Computational methods in air quality data

Zhaochen Zhu

Follow this and additional works at: https://repository.hkbu.edu.hk/etd_oa

Recommended Citation
https://repository.hkbu.edu.hk/etd_oa/402
DATE: August 21, 2017

STUDENT'S NAME: ZHU Zhaochen

THESIS TITLE: Computational Methods in Air Quality Data Assimilation

This is to certify that the above student's thesis has been examined by the following panel members and has received full approval for acceptance in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Chairman: Prof. Cheung Yiu Ming
Professor, Department of Computer Science, HKBU
(Designated by Dean of Faculty of Science)

Internal Members: Prof. Yuan Xiaoming
Professor, Department of Mathematics, HKBU
(Designated by Head of Department of Mathematics)
Prof. Liao Lizhi
Professor, Department of Mathematics, HKBU

External Members: Prof. Jing Liping
Professor
School of Computer and Information Technology
Beijing Jiaotong University

Prof. Wen Youwei
Professor
College of Mathematics and Computer Science
Hunan Normal University

In-attendance: Prof Ng Kwok Po
Chair Professor in Mathematics, Department of Mathematics, HKBU

Issued by Graduate School, HKBU
Computational Methods in Air Quality Data Assimilation

ZHU Zhaochen

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Principal Supervisor:
Prof. NG Kwok Po (Hong Kong Baptist University)

August 2017
DECLARATION

I hereby declare that this thesis represents my own work which has been done after registration for the degree of PhD at Hong Kong Baptist University, and has not been previously included in a thesis, dissertation submitted to this or other institution for a degree, diploma or other qualification.

I have read the University’s current research ethics guidelines, and accept responsibility for the conduct of the procedures in accordance with the University’s Committee on the Use of Human & Animal Subjects in Teaching and Research (HASC). I have attempted to identify all the risks related to this research that may arise in conducting this research, obtained the relevant ethical and/or safety approval (where applicable), and acknowledged my obligations and the rights of the participants.

Signature: **ZHU Zhaochen**

Date: August 2017
Abstract

In this thesis, we have investigated several computational methods on data assimilation for air quality prediction, especially on the characteristic of sparse matrix and the underlying information of gradient in the concentration of pollutant species.

In the first part, we have studied the ensemble Kalman filter (EnKF) for chemical species simulation in air quality forecast data assimilation. The main contribution of this paper is to study the sparse data observations and make use of the matrix structure of the Kalman filter updated equations to design an algorithm to compute the analysis of chemical species in the air quality forecast system efficiently. The proposed method can also handle the combined observations from multiple species together. We have applied the proposed method and tested its performance for real air quality data assimilation. Numerical examples have demonstrated the efficiency of the proposed computational method for Kalman filter update, and the effectiveness of the proposed method for NO$_2$, NO, CO, SO$_2$, O$_3$, PM2.5 and PM10 in air quality data assimilation.

For the second part, we have proposed and developed an optimization approach for data assimilation by using the gradients of the forecast of state variables. We have studied an objective function consisting of two data-fitting terms. The first term is based on the difference between the gradients of the forecast and the analysis of the state variables while the second term is based on the difference between the observations and the projected analysis of state variables. Here the existence and uniqueness of the analysis solution of the proposed objective function are shown. The solution can be calculated by using the conjugate gradient method iteratively. Experimental results based on the Community Multi-scale Air Quality (CMAQ) are presented. We then showed that the prediction performance of the proposed method is better than that of other testing methods in air quality and weather data assimilation.

Within the third part, we have set up an automatic workflow to connect the management system of the chemical transport model - CMAQ with our proposed data assimilation methods. The setup has successfully integrated the data assimilation into the management system and shown that the accuracy of the prediction has risen
to a new level. This technique has transformed the system into a real-time and high-precision system. When the new observations are available, the predictions can then be estimated almost instantaneously. Then the agencies are able to make the decisions and respond to the situations immediately. In this way, citizens are able to protect themselves effectively. Meanwhile, it allows the mathematical algorithm to be industrialized implying that the improvements on data assimilation have directly positive effects on the developments of the environment, the human health and the society. Therefore, this has become an inspiring indication to encourage us to study, achieve and even devote more research into this promising method.
First and foremost, I would like to express my deepest gratitude to my supervisor Prof. Michael K. Ng, for not only his perspicacious guidance, inspiring encouragement and financial support throughout the past four years, but also for spending extra time in valuable discussion with me and sharing his ideas on my research. His strict and devoted attitude towards scientific research, his patience and responsibility to students and his kindness and consideration in all matters will be remembered in my heart forever. Receiving an enormous amount of academic knowledge from him in the fields of data assimilation, mathematics and research, and experiencing his excellent personal charisma of sustained attention, persistent efforts and professional attitude are the greatest memories to me and I will be forever grateful to him.

I would also like to thank the Department of Mathematics for providing the excellent studying conditions and facilities and various departmental administrative staffs: Ms. Claudia Chui, Ms. Vicky Hui, Ms. Candy Li and Ms. Jenny Cheung; and the computer technicians, Ms. Tammy Lam and Mr. Cheong Wing Yeung for their kind help and assistance. I would also like to thank my classmates and friends who have given life to my journal because they have brought much happiness to me and made my life more enterprising and entertaining.

Finally, let me express my immense thanks to my family for providing me with unfailing support and continuous encouragement throughout my years of study and during the process of my researching and writing of this thesis. This accomplishment would not possibly have been achieved without the understanding and assistance of the above people.
# Table of Contents

Declaration i

Abstract ii

Acknowledgements iv

Table of Contents v

List of Figures viii

List of Tables xi

Chapter 1 Introduction 1
   1.1 Background ..................................................... 1
   1.2 Data Assimilation ............................................. 3
   1.3 The Contribution .............................................. 6
   1.4 Thesis Outline ............................................... 7

Chapter 2 Air Quality Model Simulation 11
   2.1 Air Quality Model ............................................ 11
   2.2 Photochemical Air Quality Model ............................. 13
   2.3 The Conceptual Model of Community Multi-scale Air Quality (CMAQ) 14
      2.3.1 Overview ................................................. 15
      2.3.2 Meteorology-Chemistry Interface Processor (MCIP) ......... 16
      2.3.3 Initial and Boundary Conditions (ICON and BCON) .......... 17
      2.3.4 Clear-sky Photolysis Rate Calculator (JPROC) ............. 18
   2.4 CMAQ Chemistry Transport Model (CCTM) ................... 18
2.4.1 Derivation of the Chemical Transport Model . . . . . . . . . . 20
2.4.2 Advection and Diffusion . . . . . . . . . . . . . . . . . . . . . 26
2.4.3 Deposition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
2.4.4 Gas-phase Chemistry Solvers and Process . . . . . . . . . . . 29
2.4.5 Aerosol Process . . . . . . . . . . . . . . . . . . . . . . . . . . 30
2.4.6 Cloud Mixing and Aqueous-Phase Chemistry (CLOUD) . . . 31
2.4.7 Plume-in-Grid (PING) . . . . . . . . . . . . . . . . . . . . . . 31

Chapter 3 Data Assimilation Methods
3.1 Optimal Interpolation (OI) . . . . . . . . . . . . . . . . . . . . . . 34
3.2 Variational Methods . . . . . . . . . . . . . . . . . . . . . . . . . . 36
3.2.1 3D-Var: Three-Dimensional Variation Method . . . . . . . . 37
3.2.2 Advantages of 3D-Var compared to the OI . . . . . . . . . . . 38
3.2.3 4D-Var: Four-Dimensional Variation Method . . . . . . . . . 39
3.2.4 Characteristics of 4D-Var compared to the 3D-Var and OI . . 41
3.3 Kalman Filter Methods . . . . . . . . . . . . . . . . . . . . . . . . 42
3.3.1 EKF: Extended Kalman Filter . . . . . . . . . . . . . . . . . . 42
3.3.2 EnKF: The Classical (Stochastic) Ensemble Kalman Filter with
          Perturbed Observations . . . . . . . . . . . . . . . . . . . . . 45
3.3.3 Advantage of EnKF compared to the EKF . . . . . . . . . . . 47
3.3.4 The Comparison between 4D-Var and EnKF . . . . . . . . . . 47
3.3.5 Covariance Inflation of EnKF . . . . . . . . . . . . . . . . . . 48
3.3.6 Localization of EnKF (LEnKF) . . . . . . . . . . . . . . . . . 49
3.3.7 Characteristics and Advantages of LEnKF compared to the EnKF 51
3.3.8 EnSRF: Ensemble Square-root Kalman Filter . . . . . . . . . 51

Chapter 4 Sparse Matrix Computation
4.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.2 Proposed Method . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
4.2.1 Sparse Matrix Computation . . . . . . . . . . . . . . . . . . . 54
4.2.2 Multiple Species . . . . . . . . . . . . . . . . . . . . . . . . . . 56
4.3 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
List of Figures

2.1 The overview of air quality model ........................................ 11
2.2 CMAQ chemistry-transport model (CCTM) and input processors .. 16
2.3 Meteorology preprocessor for CMAQ ................................. 16
2.4 Initial and boundary condition preprocessors for CMAQ ............ 18
2.5 CMAQ chemistry-transport model and associated preprocessors .... 18
2.6 Science process modules in CMAQ ................................. 25

4.1 Comparison of domain size ............................................. 59
4.2 CMAQ system with data assimilation (updating internal concentration) 60
4.3 CMAQ system with data assimilation (updating internal and boundary concentrations) ........................................ 61
4.4 The parts of program on reading and writing NETCDF file ........ 72
4.5 Model of concentration of pollutant in 3D ............................ 73
4.6 The observational station in cube ...................................... 74
4.7 The model of the observational stations in mesh ..................... 75
4.8 Missing value ............................................................ 76
4.9 The changes on daily average of best combination of $\lambda$ and $\gamma$ for single species in January ........................................ 84
4.10 The changes on daily average of best combination of $\lambda$ and $\gamma$ for single species in February ................................. 86
4.11 The changes on daily average of best combination of $\lambda$ and $\gamma$ for single species in March ................................. 87
4.12 The changes on daily average sample number for single species in January ........................................ 90
4.13 The changes on daily average sample number for single species in February ........................................ 91
4.14 The changes on daily average sample number for single species in March 92
4.15 The RMSEs of five pollution species with respect to number of ensembles. 94
4.16 The computational time required by data assimilation of five pollution
  species with respect to number of ensembles. ............................ 94
4.17 The RMSEs of five pollutant species for 1 month simulation in January. 95
4.18 The RMSEs of five pollutant species for 1 month simulation in February. 96
4.19 The RMSEs of five pollutant species for 1 month simulation in March. 97
4.20 The changes on daily average of best combination of \( \lambda \) and \( \gamma \) for mul-
  tiple species in January ....................................................... 98
4.21 The changes on daily average of best combination of \( \lambda \) and \( \gamma \) for mul-
  tiple species in February ..................................................... 99
4.22 The changes on daily average of best combination of \( \lambda \) and \( \gamma \) for mul-
  tiple species in March ....................................................... 99
4.23 The changes on daily average sample number for multiple species in
  January ............................................................................. 100
4.24 The changes on daily average sample number for multiple species in
  February ............................................................................ 100
4.25 The changes on daily average sample number for multiple species in
  March .................................................................................. 101
4.26 The RMSEs of multiple species with respect to number of ensembles. 102
4.27 The computational time required by data assimilation of multiple species
  with respect to the number of ensembles. ............................... 102
4.28 The RMSEs of the combination of multiple pollutant species for 1
  month simulation in January .................................................. 103
4.29 The RMSEs of the combination of multiple pollutant species for 1
  month simulation in February ................................................ 103
4.30 The RMSEs of the combination of multiple pollutant species for 1
  month simulation in March .................................................... 104

5.1 The sparse shape of \( \nabla_{\{x,y,z\}} \) and \( \nabla_{\{x,y,z\}}^T \nabla_{\{x,y,z\}} \) .................. 112
5.2 The sparse shape of \( \mathbf{M}_x \), \( \mathbf{M}_y \) and \( \mathbf{M}_z \) ................................. 112
5.3 The geometry meaning of the vector ........................................ 114
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>The geometry meaning of matrix-vector multiplication</td>
<td>114</td>
</tr>
<tr>
<td>5.5</td>
<td>The changes on daily average of $\omega$ for single species in January</td>
<td>118</td>
</tr>
<tr>
<td>5.6</td>
<td>The daily average errors for the prediction of single species by different methods</td>
<td>119</td>
</tr>
<tr>
<td>5.7</td>
<td>The changes on daily average of $\omega$ for multiple species in January</td>
<td>121</td>
</tr>
<tr>
<td>5.8</td>
<td>The daily average errors for the prediction of single species by different methods</td>
<td>122</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Mode option of netcdf.open function ........................................ 71

5.1 The average errors and improvements of CCTM with proposed method 
and other existing methods for single species in January ................. 120

5.2 The average errors and improvements of CCTM with proposed method 
and other existing methods for multiple species in January ............. 122
Chapter 1

Introduction

1.1 Background

With the gradual development of society and economy, people’s life have improved and they have become more knowledgeable. Their attitudes towards life is gradually changing from accepting basic life requirements to requesting for a more comfortable quality of life. Therefore, air quality has become a very important factor affecting people’s life at present. Within this century, several dangerous hazy and polluted weathers conditions have occurred in the UK, US and other countries which had led to traffic congestion, corrosion of buildings and also influenced the photosynthesis of vegetation and thereby threatening the people’ lives. So, in 1995, UK passed its Environment Act and published the national air quality strategy to control the standards of common air pollutants. In 1970, US passed the Amendments of the Clean Air Act which has led to the establishment of the acceptance level of the nation’s air quality standards. Meanwhile, the Hong Kong government had also set up the Environmental Protection Department (HKEPD) in 1984 to incorporate rules to control the air quality. Therefore, the daily prediction of the air quality has become the key guidance of the air quality which either helps the public to make plans for their activities in advance to avoid bad air quality, or gives a reliable reference to the government when installing policies or organizing events. We cooperated with the HKEPD to get the real air quality data of Hong Kong and to use our methods to analyze and improve the air quality predictions.
We applied the science to improve the air quality prediction. The chemical transport model is an efficient instrument based on science to handle the changes of air pollution. It can evaluate the cost and efficiency to control the way to interpret the observational data. At the same time, it can also let us understand clearly about chemical conversion in the atmosphere [13]. We can use the chemical transport model to provide data information to the government, professional organizations, and to the public, to enable society to consciously be aware that something has to be done to improve the air quality. For instance, through knowing the chemical transformation of air in the atmosphere, we can help the government to track the source of pollutants so as to reduce pollutants at its source.

Due to the complexity of air quality prediction, the chemical transport model needs vast variables as input to the process and to give the result of prediction as output. In 1970, US Environmental Protection Agency (USEPA) set up a system called the Community Multi-scale Air Quality (CMAQ) modeling system to evaluate the air quality as a whole by using state-of-the-science capabilities to model the multiple air quality problems, including fine particles, tropospheric ozone, acid deposition, toxics, and visibility degradation. Based on this idea, the CMAQ can be developed by scientific experts from different areas and become the combination of their outstanding research practical experiences. In this system, the chemical transport model is the core part to run the whole system.

Even through there are some improvements on the chemical transport model in recent years, air quality prediction is still a challenging problem because of the complex processes occurring on different scales and the strong coupling relationship across scales. Also, the air quality prediction has a large amount of uncertainties which we want to reduce, such as incomplete emission information, missing key observations in the initial and boundary conditions, lacking some scientific elements, and the worse parameterized processes [13].

Now, the HKEPD uses the CMAQ to do the air quality prediction in Hong Kong. They have considered that the accuracy is not up to standard and needed to be improved. Also, they have discovered that improving the accuracy is needed by giving the chemical transport model more constrained with the observational data.
which the CMAQ had never done before. In this thesis, we will present advanced methods based on data assimilation to improve the prediction capacity of the chemical transport model, especially on the technique to make an integer of the model and measurements.

1.2 Data Assimilation

What is data assimilation? The simplest perspective is that it is a method of combining the observations to the model state with the aim of improving the latter. Do we need the data assimilation? Why have we not chosen to use only the observations or do the interpolation to get more from them? The reason is that we want to find out more about the prediction of the future of the model state, and not just the information in current time. So, to be certain, this prediction and future ones are acquired within the biggest possibility. To achieve this, we need to rely on the model, but however, the model is not always correct. Thus, we need to update the value of the model periodically with the reality in an optimal way to get the updated prediction. This is known as the analysis.

The data assimilation techniques have been proposed and developed, see for instance [9, 30, 58, 56] and have included two mainstream methods: The first one is the variational data assimilation method. It is based on the optimization by minimizing a given objective function that measures the model for data fitting, see for instance [56, 17]. The variational data assimilation was formally introduced by the meteorological community for solving the problems on numerical weather prediction. The first attempt on forecasting the weather numerically was achieved by Lewis Fry Richardson in 1922 [42]. However, the large errors of the observation in his paper was caused by the inaccurate initial condition of the field in his analysis [60]. It indicates the start of the need for data assimilation.

Later, a major development was achieved by L. Gandin [36] in 1963. He introduced the statistical interpolation method, which inspired Kolmogorov on his earlier ideas. This is the generation of the three dimensional variational method. 3D-Var method is a regression analysis using the information of spatial distributions of co-
variance in the initial condition of model state and the true state [70]. The covariance matrices are not calculated dynamically but are predetermined. After that, a significant advance in the development of the variational data assimilation method is the four dimensional method which is established from the optimal control theory. This is based on the earlier work of G. Marchuk, Le Dimet and Talagrand [56] who first applied the theory of Lions [57] and Troltzsch [76] to the environmental modeling. The advantage of the four dimensional method is that the analysis solution not only fits the development of the dynamical equation, but also minimizes two terms: the discrepancy between the analysis state (unknown) values and observations, and the difference between the analysis state (unknown) values and forecast state values [31]. Therefore, a constrained minimization problem is solved. In both 3D-Var and 4D-Var data assimilation methods, several numerical solvers are proposed and studied for such optimization purposes. For example, adjoint methods [68] are used to deal with the huge computational cost involved in 4D-Var data assimilation [17].

Now, the variational data assimilation method is the most successfully developed to hybridize with the incremental four dimensional method. Using the ensembles to enhance the approximation of the background error covariance and this covariance matrix is evolved by the simulations of ensembles through the forecast model. Thus this method is moving forward to become a more efficient method called ensemble variational method (EnVar) [77, 83].

For the Kalman filtering approach for air quality forecast data assimilation, Kalman [52] first proposed and developed a filtering method to data assimilation. This method is then extended to nonlinear Kalman filter (for example [37]) for dealing with nonlinear systems. In order to enhance the computational efficiency, the ensemble Kalman filter [26, 29] is further developed for the data assimilation purpose. Evensen and Leeuwen [27] used it for assimilating the gridded Geosat data and Evensen [28] examined the performance with respect to the classical numerical weather model called the Lorenz equation. Houtekamer and Mitchell [43] tried to apply it on a quasi-geostrophic three-level T21 model and showed the approximation could be improved by using more ensembles. The performance will be close to the standard Kalman filter when the number reaches a large number. In recently decades, J.L. Anderson and
S.L. Anderson [2] and Miller et al. [62] considered that for the generalized ensemble Kalman filter, the distribution of error within the model may be the combination of several distributions and not just contains only a single one like the assumption of the Gaussian distribution.

For the ensemble Kalman Filter, there are several desirable properties and which are being improved and investigated by many researchers. (i) It uses the ensembles to approximate the flow-dependent forecast error covariance without a parameterized multivariate correlation of the model. (ii) Since the algorithm is independent of the model, the way of the generation of the ensembles can be varied and unconstrained. (iii) The observation can be of various kinds, for instance, the nonconventional observation - satellite radiances which are essential parts of the available observational forms [1, 24, 39] and this feature is expected to grow in the future. (iv) The accuracy will increase slowly when the number of ensemble reaches a modest value. (v) The algorithm produces the smooth analysis increments on the forecast of the model and these increments could be the source of noise and imbalance [23, 18]. (vi) The method can efficiently to handle large number of observations by using the designed parallel strategy. For example, recently, Kappenne [50] implemented the ensemble Kalman filter for a two-layer model and assimilate 2775 gridded observations in one analysis step. The process is designed to run in parallel computation but it will be very expensive if this is used significantly for the large number of observations in the standard Kalman filter. (vii) Because of the universality and superiority of the method, it has now been applied to the other research areas such as improving the weather predictions [19, 11, 82, 66, 37], for air quality prediction [31, 69, 33, 22, 44], studies in oceanography [38, 21, 15, 14, 67], for remote sensing and hydrologic modeling [46] while other possible applications can be found in [9, 51] and the references therein. After the generation of the ensemble Kalman filter, there also exist other variants like the ensemble transform Kalman filter method [4] and localized ensemble Kalman filter method [69, 63].
1.3 The Contribution

The first contribution of this thesis is to develop an efficient computational method for calculating the ensemble Kalman filter equations. Our idea is based on the fact that the number of observations is significantly less than the number of variables in estimation. Because of the sparsity in the observational equations, the huge calculation of the Kalman filter gain matrix can be avoided. The computational complexity of the resulting Kalman filter gain matrix depends on the number of non-zero entries in the observational equations and the number of observations in the system. Also, the proposed method can handle the combined observations arising from multiple species in the air quality forecast system. We then apply the proposed method and test its performance for the real air quality data assimilation. Experimental results are reported to demonstrate the efficiency of the proposed computational method for Kalman filter update. Also, the effectiveness of the proposed method is tested for NO$_2$, NO, CO, SO$_2$, O$_3$, PM2.5 and PM10. Numerical results show the usefulness of the proposed model for air quality data assimilation.

