psi(y^0) = g(y^N) \quad (7.14)$$

We denote the Jacobian matrix of the discrete time-marching operator by $\mathcal{N}'_k(y) = \partial \mathcal{N}_k / \partial y$, and the Hessian three-tensor by $\mathcal{N}''_k(y) = \partial^2 \mathcal{N}_k / \partial y^2$. The tangent linear model is

$$\delta y^k = \mathcal{N}'_k(y^{k-1}) \cdot \delta y^{k-1} \ , \quad k = 1, \dots, N \ , \quad \delta y^0 = u \quad (7.15)$$

The extended adjoint of the combined (7.13)–(7.15) for the cost function $\delta \Psi$ reads

$$\begin{aligned} \begin{bmatrix} \sigma^{k-1} \\ \lambda^{k-1} \end{bmatrix} &= \begin{bmatrix} -(\mathcal{N}'_k(y^{k-1}))^T \cdot \sigma^k - (\mathcal{N}''_k(y^{k-1}) \cdot \delta y^{k-1})^T \cdot \lambda^k \\ -(\mathcal{N}'_k(y^{k-1}))^T \cdot \lambda^k \end{bmatrix} \ , \quad (7.16) \\ \begin{bmatrix} \sigma^N \\ \lambda^N \end{bmatrix} &= \begin{bmatrix} \frac{\partial^2 g}{\partial y^2}(y^N) \cdot \delta y^N \\ \frac{\partial g}{\partial y}(y^N) \end{bmatrix} \end{aligned}$$

where $N \geq k \geq 1$. At the end of the backward in time integration (7.16) provides the gradient and the Hessian vector product

$$\lambda^0 = \left(\frac{\partial \Psi}{\partial y^0} \right)^T \ , \quad \sigma^0 = \frac{\partial^2 \Psi}{(\partial y^0)^2} \cdot u$$

Example of Computing Discrete SOA in Data Assimilation

Consider a cost functional

$$\Psi(y^0) = \frac{1}{2} (y^0 - y^B)^T B^{-1} (y^0 - y^B) + \frac{1}{2} \sum_{k=1}^N (Dy^k - z^k)^T R_k^{-1} (Dy^k - z^k) \quad (7.17)$$

where z^k are the observations available at discrete times t^k , $k = 1, \dots, N$, and D represents a linear mapping operator of states to observations.

The efficient numerical minimization of (7.17) requires the gradient of the cost function ($\lambda^0 = (\partial \Psi / \partial y^0)^T$) as well as second order derivative information in the form

of Hessian vector products ($\sigma^0 = \partial^2 \Psi / (\partial y^0)^2 \cdot u$). These derivatives are obtained via the first and second order adjoint models as follows. Consider the CTM represented compactly as (3.6). First the forward and the tangent linear models are solved together forward in time:

$$\begin{aligned}
y^0 &= y(t^0) \\
\delta y^0 &= u \\
&\text{Save } y^0, \delta y^0 \text{ on tape} \\
&\text{FOR } k = 1, 2, \dots, N-1, N \text{ DO} \\
&\quad y^k = \mathcal{N}_k(y^{k-1}) \\
&\quad \delta y^k = \mathcal{N}'_k(y^{k-1}) \cdot \delta y^{k-1} \\
&\quad \text{Save } y^k, \delta y^k \text{ on tape} \\
&\text{END FOR}
\end{aligned} \tag{7.18}$$

Next the first and second order adjoint models are solved together, backward in time:

$$\begin{aligned}
\sigma^N &= 0 \\
\lambda^N &= 0 \\
&\text{Read } y^N, \delta y^N \text{ from tape} \\
&\text{FOR } k = N, N-1, \dots, 2, 1 \text{ DO} \\
&\quad \lambda^k = \lambda^k + D^T R_k^{-1} \cdot (D y^k - z^k) \\
&\quad \sigma^k = \sigma^k + D^T R_k^{-1} D \cdot \delta y^k \\
&\quad \text{Read } y^{k-1}, \delta y^{k-1} \text{ from tape} \\
&\quad \lambda^{k-1} = -(\mathcal{N}'_k(y^{k-1}))^T \cdot \lambda^k \\
&\quad \sigma^{k-1} = -(\mathcal{N}'_k(y^{k-1}))^T \cdot \sigma^k - (\mathcal{N}''_k(y^{k-1}) \cdot \delta y^{k-1})^T \cdot \lambda^k \\
&\text{END FOR} \\
\lambda^0 &= \lambda^0 + B^{-1} \cdot (y^0 - y^B) \\
\sigma^0 &= \sigma^0 + B^{-1} \cdot \delta y^0 .
\end{aligned} \tag{7.19}$$

7.2.3 Implementation of SOA for Chemistry System

The implementation of numerical integrators for chemistry can be done automatically using the Kinetic PreProcessor KPP software tools [12]. KPP was extended [11, 51] to produce a rapid and efficient implementation of the code for sensitivity

analysis of chemical kinetic systems. KPP builds Fortran77, Fortran90, C, or Matlab simulation code for chemical systems with chemical concentrations changing in time according to the law of mass action kinetics. KPP generates the following building blocks:

1. *Fun*: the time derivative of concentrations;
2. *Jac*, *Jac_SP*: Jacobian of *Fun* in full or in sparse format;
3. *KppDecomp*: sparse LU decomposition for the Jacobian;
4. *KppSolve*, *KppSolveTR*: solve sparse system with the Jacobian matrix and its transpose;
5. *Jac_SP_Vec*, *JacTR_SP_Vec*: sparse Jacobian (transposed or not) times vector;
6. The stoichiometric matrix *STOICM*;
7. *ReactantProd*: vector of reaction rates;
8. *JacReactantProd*: the Jacobian of the above;
9. *dFun_dRcoeff*: derivatives of *Fun* with respect to reaction coefficients (in sparse format);
10. *dJac_dRcoeff*: derivatives of *Jac* with respect to reaction coefficients times user vector;
11. *Hess*: the Hessian of *Fun*; this 3-tensor is represented in sparse format;
12. *Hess_Vec*, *HessTR_Vec*: Hessian (or its transpose) times user vectors; same as the derivative of Jacobian (transposed) vector product times vector.

In [11,51] the authors show how these KPP building blocks can be used to implement very efficiently code for direct and adjoint sensitivity analysis of chemical systems.

7.2.4 Discrete SOA for Transport System

The transport equation is solved using a directional x, y, and z split approach. The basic numerical techniques solve the one-dimensional transport equation

$$\frac{\partial c}{\partial t} = -u \frac{\partial c}{\partial x} + \frac{1}{\rho} \frac{\partial}{\partial x} \left(\rho K \frac{\partial c}{\partial x} \right), \quad c(t, x_{in}) = c_{in}(t), \quad K \frac{\partial c}{\partial x} \Big|_{x_{out}} = 0 \quad (7.20)$$

in STEM the horizontal advection term is discretized by the third order upwind finite difference formula [31]

$$- \left(u \frac{\partial c}{\partial x} \right) \Big|_{x=x_i} = \begin{cases} u_i (-c_{i-2} + 6c_{i-1} - 3c_i - 2c_{i+1}) / (6\Delta x), & \text{if } u_i \geq 0 \\ u_i (2c_{i-1} + 3c_i - 6c_{i+1} + c_{i+2}) / (6\Delta x), & \text{if } u_i < 0 \end{cases} \quad (7.21)$$

The diffusion terms are discretized by the second order central differences

$$\frac{1}{\rho} \frac{\partial}{\partial x} \left(\rho K \frac{\partial c}{\partial x} \right) \Big|_{x=x_i} = \frac{(\rho_{i+1} K_{i+1} + \rho_i K_i)(c_{i+1} - c_i) - (\rho_i K_i + \rho_{i-1} K_{i-1})(c_i - c_{i-1})}{2\rho_i \Delta x^2} \quad (7.22)$$

For the inflow boundary the advection discretization drops to the first order upwind scheme, which makes the order of consistency of the whole scheme quadratic for the interior points of the domain. For the outflow boundary the advection discretization also drops to the first order upwind scheme.

The space semi-discretization leads to the linear ordinary differential equation

$$\frac{dc}{dt} = A(t) c(t) + B(t), \quad (7.23)$$

where the matrix $A(t)$ depends on the wind field, the diffusion tensor, and the air density but it does not depend on the unknown concentrations (for the discretization schemes under consideration). The vector $B(t)$ represents the Dirichlet boundary conditions.

The forward system is advanced in time from t^n to $t^{n+1} = t^n + \Delta t$ using Crank-Nicholson

$$c^{n+1} = \left(I - \frac{\Delta t}{2} A(t^{n+1}) \right)^{-1} \left[\left(I + \frac{\Delta t}{2} A(t^n) \right) c^n + \Delta t \frac{B(t^n) + B(t^{n+1})}{2} \right] \quad (7.24)$$

The chosen discretization leads to pentadiagonal matrices and systems which can be solved very efficiently.

Equation (7.24) represents the forward discrete model for horizontal transport. The corresponding adjoint system is then advanced backwards in time using the discrete adjoint formulation

$$\lambda^n = \left(I + \frac{\Delta t}{2} A^T(t^n) \right) \left(I - \frac{\Delta t}{2} A^T(t^{n+1}) \right)^{-1} \lambda^{n+1} \quad (7.25)$$

Equation (7.25) is a consistent time discretization of the continuous adjoint equation. Because of the linear discretization the second order adjoint formula obtained by taking the variation of (7.25)

$$\sigma^n = \left(I + \frac{\Delta t}{2} A^T(t^n) \right) \left(I - \frac{\Delta t}{2} A^T(t^{n+1}) \right)^{-1} \sigma^{n+1} \quad (7.26)$$

This means that the same adjoint transport routines are used for both the first and the second order adjoint solutions. Moreover, it is possible to reuse the LU decomposition from the first order adjoint (7.25) in the second order adjoint calculation (7.26).

The vertical advection term is discretized by the first order upwind finite difference formula

$$- \left(w \frac{\partial c}{\partial z} \right) \Big|_{z=z_i} = \begin{cases} -w_i (c_i - c_{i-1}) / (z_i - z_{i-1}) , & \text{if } w_i \geq 0 \\ -w_i (c_{i+1} - c_i) / (z_{i+1} - z_i) , & \text{if } w_i < 0 \end{cases} \quad (7.27)$$

The vertical diffusion is discretized by the second order central differences. Note that the vertical grid is not uniform. The top boundary condition is Dirichlet for inflow and Neumann for outflow (i.e. zero diffusive flux through the top outflow boundary). This is similar to the horizontal advection case.

The ground level boundary condition considers the flow of material given by surface emission rates Q and by deposition processes with deposition velocity V . The vertical wind speed at ground level is $w_1 = 0$. The ground boundary condition reads

$$-K \frac{\partial c}{\partial z} \Big|_{z=\text{ground}} = Q - Vc , \quad (7.28)$$

where K is the vertical eddy diffusivity. The ground level concentration is discretized in space as

$$c'_1 = \frac{(\rho_2 K_2 + \rho_1 K_1)(c_2 - c_1)}{2\rho_1(z_2 - z_1)\Delta z_1} - \frac{Vc_1 - Q}{\Delta z_1} \quad (7.29)$$

where Δz_1 is the height of the first layer.

This space semi-discretization leads to the linear ODE

$$c'(t) = A(t)c(t) + B(t)e_N + \frac{Q(t)}{\Delta z_1}e_1 \quad (7.30)$$

where the entry $A_{1,1}$ accounts now also for the deposition velocity; B accounts for the top boundary and Q accounts for ground emissions. Here e_j is the j^{th} column of the identity matrix.

Using Crank-Nicholson time stepping for the concentrations and forward Euler timestepping for the boundaries and the ground emissions the forward discrete model for vertical transport reads

$$c^{n+1} = \left(I - \frac{\Delta t}{2}A(t^{n+1}) \right)^{-1} \left[\left(I + \frac{\Delta t}{2}A(t^n) \right) c^n + \Delta t \left(B(t^n)e_N + \frac{Q(t^n)}{\Delta z_1}e_1 \right) \right] \quad (7.31)$$

Note that in practice the emission intensities and top boundary values are constant over discrete time intervals (e.g. hourly) and the above forward Euler integration within such an interval is equivalent to Crank Nicholson. The corresponding first order discrete adjoint model is of the form (7.25), and the second order adjoint of the form (7.26). The same discrete adjoint vertical transport routine can be used for both the first and the second order adjoint solutions.

7.3 Validation of the 3D Second Order Adjoints

7.3.1 CPU time for Second Order Adjoints Calculation

We have implemented second order adjoint capabilities in the STEM model according to (7.18)–(7.19). The chemistry is solved using Rosenbrock methods implemented efficiently using KPP. The second order adjoints for transport are implemented as discussed in the corresponding Section. The CPU times associated with the first and second order adjoint calculation are reported in Table 7.1. We see that the CPU time needed for a second order adjoint calculation is less than twice the time for a first order adjoint calculation.

Simulation	CPU time	Scaled time
FWD only	35.8 min	1
FWD followed by FOA	83.92 min	2.34
FWD + TLM followed by FOA + SOA	127.6 min	3.56

Table 7.1: CPU times for a 12 hours three-dimensional chemistry and transport simulation. FWD denotes the forward model, TLM the tangent linear model, FOA the first order adjoint, and SOA the second order adjoint. Shown are the wall clock times and the times relative to the forward model run.

7.3.2 Validation of Tangent Linear Model

To compute second order adjoints we have to run Tangent Linear Model (TLM) to obtain perturbation of solutions in different times. Therefore the first step of validation is to guarantee that TLM is correct. We use the cost function defined in (7.17) for this experiment .

1. **First Run (reference):**

$$y(t_0) \xrightarrow{FWD} y(t_F)$$

2. **Second Run (perturbed):**

$$\begin{aligned} \overline{y(t_0)} &= y(t_0) + \varepsilon \cdot \delta y(t_0) \xrightarrow{FWD} \overline{y(t_F)} \\ \delta y(t_0) &\xrightarrow{TLM} \delta y(t_F) \end{aligned}$$

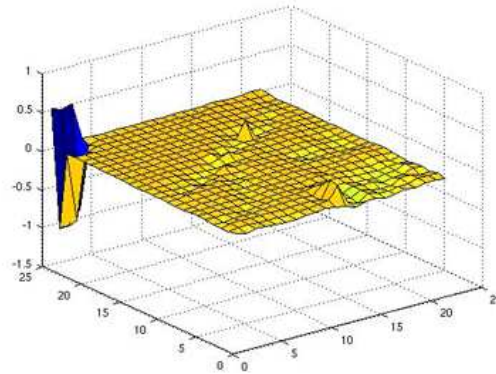
3. **Validation:**

$$\text{Check: } \delta y(t_F) \approx \frac{\overline{y(t_F)} - y(t_F)}{\varepsilon}$$

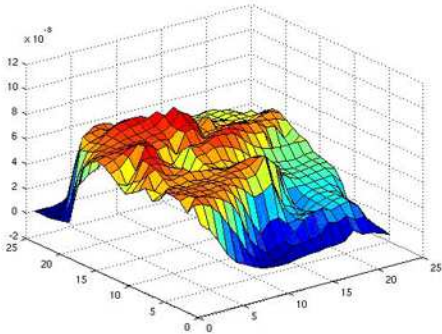
Results:

We run simulations of 1 hour, 4 hours and 8 hours, and plot corresponding values for species ozone on the first layer of the domain. The Figures 7.1 are from 8 hours' simulation and show the relative difference between $\frac{\overline{y(t_F)} - y(t_F)}{\varepsilon}$ and $\delta y(t_F)$ in (a),

as well as $\frac{\overline{y(t_F)} - y(t_F)}{\epsilon}$ in (b) and $\delta y(t_F)$ in (c). (a) implies that finite difference for the forward model approximates the tangent linear model on the entire layer except for some grid points situated at the corner of our domain. It is reasonable given that boundary conditions and emissions may be not accurately set for the STEM model. Besides, similarity of (b) and (c) also demonstrate correctness of tangent linear model.



(a) relative difference



(b) finite difference

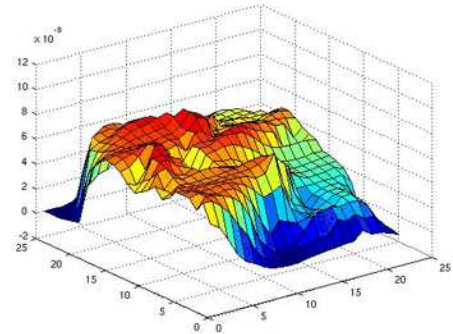
(c) perturbation of concentration at t^F

Figure 7.1: Validation of the tangent linear model for the three-dimensional chemistry transport model against finite difference of forward model.

7.3.3 Validation of Second Order Adjoints

We now go further to validate the correctness of the three-dimensional second order adjoints against finite differences of first order adjoints. The cost function for this experiment is (7.17).

1. **First Run (reference):**

(a) Forward:

$$\begin{aligned} y(t_0) &\xrightarrow{FWD} y(t_F) \\ \delta y(t_0) &\xrightarrow{FWD} \delta y(t_F) \end{aligned}$$

(b) Then backward:

$$\begin{aligned} \lambda(t_0) &\xleftarrow{FOA} \lambda(t_F) \\ \sigma(t_0) &\xleftarrow{SOA} \sigma(t_F), \text{ where } \sigma(t_0) = \text{Hessian} \times \delta y(t_0) \end{aligned}$$

2. **Second Run (perturbed):**

(a) Forward:

$$\overline{y(t_0)} = y(t_0) + \varepsilon \cdot \delta y(t_0) \xrightarrow{FWD} \overline{y(t_F)}$$

(b) Then backward:

$$\overline{\lambda(t_0)} \xleftarrow{FOA} \overline{\lambda(t_F)}$$

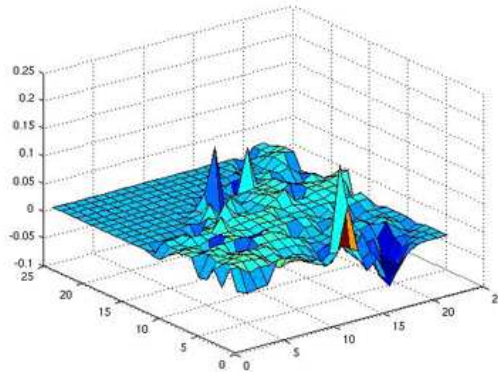
3. **Validation:**

$$\text{Check: } \sigma(t_0) \approx \frac{\overline{\lambda(t_0)} - \lambda(t_0)}{\varepsilon}$$

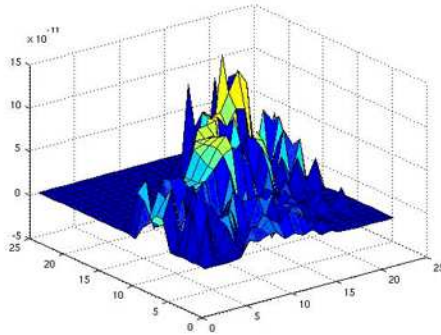
Results: We run two 8-hour simulations (one for reference and the other for perturbation) with the all species perturbed in the entire domain. The Figure 7.2(b) shows the finite difference $\Delta\lambda$, and (c) illustrates the second order adjoint *sigma* for the reference run. The relative difference between (b) and (c) is displayed in (a). The values of both (b) and (c) are for species ozone, and on the first layer of the computational domain. They look very similar, which proves correctness of second order adjoints.

7.3.4 Validation of Hessian Symmetry

Some of the applications discussed next, such as Daniel's conjugate gradient method, require the Hessian to be symmetric. Second order adjoint methodology provides Hessian-vector products. Even if the full Hessian of the cost function is not available we are able to check its symmetry as follows.



(a) relative difference



(b) finite difference

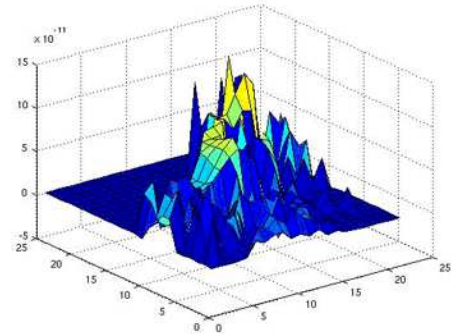
(c) σ at initial time $\sigma(t_0)$

Figure 7.2: Validation of the second order adjoint for the three-dimensional chemistry transport model against finite difference of first order adjoints.

Compute Hessian-vector products for two vectors δy^1 and δy^2 , starting from the same set of initial concentrations y^0 ,

$$\sigma^1 = H(y^0) \cdot \delta y^1, \quad \sigma^2 = H(y^0) \cdot \delta y^2,$$

and take the dot product of the second order adjoints with the other perturbation. The two dot products are equal if the computed Hessian is symmetric

$$(\delta y^2)^T \cdot \sigma^1 = (\delta y^2)^T \cdot \sigma^1 H(y^0) \cdot \delta y^1 = (\delta y^1)^T \cdot \sigma^1 H(y^0) \cdot \delta y^2 = (\delta y^1)^T \cdot \sigma^2.$$

In our experiments we set the first vector to be the set of initial conditions, $\delta y^1 = y^0$. The second vector is chosen in two ways. First, δy^1 is advanced from t^0 to t^F using the tangent linear model and δy^2 is taken to be the solution of the tangent linear model at the final time. Second, δy^2 is taken to be a vector with random entries

(scaled element-wise by y^0 to preserve the relative magnitude among concentrations of different species).

For each method we run 1 hour, 4 hours and 8 hours simulations. The results are shown in Table 7.2. The two products are close to each other in both methods, which indicates that the Hessian (computed by the second order adjoint method) is symmetric. Small differences are acceptable considering the large size of the vectors (10^7).

Method	p1/p2	1h	4h	8h
1	$(\delta y^1)^T \cdot \sigma^2$	2.5837e+4	1.8897e+5	3.2224e+5
	$(\delta y^2)^T \cdot \sigma^1$	2.4010e+4	1.8806e+5	3.1812e+5
2	$(\delta y^1)^T \cdot \sigma^2$	1.3012e+4	9.8862e+4	1.8372e+5
	$(\delta y^2)^T \cdot \sigma^1$	1.3012e+4	9.8705e+4	1.8316e+5

Table 7.2: Checking Hessian for Symmetry.

Chapter 8

Applications of Second Order Adjoint

8.1 Sensitivity Analysis

For second order sensitivity analysis, we first consider the SAPRC-99 atmospheric chemistry mechanism [7, 8] which includes the gas-phase atmospheric reactions of volatile organic compounds (*VOCs*) and nitrogen oxides (*NO_x*) in urban and regional settings. The chemical mechanism was developed at University of California, Riverside by Dr. W.P.L. Carter for use in airshed models for predicting the effects of *VOC* and *NO_x* emissions on tropospheric secondary pollutants formation such as ozone (*O₃*), peroxyacetyl nitrate (*PAN*), etc. In our analysis we consider the condensed fixed-parameter version of the SAPRC-99 mechanism which takes into consideration 235 reactions among 81 variable chemical species (in addition *O₂*, *H₂*, *CH₄*, and *H₂O* concentrations are considered fixed), and is currently incorporated into the three-dimensional regional-scale model STEM-II.