The second contribution is the improvements on the variational data assimilation method. It is motivated in the image processing applications [54, 32, 53]. The image gradient values are used to generate processed images with suitable boundary conditions given in the image processing applications. For example, the image gradient values are used to control the stitching of several images with their given overlapping regions. Numerical results reported in these image processing applications have shown that the use of image gradients is very promising and useful. To the best of our knowledge, this is the first attempt to develop an optimization approach for data assimilation by using the gradients of the forecast of state variables. We consider an objective function consisting of two data-fitting terms. The first term is based on the difference between the gradients of the forecast and the analysis (unknown) of state variables, and the second term is based on the difference between the observations and the projected analysis of state variables. We then show the existence and uniqueness of the analysis solution of the proposed objective function. The analysis solution can be computed by solving a linear system and can be calculated by using the conjugate gradient method iteratively. Also, numerical examples for the air quality data
assimilation are tested, and we demonstrate that the prediction performance of the proposed method is better than that of other testing methods for these examples.

The third contribution is that we then set up an automatic workflow to connect the management system of the chemical transport model - CMAQ with our proposed data assimilation methods. It successfully integrates the data assimilation into the management system and shows that the accuracy of prediction has arisen to a new level. This technique turns the system into a real-time and high-precision system. When the new observation is available, the prediction can be estimated very rapidly. Then the various agencies could make the decisions and respond to the situations immediately allowing the citizens as to protect themselves wisely and effectively. In the meanwhile, it makes the mathematic algorithm to be applied to industries which implies that the improvements on data assimilation have direct positive effects on the development of the environment, the human health and the society.

1.4 Thesis Outline

This thesis is organized as follows.

The main goal of Chapter 2 is to establish the concept of the air quality model simulation and understand the most effective way to employ the proposed method on it with real data. According to the requirements, the air quality model has been developed into three branches. The photochemical modeling has become the most popular air quality model. Based on the chemical, physical and mathematical rules, scientists have set up the air quality management system, namely, the Community Multi-scale Air Quality (CMAQ) system. They have used it to prepare all outer related information for the chemical transport model and simulate the chemical reactions and gas transportations in order to predict the air quality. It has progressed to the third generation and to be a highly integrated and complex system which has many scientific components to describe the varied processes within the simulation. Since the model is still being developed and has not yet reached the perfect level, the global concentration of the pollutant is blurred compared to the actual situation. The model contains numerous noises so that the accuracy is still kept to the lowest level.
After understanding the principle of model and the procedure of program, with the need of increasing the accuracy of the prediction of the chemical transport model from HKEPD and the aim of improving the life qualities of citizens, we began to research the possibility of applying the data assimilation on the chemical transport model and the improvements on the data assimilation method in the subsequent chapters.

The principles of data assimilation method are presented in Chapter 3 and our improvements are working on them. Up to now, the data assimilation method is developed into two mainstream directions: the first one is the Kalman filter method based on the estimation theory and the second one is the variational method based on the optimization theory. Variational method is started from the least square problem which contains two fitting terms in a time point. It considers the differences between the analysis solution and the forecast, and the differences between the observations and projected analysis solution. A method known as the three dimensional variational method (3D-Var) extends it a litter further by using the background covariance and observation covariance to weight the corresponding differences, respectively. This extension makes the analysis solution deviate more on the reliable background variable and observation and less on the non-confident ones. After that, the influence of the prediction estimated from the 3D-Var method in a period of future time is started to be considered. That allows for the generation of the four dimensional variational method (4D-Var). Compared to the three dimensional method, the four dimensional method would balance the performance either at the current time point or the several following time points. It guarantees the analysis solution will be completely consistent with the forecast model and observations within the time interval. Compared to the variational data assimilation, the Kalman filter method is more frequently used because of its succinct form and separable structure. The Kalman filter is derived from the estimation theory under the assumption of owing Gaussian noise. Soon, with the need of the nonlinear problem, it is extended to the extended Kalman Filter (EKF). The cost function is very similar to the variational method at this moment expect for the strategy on noise. Later, the increasing number of variables in the numerical forecast model makes the EKF unaffordable so that the algorithm starts to use the Monte Carlo method to reduce the number of samples. Besides, this improve-
ment also meet the demand of solving the "Big Data" problem in the present decade. That leads to this method being developed very quickly with many improvements in this decade. Especially, in the improvements of the covariance inflation and localization. Covariance inflation will correct the underestimation problem and balance the relationship between the background and observation covariance, as well as allow localization to keep the background covariance away from the spurious correlations caused by sampling errors.

Chapter 4 presents our proposed method on the localized ensemble Kalman filter and the efforts on setting up the linkage between CMAQ and data assimilation method. We develop a very efficient computational method for calculating the localized ensemble Kalman filter. By using the feature of sparsity of observations compared to the background variables, the huge operations and the storage of generating the background covariance matrix can be reduced without losing the accuracy. Also, we have set up a strategy to select the parameters dynamically to get the smallest RMSE so that the accuracy of algorithm for the real implement can be improved. This computational method is available for several multiple species which contain the common single species. The situation becomes complex when they have coupled relationship. Because if this method is not proper, the concentration of a single species will be updated several times and that will cause conflicts. In the implement process, some modifications on the configure files and running codes are presented. These changes make the CMAQ system and data assimilation method adapt to each other. Running as an automatic workflow until reaches the stop criterion set beforehand. This is a strategy that the Multivariate Normal Distribution is used to generate the random noises for the ensembles. It lets the noises generated from the model error covariance and builds more connections between the noises and ensembles, in other words, making the background covariance more accurate. The same measure is applied on the random noises for observations as well. In the last section of this chapter, some numerical tests and their corresponding performances are shown to evaluate the effectiveness of the proposed method and its superiority on the computational operation. We have ran the simulations of single species and multiple species for January, February and March, respectively.
In Chapter 5, we have raised a novel method on the variational data assimilation method. The novel method is inspired by image stitching. We are trying to use the underlying information (gradient) of the concentration to develop an optimization approach for data assimilation. This optimization approach is formed by two terms: the first term is related to the difference between the gradient of analysis solution and forecast and the second term is constructed by the discrepancy between the projected analysis solution and the observations. This method also has the design for several multiple species. Similar with the proposed method in Chapter 4, it creates the possibility to update the single species within the coupled observations of multiple species simultaneously. The algorithm needs to do the matrix-vection multiplication in each iteration, so it may become a very low efficient method if this process is costly. Two strategies of reduction are demonstrated in the implementation part, one is considering the reduction from the sparse characteristics of discrete matrix, another one shows a new direction which treats the process of reduction as a transformation on the geometric space. The latter is faster than the former and it is proved by this section. After that, we show the performances on this optimization approach. Running the simulations in January, February and March to show the improvements during different months and to show its universality for both single species and multiple species. In the meanwhile, comparing the RMSE and time of optimization approach with the simulation of the CCTM case and CCTM with 3D-Var case on the pollutant species demonstrates its superiority on accuracy and economical efficiency.

The summary of the thesis and the directions of future work are concluded in the Chapter 6.
Chapter 2

Air Quality Model Simulation

2.1 Air Quality Model

Air quality model is a numerical tool using mathematical and numerical techniques to simulate the physical and chemical transformations affecting the air pollutants in the atmosphere. It establishes the linkage between emissions, meteorology, atmospheric concentrations, deposition and other factors collected from varied sources and describes their causal relationships shown in Figure 2.1. Based on the inputs of meteorological data and emission materials, the model is designed to characterize the primary pollutants emitted directly into the atmosphere. Moreover, in some cases, the secondary pollutants are considered, and they are formed via a series of complex chemical reactions within the atmosphere.

![Figure 2.1: The overview of air quality model](image)
The air quality monitor provides some important and quantitative information about the ambient concentration of the pollutants. However, it can only describe the air quality at the specific locations and times, but cannot offer any guidance to identify the cause of the air quality problems. Instead, the air quality model gives a more complete and deterministic description of the air quality problems: such as the emission and meteorological sources, the physical and chemical processes and the predicted results. From the analysis of these factors, it determines the causes of the problems and helps to ease the process of the implementation of observational stations.

Therefore, the air quality model plays a very important role in determining nature and science, not only because of its capacity on evaluating the influences and importance of the processes for the pollutant, but also it can quantify the deterministic relationship between the emission and their concentration, from point to point as well as from the past to the future. The model is widely used and already involved in the air quality management system of the regulatory agencies to control the air pollutants, including the identification of the sources that contribute to the air pollutant and the assessment of the effectiveness of abatement strategies. For instance, the air quality model can be used during the permitting period to verify whether the new source (factory, parking lot, office building, shopping center et al.) will exceed the ambient air quality standards, and if necessary, extra control measure will be implemented.

The most common air quality models include the following:

- **Dispersion Modeling** - This model is frequently used during the permitting periods to estimate the concentration of the pollutant at targeted receptors around the source of emission at ground-level.

- **Photochemical Modeling** - This model is frequently used in the regulatory or policy assessment on the influence of all emission sources by simulating the changes in the concentration of pollutant and the deposition via a series of inert processes and chemical reactive processes at large spatial scales.

- **Receptor Modeling** - This model is an observational technique that uses the
chemical and physical features of the gas and particles measured at source and receptor to determine the existence of the pollutant at the receptor and the contribution from the source to the receptor.

2.2 Photochemical Air Quality Model

According to the versatility and scalability of the photochemical modeling, the administrative department would prefer to use it as the research and forecast tool for the local atmospheric phenomenon. Basically, two types of the photochemical air quality models are frequently used which are the Lagrangian trajectory model and the Eulerian grid model. The Lagrangian trajectory model uses a moving frame of reference and is adopted by the air quality management system in early times because of less computing work is needed. However, its disadvantage is obvious as well, the physical presentation of this model is incomplete. Which leads to the generation of the Eulerian grid model. Eulerian grid model manages the information of the ground in a fixed coordinate system. It is widely used in most of the air quality management systems now because of its improved and full description of the physical processes in the atmosphere.

The common photochemical air quality models are listed in the following:

- **Community Multi-scale Air Quality (CMAQ)** - CMAQ is an open source project initiated by the Atmospheric Science Modeling Division of the United States Environmental Protection Agency (USEPA) to manage the simulation of air quality model for multiple pollutants at multiple scales. It is supported and distributed by the CMAS Center, and uses the science and multi-processor computing techniques so as to better probe, understand and simulate with quick delivery of the datum regarding the chemical and physical transformations of ozone, particulates, toxins, and acid deposition in the atmosphere.

- **Comprehensive Air Quality Model with Extensions (CAMx)** - CAMx is one of the Eulerian dispersion and photochemical models. This framework is an open-source package available online and built with the purpose of integrating the assessment on gaseous and particulate air pollutants. The model
has the power to simulate the air quality over several geographic scales with a wide variety of inert processes and chemical active process. The functions of analyzing on sensitivity, process, and source-receptor are provided as well.

- **Regional Modeling System for Aerosols and Deposition (REMSAD)** - The REMSAD is developed by Systems Applications International (SAI)/ICF International to obtain a deep understanding on the emission, dispersion and deposition processes relevant to regional mist, particulate and other gaseous pollutants, such as the toxins and soluble acidic components.

- **Urban Airshed Model Variable Grid (UAM)** - The UAM modeling system is a pioneering effort in the photochemical air quality modeling and widely used for studying the air quality on ozone. The model has been developed and maintained by SAI since the early 1970s. By now, it has achieved nearly continuous cycles of application, performance evaluation, update, extension, and improvement.

This research project is based on the concentration and observational data provided by Hong Kong Environment Protection Department (HKEPD), so the research work will be established on their well-studied photochemical air quality modeling - CMAQ.

### 2.3 The Conceptual Model of Community Multiscale Air Quality (CMAQ)

In 1970, USEPA was set up by the US government and approval was granted to make the National Ambient Air Quality Standards (NAAQS) a part of the Clean Air Act. These standards protect the human health and the environment from high concentration of pollutants, like ozone and particulate matter. Since the NAAQS is frequently needed to establish strategies to control the source of air pollutants and that USEPA is expected to evaluate the effectiveness of these control strategies on a variety of complex chemical pollutants. They have decided to unite in an effort to build a system that can model and simulate the air quality range from the local
district to any region or state in the US. Also, it has the capacity to estimate the various control strategies to improve air quality and to make sure that the method used are effective and does not incur huge resources. Therefore, they set up the Community Multi-scale Air Quality (CMAQ) management system.

Up to now, this system has been developed to the third generation, and its applications can range from an administrative and policy analysis to interact research of pollutants in the chemical and physical aspects. It’s a three-dimensional Eulerian atmospheric chemistry and transport modeling system to simulate ozone, particulate matter (PM), toxic airborne pollutants, visibility and acidic and nutrient pollutant species throughout the troposphere. CMAQ is designed as ”one-atmosphere” but can handle the complex coupling relationships between some air quality problems simultaneously across multi-scales ranging from the local to the hemispheric. On top of that, the CMAQ source code has the characters of being transparent and modular, which makes it convenient to be enhanced by the community developers from different research areas.

2.3.1 Overview

CMAQ mainly contains four input processors (Meteorology-Chemistry Interface Processor (MCIP), Initial and Boundary Conditions (ICON, BCON) and Clear-sky Photolysis Rate Calculator (JPROC)) and the key simulation model called the CMAQ Chemistry-Transport Model (CCTM). The relationships between the input processors and the CCTM are shown in Figure 2.2.
Figure 2.2: CMAQ chemistry-transport model (CCTM) and input processors

CMAQ uses the MCIP processor to prepare the meteorological areas for CCTM. The ICON and BCON processors generate the initial and boundary conditions for CCTM simulation, respectively. JPROC computes the photolysis rates which are used when simulating photochemical reactions in the CCTM. Emissions for CMAQ must be prepared with a modeling system that generates emissions for the direct input to the CCTM (currently, Sparse Matrix Operator Kernel Emissions (SMOKE)).

2.3.2 Meteorology-Chemistry Interface Processor (MCIP)

MCIP uses the outcome from Fifth-Generation NCAR / Penn State Mesoscale Model (MM5) or Weather Research and Forecasting (WRF) model to create netCDF-formatted as the input meteorology data for SMOKE (the emissions processor that computer emissions inputs for CCTM) and CCTM (see Figure 2.3)

Figure 2.3: Meteorology preprocessor for CMAQ

MCIP prepares and evaluates all the meteorological information that is required
for SMOKE and CCTM. It also has the function to tidy the cells in uniform from the horizontal boundary of the domain defined by the meteorology models. Moreover, it could resize the horizontal area and vertical resolution to be the subset of the previous domain.

### 2.3.3 Initial and Boundary Conditions (ICON and BCON)

ICON generates a grid binary netCDF file that contains the concentrations of chemical species in the first hour for the modeling domain. It gets the necessary information from an ASCII file of vertically resolved concentration profiles (distributed with CMAQ) or from an existing CCTM output file. If the ASCII file does not have the same vertical definition as CCTM configuration, ICON will interpolate the vertical structure to make it consistent with the CCTM’s. Using an existing CCTM output file happens often when generating the new initial state of subdomain from the original domain in nested case (subdomain is a part of it), or getting the result from a coarse grid to a fine one.

Similarly, BCON also generates a gridded binary netCDF file that contains the horizontal boundary information of the modeling domain. This boundary condition could be static or time-varying. Same as ICON, the boundary condition is taken from an ASCII file or an existing CCTM output. From the ASCII file, BCON will interpolate the vertical resolution consistent with the CCTM configuration. There are also some differences from ICON, BCON can generate time-varying boundary conditions. A dynamic boundary condition is usually extracted from an existing CCTM output either from a coarse-grid simulation for nested simulation or a global-scale case. The ASCII profile can also support to generate dynamic boundary conditions, but generally, it is used to get static information. (see Figure 2.4)
2.3.4 Clear-sky Photolysis Rate Calculator (JPROC)

The job of JPROC is to calculate specific clear-sky photolysis rates for the targeted chemical mechanism at fixed altitudes, solar hourly angles, and latitude bands from tabulated absorption cross-section and quantum yield (CSQY). Since the CMAQ has CSQY which contains default chemical mechanism setting, we can handle the CSQY file directly simply by adding or updating.

2.4 CMAQ Chemistry Transport Model (CCTM)

Figure 2.5 shows the relationships among the CCTM and other parts in the Community Multi-scale air quality (CMAQ) modeling system. It acquires the sufficient information such as the meteorology data, the emission details, the initial and boundary conditions of the domain, and the photolysis rates to simulate the chemical reactions and transport processes for the pollutant.
As shown in the figure, there are several scientific components within the CCTM as well. Moreover, they are the most complex and important parts of the air quality modeling. Thus, we will give the detailed descriptions on their meanings and mathematical derivations in the following sections.

The scientific description on the chemistry transport model is started from the fundamental equation as shown in Equation (2.1) of atmospheric dynamics and thermodynamics in a generalized coordinate system.

\[
\frac{\partial (c_i J_z)}{\partial t} + h^2 \nabla_{\hat{x}\hat{y}} \cdot \left( \frac{c_i J_z \hat{v}_{\hat{x}\hat{y}}}{h^2} \right) + \frac{\partial (c_i J_z \hat{v}_z)}{\partial \hat{z}} = J_z T_{c_i},
\]

where the subscript \( i \) is the species of the tracer pollutant, \( c_i \) is the concentration of tracer species \( i \) in density units (e.g., kg/m\(^3\)), \( J_z \) is the vertical Jacobian in generalized coordinate \( \hat{z} \), \( h \) is the map scale factor of the generalized coordinates \( \hat{x} \) and \( \hat{y} \) to the Cartesian coordinates \( x \) and \( y \), \( \hat{v}_{\hat{x}\hat{y}} \) and \( \hat{v}_z \) are the horizontal and vertical wind components in the generalized coordinates, and \( T_{c_i} \) is the source or sink term.

Before doing the derivation, there are some assumptions that should be set first [64]:

- **Assumption 1**: The concentrations of the tracer species are small and they cannot have the counteractive effect to the meteorological condition. Thus, their conservation equations can be solved independently by the Navier-Stokes and energy equations. This assumption is valid when the heat energy from chemical reaction could influence the temperature of the medium or the pollutant becoming concentrated in atmosphere caused by the effects of radiation.

- **Assumption 2**: The concentration and velocities of the varied species in atmosphere are experiencing the turbulent diffusion and turbulent quantities. Since the molecular diffusion is so small when compared to the turbulent diffusions that it could be ignored.

- **Assumption 3**: The tensor in metric scale is not a turbulent variable which is defined as the coordinate transformation rule. It means the coordinate could be defined on the Reynolds averaged quantities and we could incrementally define the vertical grid with a time-dependent vertical coordinate.
• **Assumption 4**: The averaging process of the ensemble is based on the ergodic hypothesis which means the time average of a property can take the place of the ensemble average of the property.

• **Assumption 5**: The turbulent diffusion is assumed as fixed within the period of averaging time of interest.

• **Assumption 6**: The source term $T_{c_i}$ is certain for all practical processes and no turbulent component exists in the term.

• **Assumption 7**: The fluctuation on the concentration of species in chemical reaction is negligible, so in the other words, the interaction effects among the tracer species could be ignored.

• **Assumption 8**: The motions in large-scale are modeled as quasi-horizontal so they could be dealt in horizontal and vertical directions respectively.

### 2.4.1 Derivation of the Chemical Transport Model

To make the species continuity equation useful for the simulation of the air quality model, we need to decompose it to the terms of the mean and turbulent component and leave the Jacobian and mapping scale unchanged. Using the Reynolds decomposition on the concentration of species, the processes are expressed as:

\[
\begin{align*}
    c_i &= \bar{c}_i + c'_i, \\
    r_i &= \bar{r}_i + r'_i, \\
    \bar{c}_i + c'_i &= \bar{r}_i \bar{\rho} + r'_i \bar{\rho} + \bar{r}_i \rho' + r'_i \rho',
\end{align*}
\]

where $r_i = \frac{c_i}{\rho}$ is the ratio of the density of the tracer species with respect to the mixture, and the stochastic quantity of the species is divided into the mean ($\bar{}$) and the turbulent ($'$). **Assumption 5** implies the turbulent component is fixed and has the zero mean in the period of averaging time. Learning from [79], the mean and turbulent component of the species and the relationship between the mean and
turbulent could be approximated as:

\[ \bar{c}_i \approx \bar{r}_i \bar{\rho}, \quad (2.5) \]

\[ c'_i = r'_i \bar{\rho} + \bar{r}_i \rho' + r'_i \rho', \quad (2.6) \]

\[ \frac{r'_i \rho'}{\bar{r}_i \bar{\rho}} \approx \frac{(\rho')^2}{(\bar{\rho})^2} \ll 1. \quad (2.7) \]

After employing the Reynolds decompositions on the velocity components and the concentration of species, the ensemble average of mass conservation equation becomes:

\[
\frac{\partial (\bar{c}_i J_z)}{\partial t} + h^2 \nabla_{x \hat{y}} \cdot \left[ \frac{(\bar{c}_i + c'_i) (\bar{v}_{x \hat{y}} + \bar{v}'_{x \hat{y}}) J_z}{h^2} \right] \\
+ \frac{\partial}{\partial \hat{z}} \left[ (\bar{c}_i + c'_i) (\bar{v}_{\hat{z}} + \bar{v}'_{\hat{z}}) J_z \right] = J_z T_{c_i},
\]

(2.8)

where the \( J_z = J_{\hat{z}} \) and \( T_{\hat{z}} = T_{\hat{z}} \) are based on the Assumptions 3 and 6, respectively.