The 24 hours simulation interval starts at $t^0 = 12pm$ and ends at $t^F = 12pm$ the next day. We consider second order adjoints of two methods, SDIRK-4 and RODAS. All simulations are carried out with $rtol=1e-5$, $atol=1e-3$ molec/cm³.

We consider two different cost functions. The first one is the *PAN* concentration

at the final time, the second is half the O_3 concentration squared at the final time

$$\Psi^1 = PAN(t^F) \quad , \quad \text{and} \quad \Psi^2 = \frac{1}{2}O_3^2(t^F)$$

The initial NO_X concentrations are perturbed from their reference values as follows

$$NO(t^0) \leftarrow (1 + \varepsilon) \cdot NO^{\text{reference}}(t^0) \quad , \quad NO_2(t^0) \leftarrow (1 + \varepsilon) \cdot NO_2^{\text{reference}}(t^0) \quad (8.1)$$

For each cost function and each perturbation we compute the first order adjoints $\lambda^{1,2}(\varepsilon)$. For the reference solution we also compute the first and second order adjoints ($\lambda^{1,2}(0)$ and $\sigma^{1,2}$ respectively). Using the relation $\sigma^{1,2} \approx \lambda^{1,2}(\varepsilon) - \lambda^{1,2}(0)$ we validate our implementation by checking the second order adjoint against the finite difference of first order adjoints. Specifically we compute the RMS norm of the relative error for all n components

$$ERR^{1,2} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\lambda_i^{1,2}(\varepsilon) - \lambda_i^{1,2}(0) - \sigma_i^{1,2}}{\max(|\sigma_i^{1,2}|, tol)} \right)^2} \quad (8.2)$$

These relative errors are reported in Table 8.1. We see that for both cost functions and for both methods the agreement between the second order adjoint and the finite difference of first order adjoints is improved with decreasing the perturbation magnitude. The agreement for the RODAS method results on the first cost function is excellent.

ε	$\Psi^1 = PAN$		$\Psi^2 = 0.5 O_3^2$	
	SDIRK-4	RODAS	SDIRK-4	RODAS
0.1	1.15E-01	4.22E-07	1.22E-01	1.30E-01
0.01	2.99E-03	3.21E-09	7.90E-03	1.36E-02

Table 8.1: Validation of the second order adjoints against finite differences of first order adjoints. The RMS norm of the relative difference decreases for smaller perturbations.

Next we show how second order adjoints can be used in sensitivity analysis, and can extend the range of validity of sensitivity analysis for highly nonlinear chemical systems. The results are shown in Figure 8.1. The changes of PAN concentrations at the end of the 24 hours interval are nonlinear with respect to

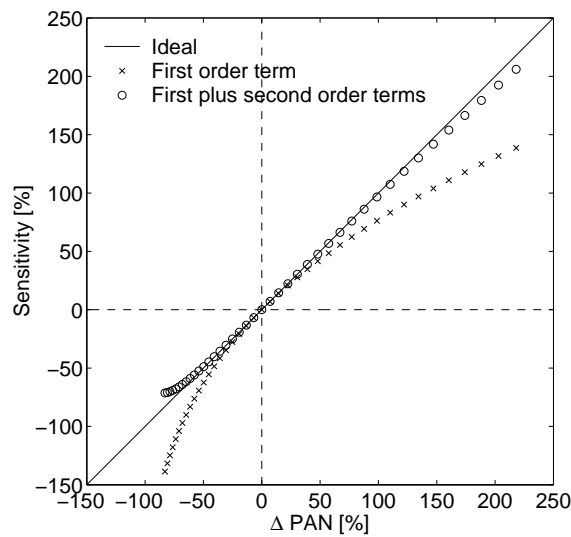


Figure 8.1: Sensitivity of final *PAN* concentration with respect to the initial concentrations of *NO* and *NO*₂. The changes in *PAN* concentration for different changes in the initial conditions Δc^0 are shown against the first order approximation ($\lambda^T \cdot \Delta c^0$, marked with “x”) and against the second order approximation ($\lambda^T \cdot \Delta c^0 + 1/2 \cdot \sigma^T \cdot \Delta c^0$, marked with “o”). The first order sensitivity analysis is inaccurate for this highly nonlinear system, and the second order sensitivity analysis predicts much better the change in *PAN*.

the initial concentrations of *NO* and *NO*₂. We let the initial *NO* and *NO*₂ initial concentrations vary according to (8.1) within $\pm 40\%$ from their reference values ($\varepsilon \in [-0.4, +0.4]$). The change in the final *PAN* concentration is predicted by first and second order Taylor series about the reference initial concentrations. The first and the second terms in the Taylor series are obtained using the first and the second order adjoints respectively:

$$\begin{aligned}
 \Psi^1(c^0) &= PAN(t^F)|_{c(t^0)=c^0}, \quad \Psi^1(c^0 + \Delta c^0) = PAN(t^F)|_{c(t^0)=c^0 + \Delta c^0}, \\
 \Delta PAN &= \Psi^1(c^0 + \Delta c^0) - \Psi^1(c^0) \\
 &= \left(\nabla_{c^0} \Psi^1(c^0) \right)^T \cdot \Delta c^0 + \frac{1}{2} (\Delta c^0)^T \cdot \left(\nabla_{c^0, c^0}^2 \Psi^1(c^0) \right) \cdot \Delta c^0 + \dots \\
 &= \lambda^T \cdot \Delta c^0 + \frac{1}{2} \sigma^T \cdot \Delta c^0 + \dots
 \end{aligned}$$

We see in Figure 8.1 that the first order approximation is poor for large perturbations, while the second order approximation continues to work well for large deviations from reference.

8.2 Optimization

We have validated the correctness of second order adjoints in the STEM and also checked the symmetry of Hessian previously. In the following we will use the STEM to perform data assimilation. We first discuss some optimization methods that use second order information in the form of Hessian-vector products.

8.2.1 Daniel's Nonlinear Conjugate Gradient Method

Daniel's nonlinear conjugate gradients method [13–15] uses explicit Hessian-vector products in the calculation of the new search direction. This approach has been traditionally considered impractical for large scale optimization problems due to the need for second order information [26]. Since second order adjoints can provide Hessian-vector products efficiently we revisit Daniel's method and use it to solve the data assimilation problem (7.17).

We next describe Daniel's method and show how it can be efficiently implemented using a single forward and backward model run (during which both first and second order adjoints are computed). In the first step one computes the gradient via one first order adjoint model run, and initializes the product of the Hessian and the search direction by either running the second order adjoint model or by approximating the Hessian with the identity matrix:

Initialization (we have $x_0 = y^B$)

- 0.1 Compute in a forward-backward run $g_0 = \left(\partial\Psi/\partial y(x_0)\right)^T$
- 0.2 Set $d_0 = -g_0$
- 0.3 Compute in another forward-backward run $v_0 = \partial^2\Psi/\partial y^2(x_0) \cdot d_0$
(or let $v_0 = d_0$)

For each iteration one constructs the one-dimensional quadratic model along the search direction, updates the point in state space, updates the gradient, the

search direction, and the product between the Hessian and the search direction. The computational cost at each step is dominated by one forward-backward run with the gradient evaluated by first order adjoint and two Hessian-vector products evaluated by the second order adjoint. Note that two Hessian-vector products can be computed simultaneously in a single backward run, and computational savings are possible by reusing the LU decompositions.

For $k \geq 1$ (we have $x = x_k$, d_k , $g_k = \partial\Psi/\partial y(x_k)$, and $v_k = \partial^2\Psi/\partial y^2(x_k) \cdot d_k$)

- | |
|---|
| <p>k.1 Find α_k via line-search such that $\Psi(x_k + \alpha d_k) \leq \Psi(x_k) + c_1 \alpha g_k^T d_k$</p> <p>k.2 Update the solution: $x_{k+1} = x_k + \alpha_k d_k$</p> <p>k.3 Compute in a single forward-backward run:</p> $g_{k+1} = \left(\partial\Psi/\partial y(x_{k+1}) \right)^T$ $a_{k+1} = \partial^2\Psi/\partial y^2(x_{k+1}) \cdot g_{k+1}$ $b_{k+1} = \partial^2\Psi/\partial y^2(x_{k+1}) \cdot d_k$ <p>k.4 Compute $\beta_k = (g_{k+1}^T v_k) / (d_k^T v_k)$ (which ensures that $d_{k+1}^T \partial^2\Psi/\partial y^2(x_k) d_k = 0$)</p> <p>k.5 Update the search direction: $d_{k+1} = -g_{k+1} + \beta_k d_k$</p> <p>k.6 Update the product: $v_{k+1} = \partial^2\Psi/\partial y^2(x_{k+1}) \cdot d_{k+1} = -a_{k+1} + \beta_k b_{k+1}$</p> |
|---|

Here we denote by x_k the vector of initial concentrations y^0 after k optimization iterations.

Integrating Line Search in Daniel's method

For step k.1, we implemented a basic backtracking line search function, which is based on Wolfe conditions.

a) sufficient decrease condition

$$f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T d_k \quad (8.3)$$

and b) curvature condition

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k \quad (8.4)$$

where $0 < c_1 < c_2 < 1$. d_k is search direction. $f(x_k)$ is the value of cost function in the k^{th} iteration while $f(x_k + \alpha_k d_k)$ is the value of cost function with new x . ∇f_k is the gradient of the cost function.

The sufficient decrease condition alone is not sufficient to ensure reasonable progress along the given search direction. But if we choose the candidate step lengths appropriately, as stated in the backtracking line search algorithm, it is safe to dispense the curvature condition. Backtracking line search goes as follows (with input $f(x_k), \alpha_k, \nabla f_k, d_k$):

For $k = 1, 2, \dots$
k.1 Set $\rho, c \in (0, 1)$
k.2 Evaluate $f(x_k + \alpha_k d_k)$
k.3 If $f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k \nabla f_k^T d_k$ then return with the current α_k Else $\alpha_{k+1} = \rho \alpha_k$, go to k.2

In practice, we set $\rho = 0.5, c = 10^{-4}$ as usually suggested. Also when $\alpha_k = 0.125 * \alpha_1$ (i.e., after three times' loop), we terminate the line search because otherwise, too many forward model calls will consume much time, and small step cannot bring about decent decrease, if any.

8.2.2 Hessian Free Newton

Hessian Free Newton method is discussed in the previous chapter about optimization methods using first order derivatives. Here the HFN is actually a modified HFN, in which we use the second order adjoints to replace the original finite difference scheme, to obtain Hessian-vector product.

The minimization of (7.17) can be carried out in principle using Newton's method. With x_k denoting the value of the solution after k Newton iteration the process is

$$\begin{aligned} \left(\frac{\partial^2 \Psi}{\partial y^2} (x_k) \right) \cdot \Delta x &= \left(\frac{\partial \Psi}{\partial y} (x_k) \right)^T \\ x_{k+1} &= x_k - \Delta x \end{aligned} \quad (8.5)$$

Each iteration requires to solve a linear system. The system matrix is the Hessian and the right hand side vector the gradient computed at the current iterate. Since the Hessian is a large symmetric matrix, a sensible approach is to solve the system using the linear conjugate gradients iterative method. The linear system solution

(8.5) needs to be only as accurate as the solution of the nonlinear system. Therefore one can stop the conjugate gradient process after only a few iterations.

Consequently the process involves two iterations: the outer iteration is Newton iteration to update solution and the inner iteration requires to solve a linear system. Each outer Newton iteration requires several inner iterations of the linear conjugate gradients to solve (8.5). Each of the inner iterations performs one forward integration of the forward and the tangent linear models (7.18), followed by one reverse integration of the first and second order adjoint models (7.19). The computational cost of each inner iteration is therefore relatively expensive.