And the ensemble average of Reynolds flux terms \( \bar{c}'_i \bar{v}'_{\{x \hat{y}, \hat{z}\}} \) can be represented by Equation (2.6) and get the approximations:

\[ \bar{c}'_i \bar{v}'_{\hat{z}} \approx \bar{\rho} r'_i \bar{v}'_{x \hat{y}} + \bar{r}_i \rho' \bar{v}'_{\hat{z}} \approx \bar{\rho} r'_i \bar{v}'_{x \hat{y}}, \quad (2.9) \]

\[ \bar{c}'_i \bar{v}'_{\hat{z}} \approx \bar{\rho} r'_i \bar{v}'_{\hat{z}} + \bar{r}_i \rho' \bar{v}'_{\hat{z}} \approx \bar{\rho} r'_i \bar{v}'_{\hat{z}}, \quad (2.10) \]

\[ \bar{r}_i \rho' \bar{v}'_{x \hat{y}} \ll 1, \quad (2.11) \]

\[ \bar{r}_i \rho' \bar{v}'_{\hat{z}} \ll 1. \quad (2.12) \]

Since the \( \rho \) hardly changes, the terms in Equations (2.11) and (2.12) are much smaller than 1 so that they could be ignored in the approximations. And the Equation (2.7) implies the second order term of \( \rho \) in (2.6) is negligible and then we rewrite the Equation (2.8) with Equations (2.5), (2.9) and (2.10):

\[
\frac{\partial (\bar{c}_i J_z)}{\partial t} + h^2 \nabla_{x \hat{y}} \cdot \left[ \frac{\bar{c}_i \bar{v}'_{x \hat{y}} J_z}{h^2} \right] + \frac{\partial (\bar{c}_i \bar{v}_z J_z)}{\partial \hat{z}} \\
+ h^2 \nabla_{x \hat{y}} \cdot \left[ \frac{\bar{r}_i \bar{v}'_{x \hat{y}} J_z}{h^2} \right] + \frac{\partial (\bar{r}_i \bar{v}_z J_z)}{\partial \hat{z}} = J_z T_{c_i},
\]

(2.13)

and the ensemble average of turbulence flux terms \( \bar{r}_i \bar{v}'_{\{x \hat{y}, \hat{z}\}} \) can be parameterized.
further by the eddy diffusion concept (K-theory):

\[
\frac{\partial}{\partial x} \mathbf{v}_x = -\left( \hat{K}^{11} \frac{\partial \bar{r}_i}{\partial x} + \hat{K}^{12} \frac{\partial \bar{r}_i}{\partial y} + \hat{K}^{13} \frac{\partial \bar{r}_i}{\partial z} \right),
\]

\[
\frac{\partial}{\partial y} \mathbf{v}_y = -\left( \hat{K}^{21} \frac{\partial \bar{r}_i}{\partial x} + \hat{K}^{22} \frac{\partial \bar{r}_i}{\partial y} + \hat{K}^{23} \frac{\partial \bar{r}_i}{\partial z} \right),
\]

\[
\frac{\partial}{\partial z} \mathbf{v}_z = -\left( \hat{K}^{31} \frac{\partial \bar{r}_i}{\partial x} + \hat{K}^{32} \frac{\partial \bar{r}_i}{\partial y} + \hat{K}^{33} \frac{\partial \bar{r}_i}{\partial z} \right),
\]

where \( \hat{K}^{jl} \) refers to the eddy diffusion tensor in generalized coordinate. If we presume the eddy diffusion tensor in Cartesian coordinate is diagonal, consequently, the eddy diffusion tensor in generalized coordinate will become:

\[
\hat{K} = \begin{pmatrix}
  h^2 K_{xx} & 0 & h \frac{\partial z}{\partial x} K_{xx} \\
  0 & h^2 K_{yy} & h \frac{\partial z}{\partial y} K_{yy} \\
  h \frac{\partial z}{\partial x} K_{xx} & h \frac{\partial z}{\partial y} K_{yy} & \frac{\partial z}{\partial x} K_{xx} + \frac{\partial z}{\partial y} K_{yy} + \frac{\partial z}{\partial z} K_{zz}
\end{pmatrix},
\]

where \( K_{xx} = \hat{K}^{11}, K_{yy} = \hat{K}^{22} \) and \( K_{zz} = \hat{K}^{33} \) are the diagonal components of eddy diffusion tensor in Cartesian coordinate.

So the diffusion terms \( \frac{\partial}{\partial x} \mathbf{v}_x \) in Equation (2.13) can be parameterized with the eddy diffusion theory Equation (2.14) as the followings:

\[
\frac{1}{\bar{\rho}} \nabla \cdot \frac{\partial}{\partial t} \mathbf{v}_x = h^2 \frac{\partial}{\partial x} \left[ \frac{\bar{\rho} \mathbf{J}_z}{h^2} \right] = h^2 \frac{\partial}{\partial x} \left[ \frac{\bar{\rho} \mathbf{J}_z}{h^2} \left( \hat{K}^{11} \frac{\partial \bar{r}_i}{\partial x} + \hat{K}^{13} \frac{\partial \bar{r}_i}{\partial z} \right) \right]
\]

\[
+ h^2 \frac{\partial}{\partial y} \left[ \frac{\bar{\rho} \mathbf{J}_z}{h^2} \left( \hat{K}^{22} \frac{\partial \bar{r}_i}{\partial y} + \hat{K}^{23} \frac{\partial \bar{r}_i}{\partial z} \right) \right],
\]

and

\[
\frac{\partial}{\partial \bar{\rho}} \frac{\partial}{\partial \bar{\rho} \mathbf{J}_z} \frac{\partial}{\partial \bar{\rho} \mathbf{J}_z} = \frac{\partial}{\partial \bar{\rho}} \left[ \bar{\rho} \mathbf{J}_z \left( \hat{K}^{31} \frac{\partial \bar{r}_i}{\partial x} + \hat{K}^{32} \frac{\partial \bar{r}_i}{\partial y} + \hat{K}^{33} \frac{\partial \bar{r}_i}{\partial z} \right) \right].
\]

Rewriting the Equation (2.13) with Equations (2.16) and (2.17) in separate form of diagonal and off-diagonal components, in addition to the detailed description of source term \( \mathbf{J}_z \mathbf{T}_{ci} \). We can obtain the final governing atmospheric equation on chemical transport model in generalized coordinate (the turbulent flux terms are presented with the eddy diffusion K-theory) [61, 72, 71, 73]:

22
\[
\frac{\partial (\bar{c}_i J_z)}{\partial t} + h^2 \nabla \cdot \left[ \frac{\bar{e} \bar{v} \bar{g} J_z}{h^2} \right] + \frac{\partial (\bar{c}_i \bar{v} \bar{z})}{\partial \bar{z}}
\]

(a) \hspace{1cm} (b) \hspace{1cm} (c)

\[
- h^2 \frac{\partial}{\partial \bar{x}} \left[ \frac{\bar{\rho} J_z}{h^2} \left( \hat{K}_{11} \partial \bar{r}_i \right) \right] - h^2 \frac{\partial}{\partial \bar{y}} \left[ \frac{\bar{\rho} J_z}{h^2} \left( \hat{K}_{22} \partial \bar{r}_i \right) \right] - \frac{\partial}{\partial \bar{z}} \left[ \bar{\rho} J_z (\hat{K}_{33} \partial \bar{r}_i) \right]
\]

(d) \hspace{1cm} (e)

\[
- h^2 \frac{\partial}{\partial \bar{z}} \left[ \bar{\rho} J_z \left( \hat{K}_{13} \partial \bar{r}_i \right) \right] - h^2 \frac{\partial}{\partial \bar{y}} \left[ \bar{\rho} J_z \left( \hat{K}_{23} \partial \bar{r}_i \right) \right]
\]

(f)

\[
- \frac{\partial}{\partial \bar{z}} \left[ \bar{\rho} J_z (\hat{K}_{31} \partial \bar{r}_i + \hat{K}_{32} \partial \bar{r}_i) \right]
\]

(g)

\[
= J_z R_{c_i} (\bar{c}_1, \ldots, \bar{c}_N) + J_z T_{c_i} + \left. \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{clim}} + \left. \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{aero}} + \left. \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{ping}}
\]

(h) \hspace{1cm} (i) \hspace{1cm} (j) \hspace{1cm} (k) \hspace{1cm} (l)

(2.18)

The meaning of each component is as follows:

(a) Rate of change on concentration of species.

(b) Horizontal advection.

(c) Vertical advection.

(d) Horizontal eddy diffusion (diagonal terms).

(e) Vertical eddy diffusion (diagonal terms).

(f) Horizontal eddy diffusion (off-diagonal terms).

(g) Vertical eddy diffusion (off-diagonal terms).

(h) Formation or decomposition process of the chemical reaction.
(i) Emissions.

(j) Clouds mixing and aqueous-phase chemical process.

(k) Aerosol process.

(l) Plume-in-grid process.

It should be stated that the deposition process is included in the vertical diffusion component since it is a flux boundary condition at the bottom layer.

We can further express the turbulence flux terms in a more simple form:

\[ \frac{r_i' v_j'}{h} \{
\begin{array}{l}
\xi = x, y, z
\end{array}
\} = \hat{F}_1 c_i; \quad \frac{r_i' v_j'}{h} \hat{v}_x = \hat{F}_2 c_i; \quad \frac{r_i' v_j'}{h} \hat{v}_z = \hat{F}_3 c_i, \quad (2.19) \]

and the \( \hat{F}^{(1,2,3)}_{c_i} \) suggest they have ready contained the off-diagonal terms. Then the final governing equation of trace species Equation (2.18) will be rewrite with the simplified turbulence flux terms as:

\[ \frac{\partial (\bar{c}_i J_z)}{\partial t} + h^2 \nabla_{\hat{x} \hat{y}} \cdot \left[ \frac{\bar{c}_i \hat{v}_{\hat{x} \hat{y}} J_z}{h^2} \right] + \frac{\partial (\bar{c}_i \hat{v}_{\hat{z}} J_z)}{\partial \hat{z}} \]

\[ + \frac{h^2}{\partial \hat{x}} \left[ \frac{\bar{\rho} J_z}{h^2} \hat{F}_{c_i}^{(1,2)} \right] + \frac{h^2}{\partial \hat{y}} \left[ \frac{\bar{\rho} J_z}{h^2} \hat{F}_{c_i}^{(2)} \right] + \frac{\partial (\bar{\rho} J_z \hat{F}_{c_i}^{(3)})}{\partial \hat{z}} = \hat{\tau} R c_i (\bar{\rho} 1, \ldots, \bar{\rho} N) + \sqrt{\hat{\tau}} T c_i + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{clu}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{aero}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{ping}}. \quad (2.20) \]

If the domain is in a gentle topography, the terms involved in the horizontal gradients with respect to the vertical coordinate can be ignored and the vertical diffusion term will purely related to the orthogonal Cartesian coordinate system at present. In this situation, the Equation (2.20) will become [78]:

\[ \frac{\partial (\bar{c}_i J_z)}{\partial t} + \sqrt{\hat{\tau}} T c_i + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{clu}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{aero}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{ping}} = \left| \frac{\partial (\bar{\rho} \sqrt{\hat{\tau}} \hat{F}_{c_i}^{(3)})}{\partial \hat{z}} \right|_{\text{chem}} + \left| \frac{\partial (\bar{\rho} \sqrt{\hat{\tau}} \hat{F}_{c_i}^{(3)})}{\partial \hat{z}} \right|_{\text{adv}} + \left| \frac{\partial (\bar{\rho} \sqrt{\hat{\tau}} \hat{F}_{c_i}^{(3)})}{\partial \hat{z}} \right|_{\text{diff}}, \quad (2.21) \]

\[ \Rightarrow \quad \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{clu}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{aero}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{ping}} = \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{chem}} + \sqrt{\hat{\tau}} T c_i + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{clu}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{aero}} + \left| \frac{\partial (\bar{c}_i J_z)}{\partial t} \right|_{\text{ping}}. \quad (2.22) \]
where $c_i^* = \sqrt{\tau} \hat{c}_i = \frac{J_z}{h^2} \hat{c}_i$. Equation (2.22) explicitly identifies the numerical descriptions of the science process modules implemented in the CMAQ and Figure 2.6 shows the linkages of the CMAQ science processors [8] among the interface processes, science processes, data provider modules and driver module.

![Science process modules in CMAQ](image)

Figure 2.6: Science process modules in CMAQ

The Interface process modules are shown in the rectangular boxes. At the same time, science process modules are shown in circles and responsible for updating the concentration field directly after the data provider modules following their routines to offer suitable environmental input data to the science process modules. In addition, CMAQ driver module works on solving the numerical integration of the coupled and decouple terms across the science processes. It is to be noted that the concentrations are linked with solid lines and other environmental data with broken lines.
2.4.2 Advection and Diffusion

The transport process of pollutant includes two terms: advection and sub-grid-scale diffusion. Advection is caused by the mean wind fields while diffusion is due to the turbulence of the mixed pollutants in sub-grid-scale. If a pollutant is mainly controlled by the advection, its concentration will be maintained stable in a long distance. On the contrary, if the diffusion is the domination for a pollutant, a quickly mixing producer of the pollutants will occur around the emission and it will lead to the many changes on the concentration of the pollutant.

- Advection

The Advection is divided into horizontal and vertical components. The difference is because the mean wind field is always horizontal and the vertical movement is the interactive effect of thermodynamics and dynamics. Advection follows the rule of mass conservation of the species continuity equation and its data is kept consistent with the meteorology input data in the whole simulations of thermodynamics and dynamics. Equation (2.23) is the advection process in the governing equation of chemical transport model,

\[
\left. \frac{\partial (c_i^*)}{\partial t} \right|_{\text{adv}} = -\nabla \hat{x} \hat{y} \cdot (c_i^* \vec{v}_{\hat{x}\hat{y}}) - \frac{\partial (c_i^* \vec{v}_z)}{\partial \hat{z}} .
\]  

(2.23)

The Horizontal Advection module uses the piecewise parabolic method (PPM). It describes the advected scaler in the finite-volume subgrid and the distribution of each grid internal is based on the parabola. PPM is a monotonic and positive-definite scheme so that it assures the input values will never become negative. The module solves the horizontal advection as:

\[
\left. \frac{\partial (c_i^*)}{\partial t} \right|_{\text{hadv}} = -\nabla \hat{x} \hat{y} \cdot (c_i^* \vec{v}_{\hat{x}\hat{y}}) .
\]  

(2.24)

The Vertical Advection is derived from the PPM of horizontal advection by using the driving meteorological model’s air density. And the first assumption of vertical advection is that there is no mass-exchange from the bottom and the top of the domain. Also, in the simulation, the steepening procedure will be applied when CCTM detects a strong gradient in the distribution of the pollutant. The
module solve for the vertical advection with boundary conditions \( \hat{v}_z = 0 \) at the bottom and the top of the air quality model is:

\[
\frac{\partial (c_i^*)}{\partial t}
\bigg|_{\text{vadv}} = -\frac{\partial (c_i^* \hat{v}_z)}{\partial \hat{z}} .
\tag{2.25}
\]

We try to emphasize the difference between horizontal and vertical advection in the advection process. However, practically, they are considered and dealt with in an integral component for solving the physical advection process on the tracer species.

- **Diffusion**

The atmospheric diffusion is divided into vertical and horizontal components. The difference is that the vertical diffusion mainly expresses the thermodynamics influences on the turbulence effect by the energy exchange in air-surface, in the meantime, the horizontal diffusion stands for the mixing situation in a sub-grid scale caused by wind fluctuations. We start from the diffusion term in the equation of chemical transport model in the same concentration units as used in advection:

\[
\frac{\partial (c_i^*)}{\partial t}
\bigg|_{\text{diff}} = -\frac{\partial (\bar{\rho} \sqrt{\hat{\tau}} \hat{F}^3_{c_i})}{\partial \hat{z}} - \nabla_{\hat{x}\hat{y}} \left[ \bar{\rho} \sqrt{\hat{\tau}} \hat{F}_{c_i} \right] + \sqrt{\hat{\tau}} T_{c_i} .
\tag{2.26}
\]

Asymmetric Convective Method (ACM) is employed to present the **Vertical Diffusion**. It assumes that, under convective conditions (warm surface), the heated air will be vertically transported by buoyancy and mixes with ambient air at each level until the rising rate of the temperature decreases to that of the ambient temperature. That results in the air moving fast in the narrow up-direction but moving slowly in the broader down-direction, and this determines a asymmetric phenomenon. On the other hand, if the surface is cool, the condition will be non-convective. The vertical diffusion will be presented by an eddy diffusivity. The eddy diffusivity uses a local mixing scheme and is estimated from the planetary boundary layer (PBL) similarity-based algorithm. The numerical description of vertical diffusion is shown in Equation (2.27)

\[
\frac{\partial (c_i^*)}{\partial t}
\bigg|_{\text{vdiff}} = -\frac{\partial (\bar{\rho} \sqrt{\hat{\tau}} \hat{F}^3_{c_i})}{\partial \hat{z}} + \sqrt{\hat{\tau}} T_{c_i} .
\tag{2.27}
\]
We note that the emission $T_c$ [5, 79] is included either in vertical diffusion process or the gas-phase chemistry process.

The **Horizontal Diffusion** is put into practice with an eddy diffusion similar to the vertical diffusion under non-convective conditions. This eddy diffusion is generated from the local wind deformation in grid-cell scale and assumed to be uniform but is dependent on the resolution of the model. It will become large in the area where the diffusion is smaller compared to the advection in a high-resolution case. The scientific part of the horizontal diffusion is expressed in Equation (2.28)

$$\left. \frac{\partial (c^*_i)}{\partial t} \right|_{\text{hdiff}} = -\nabla \bar{z} [\bar{\rho} \sqrt{\tau} \hat{F}_{c_i}] .$$

(2.28)

When the turbulent fluxes $\hat{F}_{c_i}^{(1,2,3)}$ are presented by the eddy diffusion $K$-theory, the contribution of the off-diagonal (the interaction terms in cross directions) diffusion components are clearly shown in Equation (2.18) as:

$$\left. \frac{\partial (c^*_i)}{\partial t} \right|_{\text{cdiff}} = \frac{\partial}{\partial \bar{x}} \left[ \bar{\rho} \sqrt{\tau} (\hat{K}^{13} \frac{\partial \bar{r}_i}{\partial \bar{z}}) \right] + \frac{\partial}{\partial \bar{y}} \left[ \bar{\rho} \sqrt{\tau} (\hat{K}^{23} \frac{\partial \bar{r}_i}{\partial \bar{z}}) \right] + \frac{\partial}{\partial \bar{z}} \left[ \bar{\rho} \sqrt{\tau} (\hat{K}^{31} \frac{\partial \bar{r}_i}{\partial \bar{x}} + \hat{K}^{32} \frac{\partial \bar{r}_i}{\partial \bar{y}}) \right].$$

(2.29)

When the domain has the significant feature on topographic, the off-diagonal diffusion terms can be applied. However, for the CMAQ, these terms are not included. As a result, the diffusion module in air quality is will just rely on the diagonal diffusion terms as follows:

$$\left. \frac{\partial (c^*_i)}{\partial t} \right|_{\text{hdiff}} = \frac{\partial}{\partial \bar{x}} \left[ \bar{\rho} \sqrt{\tau} (\hat{K}^{11} \frac{\partial \bar{r}_i}{\partial \bar{x}}) \right] + \frac{\partial}{\partial \bar{y}} \left[ \bar{\rho} \sqrt{\tau} (\hat{K}^{22} \frac{\partial \bar{r}_i}{\partial \bar{y}}) \right],$$

(2.30)

$$\left. \frac{\partial (c^*_i)}{\partial t} \right|_{\text{vdiff}} = \frac{\partial}{\partial \bar{z}} \left[ \bar{\rho} \sqrt{\tau} (\hat{K}^{33} \frac{\partial \bar{r}_i}{\partial \bar{z}}) \right].$$

(2.31)

### 2.4.3 Deposition

As was mentioned, the deposition process [16, 12] is a loss on the mass of pollutant resulting from the vertical diffusion process at the bottom of the air quality model. It is estimated by a flux boundary condition and affects the concentration in the lowest vertical layer of the domain. Considering the deposition process as a diffusion flux,
we can link the boundary condition in the generalized coordinate with that in the Cartesian coordinate,

$$
\hat{F}_c^3\big|_{\text{dep}} = \left( \frac{\partial \hat{z}}{\partial x} \right) \hat{F}_c^x\big|_{\text{dep}} + \left( \frac{\partial \hat{z}}{\partial y} \right) \hat{F}_c^y\big|_{\text{dep}} + \left( \frac{\partial \hat{z}}{\partial z} \right) \hat{F}_c^z\big|_{\text{dep}}.
$$

(2.32)

Since $\hat{F}_c^x\big|_{\text{dep}}$ and $\hat{F}_c^y\big|_{\text{dep}}$ do not exist, the effect of deposition in dry air is accounted on the following:

$$
\frac{\partial (c_i^*)}{\partial t}\big|_{\text{dep}} = \frac{\partial (\rho \sqrt{\tau} \bar{r}_i)\big|_{\text{chem}}}{\partial t} = -\frac{\partial (\rho \sqrt{\tau} \hat{F}_c^z\big|_{\text{dep}})}{\partial \hat{z}} = -\frac{\rho \sqrt{\tau} \left( \frac{\partial \hat{z}}{\partial z} \right) F_c^z\big|_{\text{dep}}}{\partial \hat{z}}
$$

$$
\approx -\frac{\partial (\hat{z})_{\text{dep}}}{\partial \hat{z}} (\rho \sqrt{\tau} \bar{T}_c)\big|_{\text{layer1}} = -\frac{V_{\text{dep}} (\rho \sqrt{\tau} \bar{T}_c)\big|_{\text{layer1}}}{h_{\text{dep}} c_i^*\big|_{\text{layer1}}},
$$

(2.33)

where $h_{\text{dep}} = (\frac{\partial \hat{z}}{\partial \hat{z}})_{\text{dep}} (\Delta \hat{z})_{\text{dep}}$ is the height of the lowest layer and $V_{\text{dep}}$ is the deposition velocities at the middle of the lowest layer of the tracer species.

### 2.4.4 Gas-phase Chemistry Solvers and Process

The rate of change of $c_i$ on gas-phase chemistry process is given by:

$$
\frac{\partial (c_i^*)}{\partial t}\big|_{\text{chem}} = \sqrt{\tau} R_{c_i}(c_1, \ldots, c_N) + \sqrt{\tau} T_{c_i},
$$

(2.34)

where $R_{c_i}$ and $T_{c_i}$ refer to the chemistry reactions and emission, respectively.

To simulate the gas-phase chemistry $R_{c_i}$ in troposphere, CCTM has developed various modules range from simple linear and nonlinear system for the aim of engineering to the comprehensive chemistry descriptions on finding the trace of pollutant in air quality protection. The existing photolysis mechanisms [81, 49, 74] in CMAQ to simulate the gas-phase chemistry is: CB6, CB05, SAPRC-07, SAPRC-09, and RACM.

CCTM is a differential equation from Equation (2.34) governing the chemical reaction kinetics and species conservation. It needs the numerical technique to be solved to get the concentration of species, the rate of formation or the depletion. Currently, the CCTM has three options for solving the chemical transformations: they are the Rosenbrock solver (ROS3), the Euler Backward Iterative solver (EBI), and the Sparse Matrix Vectorized GEAR (SMVGEAR) solver.
2.4.5 Aerosol Process

The aerosols are referred as particulate matter (PM) [7] in the air quality community. It can be either the primary pollutant emitted from nature and industry or the secondary pollutant which has reacted chemically from existing or other new primary particulates. The cloud process also provides the suitable conditions for the formation of the particulate matter.

Solving the chemistry reactions and dynamics changes on concentrations of aerosol species are processed by Equation (2.35):

\[
\left. \frac{\partial (c^*_i)}{\partial t} \right|_{\text{aero}} = \sqrt{\tau} R_{\text{aero}_i}(c_1, \ldots, c_N) + \sqrt{\tau} T_{\text{aero}_i} - \hat{v}_g \frac{\partial c^*_i}{\partial z},
\]

where \( R_{\text{aero}_i} \) indicates the formation and growth and depletion processes on particles while \( T_{\text{aero}_i} \) represents the emission. Besides, \( \hat{v}_g \) is the velocity of contravariant sedimentation.

CCTM presents the particle size distribution as the superposition of three log-normal sub-distributions, known as modes. There are two modes for the PM2.5 (the particulate matters whose diameter are less than or equal to 2.5mm). They are Aitken and accumulation. Besides PM2.5, there is a coarse mode for the particulate matters whose diameter are less than or equal to 10mm. Therefore, the PM10 is the combination of the PM2.5 and a coarse mode.

The key functions of the aerosol processes for CCTM are:

1. The removal process of aerosol by the size-dependent dry deposition;
2. The removal process and interaction of the aerosol-cloud droplet by precipitation;
3. The formation of the new particle in a sulfuric acid/water vapour system by binary homogeneous nucleation;
4. The production of an organic aerosol term from gas-phase precursors;
5. The growth of coagulation and condensation on particle.
2.4.6 Cloud Mixing and Aqueous-Phase Chemistry (CLOUD)

Clouds are related to the aqueous chemical reactions, vertical mixing process and the wet deposition of pollutants in CCTM so that it becomes a very important component in the air quality model. In the meanwhile, for the physical aspect, the clouds will influence the solar radiation on the air and the surface which affects the photochemical process [12, 74, 47] of pollutants and the flux of biogenic emissions. Thus, the clouds module of CCTM will be represented as some functions based on the chemical and physical processes. The clouds module is generally modeled in two types: sub-grid nonprecipitating clouds and grid-resolved clouds,

\[
\frac{\partial(c_i^+)}{\partial t} \bigg|_{\text{clld}} = \frac{\partial(c_i^+)}{\partial t} \bigg|_{\text{subclld}} + \frac{\partial(c_i^+)}{\partial t} \bigg|_{\text{resclld}}. 
\]

(2.36)

The Meteorological module (MCIP) will responsible for the second type - grid-resolved clouds and only considers the effects of the scavenging and aqueous-phase chemistry in Equation (2.37),

\[
\frac{\partial(c_i^+)}{\partial t} \bigg|_{\text{resclld}} = \frac{\partial(c_i^+)}{\partial t} \bigg|_{\text{scav}} + \frac{\partial(c_i^+)}{\partial t} \bigg|_{\text{aqchem}}. 
\]

(2.37)

The rest one type is calculated in the CCTM. It redistributes the pollutants in vertical direction, calculates the in-cloud and precipitation scavenging, propagates the aqueous chemistry calculations, and performs the wet deposition process,

\[
\frac{\partial(c_i^+)}{\partial t} \bigg|_{\text{subclld}} \sim f(\text{red}, \text{scav}, \text{aqchem}, \text{wdep}). 
\]

(2.38)

2.4.7 Plume-in-Grid (PING)

At present, nitric oxide, ozone, sulfide gases, aerosols and acidic species are largely discharged from major point sources, industrial area sources, and moving vehicles. The emissions from major point source especially cannot be estimated accurately in a narrow space by using the regional Eulerian air quality models. Thus, a new approach, called the plume-in-grid (PING) is generated and employed in CCTM. It offers a better practical treatment on simulating the physical emitting and chemical reaction processes of the major point sources in sub-grid scale.

\[
\frac{\partial c_p}{\partial t} = \frac{\partial c_p}{\partial t} \bigg|_{\text{disp}} + \frac{\partial c_p}{\partial t} \bigg|_{\text{emis}} + \frac{\partial c_p}{\partial t} \bigg|_{\text{chem}} + \frac{\partial c_p}{\partial t} \bigg|_{\text{dep}}, 
\]

(2.39)
where $\partial c_p$ is the concentration of the sub-grid plume and the rate of change terms with subscripts disp, emis, chem, and dep represent the effects of plume dispersion, point source emissions, plume chemistry, and dry deposition in the plume, respectively. When the simulation of phase of plume in sub-grid scale has been completed, the PING module will update the concentrations of the grid scale with:

$$\frac{\partial c_i}{\partial t} \bigg|_{\text{ping}} = \frac{\delta V_p}{\delta V} \frac{\partial (c_p - c_{ibg})}{\partial t},$$  

where $c_{ibg}$ is the background concentration and $\delta V_p$ is the volume of plume in a volume of grid cell $\delta V$. Currently, only the gaseous substances are dealt with the PING module.
Chapter 3

Data Assimilation Methods

A numerical model determines how a model state in a particular time changes into the model state at a later time. We hope the numerical model is a perfect representation of the actual system in order to make a perfect forecast of the future state of the actual system. This rarely happens in practice, but with the accumulation of uncertainty, simulation may go further away from the truth. Thus, we need some certain values to adjust the model along the right path. Certainly, it is better to use all established observations as the model state to do the forecast, but to be practical case, this is costly, or even impossible in some situations. Therefore, overall, we can only use some established or known observations to adjust the value of the model to improve the prediction of that particular model. This method is called the Data Assimilation method. Data assimilation is the process whereby the integrating model forecast value and measurements to get an optimal estimation of the situation. The application of data assimilation arises in many fields of geosciences, and is important mainly for use in weather forecasting, air quality prediction and hydrology. For air quality, it is an essential tool for the analysis and estimating activities.

Ever since the numerical model has propagated the information from past model value to the current time and then is combined with current observations of the actual system using a data assimilation method. Most commonly, this leads to the numerical modeling system which will alternately performing a numerical forecast and a data analysis. This is known as analysis/forecast cycling. The forecast step does the simulation from the previous analysis value up to the current one is frequently called
the forecast value or background value \( (f) \). Then the analysis step considers this forecast value and measurements altogether to get the estimation value \( (a) \) on truth. Generally, a weight mean is used to mix them: estimating the uncertainty of the model value and the measurements to determine the weight factors. Data assimilation is always a multivariate problem, so evaluating the relationships between the variables is another necessary approach. Moreover, the observations are from the actual state rather than the model, and it leads to an incremental analysis to ascertain which and how much should be added to the background. At times, compared to the model value, these increments are so minor that the simulation of models will ”devour” these small changes because of some unstable models. At such times, we need to use some filters to avoid the problem.

### 3.1 Optimal Interpolation (OI)

Optimal Interpolation (OI) can be seen as an approximation of the Kalman filter where the state error covariance matrix is not propagated using the dynamics of the physical model but the matrix may change with time in other ways. Typically, when they were running OI, the Numerical Weather Prediction Centers had a different covariance matrix for each month typically. OI reaches its limits when the dynamics of the day significantly determine the covariance errors. OI is still in use in several oceanographic operational centers. There are two approaches to properly form the (static) covariance matrix.

Without loss of the generality for each tracer species, the truth of the tracer species is presented as \( x^i \) and will omit the subscript which denotes the species \( i \). After that, we first give three estimates on the truth \( x^i \) in a discrete-time system first. They are:

\[
\begin{align*}
f &= x^i + w^f, \\
a &= x^i + w^a, \\
y &= H(x^i) + v,
\end{align*}
\]  

(3.1)

where \( x^i \) means the truth(t) of the model state \( x, f \in \mathbb{R}^n \) is the forecast or the
background state of the air quality model; \( a \in \mathbb{R}^n \) is the best estimation of the truth \( x^t \); \( y \in \mathbb{R}^m \) is the observations; \( w^f, \ w^a \in \mathbb{R}^n \) and \( v \in \mathbb{R}^m \) are the noises within the background state by estimation and observation, respectively; and \( H \in \mathbb{R}^{m \times n} \) is the mapping operator (or forward model operator) from the model space to the observational space. To pursue the best estimation of \( a \) on \( x \) from \( f \), we still need the additional assumptions of:
\[
\begin{align*}
(w^f)(w^f)^T &= F , & (w^a)(w^a)^T &= A , & (v)(v)^T &= V , \\
(w^f)(v)^T &= 0 , & (w^a)(v)^T &= 0 ,
\end{align*}
\]
where \( F \in \mathbb{R}^{n \times n} \), \( A \in \mathbb{R}^{n \times n} \) and \( V \in \mathbb{R}^{m \times m} \) are the background error covariance matrix, estimation error covariance matrix and observation error covariance matrix respectively.

Now, the analysis equation of optimal interpolation is assumed as
\[
a = f + W(y - H(f)) ,
\]
(3.2)
and \( W \in \mathbb{R}^{n \times m} \) is the weight matrix. By using the Equation (3.1) and the additional assumptions, the errors of analysis equation Equation (3.2) becomes:
\[
\begin{align*}
x^t + w^a &= x^t + w^f + W[H(x^t) + v - H(x^t + w^f)] \\
w^a &= w^f + W[H(x^t) + v - H(x^t + w^f)] \\
w^a &\approx w^f + W[H(x^t) + v - H(x^t) - Hw^f] \\
w^a &\approx w^f + W(v - Hw^f) .
\end{align*}
\]
(3.3)
To get the third line denotation in Equation (3.3), we need do an approximation on the third term in the bracket with the Taylor series truncated after the linear term. Here the new notation \( H \in \mathbb{R}^{m \times n} \) is called ”Tangent Linear Forward Model operator” and is defined as:
\[
H = \frac{\partial H}{\partial x} \bigg|_f ,
\]
(3.4)
where \( H \) is the derivative of the forward model operator with respect to the model state. Therefore, we have performed a linearization of the nonlinear observational
operator in the model state, assuming the truth is not too far from the forecast. In order to form the error covariance in the analysis step, we multiply the Equation (3.3) with its transpose and apply the expectation operator:

\[ w^a(w^a)^T = [w^f + W(v - Hw^f)][w^f + W(v - Hw^f)]^T \]

\[ A = w^f(w^f)^T + [W(v - Hw^f)][W(v - Hw^f)]^T + W(v - Hw^f)(w^f)^T + w^f[W(v - Hw^f)]^T \]

\[ A = F + W(V + HFH^T)W^T - FHW^T - WHF. \]  \hspace{1cm} (3.5)

To obtain the minimum estimation error covariance of \( A \), we take a derivative on its trace with respect to the \( W \),

\[ 0 = d\text{Tr} A \over dW = 2W(V + HFH^T) - 2FH^T, \] \hspace{1cm} (3.6)

and finally get

\[ W = FH^T(HFH^T + V)^{-1}. \] \hspace{1cm} (3.7)

In conclusion, the algorithm of optimal interpolation (OI) of the data assimilation method including the two steps (forecast step and analysis step) within three equations: the analysis equation (3.2); the weight matrix (3.7); and the simulation on air quality model.

**Forecast step:**

\[ f = L(a). \]

**Analysis step:**

\[ a = f + W(y - H(f)) \]

\[ W = FH^T(HFH^T + V)^{-1}. \]

### 3.2 Variational Methods

The variational method is a mainstream method within the data assimilation methods. It considers a cost function \( P \) to measure the misfit among the model state \( x \) to the background state \( f \) and the observations \( y \). In the variational method, the aim is...
to find the specific $x$ that gives the minimum $P$ by using the least squares method. The $x$ that achieves this minimum is known as estimation $a$. The simplest version of the variational method is:

$$P(x) = \frac{1}{2}(x - f)^T(x - f) + \frac{1}{2}[y - H(x)]^T[y - H(x)] ,$$

(3.8)

and

$$a = \arg\min_x P(x) = \frac{1}{2}(x - f)^TF^{-1}(x - f) + \frac{1}{2}[y - H(x)]^TV^{-1}[y - H(x)] .$$

(3.9)

### 3.2.1 3D-Var: Three-Dimensional Variation Method

However, since Equation (3.8) does not consider the uncertainties in $f$ and $y$, the result would not be good when the tendencies among the variables are differ a lot. After taking the background error covariance $F$ and observation error covariance $V$ into account to measure the uncertainties, the new cost function will be as follows:

$$P(x) = \frac{1}{2}(x - f)^TF^{-1}(x - f) + \frac{1}{2}[y - H(x)]^TV^{-1}[y - H(x)] ,$$

(3.10)

and

$$a = \arg\min_x P(x) = \frac{1}{2}(x - f)^TF^{-1}(x - f) + \frac{1}{2}[y - H(x)]^TV^{-1}[y - H(x)] .$$

(3.11)

When $H$ has the same tangent linear hypothesis same as it in Equation (3.3), the 3D-Var method is equivalent to the Optimal Interpolation (OI) and could be solved in the level of the matrix as well.

However, another way exists to deal with the problem. Here, this tangent linear operate as $H$:

$$x = x^t + \delta x ,$$

$$H(x) = H(x^t) + H\delta x .$$

(3.12)
The gradient of $P$ will be like

$$\nabla_x P = F^{-1}(x - f) - H^T V^{-1}[y - H(x)], \quad (3.13)$$

and the solution can be solved using the gradient decent method.

Moreover, if $H$ is a linear model. Then at this time,

$$Hx = Hx^t + H\delta x, \quad (3.14)$$

and then assume that $\delta x = w^a$, $\nabla_a P = 0$,

$$0 = \nabla_a P = F^{-1}(a - f) - H^T V^{-1}[y - Ha].$$

The result is that the solution of (3.10) can be found by solving a linear system:

$$(F^{-1} + H^T V^{-1}H)a = F^{-1}f + H^T V^{-1}y, \quad (3.15)$$

with the conjugate method or newton method and so on.

### 3.2.2 Advantages of 3D-Var compared to the OI

Even though OI and 3D-Var are equivalent when the forward model $H$ is a linear model, there still exists some differences in the practical implement of the methods. The advantages of the 3D-Var method compared to the OI are:

- The 3D-Var method has more possibilities of pursuing the global performance than OI because OI only extracts parts of the information of background state and mixes them with the information from observations and then the whole background is updated with the feedback of these processed information but the 3D-Var has the option to obtain an accurate globe solution.

- The 3D-Var method can be treated as a linear system sometimes as there are some classical numerical methods to solve it quickly and efficiently. It is unlike the OI, which may have required much resources to be spent on the explicitly inverse process.

- The 3D-Var has the potential to easily include more constrains. In some cases, we can pursue an inaccurate solution where the situation of the equation has a large number of constrains but unfortunately this can’t be done by the OI.
3.2.3 4D-Var: Four-Dimensional Variation Method

Four-dimensional variational data assimilation is an extension of 3D-Var. It obtains the best estimation from considering its effect within a moment to letting it win within a period of future time.

Equation (3.16) shows that 4D-Var measures the difference between the forecast values and the observations in the observational space for a period of time in the future. It makes the estimation direct more attention to the future development which is more reasonable to enhance the aim of prediction.

\[
\begin{align*}
P(x) &= \frac{1}{2}(x - f)^TF^{-1}(x - f) \\
   &\quad + \frac{1}{2}\sum_{i=0}^{p}(y_i - H_i[L_0\rightarrow_i(x)])^TV_i^{-1}(y_i - H_i[L_0\rightarrow_i(x)]) \\
   &= P_b(x) + P_o(x)
\end{align*}
\]

and

\[
\begin{align*}
a &= \arg\min_x P(x) = \frac{1}{2}(x - f)^TF^{-1}(x - f) \\
   &\quad + \frac{1}{2}\sum_{i=0}^{p}(y_i - H_i[L_0\rightarrow_i(x)])^TV_i^{-1}(y_i - H_i[L_0\rightarrow_i(x)])
\end{align*}
\]

where \(L_0\rightarrow_i\) is the predefined chemical transport model operator from time 0 to time \(i\), \(p\) is the length of time interval after the analysis time. Thus, 4D-Var is a nonlinear constrained optimization problem on operator \(L_{k\rightarrow i}\) and is difficult to solve in general cases. Fortunately, there are two hypotheses to make it easier:

- **Causality**: The forecast model \(L_0\rightarrow_i\) can be described as the product of intermediate forecast steps \((L_0, \ldots, L_i)\), which implies the causality of nature. Generally, it starts from the model state \(x\) as the initial condition and the rest of the forecast states are generated with the corresponding \(f_i = L_i(f_{i-1})\) or by the recurrence:

\[
f_i = L_i(L_{i-1}(L_{i-2}(\ldots(L_1(L_0x))\ldots)))
\]

- **Tangent linear hypothesis**: Since the cost function of 4D-Var is quadratic
and $H$ has the tangent linear hypothesis. Then the $L$ could be linearized as:

$$y_i - H_i[L_{0\rightarrow i}(x)] = y_i - H_i[L_{0\rightarrow i}(x^T)] - H_i L_{0\rightarrow i} \delta(c),$$

where $L$ is the tangent linear model (the differential of $L$). It is similar as the tangent linear hypothesis $H$ in Equation (3.3). It implies that the tangent linear hypothesis is applied not only on the data assimilation system, but also on the model.

These two hypothesizes simplify the problem to a unconstrained quadratic problem and make it easier to solve by using numerical methods.

In order to get the minimum of $P(x)$, we need to find the best estimation $a$ on model state $x$ to make the $\nabla_a P = 0 = \nabla_a P_b - \nabla_a P_o$. And for the first term $\nabla_x P_b$, the same strategy is applied as the first term in Equation (3.13), $\nabla_x P_b = F^{-1}(x - f)$.

The complex one is the second term $\nabla_x P_o$. In the following, we will give detailed procedures to get the result of it.

1. Store each forecast $f_i$ in the time window

   $$f_i = L_i(L_{i-1}L_{i-2} \ldots (L_1(L_0 x))), \quad i = 0, \ldots, p;$$

2. Store each “normalized innovation” in the time window

   $$d_i = V_i^{-1}(y_i - H_i(f_i)), \quad i = 0, \ldots, p;$$

3. The calculation of $\nabla_x P_o$ is

   $$-\nabla_x P_o = \sum_{i=0}^{p} \nabla_x P_o^i$$

   $$= \sum_{i=0}^{p} L_i^T L_i^T L_{i-1}^T \ldots L_0^T d_i$$

   $$= L_0^T d_0 + L_1^T [H_1^T d_1 + L_2^T [H_2^T d_2 \ldots [L_p^T H_p^T d_p] \ldots] \ldots ].$$

   (3.19)

In the above step, numerically, we can set up an adjoint variable $\tilde{x}_i = 0$, and then add the adjoint forcing $H_i^T d_i$. After that is performing the adjoint integration by multiplying it with the corresponding model operator $L_i^T$. Taking
this analogy \( \tilde{x}_{i-1} = L_i^T[\tilde{x}_i + H_i^T d_i] \) until we have reached the end of recurrence.