For the numerical experiments we use the hybrid code of Morales and Nocedal [44] to test the stand-alone HFN method. This code also implements an enriched optimization algorithm that allows to interlace L-BFGS and HFN iterations and use the information collected by one type of iteration to improve the performance of the other. In the numerical experiments reported here we alternate five L-BFGS iterations with one HFN iteration.

8.2.3 Optimization Results

Data assimilation experiments use a STEM model simulation of air quality in Northeastern United States. The 12 hours data assimilation window starts at 12 GMT (8 EDT) on July 20th, 2004. The settings are the same as mentioned in Test Case Two, Chapter 5. We assess the performance of five optimization methods used to minimize the cost function (7.17). L-BFGS and the Fletcher-Reeves Nonlinear Conjugate Gradients (FR-CG) methods require only first order derivative information. Daniel Nonlinear Conjugate Gradients (Daniel-CG), HFN and the hybrid methods require second order derivative information. Since L-BFGS is considered the gold standard in variational data assimilation we will use its solution as a reference.

When solving real large-scale variational data assimilation problems with very expensive evaluations of the function, the gradient, and the Hessian-vector products the optimization process is typically not run to convergence. In practice the number of iterations is predefined (based on an estimate of the feasible computational time). Following this approach in our numerical experiments each method is allowed to

take a fixed number of ten iterations. Each iteration of L-BFGS finds a new solution point (“NEW_X”), and can use multiple model runs during the line search. For HFN we consider ten Newton (outer) iterations; each iteration finds a new solution point, and can use multiple inner (linear conjugate gradients) iterations. For the Fletcher-Reeves and the Daniel nonlinear conjugate gradients each iteration produces a new solution point.

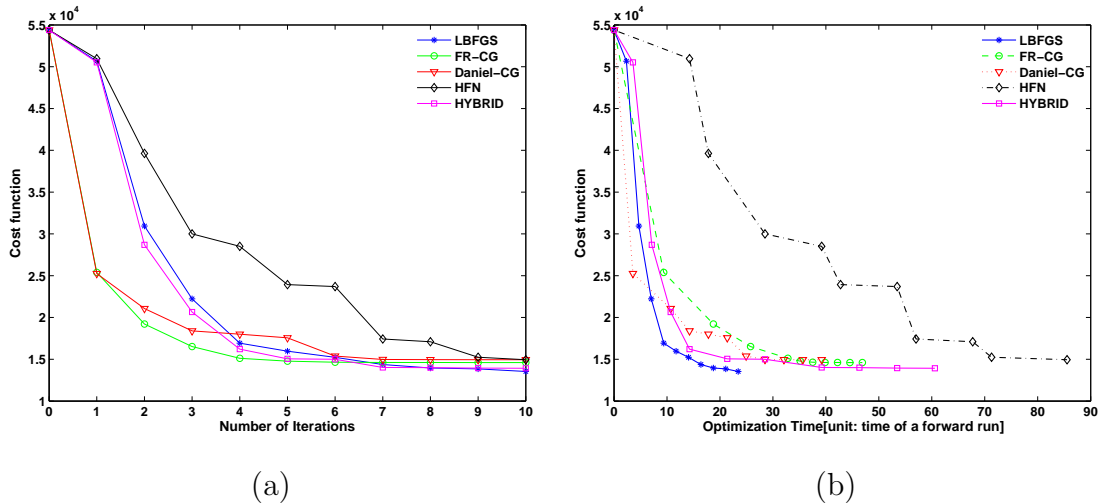


Figure 8.2: (a) Decrease of the cost function vs. number of iterations. (b) Decrease of the cost function vs. scaled CPU time.

The decrease of the cost function versus the number of iterations is reported in Figure 8.2(a). All methods are able to drive the cost function from a value of about 55000 down to about 14000 after ten iterations. Beyond ten iterations further decrease in the cost function is small, indicating that all solutions have approached the optimum.

The decrease of the cost function versus the computational time is reported in Figure 8.2(b). On the abscissa we use scaled time units, where one unit is the cpu time of one forward run (with only the nonlinear model, and without any derivative calculations). The cost of each optimization is estimated based on the number of model runs and the relative timings for the first and second order adjoint calculations given in Table 7.1.

The results in Figure 8.2(b) indicate that that L-BFGS method is the most efficient method. Daniel’s CG method performs better than FR-CG, especially

during the first few iterations. HFN converges toward the solution in a small number of outer iterations, but at the cost of many inner iterations. This makes the total computational cost of HFN to be the highest among all methods. The hybrid method starts with five L-BFGS iterations, and during them its performance is similar to that of L-BFGS. After HFN is called the hybrid method becomes slightly slower than L-BFGS.

The quality of each optimization solution is measured by the norm of the gradient of the cost function, and by the R^2 correlation factor and root mean square (RMS) difference between the observations and the model predictions (when the model is initialized with the solution of the optimization process $y^0 = y^a$).

	BG	LBFGS	FR-CG	Daniel-CG	HFN	HYBRID
$\ \partial\Psi/\partial y\ $	4147.38	493.09	757.70	490.61	795.83	559.46
RMS	24.76	11.94	12.67	12.68	12.93	12.24
R^2	0.15	0.68	0.65	0.64	0.64	0.67

Table 8.2: The quality of different optimized solutions measured by the norm of gradient, the correlation coefficient, and root mean square distance between model predictions and observations.

Table 8.2 shows the norm of the gradient of the cost function, the R^2 correlation factor, and the RMS difference between observations and model predictions when initialized with the background and with each of the optimization solutions. A good solution has a small norm of gradient, a small RMS difference between observations and model predictions, as well as a large correlation coefficient between observations and model predictions.

The results in Table 8.2 indicate that all optimized solutions show a considerable improvement from the background state. Model predictions are much closer to the observations (in both the R^2 and the RMS metrics) when the simulation is initialized with any of the optimal solutions. The norm of gradient indicates that the L-BFGS and Daniel solutions are the closest to the optimum, while the HFN solution is the farthest. Overall the L-BFGS solution is slightly better than the other, and considering the computational time we conclude that L-BFGS performs best on the

data assimilation problem under consideration.

The scatter and quantile-quantile plots of Figure 8.3 also illustrate that the correlation between model predictions (represented on the y-axes) and observations (represented on the x-axes) increases considerably from $R^2 = 0.15$ for the original model to $R^2 = 0.68$ after L-BFGS assimilation. Figure 8.3 (a) shows a large spread of the scatter plot and a visible biased quantile-quantile plot. However, after data assimilation, we can observe in Figure 8.3 (b) a clustered scatter plot, as well as a quantile-quantile plot much closer to midline.

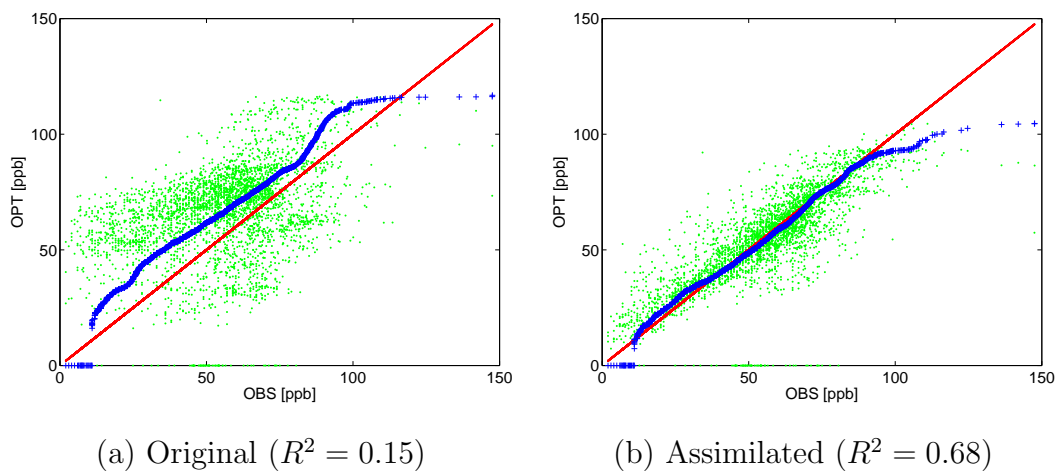


Figure 8.3: Scatter plots and quantile-quantile plots of model-observations agreement: (a) before data assimilation, and (b) after data assimilation.

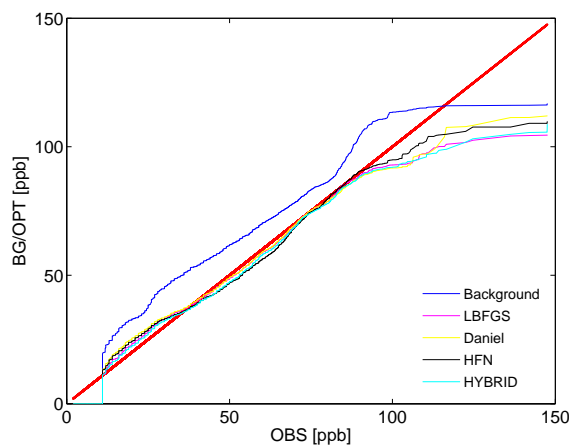


Figure 8.4: Quantile-Quantile plot for Background, L-BFGS, Danile, HFN and hybrid methods.

Figure 8.4 gives a contrast between the Background, L-BFGS, Daniel-CG, HFN

and hybrid methods. We see that L-BFGS, Daniel-CG, HFN and hybrid quantile-quantile plots overlap with the ideal line for most of the range of values, showing a considerable decrease in model results bias.

The ground level ozone fields at 1pm EDT of July 20, 2004 using L-BFGS-B solutions as initial conditions are shown in Figure 8.5. Visually there is a better agreement between model predictions and observations after assimilation, especially near the West boundary.

We also plot the absolute value of difference at initial time between L-BFGS solution and the background concentrations, L-BFGS and Daniel's method, L-BFGS and HFN, as well as L-BFGS and hybrid in Figure 8.6. There is some distinction between L-BFGS optimized solution and background values in the range $0ppb$ to $40ppb$, but much smaller difference between L-BFGS and other optimization solutions, ranging from $0ppb$ to $8ppb$. L-BFGS solution has relatively large difference from Daniel's solution in the New York City and its surrounding area, as well as the boundary crossing Ohio. HFN gives different solution from L-BFGS also on the west boundary, and area between eastern Pennsylvania and western Connecticut. Hybrid matches best with L-BFGS solution.

To show the time evolution of ground level ozone concentrations we select four AirNow stations A–D, shown in Figure 5.8 (b). The ozone time series initialized using the background and L-BFGS solution at these four stations has already been illustrated in Figure 5.11.

To show the differences between L-BFGS and Daniel's solutions, L-BFGS and HFN solutions, as well as the difference between L-BFGS and hybrid solutions, we plot in Figure 8.7 time series at the station C. From the figure, we cannot tell much differences between different optimized solutions, which also implies that all these methods generate similar solutions. L-BFGS, Daniel, HFN, and hybrid time series lines all show better estimation than background line to observations. Daniel and hybrid lines are more close to L-BFGS line, while HFN is slightly biased from L-BFGS line.

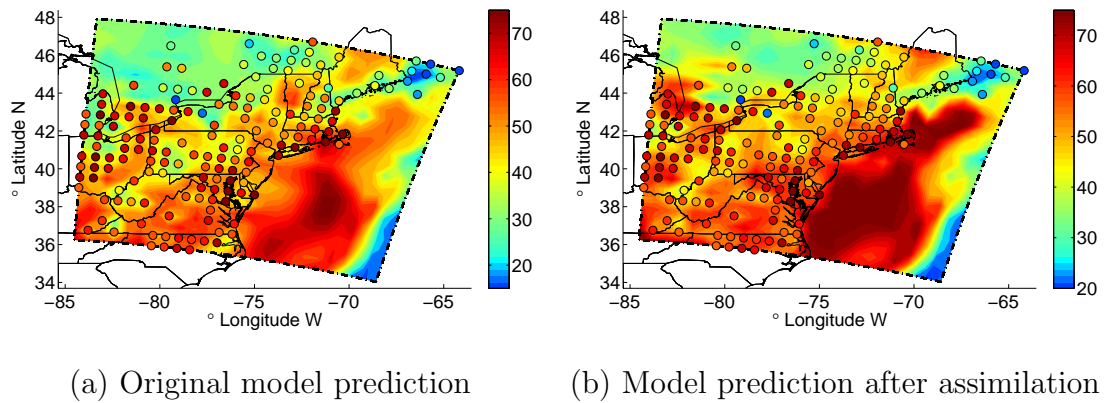


Figure 8.5: Ground level ozone distribution in northeastern U.S. at 1pm EDT on July 20, 2004. (a) before data assimilation, and (b) after data assimilation.

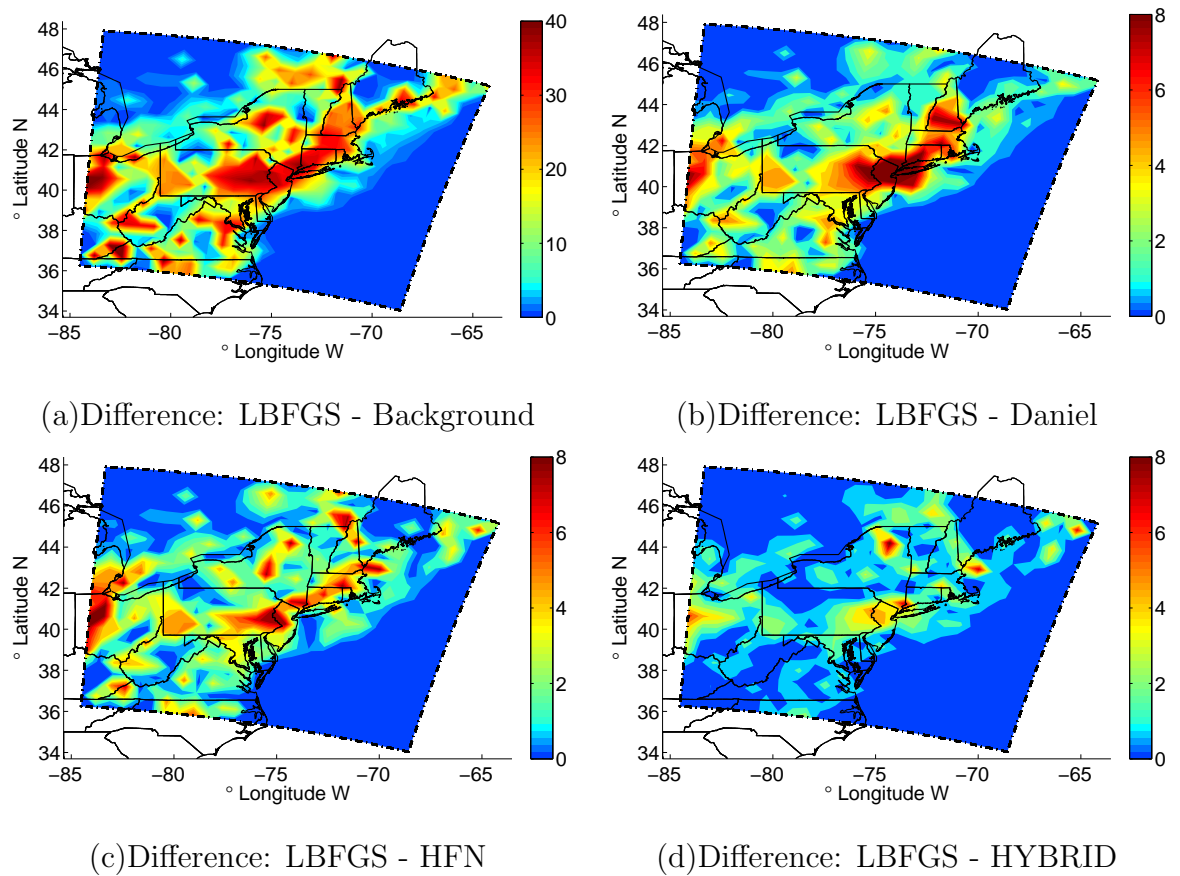


Figure 8.6: Difference between optimization solutions.

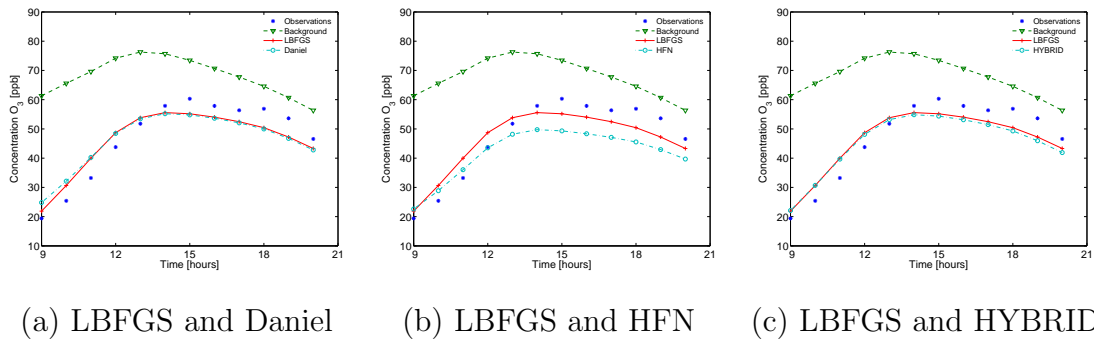


Figure 8.7: Time series of ozone concentrations at station C for L-BFGS, Daniel, HFN and hybrid solutions.

8.3 Uncertainty Quantification

Recall that when the model is linear, and the background and observation uncertainties are Gaussian, the a posteriori probability density of the initial state is also Gaussian (with mean y^a and covariance $P^a(t^0)$, $y^0 \in \mathcal{N}(y^a, P^a(t^0))$). In this case the cost function (7.17) is quadratic and represents the negative logarithm of the a posteriori Gaussian probability density function

$$\begin{aligned} \Psi(y^0) &= -\log p^a(y^0), \\ p^a(y^0) &= \text{const} \times \exp\left(-\frac{1}{2}(y^0 - y^a)^T (P^a(t^0))^{-1} (y^0 - y^a)\right) \end{aligned}$$

It is easy to see that the Hessian of the cost function equals the inverse of the a posteriori covariance matrix, $\partial^2 \Psi / \partial y^2 = (P^a(t^0))^{-1}$.

For nonlinear models with non-Gaussian uncertainty probability densities one solves the nonlinear minimization problem (7.17) to obtain the analyzed initial condition

$$y^a = \arg \min_{y^0} \Psi(y^0)$$

The Hessian of the cost function (7.17), evaluated at the optimal initial condition y^a , offers an approximation of the a posteriori covariance matrix of the uncertainty in the analyzed initial conditions:

$$P^a(t^0) \approx \left(\frac{\partial^2 \Psi}{\partial y^2}(y^a) \right)^{-1} \quad (8.6)$$

We expect this to be a good approximation if the errors are relatively small, if

their propagation in time obeys the tangent linear model, and if the distribution of uncertainty is not far from Gaussian.

Our goal is now to characterize the a posteriori errors, i.e., to quantify the uncertainty in the initial state y^a after the assimilation of observations. For this let (λ_i^P, v_i) , $i = 1, \dots, n$, be the eigenvalue-eigenvector pairs of the a posteriori covariance matrix $P^a(t^0)$. The eigenvectors are orthogonal to each other (because of symmetry) and have norm one. Moreover, all the eigenvalues are non-negative $\lambda_i^P \geq 0$.

Under the Gaussian assumption the a posteriori error in the initial condition is a Gaussian random process which can be described in terms of the eigenvalues and eigenvectors of the covariance matrix

$$Err = y^0 - y^a = \sum_{i=1}^n \xi_i \sqrt{\lambda_i^P} v_i, \quad \xi_i \in \mathcal{N}(0, 1), \quad (8.7)$$

where ξ_i are independent Gaussian random variables. The principal components $\sqrt{\lambda_i^P} v_i$ of the a posteriori error are along the directions of the largest eigenvalues of the covariance matrix. According to (8.6) the largest eigenvalues of the covariance matrix are (approximated by) the inverses of the smallest Hessian eigenvalues $\lambda_i^P = 1/\lambda_i^H$, while the corresponding eigenvectors are the same. To characterize the a posteriori error we estimate its principal components (8.7) from the Hessian eigenvalues and eigenvectors as follows.

The largest five and the smallest five eigenvalues of the Hessian of the cost function, evaluated at the minimum argument y^a , are computed using the ARPACK package [37]. The second adjoint model is used to provide the Hessian-vector products required by ARPACK. These eigenvalues are reported in Table 8.3. The inverses of the Hessian eigenvalues approximate the eigenvalues of the a posteriori covariance matrix $P^a(t^0)$ and are also reported in Table 8.3. These eigenvalues represent variances of the principal components (equation 8.7) in (molecules of O_3 per cm^3 of air)². The square root of the covariance eigenvalues represent the standard deviations of each of the principal components; we report the standard deviations in the more convenient units of parts-per-billion (ppb). The conversion is done by dividing the concentration to the ground level air density ($\rho = 2.4 \times 10^{19}$ mlc/cm³) and multi-

plying the results by 10^9 . We see that the error is dominated by the first principal component (along which the standard deviation is 47 ppb).

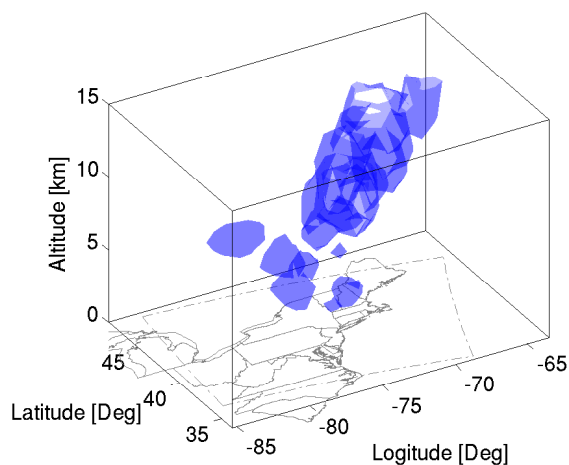
	First	Second	Third	Fourth	Fifth
$\lambda_{\text{small}}^H (\text{mlc}/\text{cm}^3)^{-2}$	7.54×10^{-25}	1.15×10^{-23}	4.04×10^{-23}	8.47×10^{-23}	1.42×10^{-22}
$\lambda_{\text{large}}^P (\text{mlc}/\text{cm}^3)^2$	1.33×10^{24}	8.70×10^{22}	2.48×10^{22}	1.18×10^{22}	7.04×10^{21}
$\sqrt{\lambda_{\text{large}}^P}$ (ppb)	47	12	7	4	3
$\lambda_{\text{large}}^H (\text{mlc}/\text{cm}^3)^{-2}$	4.17×10^{-22}	3.79×10^{-22}	3.34×10^{-22}	2.76×10^{-22}	2.12×10^{-22}
$\lambda_{\text{small}}^P (\text{mlc}/\text{cm}^3)^2$	2.40×10^{21}	2.64×10^{21}	3.00×10^{21}	3.62×10^{21}	4.72×10^{21}
$\sqrt{\lambda_{\text{small}}^P}$ (ppb)	2.04	2.14	2.28	2.51	2.86

Table 8.3: The smallest and largest five eigenvalues of the Hessian and the corresponding eigenvalues of the a posteriori covariance matrix.