In the end, \( \tilde{x}_0 = \nabla_x P_o \), which is the required result.

After getting the \( \nabla_x P_b \) and \( \nabla_x P_o \), we can obtain \( \nabla_x P \) and use the gradient decent method to solve. Further, if the forecast model \( L \) and forward operator \( H \) are linear model, then we state that \( \nabla_x P = 0 \) as a linear system to solve by using the conjugate gradient method, newton method and other classical numerical methods.

### 3.2.4 Characteristics of 4D-Var compared to the 3D-Var and OI

Compared to the 3D analysis algorithm in a sequential assimilation system, the 4D-Var has the following characteristics:

- 4D-Var works under the assumption that the forecast model is perfect, or the problem will occur only when the error of model is very large. For the 3D analysis methods (3D-Var and OI), that is not a problem.

- 4D-Var is required to do many forecast simulations of \( L \) on the current model state \( x \) which depends on the length of the time window for every iteration. Also, it needs the adjoint of each \( L \) within the time window. In other words, the computation can be very expensive if the model is complex. On the contrary, the computations on 3D-Var and OI are cheap and feasible.

- For the application on a real-time system, the 4D-Var method needs to wait for the observations within the time interval to be ready before entering the analysis step. While, the Kalman filter method and other sequential assimilation method can figure out the prediction very rapidly after the current observation is available. It means the 4D-Var will delay the prediction.

- 4D-Var will use the model state \( x \) as the initial condition to do the forecast and balances the errors between the forecasts and the observations. This guarantees the estimation will be completely consistent with the forecast model and observation within the time internal. Thus, it is very suitable for the classical numerical forecast system.
3.3 Kalman Filter Methods

We have discussed that the Optimal Interpolation minimizes the expected analysis error covariance, but the 3D-Var and 4D-Var methods, which solve essentially the same problem minimize the cost function. In these methods, the forecast (or background) error covariance matrix is estimated once and for all, as if the forecast errors were statistically stationary.

Some researches have shown, however, that there is large day-to-day variability in the model forecast error (with a time scale of a few days) and the variability is about as large as the average error. It points to the importance of the "errors of the day", which on the large scales are dominated by the baroclinic instabilities of synoptic time scales, and which are ignored when the forecast error covariance is assumed to be constant. In this section we will give a brief introduction to the more advanced (and much costlier) schemes that include, at least implicitly, the evolution of the forecast error covariance.

3.3.1 EKF: Extended Kalman Filter

In this section, we will present a popular data assimilation method - Kalman filter method. Considering the dynamical forecast model is a sequential system and using time \( k \) as the subscript. The model is thought to be imperfect and presented as:

\[
\begin{align*}
\mathbf{f}_k &= L_k (\mathbf{a}_{k-1}) + \mathbf{q}_k, \\
\mathbf{y}_k &= H_k (\mathbf{x}_k^t) + \mathbf{v}_k,
\end{align*}
\]

where \( L_k \) is the forecast model same as Equation 3.18 and \( \mathbf{q}_k \in \mathbb{R}^n \) is the forecast model error. It is assumed that \( \mathbf{q}_k \) and \( \mathbf{v}_k \) are stationary zero-mean white noises processed with covariance matrices \( \mathbf{Q}_k \in \mathbb{R}^{n \times n} \) and \( \mathbf{V}_k \) of Gaussian distribution. And we learn from the previous sections that:

\[
\begin{align*}
\mathbf{f}_k &= \mathbf{x}_k^f + \mathbf{w}_k^f, \\
\mathbf{a}_{k-1} &= \mathbf{x}_{k-1}^f + \mathbf{w}_{k-1}^a .
\end{align*}
\]
Thus the relationship among the forecast model error $q_k$, forecast state error $w^f_k$ and estimation state error $w^a_k$ are derived as:

\[
\begin{align*}
  f_k &= L_k(a_{k-1}) + q_k \\
  x^f_k + w^f_k &= L_k(x^f_{k-1} + w^a_{k-1}) + q_k \\
  L_k(x^f_{k-1}) + w^f_k &= L_k(x^f_{k-1} + w^a_{k-1}) + q_k \\
  L_k(x^f_{k-1}) + w^f_k &\approx L_k(x^f_{k-1}) + L_kw^a_{k-1} + q_k \\
\end{align*}
\]

(apply Tangent linear hypothesis)

\[
  w^f_k = L_kw^a_{k-1} + q_k ,
\]

where $w^f_k$, $w^a_k$ are assumed as zero-mean and noises with Gaussian distribution as well.

The Kalman filter theory is started from a probability density function $f(x_k)$ and a likelihood function $f(y_k|x_k)$ for the measurement $y_k$. Also, we have the Bayesian theorem:

\[
f(x_k|y_k) \propto f(x_k)f(y_k|x_k) .
\]  

(3.24)

Thus, the posterior density for $x_k$ given the observation $y_k$, is the product of the prior density for $x_k$ and the likelihood for the observation $y_k$.

Considering that the distribution of the forecast in Equation (3.22) and observation in Equation (3.21) are in the Gaussian statistics so as to we can define the prior density $f(x_k)$ and likelihood for the observation $f(y_k|x_k)$ as:

\[
f(x_k) \propto \exp\left(-\frac{1}{2}[x_k - f_k]^T F^{-1}_k [x_k - f_k]\right) ,
\]  

(3.25)

and

\[
f(y_k|x_k) \propto \exp\left(-\frac{1}{2}[H(x_k) - y_k]^T V^{-1}_k [H(x_k) - y_k]\right) .
\]  

(3.26)

Thus, the posterior density $f(x_k|y_k)$ will be written as:

\[
f(x_k|y_k) \propto \exp\left(-\frac{1}{2}P(x_k)\right) ,
\]  

(3.27)

where

\[
P(x_k) = [x_k - f_k]^T F^{-1}_k [x_k - f_k] + [y_k - H(x_k)]^T V^{-1}_k [y_k - H(x_k)] .
\]  

(3.28)
Soon, you will find that this equation is the same as Equation (3.10) at time $k$. The least square solution $a_k$ is the minimum of $P(x_k)$ and the maximum of $f(x_k|y_k)$ which means it is the maximum likelihood on the estimation. This conclusion is always right when all the errors are in the normal distribution.

As a matter of fact, the derivation of analysis state of the extended Kalman filter is same as that of OI, meanwhile, the weight matrix $W$ of OI is same as the Kalman gain $K$ of EKF, for simplicity, so we won’t present the procedures for EKF again (the procedures are same as the OI). However, the biggest differences between the Kalman filter and the OI is the forecast step (physical step). The Kalman Filter does not assume the background $F$ is stationary statistics, instead, will dynamically propagate the background error covariance in time.

Thus, the optimal estimation of $x_k$ of extended Kalman Filter can be calculated by the following analysis step:

\begin{align}
K_k &= F_k H_k^T (H_k F_k H_k^T + V)^{-1}, \\
a_k &= f_k + K_k(y_k - H_k(f_k)) \\
A_k &= (I - K_k H_k) F_k.
\end{align}

At the beginning, we have assumed the model is not perfect in Equation (3.20) and the relationship of model error, forecast error and observation error are shown clearly in Equation (3.24). Therefore, the forecast step could be updated as

\begin{align}
F_{k+1} &= L_k A_k L_k^T + Q_k, \\
f_{k+1} &= L_{k+1}(a_k) + q_k.
\end{align}

In (3.29), $F_k H_k^T$ is the cross-covariance matrix of the forecast error $(x_k^t - f_k)$ and the forecast measurement error $(y_k - H_k(f_k))$: \( E[(x_k^t - f_k)(y_k - H_k(f_k))^T] \), and $H_k F_k H_k^T + V_k$ is the covariance matrix of the forecast measurement error: \( E[(y_k - H_k(f_k))(y_k - H_k(f_k))^T] \).

The EKF is treated as the “gold standard” in the data assimilation area and is always compared with other new data assimilation methods to evaluate their capabilities. The reasons are in the following.
• Even when the system begins with a poor initial guess of state of the atmospherical phenomena, the performance of EKF will still offer the Best Linear Unbiased Estimation (BLUE) on the state of the atmospherical phenomena with its applicable background error covariance after one or two weeks simulation.

• The dynamic propagation on forecast error covariance matrix guarantee the flow-dependent error is taken in consideration.

Unfortunately, the EKF also has some deficiencies.

• The computational burden on EKF is very large, because the propagation on forecast error covariance needs to do the $L_k A_k L_k^T$ and its operation is $O(n^3)$. Usually, the forecast model of data assimilation has a large number of variables (more than $10^6$). Thus, the common solution is to use the low order model or to update it infrequently.

• If the system is unstable and the observation is insufficient, the Tangent linear operator may become inaccurate making the solution shift.

3.3.2 EnKF: The Classical (Stochastic) Ensemble Kalman Filter with Perturbed Observations

The ensemble Kalman filter is a suboptimal estimator where covariance matrices in (3.32) are estimated by ensembles. The background state is perturbed by $f_k + q^{(e)}_k$ for $e = 1, 2, \ldots, q$ and we denote $q$ ensembles at time step $k$ by $f^{(1)}_k, f^{(2)}_k, \ldots, f^{(q)}_k$, and observations is perturbed by $y_k + v^{(e)}_k$ for $e = 1, 2, \ldots, q$ and we denote $q$ ensembles at time step $k$ by $y^{(1)}_k, y^{(2)}_k, \ldots, y^{(q)}_k$. Now the forecast ensemble set is $\tilde{f}_k = [f^{(1)}_k, f^{(2)}_k, \ldots, f^{(q)}_k]$ and the ensemble mean can be calculated as follows: 

$$f_k = \frac{1}{q} \sum_{e=1}^{q} f^{(e)}_k.$$

The sample matrix of the forecast error and observation error

$$\Phi_k = \frac{1}{\sqrt{q-1}} [f^{(1)}_k - \bar{f}_k, f^{(2)}_k - \bar{f}_k, \ldots, f^{(q)}_k - \bar{f}_k],$$

$$Z_k = \frac{1}{\sqrt{q-1}} [y^{(1)}_k - \bar{y}_k, y^{(2)}_k - \bar{y}_k, \ldots, y^{(q)}_k - \bar{y}_k],$$

are calculated and formed.
In the ensemble Kalman filter [26, 29], the observation is perturbed by \( y_k + v_k^{(e)} \) for \( e = 1, 2, \cdots, q \). where \( v_k^{(e)} \) is a zero-mean random variable with a normal distribution and covariance \( V \). By using \( f_k^{(e)} \), we obtain the analysis estimate of the state:

\[
a_k^{(e)} = f_k^{(e)} + \hat{K}_k( y_k + v_k^{(e)} - H_k(f_k^{(e)})) ,
\] (3.34)

where \( \hat{K}_k \) is the estimate of \( K_k \) by using the ensembles. We note that the cross-covariance matrix of the forecast error and the covariance matrix of the measurement error are estimated by \( \Phi_k \Phi_k^T \) and \( Z_kZ_k^T \) respectively. Now the estimated Kalman gain matrix is computed as follows:

\[
\hat{K}_k = (\Phi_k \Phi_k^T)H_k^T(H_k \Phi_k \Phi_k^T H_k^T + Z_kZ_k^T)^{-1} ,
\] (3.35)

where \( \Phi_k \Phi_k^T H_k^T \) is the cross-covariance matrix of the forecast error \( (\hat{f}_k - \bar{f}_k) \) and the forecast measurement error \( (\bar{y}_k - H_k(\bar{f}_k)) \):

\[
\mathcal{E}[(\bar{f}_k - \hat{f}_k)(\bar{y}_k - H_k(\hat{f}_k))^T] ,
\] (3.36)

and \( H_k \Phi_k \Phi_k^T H_k^T + \hat{V}_k \) is the covariance matrix of the forecast measurement error:

\[
\mathcal{E}[(\bar{y}_k - H_k(\hat{f}_k))(\bar{y}_k - H_k(\hat{f}_k))^T] .
\] (3.37)

In conclusion, the analysis step and forecast step of EnKF are:

**forecast step:**

\[
f_k^{(e)} = L_k(a_{k-1}^{(e)}) + q_k^{(e)} .
\] (3.38)

**analysis step:**

\[
y_k^{(e)} = y_k + v_k^{(e)} ,
\]

\[
\hat{K}_k = \Phi_k \Phi_k^T H_k^T(H_k \Phi_k \Phi_k^T H_k^T + Z_kZ_k^T)^{-1} ,
\] (3.39)

\[
a_k^{(e)} = f_k^{(e)} + \hat{K}_k( y_k^{(e)} - H_k(f_k^{(e)})) ;
\]

It should be stated that when the observations are uncorrelated in the practical case, it could be employed in analysis step in the another way. They would be assimilated serially, i.e., incorporated one by one in a time point. In this situation, the covariance in Equation (3.35) will be calculated for each observation successively until the analysis step of the last observation is finished. During this process, the
previous estimate state will be the forecast state of current iteration. Moreover, when we are dealing with a single observation, observation error covariance $V$ will become a scalar, same as the forecast error covariance $H_k \Phi_k \Phi_k^T H_k^T$. Thus, the matrix inversion will not exist and be involved in the evaluation of the gain matrix.

### 3.3.3 Advantage of EnKF compared to the EKF

1. Since the ensemble Kalman Filter makes use of ensembles to estimate the background covariances matrix, there is no need to do the propagation from estimation error covariance to the forecast error covariance. The computational workload can be reduced significantly compared with the extended Kalman Filter method.

2. The EnKF only needs to store $s$ forecast state vectors and the generation of ensemble background covariance by several samples instead of large number of state vectors in EKF, and it can save a lot of storage and operations. Through, the inversion of the innovation error covariance matrix is still required to compute the Kalman gain which is the same as EKF. But the dimension of this matrix $m \times m$ is not a big matrix in real case. Moreover, some methods are developed based on the EnKF to reduce the dimension of the innovation error covariance for easy computing just like the localization technique, divide and conquer strategy, parallel computational technique and so on.

3. A good perturbation of model error distribution $q_k$ may improve the performance of estimation a lot.

### 3.3.4 The Comparison between 4D-Var and EnKF

#### Advantages of EnKF

- The EnKF is described in matrix form which makes it easier to implement.

- No need for the adjoint model which causes an abundance of computation in 4D-Var.
• The EnKF is constructed by several matrices, and they can be improved separately by using numerical and statistic methods. Also, the computation of algorithm can be reduced by the classical numerical methods.

• The EnKF is using the ensemble to assimilate the data as some processes are not related with each other. It is a potential design to compute the method by parallel approach.

• Applying the localization in EnKF can weaken the influence of long-distance problem in samples and increase the capacity to adapt more observations.

• The blurry samples in EnKF may leads to the right or close model state with respect to the truth.

Advantages of 4D-Var

• 4D-Var can easily assimilate different sources of observations.

• 4D-Var has the ability to handle the case of the observation and model with non-Gaussian errors.

• 4D-Var gives the accurate covariance of model in time interval and follows the better tendency since it needs the propagation in each iteration.

• Tunning the weight between the least square terms, 4D-Var has the option to acquire the optimal global solution.

• If the forecast model is linear, there exist many classical iteration method to be employed. That makes the computation very efficient and the research on numerical methods will bring the speed to 4D-Var as well.

3.3.5 Covariance Inflation of EnKF

EnKF covariance is generated from the limited number of ensembles compared to the large number of samples in EKF. Thus there is no doubt that the matrix is rank deficient which means the number of linearly independent vectors are smaller than the order of the matrix. Up to now, there have been two mainstream methods to fix this problem in the analysis process.
On the one hand, covariance inflation is one method on correcting the underestimation problem in the forecast error covariance matrix. In paper [2], Anderson and Anderson introduce the inflation technique on each ensemble of forecast error covariance and the deviation is designed by percentage. When the assimilation process enter into the analysis step, the deviation of the forecast with respect to the mean of forecast will be artificially increased by a parameter, known as inflation factor $\alpha$:

$$\hat{f}_k^{(e)} = \alpha(f_k^{(e)} - \bar{f}_k) + \bar{f}_k,$$

then the $\hat{f}_k^{(e)}$ will replace the original forecast ensembles in Equation (3.35) to continue the analysis process.

Learning from the research paper, the $\alpha$ is normally chosen slightly bigger than one. However, the specific optimal number of inflation factor is dependent on the actual situation and the size of ensemble [48]. From the experience, Hamill [48] tested that the 1% increase in inflation factor will get the optimal solution according to the performance of estimation. Meanwhile, Whitaker and Hamill [80] concluded that the best inflation factors is 7% for EnKF and 3% for EnSRF. So, choosing a suitable inflation factor relies on varied elements and needs to be researched further. For instance, the inflation factor may depend on the dynamical forecast model, which is the type of Kalman filter related method or the distance matrix mentioned in a later section. This method is commonly used in covering the problem of inbreeding, like in [80, 48, 3, 65] and may lead to the damage on the balance of forecast system in [3]. The inflation factors actually have still not solved the long-distance problem about the spurious correlation in the forecast error covariance, and so, this problem will be handled in the next section.

3.3.6 Localization of EnKF (LEnKF)

On the other hand, the ensemble Kalman filter may suffer from spurious correlations caused by sampling errors in the estimation of the covariance matrices. Several methods have been proposed to improve the performance of the ensemble Kalman filter. One way is to impose the localization in the ensemble Kalman filter by using the decorrelation function with local support [34]. The decorrelation function can be
chosen such that its value is monotonically decreasing with respect to the distance from the origin [44, 40]. In [10], Adrian et al. had considered to use the following matrix $D$ for this decorrelation purpose. The $(i, j)$-th entry of $D$ is given by

$$[D]_{i,j} = \exp \left( - \frac{d_{i,j}}{\gamma} \right)^2,$$

(3.41)

where $d_{i,j}$ refers to the distance between the $i$-th and $j$-th variables of the state vector, and $\gamma$ is a positive parameter to control the influence of the decorrelation. For instance, in the air quality forecast system, when the $i$-th variable refers to the value of the $i$-th grid point, the distance between the $i$-th and $j$-th variables corresponds to the physical distance between these two grid points. Instead of using (3.35), the estimation of the Kalman gain matrix is given by

$$\hat{K}_k := (D \circ \Phi_k \Phi_k^T) H_k^T \left( H_k (D \circ \Phi_k \Phi_k^T) H_k^T + Z_k Z_k^T \right)^{-1},$$

(3.42)

where $\circ$ refers to the entry-wise multiplication of two matrices. We remark that the Kalman gain matrix in (3.42) employs model space localization.

On the other hand, Houtekamer and Mitchell [44] studied observation space localization. The Kalman gain matrix is given by

$$\hat{K}_k := (D_1 \circ [(\Phi_k \Phi_k^T) H_k^T]) \left( D_2 \circ [H_k (\Phi_k \Phi_k^T) H_k^T] + Z_k Z_k^T \right)^{-1},$$

(3.43)

where $D_1$ refers to the decorrelations defined as (3.41) between the model grid points and the observations with the physical distances and $D_2$ considers the decorrelations for the observed data. The advantage of using (3.43) is that the computational cost of computing the analysis can be significantly reduced as $D_2$ is operated on the observation covariance matrix $H_k (\Phi_k \Phi_k^T) H_k^T$. It is obvious that (3.42) and (3.43) are different models [20]. In general, numerical examples have shown that the prediction performance of using (3.42) is better than that of using (3.43), see [20, 45, 59]. In [59], it is found that the prediction performance can depend on the degree of used decorrelation. For example, when the degree of decorrelation (localization) is large, the performance of using (3.42) can be better than that of using (3.43).
3.3.7 Characteristics and Advantages of LEnKF compared to the EnKF

Compared to the ensemble Kalman filter, the LEnKF has some extra characteristics and advantages.

- In the local ensemble Kalman filter, we update the globally ensembles vectors by using them locally. The ensembles are responsible for providing the errors distribution within the local cubes.

- This localization effect largely reduces the number of ensembles for the need of ensemble Kalman filter and enlarges the rank of background covariance matrix, at most $m$.

- It allows the observation in the cube to be used simultaneously and data assimilation can computes the analysis in the cube independently.

- It prevents the long-distance correction effect that existed in the EnKF. This corrections is due to the subsampling problem (not physical or statistical problems), especially for the distant grid points.

3.3.8 EnSRF: Ensemble Square-root Kalman Filter

The classical ensemble Kalman Filter and localized ensemble Kalman Filter which will be mentioned later, their observational ensembles are perturbed by some normal distribution. However, there also exist a class of ensemble Kalman Filter related algorithms don’t include the perturbations in the observational ensembles. The reason of the existence is that no perturbation on observations means no sampling errors are brought in by the addition of perturbations of observation.

These algorithms include the ensemble square-root Kalman Filter (EnSRF) raised by Whitaker and Hamill [80], ensemble adjustment filter (EAKF) proposed by Anderson [75] and the transform Kalman Filter (ETKF) presented by Bishop [4]. They are all summarized by Tippett [75].