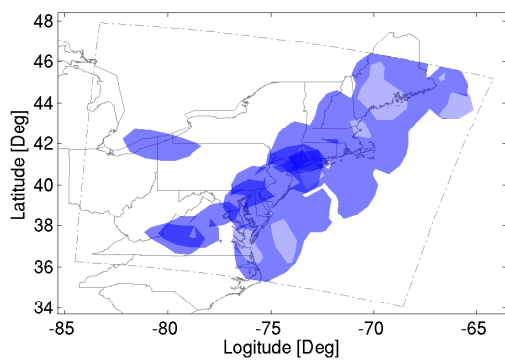
To visualize the spatial distribution of the error we plot the 2 ppb isosurfaces of the first three principal error components $\sqrt{\lambda_1^P} v_1$, $\sqrt{\lambda_2^P} v_2$, and $\sqrt{\lambda_3^P} v_3$ in Figure 8.8, Figure 8.9, and Figure 8.10 respectively. The unit conversion from mlc/cm^3 to ppb is done using the appropriate air density in each vertical layer. As expected for the first principal error component the 2 ppb isosurface covers the largest area, see Figure 8.8. For all three principal components the error is located at high altitudes. This can be explained by the dense observational network at the ground level used in this data assimilation study, see Figure 5.8; the assimilation of these observations reduces the uncertainty in ozone initial concentrations at low altitudes. In contrast the number of observations at high altitudes is low and considerable uncertainty remains after data assimilation. One possible conclusion is that more high altitude observations are needed to further reduce the global level of uncertainty.

8.4 Directions of Fastest Error Growth

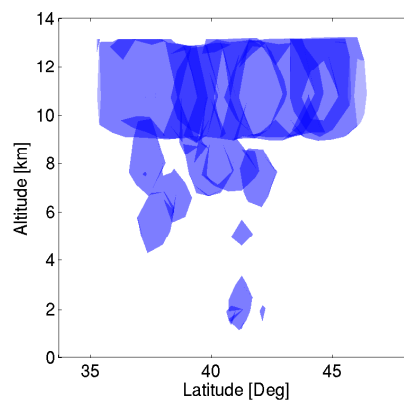
We now look into the problem of how uncertainties propagate forward in time through the model. Specifically we want to estimate which perturbations at the initial time grow to have the largest impact on the solution accuracy at the final time. These “directions of maximal error growth” [2] are important in several applications. First, in order to have an accurate forecast (an accurate solution at the final time) one needs to reduce the uncertainty in the initial state along these directions [39]. New observations added to increase the accuracy of the simulation



(a) 3D View, 2 ppb Error

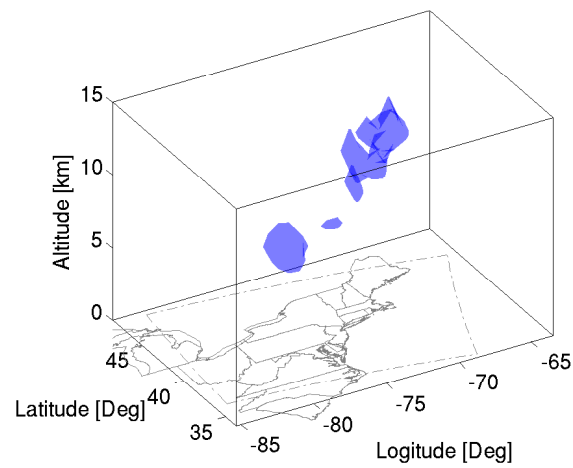


(b) Top View, 2 ppb Error

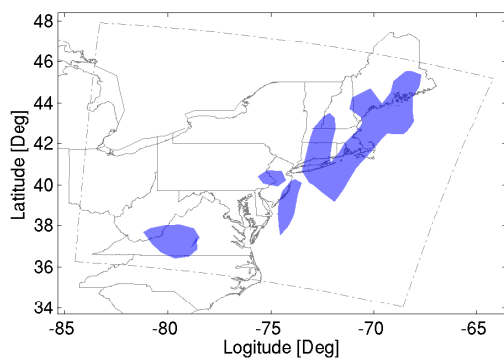


(c) East View, 2 ppb Error

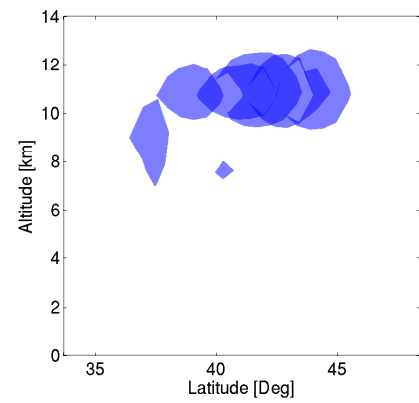
Figure 8.8: First principal component of the error in the initial ozone field. The 2 ppb error isosurface is shown in (a) 3D View, (b) Top View, and (c) East View.



(a) 3D View, 2 ppb Error

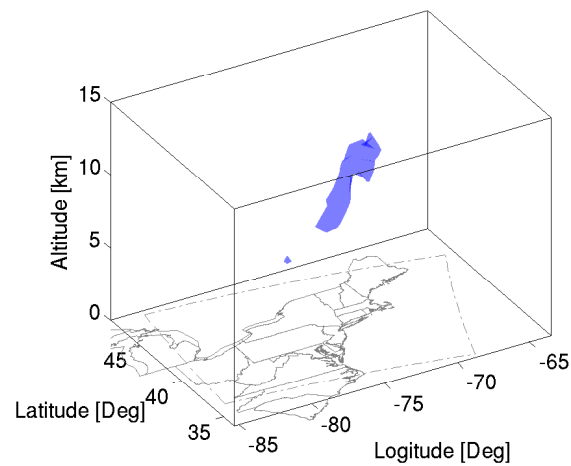


(b) Top View, 2 ppb Error

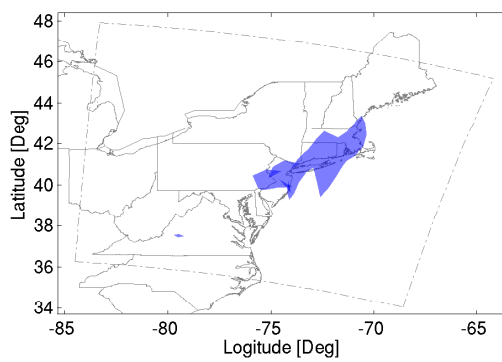


(c) East View, 2 ppb Error

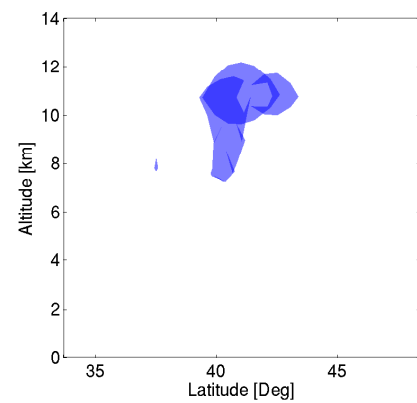
Figure 8.9: Second principal component of the error in the initial ozone filed. The 2 ppb error isosurface is shown in (a) 3D View, (b) Top View, and (c) East View.



(a) 3D View, 2 ppb Error



(b) Top View, 2 ppb Error



(c) East View, 2 ppb Error

Figure 8.10: Third principal component of the error in the initial ozone field. The 2 ppb error isosurface is shown in (a) 3D View, (b) Top View, and (c) East View.

(through data assimilation) are most useful if placed along these directions [38]. Next, in a Monte Carlo approach, a small ensemble of runs can represent well the uncertainty in a large-dimensional system if it is initialized with perturbations along the directions of maximal error growth [1].

Following (equation 3.14) and (equation 3.24), we denote by \mathcal{N}' , \mathcal{N}'^* the tangent linear and the adjoint model solution operators on the interval $[t^0, t^F]$. The model is initialized at t^0 with the optimal state y^a (for which the error covariance is $P^a(t^0)$). Perturbations (small errors) in the initial conditions δy^0 propagate forward in time according to the tangent linear model (7.15), and grow at the final time to

$$\delta y(t^F) = \mathcal{N}' \delta y^0 . \quad (8.8)$$

The error covariance matrix $P^a(t^0)$ evolves into the forecast error covariance matrix at t^F

$$P^f(t^F) = \mathcal{N}' \cdot P^a(t^0) \cdot \mathcal{N}'^* .$$

The principal components of the forecast uncertainty (uncertainty at the final time) are along the dominant eigenvectors of the forecast error covariance matrix $P^f(t^F)$. We want to find the directions δy^0 at the initial time which grow through (8.8) into the dominant eigenvectors of P^f at the final time. We have that:

$$\begin{aligned} P^f(t^F) \delta y(t^F) = \lambda_{\max} \delta y(t^F) &\Leftrightarrow (\mathcal{N}' \cdot P^a(t^0) \cdot \mathcal{N}'^*) \mathcal{N}' \delta y^0 = \lambda_{\max} \mathcal{N}' \delta y^0 \\ &\Leftrightarrow \mathcal{N}'^* \mathcal{N}' \delta y^0 = \lambda_{\max} (P^a(t^0))^{-1} \delta y^0 \end{aligned}$$

The inverse covariance matrix can be approximated by the Hessian of the cost function (8.6). We see that the dominant eigenvectors in this case are the solution of the generalized eigenvalue problem

$$\mathcal{N}'^* \mathcal{N}' \delta y^0 = \lambda_{\max} \left(\frac{\partial^2 \Psi}{\partial y^2}(y^0) \right) \delta y^0 \quad (8.9)$$

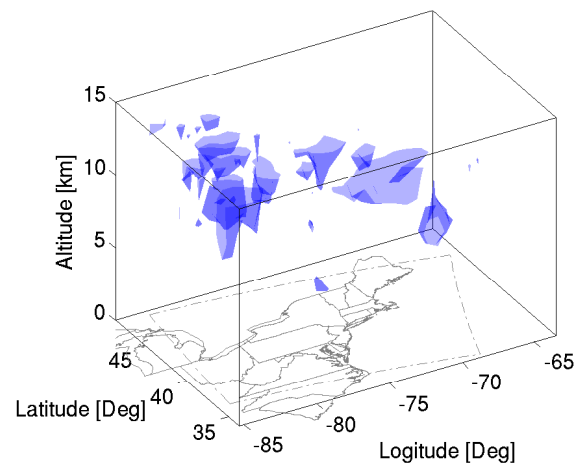
The generalized eigenvectors in (8.9) are called *Hessian singular vectors* in the data assimilation literature [1]. The matrix times vector products $\mathcal{N}'^* \mathcal{N}' \delta y^0$ needed to evaluate the left hand side are computed by one forward integration of the TLM ($\delta y(t^F) = \mathcal{N}' \delta y^0$) followed by one backward integration of the adjoint ($\mathcal{N}'^* \delta y(t^F)$). The adjoint variable is initialized with the final value of the TLM integration. The

Hessian times vector products needed to evaluate the right hand side are obtained by the second order adjoint.

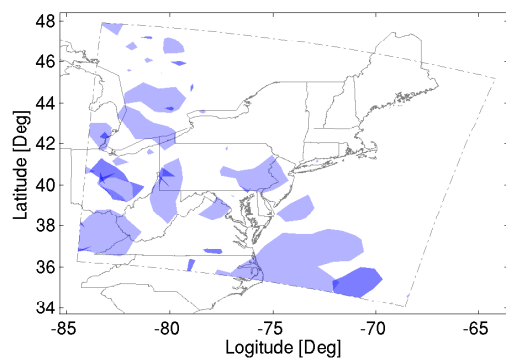
For numerical experiments we run the STEM model for 8 hours. The simulation is initialized with the optimal solution of the data assimilation problem given by L-BFGS. The dominant generalized eigenvalues (8.9) are computed using the JDQZ package which implements a Jacobi-Davidson algorithm [56]. Table 8.4 shows the largest five generalized eigenvalues (8.9). We see that the fifth generalized eigenvalue is two orders of magnitude smaller than the first. few directions at the initial time have a large impact on the final time uncertainty. Figure 8.11 presents the Hessian singular vector associated with the largest generalized eigenvalue in Table 8.4. Most of the area is at high altitudes, which is not surprising given that most of the uncertainty in the ozone field is at high altitudes.