This method is only valid on analyzing the uncorrelated observations and incorporating them with analysis step one by one, so it is a serial method.
forecast step:

\[ f_k^{(e)} = L_k(a_{k-1}^{(e)}) + q_k^{(e)} \] ; \hspace{1cm} (3.44)

analysis step

\[ \bar{a}_k = \bar{f}_k + \bar{K}_k[y_k - H_k(\bar{f}_k)] , \]
\[ \delta a_k^{(e)} = \alpha(I - \beta K_k H_k) \delta f_k^{(e)} , \]
\[ \hat{K}_k = \Phi_k \Phi_k^T H_k^T (H_k \Phi_k \Phi_k^T H_k^T + \hat{V})^{-1} \] ;

where \( \alpha \) is the inflation factor and will be discussed later; \( \delta a_k^{(i)} \) and \( \delta f_k^{(i)} \) are the deviations compared to the their own means for each estimation ensemble and forecast ensemble, respectively. And the parameter \( \beta \) is calculated as

\[ \beta = \left[ 1 + \sqrt{\hat{V}(H_k \Phi_k \Phi_k^T H_k^T + \hat{V})^{-1}} \right]^{-1} . \] (3.46)

This equation for the \( \beta \) is also just valid for the analysis step of single observation fitted for the assumption of algorithm. In this situation, both the numerator and denominator inside the square root will be the scalars and its evaluation of \( \beta \) will be easy.

And for the background forecast covariance \( \Phi_k \Phi_k^T H_k^T \) and \( H_k \Phi_k \Phi_k^T H_k^T \) in EnSRF, they are very similar to the Equation (3.36) and Equation (3.37), and can replace the \( \bar{y}_k \) with \( y_k \) of this algorithm. Since the algorithm contains one single observation in a iteration, \( F_k H_k^T \) will be a vector and \( H_k F_k H_k^T \) is a scalar. Same as the EnKF in the uncorrelated observation case, after the analysis step for one observation is finished, the estimation will become the new forecast state for the next one observation case and the analysis step will be soon be repeated again. After all the observations at the given time are analyzed, the new forecast ensembles will be proceeded from these estimation ensembles and at the same time the analysis cycle will be repeated with the new observation(s), .
Chapter 4

Sparse Matrix Computation

4.1 Motivation

Data assimilation is a very popular concept in estimating the state of big data by using limited observations from varied sources. Thus, it is frequently used in engineering big data problems, for instance, atmospheric chemistry, oceanic biochemistry, glaciology, seismology, oil industry, nuclear fusion, medicine, agronomy, etc. And the applications of the data assimilation to these problems show very good performances on the improvements of accuracy of estimation which could be treated as the optimal initial condition for the next physical cycle or the reliable future state.

In Chapter 2, we mentioned the CMAQ Chemical Transport Model (CCTM, one kind of atmospheric chemistry model) which is commonly used in simulating the air quality by many academic and government agencies. It is developed to the 3rd generation in recent years, but still doesn’t reach the level of perfect model. Thus the simulation process of this model still has many unknown noises and they have led to inaccurate forecast results. Because of the inaccurate forecast results, we have begun to worry about our life quality and health. That has pushed us to employ some methods to largely increase the precision of prediction in order to protect ourselves. In this thesis, we have used and developed the data assimilation method to improve the accuracy of model with existing observations. Besides, we are handling the big data simulated by the chemical transport model, so we won’t study deeper into its scientific components which are constructed by physical and chemical processes.
As mentioned before, the data assimilation methods are indeed the mathematical algorithms. It is developed for two mainstream mathematical methods: the Kalman filter method based on the estimation theory and variational method deriving from the optimization theory. In this chapter, we will introduce our proposed method which is based on the Kalman filter method.

4.2 Proposed Method

4.2.1 Sparse Matrix Computation

Learning from the classical equation (3.42) of LEnKF in Section 3.3.6, we note that the most expensive computational part is to build the ensemble Kalman gain matrix \( \hat{K}_k \). For simplicity, we ignore the subscript index \( k \) in the following discussion. In the calculation, it involves two \( n \)-by-\( n \) matrices \( D \) and \( \Phi \Phi^T \) where \( n \) is the number of variables in the state vector. In practical applications, \( n \) could be very large. It is not suitable to store these two matrices in the implementation. In matrix-vector multiplication \( H(D \circ \Phi \Phi^T) \), we need to generate the column of \( D \circ \Phi \Phi^T \) one by one, and perform the multiplication of the row of \( H \). The total cost is of \( O((m + q)n^2) \).

When the number of observations and the number of ensembles are small compared with \( n \), the main cost of ensemble Kalman filter per iteration is of \( O((m+q)n^2) \) which is very expensive when the number \( n \) of variables is very large.

Now let us discuss a case that the number of arithmetic operations can be reduced by considering the structure of the ensemble Kalman gain matrix. We consider \( H \) is very sparse, i.e., there are a lot of zeros in the matrix. For example, we investigate the air quality forecast system in Hong Kong, there are \( n = 142376 \) variables corresponding to the grid points for calculating the concentration of species in a three-dimensional space. However, there are about \( m = 112 \) observations for these variables. It implies that there are a lot of zeros in \( H \), and the number of arithmetic operations can be significantly reduced.

Suppose there are at most \( r \) non-zeros in each row of \( H \), the number of arithmetic operations required to generate \( H(D \circ \Phi \Phi^T)H^T \) is of \( O(qr^2m^2) \) which is independent of \( n \). The cost is summarized in the following steps.

54
1. Let the positions of non-zero entries of the \(i\)-th row of \(H\) be \(p_{i,1}, p_{i,2}, \ldots, p_{i,r}\). Correspondingly, the positions of non-zero entries of the \(j\)-th column of \(H^T\) are \(p_{j,1}, p_{j,2}, \ldots, p_{j,r}\).

2. In order to compute the \((i, j)\)-th entry of \(H(D \circ \Phi \Phi^T)H^T\), we need to generate \(i\)-th row of \(H(D \circ \Phi \Phi^T)\) and use \(j\)-th column of \(H^T\) for multiplication, i.e., we only need to generate the \(p_{j,1}, p_{j,2}, \ldots, p_{j,r}\)-th entries of the \(i\)-th row of \(H(D \circ \Phi \Phi^T)\).

3. In order to generate the \(p_{j,1}, p_{j,2}, \ldots, p_{j,r}\)-th entries of the \(i\)-th row of \(H(D \circ \Phi \Phi^T)\), we multiply the \(i\)-th row of \(H\) and the \(p_{j,1}, p_{j,2}, \ldots, p_{j,r}\)-th columns of \(D \circ \Phi \Phi^T\). Note that there are only \(r\) non-zero entries at the \(i\)-th row of \(H\). Therefore, we only need to generate the \(p_{i,1}, p_{i,2}, \ldots, p_{i,r}\)-th entries at the \(p_{j,1}, p_{j,2}, \ldots, p_{j,r}\)-th columns of \(D \circ \Phi \Phi^T\).

4. According to (3.41), it requires \(O(r^2)\) arithmetic operations to compute the \(p_{i,1}, p_{i,2}, \ldots, p_{i,r}\)-th entries at the \(p_{j,1}, p_{j,2}, \ldots, p_{j,r}\)-th columns of \(D\).

5. Since \(\Phi\) is an \(n\)-by-\(q\) matrix, it requires \(O(qr^2)\) arithmetic operations to compute the \(p_{i,1}, p_{i,2}, \ldots, p_{i,r}\)-th entries at the \(p_{j,1}, p_{j,2}, \ldots, p_{j,r}\)-th columns of \(SS^T\).

6. The cost of calculating the \((i, j)\)-th entry of \(H(D \circ \Phi \Phi^T)H^T\) is \(O(qr^2)\) arithmetic operations. Therefore the complexity of computing the \(m\)-by-\(m\) matrix \(H(D \circ \Phi \Phi^T)H^T + \frac{1}{\alpha}V\) is \(O(qr^2m^2)\).

By using the similar method and argument, \((D \circ \Phi \Phi^T)H^T\) can be obtained in \(O(qrmn)\) arithmetic operations. The inverse of \(H(D \circ \Phi \Phi^T)H^T + \frac{1}{\alpha}V\) can be computed in \(O(m^3)\) arithmetic operations. According to (3.34) and (3.42), the resulting vector 
\[
(D \circ \Phi \Phi^T)H^T \left( H(D \circ \Phi \Phi^T)H^T + \frac{1}{\alpha}V \right)^{-1} (y + v^{(e)} - Hf^{(e)})
\]
can be computed in \(O(qrm(rm+n))\) arithmetic operations. As a summary, by making use of the sparsity of \(H\) (\(r \ll n\)), and the number \(m\) of observations are significantly smaller than the number \(n\) of variables (\(m \ll n\)), the cost of performing the ensemble Kalman filter in (3.34) is \(O(qrmn)\) arithmetic operations per ensemble.

\footnote{Without loss of generality, we assume that each row has the \(r\) non-zeros entries.}
4.2.2 Multiple Species

In the air quality forecast system, the concentrations of different species are predicted via the chemistry-transport model. In the pollutants recording system, sensors detect multiple species and accumulate them together to make the observations at different time steps. The main contribution of this paper is to propose a block matrix model to manage ensemble Kalman filter for these accumulated multiple species together.

Let \( c^{(1)}, c^{(2)}, \ldots, c^{(c)} \) be the variables of \( c \) multiple species to be considered. Let \( y^{(1)}, y^{(2)}, \ldots, y^{(o)} \) be the observations of these species. Similar to Equation (3.39), we have the following block observation equations:

\[
\begin{bmatrix}
  y^{(1)} \\
  y^{(2)} \\
  \vdots \\
  y^{(o)}
\end{bmatrix} =
\begin{bmatrix}
  c^{(1)} \\
  c^{(2)} \\
  \vdots \\
  c^{(c)}
\end{bmatrix} =
\begin{bmatrix}
  H^{(1,1)} & H^{(1,2)} & \cdots & H^{(1,c)} \\
  H^{(2,1)} & H^{(2,2)} & \cdots & H^{(2,c)} \\
  \vdots & \vdots & \ddots & \vdots \\
  H^{(o,1)} & H^{(o,2)} & \cdots & H^{(o,c)}
\end{bmatrix}
\begin{bmatrix}
  c^{(1)} \\
  c^{(2)} \\
  \vdots \\
  c^{(c)}
\end{bmatrix},
\]

where \( H^{(i,j)} \) refers to the observation matrix for the \( j \)-th species to the \( i \)-th observation. It is necessary all \( H^{(i,j)} \) to be non-zero. When \( H^{(i,j)} \) is a zero matrix, it means the \( j \)-th species is not counted in the \( i \)-th observation. We remark that for simplicity, we remove the time index \( k \) in the discussion. For instance, in the air quality forecast system, there are two key observations: PM2.5 and PM10, i.e., \( o = 2 \). There are 40 multiple species counted in these two observations, i.e., \( c = 40 \). Here PM2.5 (whose diameter is smaller than 2.5\( \mu \)m) contains 33 species and PM10 (whose diameter is smaller than 10\( \mu \)m) contains all 40 species.

To work for ensemble Kalman filter for multiple species, we propose to employ the block decorrelation matrix:

\[
D = \begin{bmatrix}
  D^{(1)} & 0 & \cdots & 0 \\
  0 & D^{(2)} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & D^{(c)}
\end{bmatrix},
\]

where \( D^{(i)} \) is given in the form of (3.41). In the block matrix model, the analysis
estimate of multiple species for the $e$-th ensemble is given by

$$
\begin{bmatrix}
a^{(1,e)} \\
a^{(2,e)} \\
\vdots \\
a^{(c,e)}
\end{bmatrix} =
\begin{bmatrix}
f^{(1,e)} \\
f^{(2,e)} \\
\vdots \\
f^{(c,e)}
\end{bmatrix}
+ 
\hat{K}
\begin{bmatrix}
y^{(1)} + u^{(1,e)} \\
y^{(2)} + u^{(2,e)} \\
\vdots \\
y^{(o)} + u^{(o,e)}
\end{bmatrix}
- 
\begin{bmatrix}
H^{(1,1)} & H^{(1,2)} & \cdots & H^{(1,c)} \\
H^{(2,1)} & H^{(2,2)} & \cdots & H^{(2,c)} \\
\vdots & \vdots & \ddots & \vdots \\
H^{(o,1)} & H^{(o,2)} & \cdots & H^{(o,c)}
\end{bmatrix}
\begin{bmatrix}
f^{(1,e)} \\
f^{(2,e)} \\
\vdots \\
f^{(c,e)}
\end{bmatrix},
$$

(4.2)

where $f^{(j,e)}$ and $a^{(j,e)}$ are the forecast and analysis of the $j$-th species state $x^{(j)}$ for the $e$-th ensemble respectively, and $u^{(i,e)}$ is a noise to the $i$-th observation for the $e$-th ensemble (it is generated by a zero-mean random variable with a normal distribution and covariance $V^{(l)}$ is corresponding to the $l$-th observation). Here $\hat{K}$ is the ensemble Kalman filter matrix:

$$
\hat{K} = (D \circ \Phi \Phi^T)H^T(H(D \circ \Phi \Phi^T)H^T + \frac{1}{\alpha}V)^{-1},
$$

(4.3)

where

$$
\Phi = 
\begin{bmatrix}
\Phi^{(1)} \\
\Phi^{(2)} \\
\vdots \\
\Phi^{(c)}
\end{bmatrix}, \quad \Phi^{(i)} := \frac{1}{\sqrt{q-1}}[f^{(i,1)} - \bar{f}^{(i)}, f^{(i,2)} - \bar{f}^{(i)}, \ldots, f^{(i,q)} - \bar{f}^{(i)}],
$$

and

$$
V = 
\begin{bmatrix}
V^{(1)} & 0 & \cdots & 0 \\
0 & V^{(2)} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & V^{(o)}
\end{bmatrix}.
$$

It is interesting to note that $\Phi \Phi^T$ are the sample cross-covariance matrix among $c$ species. By using such statistical information, the Kalman filter equation can generate the state estimation of $c$ species via their accumulated observations.
Now the size of the matrix $H(D \circ \Phi \Phi^T)H^T$ is $om$-by-$om$. By using the sparsity of each $H^{(i,j)}$ and the same computational procedure in Section 2, the cost of calculating the $om$-by-$om$ matrix $H(D \circ \Phi \Phi^T)H^T + \frac{1}{a}V$ is of $(q^2r^2o^2m^2)$ which is independent of $n$. Also its inverse can be computed in $O(o^2m^3)$ arithmetic operations. Similarly, $(D \circ \Phi \Phi^T)H^T$ can be obtained in $O(qc^2romn)$ arithmetic operations. The cost of performing the ensemble Kalman filter in (4.2) is $O(qc^2rom(n + m))$ arithmetic operations. We remark that $c, o, m, q$ and $r$ are very small compared with the number of variables in the air quality forecast system. Therefore, the proposed method reduces the cost from $O(c^2omn^2)$ to $O(qc^2romn)$ per ensemble so that it’s very efficient for large-scale computation.

4.3 Implementation

4.3.1 Overview of the Connection between CCTM and DA

In the experiment, we have obtained Hong Kong’s actual data of air quality in 2010 from the HKEPD (initial condition at 2010-01-01-0:00, the hourly boundary condition of the year 2010 of 1 size, hourly emission file of the year 2010, hourly MCIP file of year 2010 and grid file of 4 sizes). The raw data covers the $1971km \times 1971km \times 17.408km$ space and use $27km$ as the horizontal grid size, sigma level strategy as the vertical grid size. In other words, the space is modeled in a $74 \times 74 \times 26$ mesh called Domain 1. We concentrate on the situation in Hong Kong, and as this area is coarse, so we have used these files of $27km$ with $9km$ grid file as input to propagate through the CCTM and to get the concentration data of $9km$. That covers the $666km \times 666km \times 17.408km$ space with $9km$ horizontal grid size (same vertical grid size as before) called Domain 2. Again, putting these files of $9km$ with $3km$ grid files into the CCTM, we figure out the concentration of $3km$. That covers the $219km \times 219km \times 17.408km$ space with $3km$ horizontal grid size (same vertical grid size as before) called Domain 3. In the end, we input these files of $3km$ with $1km$ grid files to the CCTM and output the concentration of $1km$. It contains $1km$ fine grid size and covers the $73km \times 73km \times 17.408km$ space above the Hong Kong area called Domain 4. Now, since the focus is on improving the air quality prediction of Hong Kong, of $1km$ resolution, and the range is enough
to do the simulation. Finally, we acquire the 12 initial condition files corresponding to the 1st day of 12 months and 365 hourly boundary condition files of Domain 4. 365 meteorology file sets, 365 emission files and grid file are prepared beforehand for Domain 4. Further, in the future, we will experiment on Domain 3, then Domain 2 and later even on Domain 1 to get better subarea information so that the concentration will be closer to the truth.

![Figure 4.1: Comparison of domain size](image)

Here we use information of Domain 4 area and follow the chemical transport model formulation of the CCTM,

\[
\frac{\partial (c^*)}{\partial t} + \frac{\partial (c^*\hat{v})}{\partial t} \bigg|_{\text{adv}} + \frac{\partial (c^*\hat{v}')}{\partial t} \bigg|_{\text{diff}} = \frac{\partial (c^*)}{\partial t} \bigg|_{\text{chem}} + \sqrt{\tau} Q_c + \frac{\partial (c^*)}{\partial t} \bigg|_{\text{chl}} + \frac{\partial (c^*)}{\partial t} \bigg|_{\text{aero}} + \frac{\partial (c^*)}{\partial t} \bigg|_{\text{ping}}.
\]

(4.4)

It is decided to take the internal concentration (initial condition) as \(c^*_0\), the meteorology file as \(\hat{v}_0\) and \(\hat{v}'_0\), and the emission file as \(Q_{c_0}\). Boundary condition is involved in the advection process while photolysis data affects the chemical reaction process. Adding the other additional files into this formulation, we get the predicted internal concentration result \(\hat{f}_1\), and put it into our proposed data assimilation. After being analyzed and processed (this progress is the ”analysis step” part of data assimilation), the first predicted internal concentration \(\hat{a}_1\) is figured out. We treat the result \(\hat{a}_1\) as
the input data of the next iteration. In the next iteration, first, we use the ICON component to transform the output $\dot{a}_1$ from DA to the input form of the CCTM. Second, in a similar way, propagating this $\dot{a}_1$ through CCTM to get the predicted result $\dot{f}_2$. Third, we put it into DA again to get the predicted result $\dot{a}_2$. By this analogy, the iteration will continue until reaching the time or the specified number of iteration that we have set (see Figure 4.2). Since we were able to get Hong Kong’s actual observational data of air quality in 2010, we are able to tell the difference between the result we get and the actual observational data from HKEPD. Also, we could evaluate the improvements $\{\dot{a}_1, \dot{a}_2, \cdots, \dot{a}_n\}$ (red line of Figure 4.2) compared with the forecast values $\{f_1, f_2, \cdots, f_n\}$ from the CMAQ (blue line of Figure 4.2). Furthermore, comparing the efficiency of our proposed data assimilation method with respect to the original data assimilation method is another numerical result and will be presented in a later section.

Figure 4.2: CMAQ system with data assimilation (updating internal concentration)

Moreover, since the forecast result $\dot{f}_k$ contains the same range of area as the initial condition, we could combine this $\dot{f}_k$ with the corresponding boundary condition
\( \hat{f}_k^n \) as the input of DA. Then the DA could use this combined background state with observations to update them simultaneously (see Figure 4.3) and they become \( \hat{a}_k \) and \( \hat{a}_k^n \). On the one hand, this could give the area the continuous change on the performance no matter it is internal or on the boundary. It makes them more consistent with each other when they become the input of the next iteration. On the other hand, the DA also gives the boundary condition an update while the state is approaching the observations.

Figure 4.3: CMAQ system with data assimilation (updating internal and boundary concentrations)

### 4.3.2 Configurations on the CMAQ System to meet with Data Assimilation

In order to employ the data assimilation method in CMAQ system, we need to do some modifications in the configure files and the running codes of ICON and CCTM processes, respectively. Before describing the details of the modifications, it is necessary to emphasize the working environment first. All the computations
are performed on a desktop computer with Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz with 28 processors and 256GB Memory. The system is CentOS v6.5, and the version of CMAQ is 5.0.1

The following is the updates on the configure files and running codes of ICON and CCTM processes. They are prepared to adapt to our real data and proposed data assimilation method.

**ICON configure file:**

- Updating the line of index of the configure folder from the setting of benchmark case to the setting of our real case.
  
  ```
  set APPL = V5g ⇒ set APPL = hk
  ```

- Updating the line of ModType from using the profile to using the previous concentration file.
  
  ```
  set ModType = (module profile $Revision;)
  ⇒
  set ModType = (module m3conc $Revision;)
  ```

- Updating the line of mechanism from the setting of benchmark case to the setting of our real case.
  
  ```
  set Mechanism = cb05tucl_ae6_aq
  ⇒
  set Mechanism = cb05cl_ae5_aq
  ```

**ICON running code:**

- Updating the line of index of configure folder from the setting of benchmark case to the setting of our real case.
  
  ```
  set APPL = V5g ⇒ set APPL = hk
  ```

- Updating the line of type of mechanism from the setting of benchmark case to the setting of our real case.
  
  ```
  set MECH = cb05tucl_ae6_aq
  ⇒
  set MECH = cb05cl_ae5_aq
  ```

- Updating the lines of position and name of grid file from the setting of benchmark case to the setting of our real case.