	First	Second	Third	Fourth	Fifth
λ	0.16×10^{-15}	0.12×10^{-15}	0.66×10^{-16}	0.59×10^{-16}	0.17×10^{-17}

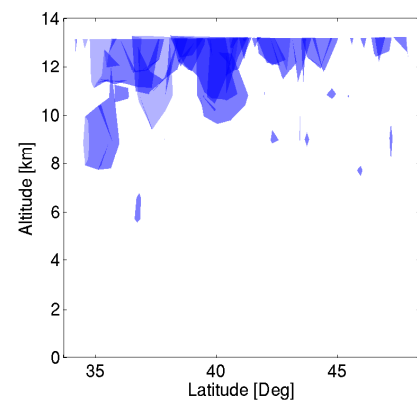
Table 8.4: The largest five Hessian singular eigenvalues.



(a) 3D View



(b) Top View



(c) East View

Figure 8.11: The 0.02 isosurface of the dominant Hessian singular vector: (a) 3D view, (b) Top view, and (c) East view.

Chapter 9

Conclusion

9.1 Conclusion

The adjoint model can be used to efficiently compute the derivatives of a functional with respect to a large number of model parameters. First order adjoints compute the first derivative of a functional, while second order adjoints provide second order information of a functional. In this research we explore applications of first order adjoints in the area of air quality analysis. Moreover, we propose the construction of second order adjoints in chemical transport models for air quality simulations.

Using first order adjoints, we perform sensitivity analysis to show the area that brings large influence on ozone concentrations over Dallas Fort Worth, Texas. We also assess the performance of L-BFGS, Fletcher Reeves Conjugate Gradient method, Hessian Free Newton method and hybrid method, using different background and both real observation data and artificial data. L-BFGS is the best of all these methods no matter which background or data set we use. We can say that L-BFGS is robust and thus may be currently the optimal candidate to perform data assimilation on our STEM chemical transport model.

The second order adjoints give the second derivatives in the form of Hessian-vector product. Usually in chemical transport models it is prohibitive to compute and store the Hessian matrix due to a large number of model states. However, with the help of second order adjoints, many applications which need second order information have become possible, and have generated plenty of results through

several applications, such as optimization, uncertainty quantification, and directions of fastest error growth.

In data assimilation, we use Daniel's method, modified Hessian Free Newton method and hybrid method to obtain optimized initial conditions. The results are encouraging in that all of these methods are able to obtain optimal states as done by first order optimization methods. Daniel's method is even faster than Fletcher Reeves method. Although it is not as efficient as L-BFGS method, it is still a promising optimization method given that L-BFGS has been improved by many people for long time. In this work, we implemented the Daniel's method with a simple backtracking line search algorithm. If some advanced line search or preconditioning scheme is incorporated in Daniel's method, it might beat L-BFGS one day. Similarly, the hybrid method is proposed in 2002, far later than L-BFGS, so there is still space for these methods to be improved. Our work gives hints on the promising future of these methods: if well written and improved, they might become powerful optimization methods to solve large-scale nonlinear problems, not only in air quality models, but in many other areas. What's more, we would like to point out that second order adjoints are important in this application. With the development of the optimization method that requires second order information, such as Daniel's method, second order adjoints give us another promising way or maybe better way to solve problems besides first order adjoints.

Second order adjoints also contribute to computing the eigenvalues of the Hessian matrix and Hessian singular vectors, which are helpful in analyzing uncertainty and identifying the directions of fastest error growth. We find that the uncertainties come more from high altitude than low altitude. The results match the reality that the ground observations help to decrease the uncertainty at that area.

9.2 Contributions

Throughout the thesis we explore adjoint methods and their applications in chemical transport models. Our goal is to analyze the air quality from all aspects by encapsulating the adjoint models in large-scale simulations.

For applications of first order adjoints we compare the direct sensitivity and adjoint sensitivity analysis in chemical transport models. By employing the adjoint sensitivity we discover the influence areas which bring about the changes of ozone concentrations in Texas. We have shown that data assimilation can improve both the forecasts over Texas and the northeastern United States. We also design different scenarios (AR background, NMC background with observation data and AR background with artificial data) to assess the performances of optimization methods like L-BFGS, HFN, FR-CG in data assimilation.

Unlike first order adjoints, second order adjoints have not been used to date in the air quality simulation area. No theoretical descriptions of construction of second order adjoints for large-scale chemical transport models were given before. We explore the computation of discrete second order adjoints in chemical transport models, and validate the correctness of computing the adjoints in the 3D STEM model. Moreover, we evaluate the CPU time to run second order adjoints on the STEM model. With second order adjoints we check the Hessian matrix for symmetry, so as to guarantee that the conjugate gradient method can be used in the data assimilation process. After that, we demonstrate the value of second order adjoints in helping conduct and improve applications like sensitivity analysis, optimization, uncertainty quantification and Hessian singular vectors.

9.3 Future Work

As we have shown, the second order adjoints are computationally feasible and important for several applications in the air quality simulation area; we will further explore other applications that rely on second order adjoint models. One research direction is to discover the targeted area which will reduce the forecast uncertainty if we put more measurements there.

For the optimization application of second order adjoints, we have shown that Daniel's method is only inferior to the L-BFGS method. In this work, Daniel's method is implemented only to show the feasibility of second order adjoints. Future work will focus on incorporating some advanced techniques in this method, so as to

improve its convergence speed.

In the aspect of construction of second order adjoints in chemical transport models, we have presented both discrete and continuous second order adjoints. Unlike first order adjoints, the consistency of discrete and continuous approaches in 3D chemical transport models hasn't been well explored. This issue is worth further study. In this work we have described discrete second order adjoints, in which the chemical reactions are integrated by KPP. Currently KPP provides Runge-Kutta, Rosenbrock, and SDIRK integrators to solve second order adjoints. The approximation accuracy and performance of these methods in chemical transport models should be compared and investigated.

Bibliography

- [1] J. Barkmeijer, R. Buizza, and T.N. Palmer. 3D-Var Hessian singular vectors and their potential use in the ECMWF Ensemble Prediction System. *Quarterly Journal of the Royal Meteorological Society*, 125:2333-2351, 1999.
- [2] J. Barkmeijer, R. Buizza, T.N. Palmer, K. Puri, and J.F. Mahfouf. Tropical singular vectors computed with linearized diabatic physics. *Quarterly Journal of the Royal Meteorological Society*, 127:685-708, 2001.
- [3] D.G.Cacuci. Sensitivity theory for nonlinear systems. I. Nonlinear functional analysis approach. *Journal of Mathematical Physics*, 22:2794-2802, 1981.
- [4] D.G.Cacuci. Sensitivity theory for nonlinear systems. II. Extensions to additional classes of responses. *Journal of Mathematical Physics*, 22:2803-2812, 1981.
- [5] G.R. Carmichael, L.K. Peters, R.D. Saylor. The STEM-II regional scale acid deposition and photochemical oxidant model - I. An overview of model development and applications. *Atmospheric Environment*, 25A:2077-2090, 1990.
- [6] G.R. Carmichael, T. Chai, A. Sandu, E.M. Constantinescu, and D. Daescu. Predicting Air Quality. *Journal of Computational Physics*, 2006. Submitted.
- [7] W.P.L. Carter. Implementation of the SAPRC-99 Chemical Mechanism into the Models-3 Framework. *Technical Report*, Report to the United States Environmental Protection Agency, January 2000.

- [8] W.P.L. Carter. Documentation of the SAPRC-99 Chemical Mechanism for VOC Reactivity Assessment. *Technical Report*, No. 92-329, and 95-308, Final Report to California Air Resources Board Contract, May 2000.
- [9] T. Chai, G.R. Carmichael, A. Sandu, Y. Tang, and D. Daescu. Chemical data assimilation of Transport and Chemical Evolution over the Pacific (TRACE-P) aircraft measurements. *Journal of Geophysical Research*, 111(D02301), 2006.
- [10] E.M. Constantinescu, T. Chai, A. Sandu, and R. Gregory. Autoregressive Models of Background Errors for Chemical Data Assimilation. *Technical Report*, TR-06-22, Computer Science, Virginia Tech.
- [11] D. Daescu, A. Sandu, and G.R. Carmichael. Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: II - Numerical Validation and Applications. *Atmospheric Environment*, 37:5097-5114, 2003.
- [12] V. Damian, A. Sandu, M. Damian, F. Potra, and G.R. Carmichael. The Kinetic Preprocessor kpp - A Software Environment for Solving Chemical Kinetics. *Computers and Chemical Engineering*, 26:1567-1579, 2002.
- [13] J.W. Daniel. The Conjugate Gradient Method for Linear and Nonlinear Operator Equations. *SIAM Journal on Numerical Analysis*, 4(1):10-26, Mar 1967.
- [14] J.W. Daniel. Convergence of the conjugate gradient method with computationally convenient modifications. *Numer. Math.*, 10:125-131, 1967.
- [15] J.W. Daniel. A Correction Concerning the Convergence Rate for the Conjugate Gradient Method. *SIAM Journal on Numerical Analysis*, 7:277-280, 1970.
- [16] R.S. Dembo, and T. Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26:190-212, 1983.
- [17] A.M. Dunker. The decoupled direct method for calculating sensitivity coefficients in chemical kinetics. *Journal of Chemical Physics*, 81:2385-2393, 1984.

- [18] H. Elbern, H. Schmidt, and A. Ebel. Variational data assimilation for tropospheric chemistry modeling. *Journal of Geophysical Research*, 102(D13):15967-15985, 1997.
- [19] H. Elbern and H. Schmidt. A 4D-Var chemistry data assimilation scheme for Eulerian chemistry transport modeling. *Journal of Geophysical Research*, 104(5):18583-18598, 1999.
- [20] H. Elbern and H. Schmidt, O. Talagrand, and A. Ebel. 4D-variational data assimilation with an adjoint air quality model for emission analysis. *Environmental Modeling and Software*, 15:539-548, 2000.
- [21] H. Elbern and H. Schmidt. Ozone episode analysis by 4D-Var chemistry data assimilation. *Journal of Geophysical Research*, 106(D4):3569-3590, 2001.
- [22] Q. Errera, and D. Fonteyn. Four-dimensional variational chemical assimilation of CRISTA stratospheric measurements. *Journal of Geophysical Research*, 106(D11):12253-12265, 2001.
- [23] M. Fisher and D.J. Lary. Lagrangian four-dimensional variational data assimilation of chemical species. *Quarterly Journal of the Royal Meteorological Society*, 121:1681-1704, 1995.
- [24] J.C. Gilbert, and J. Nocedal. Global Convergence Properties of Conjugate Gradient Methods for Optimization. *SIAM Journal on Optimization* 2:21-42, 1992.
- [25] K.W. Gwak, and G.Y. Masada. Regularization embedded nonlinear control designs for input-constrained and ill-conditioned thermal system. *Journal of Dynamic Systems, Measurement, and Control*, 126(3):574-582, 2004.
- [26] W.W. Hager, and H. Zhang. A New Conjugate Gradient Method with Guaranteed Descent and an Efficient Line Search. *SIAM Journal on Optimization*, 16:170-192, 2005.
- [27] W.W. Hager, and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2:35-58, 2006.