62
setenv GRIDDESC = $M3HOME/scripts/GRIDDESC
⇒ setenv GRIDDESC $M3DATA/EPD/GRIDDESC
setenv GRID_NAME CMAQ-BENCHMARK ⇒ setenv GRID_NAME HKPATH_1KM

- Updating the line of position and name of layer file from the setting of benchmark case to the setting of our real case.
  setenv LAYER_FILE $M3DATA/mcip/METCRO3D.Benchmark
  ⇒ setenv LAYER_FILE $M3DATA/EPD/mcip/METCRO3D.$MONTH$DAY

- Updating the line of position and name of output folder from the setting of benchmark case to the setting of our real case.
  set OUTDIR = $M3DATA/icon ⇒ set OUTDIR = $M3DATA/EPD/icon

- Updating the line of form of ICON input file from profile to CTM concentration file.
  set IC = profile ⇒ set IC = m3conc

- Updating the lines of date and name of ICON output file.
  set DATE = 2006213 ⇒ set DATE = 2010$DATE
  setenv INIT_CONC_1 "$OUTDIR/ICON$APPL$CFG$DATE -v" ⇒ setenv INIT_CONC_1 "$M3DATA/EPD/icon/ICON.$ens -v"

- Updating the line of position and name of ICON input file (the output from CCTM).
  setenv CTM_CONC_1 $M3DATA/cctm/CCTM.d1bCONC.d1b
  ⇒ setenv CTM_CONC_1 $M3DATA/EPD/cctm/CCTM.$ens

- Updating the line of time of ICON output file.
  setenv STIME 000000 ⇒ setenv STIME $TIME0000

CCTM configure file:

- Updating the line of index of configure folder from the setting of benchmark case to the setting of our real case.
  set APPL = V5g ⇒ set APPL = hk
• Updating the line of model of chemistry solver from the setting of benchmark case to the setting of our real case.

\[
\text{set ModChem = (module ebi\_cb05tucl $Revision;) } \Rightarrow \text{set ModChem = (module ebi\_cb05cl $Revision;)}
\]

• Updating the line of model of the aerosol process from the setting of the benchmark case to the setting of our real case.

\[
\text{set ModAero = (module aero6 $Revision;) } \Rightarrow \text{set ModAero = (module aero5 $Revision;)}
\]

• Updating the line of model of cloud from the setting of the benchmark case to the setting of our real case.

\[
\text{set ModCloud = (module cloud\_acm\_ae6 $Revision;) } \Rightarrow \text{set ModCloud = (module cloud\_acm\_ae5 $Revision;)}
\]

• Updating the line of type of mechanism from the setting of the benchmark case to the setting of our real case.

\[
\text{set Mechanism = cb05tucl\_ae6\_aq } \Rightarrow \text{set Mechanism = cb05cl\_ae5\_aq}
\]

CCTM running code:

• Updating the line of index of configure folder from the setting of the benchmark case to the setting of our real case.

\[
\text{set APPL = V5g } \Rightarrow \text{set APPL = hk}
\]

• Updating the line of index of mechanism from the setting of the benchmark case to the setting of our real case.

\[
\text{set MECH = cb05tucl\_ae6\_aq } \Rightarrow \text{set MECH = cb05cl\_ae5\_aq}
\]

• Updating the lines of starting date, starting time, step size, year, month and day from the setting of the benchmark case to the setting of our real case.

\[
\text{set STDATE = 2006213 } \Rightarrow \text{set STDATE = 2010$DATE$}\]
\[
\text{set STTIME = 000000 } \Rightarrow \text{set STTIME = $TIME0000$}\]
\[
\text{set NSTEPS = 240000 } \Rightarrow \text{set NSTEPS = 010000}\]
\[
\text{set YEAR = 2006 } \Rightarrow \text{set YEAR = 2010}\]
set YR = 06 \Rightarrow \text{set YR} = 10

set MONTH = 08 \Rightarrow \text{set MONTH} = \$\text{MONTH}

set DAY = 01 \Rightarrow \text{set DAY} = \$\text{DAY}

- Updating the lines of position and name of grid file from the setting of the benchmark case to the setting of our real case.

\text{setenv GRIDDESC} = \$\text{M3HOME/scripts/GRIDDESC}

\Rightarrow \text{setenv GRIDDESC} \ $\text{M3DATA/EPD/GRIDDESC}$

\text{setenv GRID\_NAME CMAQ\_BENCHMARK} \Rightarrow \text{setenv GRID\_NAME HKPATH\_1KM}

- Updating the lines of species and layer range for integral average concentration from the setting of the benchmark case to the setting of our real case.

\text{setenv AVG\_CONC\_SPCS "O3 NO CO NO2 ASO4I ASO4J NH3"}

\Rightarrow \text{setenv AVG\_CONC\_SPCS "ALL"}

\text{setenv ACONC\_BLEV\_ELEV "1 1"} \Rightarrow \text{setenv ACONC\_BLEV\_ELEV "1 26"}

- Updating the line of windblown dust to a non-existent state.

\text{setenv CTM\_WB\_DUST Y} \Rightarrow \text{setenv CTM\_WB\_DUST N}

- Updating the line of agricultural activity for windblown dust to a non-existent state.

\text{setenv CTM\_ERODE\_AGLAND Y} \Rightarrow \text{setenv CTM\_ERODE\_AGLAND N}

- Updating the line of windblown dust emissions diagnostic file to a non-existent state.

\text{setenv CTM\_DUSTEM\_DIAG Y} \Rightarrow \text{setenv CTM\_DUSTEM\_DIAG N}

- Updating the line of lightning NOx file to a non-existent state.

\text{setenv CTM\_LTNG\_NO Y} \Rightarrow \text{setenv CTM\_LTNG\_NO N}

- Updating the line to neglect the surface HONO interaction.

\text{setenv CTM\_SFC\_HONO Y} \Rightarrow \text{setenv CTM\_SFC\_HONO N}

- Updating the line of in-line biogenic emissions file to a non-existent state.

\text{setenv CTM\_BIOGEMIS Y} \Rightarrow \text{setenv CTM\_BIOGEMIS N}
• Updating the line of in-line plume rise emissions file to a non-existent state.

```
setenv CTM_PT3DEMIS Y ⇒ setenv CTM_PT3DEMIS N
```

• Updating the line of position and name of output folder from the setting of the benchmark case to the setting of our real case.

```
set OUTDIR = $M3DATA/cctm ⇒ set OUTDIR = $M3DATA/EPD/cctm
```

• Updating the line of position and name of input ocean file from the setting of the benchmark case to the setting of our real case.

```
setenv OCEAN_1 $M3DATA/ocean/CMAQ-BENCHMARK_surf.ncf ⇒ setenv OCEAN_1 $M3DATA/EPD/ocean/OCEANFILE_HK1km
```

• Updating the line of name of output file from the setting of the benchmark case to the setting of our real case.

```
set CONCfile = $EXEC.CONC.${CTM_APPL} ⇒ set CONCfile = CCTM.$ens
```

• Updating the lines of position and name of emission input file from the setting of the benchmark case to the setting of our real case.

```
set EMISpath = $M3DATA/emis ⇒ set EMISpath = $M3DATA/EPD/emis
set EMISfile = e3d.${EMISDATE}.12EUS1_279X240.cb05soa.24.ncf ⇒ set EMISfile = EMISS.$DATE.ncf
```

• Updating the lines of position and name of initial condition, boundary condition, meteorology information and photolysis rate input files from the setting of the benchmark case to the setting of our real case.

```
set ICpath = $M3DATA/icon ⇒ set ICpath = $M3DATA/EPD/icon
set ICFILE = ICON.${APPL}.${CFG}.profile ⇒ set ICFILE = ICON.$ens
set BCpath = $M3DATA/bcon ⇒ set BCpath = $M3DATA/EPD/bcon
set BCFILE = BCON.${APPL}.${CFG}.profile ⇒ set BCFILE = BCON.$ens
set EXTN = Benchmark ⇒ set EXTN = $MONTH$DAY
set METpath = $M3DATA/mcip ⇒ set METpath = $M3DATA/EPD/mcip
set JVALpath = $M3DATA/jproc ⇒ set JVALpath = $M3DATA/EPD/jproc
```
Automatic workflow:

After completing these preparations, the next step is to link the CMAQ system with our proposed data assimilation. Therefore, we have developed a workflow to link each part within the CMAQ system together and simulates them with data assimilation method sequentially. The simulation will keep running hour by hour for the whole year until it reaches the stop criterion which we have set beforehand. Here I will describe the workflow as a paragraph of pseudo code.
Algorithm 1 Pseudo code of workflow (Part I)

1: Source the environmental variables file of Intel Compiler and CMAQ system
2: for MONTH do ∈ 1 ～ 12
3: Prepare the concentration file ”CCTM.2010$\{MONTH\}01” which contains the information of 24 hours in the first day of the month
4: for ens do ∈ 1 ～ Num_of_Ens
5: cp ${M3DATA}/EPD/icon/ICON.2010$\{MONTH\}0100 $\{current\}$/icon/ICON.$\{ens\}$
6: end for
7: Initialize the initial conditions to be the ensembles (MATLAB code) with the techniques in Section 4.3.4
8: for DAY do ∈ 1 ～ 1
9: for TIME do ∈ 1 ～ 1
10: Replace the lines of CCTM running code in Section 4.3.2
11: for ens do ∈ 1 ～ Num_of_ens
12: Replace the line of CCTM running code about ”CONCfile” with ”CCTM.$\{ens\}”
13: Replace the line of CCTM running code about ”ICFILE” with ”ICON.$\{ens\}”
14: Execute the CCTM running code
15: end for
16: Run the analysis step code of proposed data assimilation method (MATLAB code) and use the techniques in Section 4.3.4.
17: end for
Algorithm 2 Pseudo code of workflow (Part II)

19: for TIME do ∈ 2 ~ 24
20: Replace the lines of ICON running code in Section 4.3.2
21: for ens do ∈ 1 ~ Num_of_ens
22: Replace the line of ICON running code about ”INIT_CONC_1” with ”ICON.$\{ens\}”
23: Replace the line of ICON running code about ”CTM_CONC_1” with ”CCTM.$\{ens\}”
24: Execute the ICON running code
25: end for
26: Replace the lines of CCTM running code in Section 4.3.2
27: for ens do ∈ 1 ~ Num_of_ens
28: Replace the line of CCTM running code about ”CONCfile” with ”CCTM.$\{ens\}”
29: Replace the line of CCTM running code about ”ICFILE” with ”ICON.$\{ens\}”
30: Execute the CCTM running code
31: end for
32: Run the analysis step code of the proposed data assimilation method (MATLAB code) and use the techniques in Section 4.3.4.
33: end for
34: end for
35: Update the name and position of layer file for next day
36: for DAY do ∈ 2 ~ 30 (or 28, 31)
Algorithm 3 Pseudo code of workflow (Part III)

37: for TIME do ∈ 1 ~ 24
38: Replace the lines of ICON running code in Section 4.3.2
39: for ens do ∈ 1 ~ Num_of_ens
40: Replace the line of ICON running code about ”INIT_CONC_1” with ”ICON.${ens}”
41: Replace the line of ICON running code about ”CTM_CONC_1” with ”CCTM.${ens}”
42: Execute the ICON running code
43: end for
44: Replace the lines of CCTM running code in Section 4.3.2
45: for ens do ∈ 1 ~ Num_of_ens
46: Replace the line of CCTM running code about ”CONCfile” with ”CCTM.${ens}”
47: Replace the line of CCTM running code about ”ICFILE” with ”ICON.${ens}”
48: Execute the CCTM running code
49: end for
50: Run the analysis step code of proposed data assimilation method (MATLAB code) and use the techniques in Section 4.3.4.
51: end for
52: end for
53: end for

4.3.3 Configurations on the Data Assimilation to fit the CMAQ System

The input and output process of the Data Assimilation with respect to the CMAQ system

To collaborate with the CMAQ system, Data Assimilation process needs to recognize the pattern of reading and writing on the concentration file outputted from CCTM and the file is in NETCDF format. NETCDF is a self-describing and machine-
independent data format that is based on the matrix and support to create, access and share in the array-oriented scientific data. It is hosted by the University Corporation for Atmospheric Research (UCAR) and distributed in C, Fortran, C++, Java and other languages.

In the current case, we are using the MATLAB to handle the access and update of the NETCDF file generated from CCTM. MATLAB has its own NETCDF package and contains many functions. In the subsection, I will introduce some important functions that are used in our program.

- **netcdf.open** and **netcdf.close**:
  Matlab describes the **netcdf.open** as

  \[
  \text{ncid} = \text{netcdf.open} \left( \text{source, mode} \right),
  \]

  where the **ncid** is the identity number of NETCDF file, and the **source** can be the name of a NETCDF file or the URL of a NETCDF data source. The option **mode** is chosen from Table 4.1.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>'WRITE'</td>
<td>Read-write access</td>
</tr>
<tr>
<td>'SHARE'</td>
<td>Synchronous file updates</td>
</tr>
<tr>
<td>'NOWRITE'</td>
<td>Read-only access (Default)</td>
</tr>
</tbody>
</table>

  It will be easier to employ the **netcdf.close** function. There is only one description for it which is

  \[
  \text{netcdf.close} \left( \text{ncid} \right).
  \]

  It terminates the access to the identified NETCDF file with the value of **ncid**.

- **netcdf.inqVarID**
  After getting the identity number of NETCDF file, the next step is to obtain
the identity number of the variable for the tracer species. In this process, the function `netcdf.inqVarID` employs the following command

```
varid = netcdf.inqVarID ( ncid, varname )
```

with previous `ncid` value and the name of tracer species `varname` to acquire the value of `varid`.

- **netcdf.getVar** and **netcdf.putVar**
  
  With the identity number of NETCDF file and tracer species, function `netcdf.getVar` is able to read and function `netcdf.putVar` is able to write the real data from the NETCDF file, i.e., the output of CCTM. These two functions collect the value of `ncid`, value of `varid`, starting vector of `start` and count vector of `count` to read and write the four dimensional background state of `data`.

```
data = netcdf.getVar ( ncid, varid, start, count )
netcdf.putVar ( ncid, varid, start, count, data )
```

It should be remarked that the four indexes in vectors `start` and `count` are in reverse order with respect to our common sense. The indexes are ordered successively as y-coordinate, x-coordinate, number of layer and time index. Moreover, the indexes in vector `start` are started from 0, not 1 which is different from the general concept of MATLAB. And the indexes in vector `count` are showing the count number and it includes the starting index. This sentence may be hard to understand, so I will show an example.

```matlab
for iens=1:Nens
    ncid = netcdf.open(strcat('CCTM','.',num2str(iens)),'NOWRITE');
    varid = netcdf.inqVarID(ncid,species_name{ispec});
    data(:,iens) = netcdf.getVar(ncid,varid,[0,0,0,1],[74,74,26,1]);
    netcdf.close(ncid);
end
for iens=1:Nens
    ncid = netcdf.open(strcat('CCTM','.',num2str(iens)),'WRITE');
    varid = netcdf.inqVarID(ncid,species_name{ispec});
    netcdf.putVar(ncid,varid,[0,0,0,1],[74,74,26,1],data(:,iens));
    netcdf.close(ncid);
end
```

Figure 4.4: The parts of program on reading and writing NETCDF file
The example shown in Figure 4.4 presents the program is reading/writing the data from/to several ensembles in the NETCDF format. The updating area is a three dimensional model in a $74 \times 74 \times 26$ mesh. The starting point of model is $(1,1,1)$ and the ending point is $(74,74,26)$ which refers to the starting index $[0,0,0]$ and ending index $[73,73,25]$ of the code. Since $[0,0,0]$ needs 74 number of counts in the x-coordinate and y-coordinate, and 26 in the z-coordinate, eventually, the count index become $[74,74,26]$. In the meanwhile, Figure 4.4 also demonstrates the usages of previous opening, closing and inquiring functions.

The transformation process of 3D model to the forecast vector

This is the core equation of Ensemble Kalman Filter of Data Assimilation.

$$ a^{(e)} = f^{(e)} + \hat{K}(y^{(e)} - H(f^{(e)})) . \quad (4.5) $$

Based on the practical problem, first, we set up a mesh to model the concentration of pollutant in three dimension (see Figure 4.5).

![Figure 4.5: Model of concentration of pollutant in 3D](image)

This mesh contains 26 layers and each layer has 74 rows and 74 columns. So, in
all, there are 142,376 grid points in this mesh to model the concentration of pollutant in a 3D space, we called it as the background state. We name the grid point 1 to 142,376 from the bottom left corner to the top right corner row by row, and from the first layer to the 26th layer, which means the $c^{(e)}$ is expressed as (here the superscript involved in $c^{(e)}$ is the number of grid point and $n = 142,376$)

$$
c^{(e)} = \begin{pmatrix}
c^{(1,e)} \\
c^{(2,e)} \\
c^{(3,e)} \\
\vdots \\
c^{(n,e)}
\end{pmatrix},
$$

and use the background covariance $Q$ to generate white noise $q^{(e)}$ to get $f^{(e)}$. So each ensemble of background concentration of pollutant $f^{(e)} = c^{(e)} + q^{(e)}$

$$
f^{(e)} = \begin{pmatrix}
c^{(1,e)} \\
c^{(2,e)} \\
c^{(3,e)} \\
\vdots \\
c^{(n,e)}
\end{pmatrix} + q^{(e)} = \begin{pmatrix}
c^{(1,e)} + q^{(1,e)} \\
c^{(2,e)} + q^{(2,e)} \\
c^{(3,e)} + q^{(3,e)} \\
\vdots \\
c^{(n,e)} + q^{(n,e)}
\end{pmatrix} = \begin{pmatrix}
f^{(1,e)} \\
f^{(2,e)} \\
f^{(3,e)} \\
\vdots \\
f^{(n,e)}
\end{pmatrix}.
$$

Here the second superscript of $c$ and $q$ means the number of ensemble, while the first superscript of $c$ and $q$ is the number of grid point.

**The transformation process of observational stations to observations**

In the meanwhile, to get $y$ from the concentrations collected from the observational stations, we need a strategy to get the values on the grid points of the mesh. Since we know the observation stations are in this 3D mesh whereby each one is treated as if it is in a small cube (See Figure 4.6).

![Figure 4.6: The observational station in cube](image)
Since we want the values on the neighboring grid points, we have mapped this observation value to 8 neighboring grid points first and ensure that there are more than one observation stations for one grid point in the 8 neighboring cubes like in Figure 4.7.

![Figure 4.7: The model of the observational stations in mesh](image)

We use weights to get the concentration on \( y \) (here superscript \( i \) is the number of grid point of observation). In common, the nearer point will be effected more than the further ones, so we use \( \frac{1}{d^2} \) as weight and the concentration \( y = \frac{\frac{1}{d_1} \cdot c_{obs_1} + \frac{1}{d_2} \cdot c_{obs_2}}{\frac{1}{d_1} + \frac{1}{d_2}} \).

Use this similar strategy if there are more observation stations in 8 neighboring cubes for one grid point \( y = \frac{\sum_{j=1}^{k} \frac{1}{d_j} \cdot c_{obs_j}}{\sum_{j=1}^{k} \frac{1}{d_j}} \) (\( j \) means the number of observational stations neighbor this grid point and \( k \) means the total number of observation stations around this grid point).

After getting \( y^{(1)} \ldots y^{(m)} \) (\( m \) means the total number of observations), we need to use the observation covariance \( V \) to generate observation white noise \( u^{(e)} \) on \( y \) to get ensemble \( y^{(e)} = y + u^{(e)} \) (here superscript of \( y \) and \( u \) is the number of ensembles).

This is similar to the the generating \( f^{(e)} \),

\[
y^{(e)} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{pmatrix} + u^{(e)} = \begin{pmatrix} y^{(1)} + u^{(1,e)} \\ y^{(2)} + u^{(2,e)} \\ y^{(3)} + u^{(3,e)} \\ \vdots \\ y^{(m)} + u^{(m,e)} \end{pmatrix} = \begin{pmatrix} y^{(1,e)} \\ y^{(2,e)} \\ y^{(3,e)} \\ \vdots \\ y^{(m,e)} \end{pmatrix}
\]

But for the observations, there is a very special situation. Sometimes, since some observation stations are not available, the concentration of some pollutants will be missed (see Figure 4.8, the negative value). And our solution is to ignore the missing
values and only focus on those available values. This solution will lead the number
of observations $m$ dynamically to be changed in every iteration, but it is better than
setting the unavailable values to zero to bring in the noises.

<table>
<thead>
<tr>
<th>m</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
<th>l</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
<th>q</th>
<th>r</th>
<th>s</th>
<th>t</th>
<th>u</th>
<th>v</th>
<th>w</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>201001010000 cb 129.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 cl -32767.00 -1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 cw 52.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 en 46.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 kc 53.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 kt 47.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 mb 8.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 mk 119.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 ssp 55.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 st 25.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 tc 37.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 tp 29.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 tw 49.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>201001010000 yl 41.00 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.8: Missing value

4.3.4 Techniques used in the Proposed Data Assimilation Method

Background Covariance Estimation

We use a strategy called Multivariate Normal Distribution [41] to generate the random
vector from $Q$ and $V$, so that the generation of $Q$ and $V$ will be based on their own
concentration distributions. It will try to find a way to link more elements, such as
$Q$ and $V$, with the background state and the observation concentration which will
bring the estimation much closer to the reality. That is the effect that we want.

Learning from Central Limit Theorem of Multivariate Normal Distribution, we
try to generate a normal distribution vector $N(0, I)$ in a closed interval from a uniform
distribution $U(0, 1)$ as follows,

$$
\lim_{n \to \infty} \frac{\left( \sum_{i=1}^{n} U(0, 1) \right) - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \sim N(0, I) \ .
$$

From the conclusion, in practice, $n = 12$ yields very good approximation. Here
we choose $n = 48$.