- [28] A. Hakami, J.H. Seinfeld, T. Chai, Y. Tang, G.R. Carmichael, and A. Sandu. Adjoint Sensitivity Analysis of Ozone Nonattainment over the Continental United States. *Environmental Science and Technology*, 2006.
- [29] D.K. Henze, and J.H. Seinfeld. Development of the adjoint of GEOS-Chem. *Atmospheric Chemistry and Physics Discussions*, 6:10591-10648, 2006.
- [30] B.V. Khattatov, J.C. Gille, L.V. Lyjak, G.P. Brasseur, V.L. Dvortsov, A.E. Roche, and J. Walters. Assimilation of photochemically active species and a case analysis of UARS data. *Journal of Geophysical Research*, 104:18715-18737, 1999.
- [31] W.H. Hundsdorfer, and J.G. Verwer. Numerical Solution of Time-Dependent Advection-Deffusion-Reaction Equations. *Volume 33 of Springer Series in Conmputational Mathematics*. Springer Verlag, 2003.
- [32] International Consortium for Atmospheric Research on Transport and Transformation (ICARTT). ICARTT web site: <http://www.al.noaa.gov/ICARTT>, 2006.
- [33] S. Lakshmivarahan, Y. Honda, and J.M. Lewis. Second-order approximation to the 3DVAR cost function: application to analysis/forecast. *Tellus*, 55(5):371-384, 2003.
- [34] D. Lanser and J.G. Verwer. Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modeling. *Centrum voor Wiskunde en Informatica Report*, MAS-R9805, 1998.
- [35] F.X. Le Dimet, H.E. Ngodock, B. Luong, and J. Verron. Sensitivity analysis in variational data assimilation. *J. Meteor. Soc. Japan*, 75(B):245-255, 1997.
- [36] F.X. Le Dimet, I.M. Navon, and D. Daescu. Second Order Information in Data Assimilation. *Monthly Weather Review*, 130(3):629-648, 2002.
- [37] R. Lehoucq, K. Maschhoff, D. Sorensen, and C. Yang. Arpack Software (parallel and serial). URL: <http://www.caam.rice.edu/software/ARPACK>.

- [38] M. Leutbecher, J. Barkmeijer, T.N. Palmer, and A.J. Thorpe. Potential improvement of forecasts of two severe storms using targeted observations. *Quarterly Journal of the Royal Meteorological Society*, 128:1641-1670, 2002.
- [39] W. Liao, A. Sandu, T. Chai, and G.R. Carmichael. Total Energy Singular Vector Analysis for Atmospheric Chemical Transport Models. *Monthly Weather Review*, 134(9):2443-2465, 2006.
- [40] D.C. Liu and J. Nocedal. On the Limited Memory Method for Large Scale Optimization. *Mathematical Programming*, B45(3):503-528, 1989.
- [41] Z. Liu, and A. Sandu. Analysis of Discrete Adjoints of Numerical Methods for the Advection Equation. *International Journal on Numerical Methods for Fluids*, 2007, Accepted.
- [42] G.I. Marchuk. Adjoint Equations and Analysis of Complex Systems. *Kluwer Academic Publishers*, 1995.
- [43] G.I. Marchuk, V.I. Agoshkov, and V.P. Shutyaev. Adjoint Equations and Perturbation Algorithms in Nonlinear Problems. *CRC Press*, 1996.
- [44] J.L. Morales, and J. Nocedal. Enriched Methods for Large-Scale Unconstrained Optimization. *Computational Optimization and Applications*, 21:143-154, 2002.
- [45] S.G. Nash. A survey of truncated-Newton methods. *Journal of Computational and Applied Mathematics*, 124(1-2):45-59, 2000.
- [46] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation* 24:773-782, 1980.
- [47] J. Nocedal, and S.J. Wright. Numerical Optimization, *Springer Series in Operations Research*, New York, 1999.
- [48] B.D. Ozyurt, and P.I. Barton. Cheap Second Order Directional Derivatives of Stiff ODE Embedded Functionals. *SIAM Journal on Scientific Computing*, 26(5):1725-1743, 2005.

- [49] B.D. Ozyurt, and P.I. Barton. Large-Scale Dynamic Optimization Using the Directional Second-Order Adjoint Method. *Industrial and Engineering Chemistry Research*, 44(6):1804-1811, 2005.
- [50] D.F. Parrish, and J.C. Derber. The National Meteorological Center's Spectral Statistical-Interpolation Analysis System. *Monthly Weather Review*, 120:1747-1763, 1992.
- [51] A. Sandu, D. Daescu, and G.R. Carmichael. Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: I - Theory and Software Tools. *Atmospheric Environment*, 37:5083-5096, 2003.
- [52] A. Sandu, D. Daescu, G.R. Carmichael, and T. Chai. Adjoint Sensitivity Analysis of Regional Air Quality Models. *Journal of Computational Physics*, 204:222-252, 2005.
- [53] T. Schlick, and A. Fogelson. TNPACK – A truncated Newton minimization package for large-scale problems. I: Algorithm and usage. *ACM Transactions on Mathematical Software*, 18:46-70, 1992.
- [54] T. Schlick, and A. Fogelson. TNPACK – A truncated Newton minimization package for large-scale problems. II: Implementation examples. *ACM Transactions on Mathematical Software*, 18:71-111, 1992.
- [55] Z. Sirkes, and E. Tziperman. Finite difference of adjoint or adjoint of finite difference? *Monthly Weather Review*, 49:5-40, 1997.
- [56] G.L.G. Sleijpen, J.G.L. Booten, D.R. Fokkema, and H.A. Van der Vorst. Jacobi-davidson type methods for generalized eigenproblems and polynomial eigenproblems. *B.I.T.*, 36(3):595-633, 1996.
- [57] B. Sportisse. An analysis of operator splitting techniques in the stiff case. *Journal of Computational Physics*, 161:69-91, 2000.
- [58] K.Y. Wang, D.J. Lary, D.E. Shallcross, S.M. Hall, and J.A. Pyle. A review on the use of the adjoint method in four-dimensional atmospheric-chemistry

- data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 127(576(Part B)):2181-2204, 2001.
- [59] Z. Wang, I.M. Navon, F.X. Le Dimet, and X Zou. The second order adjoint analysis: Theory and applications. *Meteorology and Atmospheric Physics*, 50(1-3):3-20, 1992.
- [60] 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.
- [61] M.J. Wooldridge, and N.R. Jennings. Intelligent Agents: Theory and Practices. *Knowledge Engineering Review* 10(2):115-152, 1995.
- [62] Y.J. Yang, M.T. Odman, and A.G. Russell. Fast three-dimensional sensitivity analysis of photochemical air quality models: an application to southern California. *Proceedings of the Air and Waste Management Association's Annual Meeting and Exhibition*, 98-WP76A(06):2, 1998.

Appendix A

Second Order Adjoint

A.1 The ODE Model, the Jacobian, and the Hessian

In this paper we consider all vectors to be column vectors. Gradients of scalar functions are by default row vectors. An upper script $(\cdot)^T$ denotes the transposition operator.

The first and second derivatives of a scalar function are

$$\Psi: \mathbb{R}^n \rightarrow \mathbb{R} \quad \Rightarrow \quad \frac{\partial \Psi}{\partial y} = \left[\frac{\partial \Psi}{\partial y_1}, \dots, \frac{\partial \Psi}{\partial y_n} \right] \quad \text{and} \quad \frac{\partial^2 \Psi}{\partial y^2} = \begin{bmatrix} \frac{\partial^2 \Psi}{\partial y_1^2} & \cdots & \frac{\partial^2 \Psi}{\partial y_1 \partial y_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \Psi}{\partial y_n \partial y_1} & \cdots & \frac{\partial^2 \Psi}{\partial y_n^2} \end{bmatrix}$$

The Jacobian of a multidimensional vector function is represented as

$$h: \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad h(y) = \begin{bmatrix} h_1(y_1 \cdots y_n) \\ \vdots \\ h_m(y_1 \cdots y_n) \end{bmatrix} \quad \Rightarrow \quad \frac{\partial h}{\partial y} = \begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \cdots & \frac{\partial h_1}{\partial y_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m}{\partial y_1} & \cdots & \frac{\partial h_m}{\partial y_n} \end{bmatrix}$$

Consider a coupled system of stiff nonlinear differential equations which constitute the *forward model*

$$\frac{dy}{dt} = f(t, y), \quad y(t^0) = y^0, \quad t^0 \leq t \leq t^F.$$

The Jacobian of the time derivative function is

$$J_{i,j}(t, y) = \frac{\partial f_i(t, y)}{\partial y_j}, \quad 1 \leq i, j \leq n.$$

The Hessian contains second order derivatives of the time derivative functions. More exactly, the Hessian is a 3-tensor such that

$$H_{i,j,k}(t, y) = \frac{\partial J_{i,j}(t, y)}{\partial y_k} = \frac{\partial^2 f_i(t, y)}{\partial y_j \partial y_k} = \frac{\partial^2 f_i(t, y)}{\partial y_k \partial y_j} = H_{i,k,j}(t, y) , \quad 1 \leq i, j, k \leq n .$$

For each component i of the ODE derivative function there is a Hessian matrix $H_{i,:,\cdot}$.

The Hessian allows to conveniently express the derivatives of the Jacobian times a user vector:

$$\begin{aligned} \frac{\partial}{\partial y} [J(t, y) \cdot u] &= \left(\frac{\partial}{\partial y_j} [J(t, y) \cdot u]_i \right)_{i,j} = \left(\frac{\partial}{\partial y_j} \left[\sum_{m=1}^n J_{i,m}(t, y) u_m \right] \right)_{i,j} \\ &= \left(\sum_{m=1}^n \frac{\partial J_{i,m}(t, y)}{\partial y_j} u_m \right)_{i,j} = \left(\sum_{m=1}^n H_{i,m,j}(t, y) u_m \right)_{i,j} \\ &= \left(\sum_{m=1}^n H_{i,j,m}(t, y) u_m \right)_{i,j} \\ &= H(t, y) \cdot u \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial y} [J^T(t, y) \cdot u] &= \left(\frac{\partial}{\partial y_j} [J^T(t, y) \cdot u]_i \right)_{i,j} = \left(\frac{\partial}{\partial y_j} \left[\sum_{m=1}^n J_{m,i}(t, y) u_m \right] \right)_{i,j} \\ &= \left(\sum_{m=1}^n \frac{\partial J_{m,i}(t, y)}{\partial y_j} u_m \right)_{i,j} = \left(\sum_{m=1}^n H_{m,i,j}(t, y) u_m \right)_{i,j} \\ &= \left(\sum_{m=1}^n u_m H_{m,i,j}(t, y) \right)_{i,j} \\ &= u^T \cdot H(t, y) \end{aligned}$$

For any vectors $u, v \in \mathbb{R}^n$ we have that

$$\begin{aligned} \frac{\partial}{\partial y} [J(t, y) \cdot u] \cdot v &= (H(t, y) \cdot u) \cdot v = \sum_{j,m=1}^n H_{i,j,m}(t, y) u_m v_j \\ &= \sum_{j,m=1}^n H_{i,m,j}(t, y) v_j u_m = (H(t, y) \cdot v) \cdot u \\ &= \frac{\partial}{\partial y} [J(t, y) \cdot v] \cdot u \end{aligned}$$

and

$$\begin{aligned}\frac{\partial}{\partial y} [J^T(t, y) \cdot u] \cdot v &= \left(u^T \cdot H(t, y) \right) \cdot v = \sum_{j,m=1}^n u_m H_{m,i,j}(t, y) v_j \\ &= \sum_{m=1}^n \left(H(t, y) \cdot v \right)_{m,i} u_m = \left(H(t, y) \cdot v \right)^T \cdot u\end{aligned}$$