After that, this $N(0, I)$ generates the random numbers between $-12$ and $12$. The
mean is 0 and the covariance is 1. We name this special standard normal distribution as \( \hat{N}(0, I) \).

Then we learn from the mean and variance of the linear transformation theorem. If a vector \( \mathbf{Z} \) has a mean \( E[\mathbf{Z}] \) and a covariance matrix \( E[(\mathbf{Z} - E[\mathbf{Z}])(\mathbf{Z} - E[\mathbf{Z}])^T] = \Sigma_\mathbf{Z} \), and both of their dimensions are \( d \). Then for

\[
\Theta = \begin{pmatrix}
\theta_{1,1} & \theta_{1,2} & \ldots & \theta_{1,d} \\
\theta_{2,1} & \theta_{2,2} & \ldots & \theta_{2,d} \\
\vdots & \vdots & \ddots & \vdots \\
\theta_{k,1} & \theta_{k,2} & \ldots & \theta_{k,d}
\end{pmatrix}
\quad \text{and} \quad
\Psi = \begin{pmatrix}
\psi_1 \\
\psi_2 \\
\vdots \\
\psi_k
\end{pmatrix},
\]

\( \mathbf{Y} = \Theta \mathbf{Z} + \Psi \) has the mean \( E[\mathbf{Y}] = \Theta E[\mathbf{Z}] + \Psi \) and the covariance matrix \( \Sigma_\mathbf{Y} = \Theta \Sigma_\mathbf{Z} \Theta^T \).

We set each element of \( \mathbf{Z} \) fitting \( \hat{N}(0, 1) \) and \( \Psi = 0 \). So the covariance \( \Sigma_\mathbf{Y} = \Theta \Theta^T \) and the mean \( E[\mathbf{Y}] = \Theta E[\mathbf{Z}] = 0 \).

Let us go back to discuss \( \mathbf{q} \) and \( \mathbf{u} \). \( \mathbf{q} \) is the white noise and could be based on the background covariance \( \mathbf{Q} \). The sample covariance of background state is \( \mathbf{Q} = \frac{1}{N-1} \sum_{s=1}^{N} (\mathbf{c}^{(s)} - \bar{\mathbf{c}})(\mathbf{c}^{(s)} - \bar{\mathbf{c}}) \). So for one day, there are 24 samples from the background state (now the samples are from 1h, 2h, 3h,..., 24h). The sample matrix should look like this

<table>
<thead>
<tr>
<th></th>
<th>1st sample</th>
<th>2nd sample</th>
<th>3rd sample</th>
<th>...</th>
<th>24th sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>point 1</td>
<td>( \mathbf{c}^{(1,1)} )</td>
<td>( \mathbf{c}^{(1,2)} )</td>
<td>( \mathbf{c}^{(1,3)} )</td>
<td>...</td>
<td>( \mathbf{c}^{(1,24)} )</td>
</tr>
<tr>
<td>point 2</td>
<td>( \mathbf{c}^{(2,1)} )</td>
<td>( \mathbf{c}^{(2,2)} )</td>
<td>( \mathbf{c}^{(2,3)} )</td>
<td>...</td>
<td>( \mathbf{c}^{(2,24)} )</td>
</tr>
<tr>
<td>point 3</td>
<td>( \mathbf{c}^{(3,1)} )</td>
<td>( \mathbf{c}^{(3,2)} )</td>
<td>( \mathbf{c}^{(3,3)} )</td>
<td>...</td>
<td>( \mathbf{c}^{(3,24)} )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>point n</td>
<td>( \mathbf{c}^{(n,1)} )</td>
<td>( \mathbf{c}^{(n,2)} )</td>
<td>( \mathbf{c}^{(n,3)} )</td>
<td>...</td>
<td>( \mathbf{c}^{(n,24)} )</td>
</tr>
</tbody>
</table>

Then we calculate the mean of 24 samples by using \( \bar{\mathbf{c}} = \frac{\sum_{s=1}^{24} \mathbf{c}^{(s)}}{24} \) and subtract the
mean from each column of matrix. Naming the following matrix

$$
\begin{pmatrix}
\begin{array}{cccc}
\mathbf{c}^{(1,1)} - \overline{\mathbf{c}}^{(1)} & \mathbf{c}^{(1,2)} - \overline{\mathbf{c}}^{(1)} & \mathbf{c}^{(1,3)} - \overline{\mathbf{c}}^{(1)} & \cdots & \mathbf{c}^{(1,24)} - \overline{\mathbf{c}}^{(1)} \\
\mathbf{c}^{(2,1)} - \overline{\mathbf{c}}^{(2)} & \mathbf{c}^{(2,2)} - \overline{\mathbf{c}}^{(2)} & \mathbf{c}^{(2,3)} - \overline{\mathbf{c}}^{(2)} & \cdots & \mathbf{c}^{(2,24)} - \overline{\mathbf{c}}^{(2)} \\
\mathbf{c}^{(3,1)} - \overline{\mathbf{c}}^{(3)} & \mathbf{c}^{(3,2)} - \overline{\mathbf{c}}^{(3)} & \mathbf{c}^{(3,3)} - \overline{\mathbf{c}}^{(3)} & \cdots & \mathbf{c}^{(3,24)} - \overline{\mathbf{c}}^{(3)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{c}^{(n,1)} - \overline{\mathbf{c}}^{(n)} & \mathbf{c}^{(n,2)} - \overline{\mathbf{c}}^{(n)} & \mathbf{c}^{(n,3)} - \overline{\mathbf{c}}^{(n)} & \cdots & \mathbf{c}^{(n,24)} - \overline{\mathbf{c}}^{(n)}
\end{array}
\end{pmatrix}
$$

as $\tilde{\mathbf{A}}$. Then the covariance matrix should be $\mathbf{Q} = \frac{1}{N-1} \tilde{\mathbf{A}} \cdot \tilde{\mathbf{A}}^T$.

We use SVD on $\tilde{\mathbf{A}}$ and then get Equation (4.7). ($\mathbf{R}$ is an orthogonal matrix, so $\mathbf{R}^T = \mathbf{R}^{-1}$.)

$$\tilde{\mathbf{A}} = \mathbf{U} \ast \mathbf{S} \ast \mathbf{R}^T. \quad (4.7)$$

So

$$\tilde{\mathbf{A}} \ast \tilde{\mathbf{A}}^T$$

$$= \mathbf{U} \ast \mathbf{S} \ast \mathbf{R}^T \ast (\mathbf{U} \ast \mathbf{S} \ast \mathbf{R}^T)^T$$

$$= \mathbf{U} \ast \mathbf{S} \ast \mathbf{R}^T \ast \mathbf{R} \ast \mathbf{S}^T \ast \mathbf{U}^T$$

$$= \mathbf{U} \ast \mathbf{S} \ast \mathbf{S}^T \ast \mathbf{U}^T$$

and

$$\tilde{\mathbf{A}}^T \ast \tilde{\mathbf{A}}$$

$$= (\mathbf{U} \ast \mathbf{S} \ast \mathbf{R})^T \ast \mathbf{U} \ast \mathbf{S} \ast \mathbf{R}^T$$

$$= \mathbf{R} \ast \mathbf{S}^T \ast \mathbf{U}^T \ast \mathbf{U} \ast \mathbf{S} \ast \mathbf{R}^T$$

$$= \mathbf{R} \ast \mathbf{S}^T \ast \mathbf{S} \ast \mathbf{R}^T. \quad (4.8)$$

Do the eigenvalue decomposition on $\tilde{\mathbf{A}}^T \ast \tilde{\mathbf{A}}$, so

$$\tilde{\mathbf{A}}^T \ast \tilde{\mathbf{A}}$$

$$= \mathbf{E} \ast \mathbf{G} \ast \mathbf{E}^T. \quad (4.9)$$

From Equations (4.8) and (4.9),

$$\mathbf{R} \ast \mathbf{S}^T \ast \mathbf{S} \ast \mathbf{R}^T = \mathbf{E} \ast \mathbf{G} \ast \mathbf{E}^T. \quad (4.10)$$
We can set a solution that
\[ R = E \quad \text{and} \quad S^T S = G. \] (4.11)

Because
\[
Q = \frac{1}{N-1} \tilde{A} \tilde{A}^T
= \frac{1}{N-1} U S R (U S R)^T
= \frac{1}{N-1} U S S^T U^T
= (\sqrt{\frac{1}{N-1} U S}) (\sqrt{\frac{1}{N-1} U S})^T. \] (4.12)

From Equation (4.7), we can then do a transformation by multiplying \( R \) to both right sides and it becomes
\[ \tilde{A} R = U S. \] (4.13)

From Equations (4.11) and (4.13), Equation (4.12) will be
\[
Q = (\sqrt{\frac{1}{N-1} U S}) (\sqrt{\frac{1}{N-1} U S})^T
= (\sqrt{\frac{1}{N-1} \tilde{A} R}) (\sqrt{\frac{1}{N-1} \tilde{A} R})^T
= (\sqrt{\frac{1}{N-1} \tilde{A} E}) (\sqrt{\frac{1}{N-1} \tilde{A} E})^T.
\]

As mentioned before, we have set a linear transformation from \( Z \) to \( Y \), \( Y = \Theta Z + \Psi \) with \( \Psi = 0 \), \( Z \sim \tilde{N}(0,1) \), then \( Y = \Theta Z \). So the covariance \( \Sigma_Y = \Theta \Theta^T \) and the mean \( E[Y] = \Theta E[Z] = 0 \). Here we set \( \Theta = \frac{1}{N-1} \tilde{A} E \), and \( Z \sim \tilde{N}(0,1) \), and the mean \( E[Y] = \Theta E[Z] = 0 \) and the covariance \( \Sigma_Y = \text{Cov}(Y) = \Theta \Theta^T = (\sqrt{\frac{1}{N-1} \tilde{A} E}) (\sqrt{\frac{1}{N-1} \tilde{A} E})^T = Q \), finally \( Y \sim N(0, Q) \).

Hence we can add this \( Y \) as the background white noise \( q \) from \( Q \) to get \( f^{(e)} \) in Equation (3.38) which is based on the background state \( f \) from 1h to 24h.

**Observation Noise Estimation**
In the similar way, we can set the observation sample matrix as follows:

\[
\begin{pmatrix}
  \text{station 1} & \text{station 2} & \text{station 3} & \cdots & \text{station m} \\
  y^{(1,1)} & y^{(1,2)} & y^{(1,3)} & \cdots & y^{(m,24)} \\
  y^{(2,1)} & y^{(2,2)} & y^{(2,3)} & \cdots & y^{(24,24)} \\
  y^{(3,1)} & y^{(3,2)} & y^{(3,3)} & \cdots & y^{(3,24)} \\
  \vdots & \vdots & \vdots & \cdots & \vdots \\
  y^{(m,1)} & y^{(m,2)} & y^{(m,3)} & \cdots & y^{(m,24)} \\
\end{pmatrix}
\]

then subtract the mean of samples \( \bar{y} = \frac{\sum_{s=1}^{24} y^{(s)}}{24} \) to get matrix \( \tilde{A}' \) as follows

\[
\begin{pmatrix}
  \text{station 1} & \text{station 2} & \text{station 3} & \cdots & \text{station m} \\
  y^{(1,1)} - \bar{y}^{(1)} & y^{(1,2)} - \bar{y}^{(1)} & y^{(1,3)} - \bar{y}^{(1)} & \cdots & y^{(m,24)} - \bar{y}^{(m)} \\
  y^{(2,1)} - \bar{y}^{(2)} & y^{(2,2)} - \bar{y}^{(2)} & y^{(2,3)} - \bar{y}^{(2)} & \cdots & y^{(24,24)} - \bar{y}^{(24)} \\
  y^{(3,1)} - \bar{y}^{(3)} & y^{(3,2)} - \bar{y}^{(3)} & y^{(3,3)} - \bar{y}^{(3)} & \cdots & y^{(3,24)} - \bar{y}^{(3)} \\
  \vdots & \vdots & \vdots & \cdots & \vdots \\
  y^{(m,1)} - \bar{y}^{(m)} & y^{(m,2)} - \bar{y}^{(m)} & y^{(m,3)} - \bar{y}^{(m)} & \cdots & y^{(m,24)} - \bar{y}^{(m)} \\
\end{pmatrix}
\]

and do the eigenvalue decomposition on \((\tilde{A}')^T \cdot \tilde{A}'\) to get \(E'\). So the \( V = (\sqrt{\frac{1}{N-1}} \cdot \tilde{A}' \cdot \tilde{A}') \times (\sqrt{\frac{1}{N-1}} \cdot \tilde{E}' \cdot \tilde{E}') \times (\sqrt{\frac{1}{N-1}} \cdot \tilde{A}' \cdot \tilde{E}') \times (\sqrt{\frac{1}{N-1}} \cdot \tilde{A}' \cdot \tilde{E}')^T \).

Now, we set \( Y' = \Theta' Z \) and \( \Theta' = \sqrt{\frac{1}{N-1}} \cdot \tilde{A}' \cdot \tilde{E}' \cdot Z \sim \tilde{N}(0,1) \), then the mean \( E[Y'] = \Theta' E[Z] = 0 \) and the covariance \( \Sigma_Y = \text{Cov}(Y') = \Theta' \Theta'^T = (\sqrt{\frac{1}{N-1}} \cdot \tilde{A}' \cdot \tilde{E}') \times (\sqrt{\frac{1}{N-1}} \cdot \tilde{A}' \cdot \tilde{E}')^T = V \), and finally \( Y' \sim N(0, V) \).

We add this \( Y' \) as the observation white noise \( u \) from \( V \) to get \( y^{(e)} \) which is based on the concentrations collected from the observational stations during 1h to 24h.

### 4.4 Numerical Results

In this section, we test the chemistry transport model to demonstrate the efficiency of the proposed sparse computational method and the effectiveness of the block matrix modeling in air quality data assimilation. The chemistry transport model solves mass balance equations in a three-dimensional domain for air quality forecast, see CMAQ [6]. The computational model can be represented by the following scheme:

\[
c_{k+1} = L_k(c_k, b_k, \hat{v}_k, Q_c), \tag{4.14}
\]
where \( L_k \) is the discrete chemistry transport model solution operator, \( c_k \) is the concentration of species at time \( k \), \( b_k \) is the boundary condition at time \( k \), \( \hat{v}_k \) is the meteorological input data at time \( k \) and \( Q_c \) is the emission input data at time \( k \). This kind of model has been used in testing for data assimilation methods, for instance see [69, 19, 25].

In our experiment, we study the air pollutant concentrations in the Hong Kong area by using (4.14). The computational domain involves \( 74 \times 74 \times 26 \) (142,376) grid points. There are five pollutant species: NO\(_2\), NO, CO, SO\(_2\), O\(_3\) to be tested for sparse matrix computation, and there are two accumulated species PM2.5 and PM10 to be studied for block matrix modeling. The number of computational variables of each single pollutant is 142,376. For PM2.5, and there are 33 pollutant species added together. For PM10, seven more pollutant species are added to those 33 species to form the resulting concentration. The numbers of computational variables of PM2.5 and PM10 are 4,698,408 and 5,695,040 respectively. In the practical setting, there are 112 grid points to have hourly measurements in the concentrations of NO\(_2\), NO, CO, SO\(_2\), O\(_3\), PM2.5 and PM10 which are collected by the Hong Kong Environmental Protection Department. We make use of these sparse observations \( y_k \) to conduct the proposed data assimilation procedure to improve the analysis \( a_k \) of pollutant concentrations. Here we compute the root mean squares error

\[
RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (a_k - x_k)^2}
\]

to evaluate the performance of the proposed data assimilation procedure, where \( N \) is the time points to be studied in our experiment.

### 4.4.1 Single Species

**The Parameters Setting**

**The Effect of \( \lambda \) (to replace the \( \frac{1}{\alpha} \)) and \( \gamma \).** We first test the effect of the parameters for the proposed data assimilation procedure. The parameters include

- \( \lambda \) for controlling the weighting between the observation error covariance matrix and the background noise covariance matrix in Equation (3.35);
• \( \gamma \) for controlling the influence of the decorrelation in Equation (3.41);

• \( s \) for the number of samples to be used for the estimation of background covariance matrix in the initialization.

We note that when the observation error \( v_k \) is large or the smallest eigenvalue of the observation error covariance matrix is large, we can use a small value of \( \lambda \).

Here we test different values of \( \lambda \) according to the concentration levels of five pollutant species. Considering that the effect of \( \lambda \) is to control the balance between the level of error of observation and background, we test the values of \( \lambda \) mainly in 5 sets of level: the first set is the smallest value with \( \lambda \) in \( \{0.00001, 0.00002, \ldots, 0.00009\} \); the second set is the 10 times of the first set with \( \lambda \) in \( \{0.0001, 0.0002, \ldots, 0.0009\} \); the third set is 100 times of the first set with \( \lambda \) in \( \{0.001, 0.002, \ldots, 0.009\} \); by this analogy, the fourth set is 1000 times of the first set and the fifth set is 10,000 times of the first set. Adding up the \( \lambda \) in \( \{100\} \), these 64 values become the selected values for \( \lambda \) of NO\(_2\), NO, O\(_3\), SO\(_2\) and CO.

On the other hand, the parameter \( \gamma \) is used to control the influence in the decorrelation matrix \( D \). When \( \gamma \) is small (or large), the entries along the diagonals of \( D \) decay very quickly (or slowly), and therefore the corresponding correlation of the pollutant at two different spatial locations is weak (or strong). Different pollutants can have different levels of correlation because their chemical effects are different when they are involved in varying atmospheric chemical reactions. In particular, we consider the values of \( \gamma \) in \( \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, \ldots, 10\} \) for nitrogen family (NO\(_2\) and NO), SO\(_2\), CO and O\(_3\).

In the experimental setting, we perform a 1-hour data assimilation (\( N = 1 \)) in the time on a day in January 2010 to evaluate the different combinations of \( \lambda \) and \( \gamma \) for testing. In the initialization, we make use of the 24 hours simulation results of the previous day to build the background covariance matrices to generate the initial ensembles. Here for simplicity, we set the number of ensembles to be nine. We compute RMSEs for different testing values of \( \lambda \) and \( \gamma \), and choose the best combination for the five single pollutants for each time. In the meanwhile, a time-saving strategy is applied here. We assume that the situation between two time points does not differ greatly at first, and then we use the previous best combination of \( \lambda \)
and $\gamma$ as the starting point for choosing the current time. It leads to the subsequent choosing process becoming very efficient.

The following Figures 4.9, 4.10 and 4.11 are the changes on the daily average of the best combination of $\lambda$ and $\gamma$ among the days in January, February and March of 2010.
Figure 4.9: The changes on daily average of best combination of $\lambda$ and $\gamma$ for single species in January
(a) The $\lambda$ of NO$_2$

(b) The $\gamma$ of NO$_2$

(c) The $\lambda$ of O$_3$

(d) The $\gamma$ of O$_3$

(e) The $\lambda$ of SO$_2$

(f) The $\gamma$ of SO$_2$

(g) The $\lambda$ of NO

(h) The $\gamma$ of NO
Figure 4.10: The changes on daily average of best combination of $\lambda$ and $\gamma$ for single species in February
The Figures 4.9, 4.10 and 4.11 show that first of all, the best \( \lambda \) are not always equal to 1 which means the \( \lambda \) (the covariance inflation technique) has the effect on improving the accuracy of EnKF. Then generally, the best \( \lambda \) for NO\(_2\), O\(_3\), SO\(_2\) and NO are smaller than 0.1. It implies that compared to the correlations within the observation covariance, the correlations within the background covariance is more reliable to estimate the increment of the model state. It is reasonable that because
the sensors of observations in real world don’t possess many connections between each other. But the projected analysis state (the approximation on projected truth) should have this kind of feature. Thus, the observation covariance could provide some useful information but these information can’t reach a big proportion. The best $\lambda$ are mainly involved in the interval $0.001 \sim 0.01$ for NO$_2$, $0.001 \sim 0.1$ for O$_3$, SO$_2$ and NO. While for the CO, the best $\lambda$ is obviously bigger than 0.1 and even bigger than 1 in sometimes. That unusual performance makes us confused at the beginning. However, after checking the values of observations of CO, we can explain this special situation. For CO, its values of observations are bigger or smaller than the values of project forecast state uniformly. Now, we are focusing on the RMSE of the projected analysis state with respect to the observations. Focusing on obtaining the smaller RMSE leads the value of projected analysis state to approach the values of observations and gives more weights on the observation covariance if the values of observations have the uniform behavior on the comparison towards the value of projected forecast state. These conclusions have the consistent performance in January, February and March.

For the background covariance, since the number of ensembles is smaller than the number of variables, it may suffer from the spurious correlations. And now the $\gamma$ is the measurement to employ the localization technique on the background covariance and using the decorrelation function to reduce the influence of spurious correlations. Learning from the Figures 4.9, 4.10 and 4.11, the best $\gamma$ for NO$_2$, O$_3$, SO$_2$, NO and CO are mainly between 1 and 10. First of all, the performances show the $\gamma$ (localization technique) has the ability to improve the accuracy of EnKF and the background covariance does suffer from the spurious correlations (if not, the best $\gamma$ should close to the zero). Second, these best $\gamma$ show that the corresponding correlations of these five single pollutants at two different spatial locations are very strong. And within these five single species, the influences of decorrelation for O$_3$, SO$_2$ and NO are stronger than ones for NO$_2$ and CO. These conclusions are consistent in January, February and March.

The Effect of Background Covariance Matrices

In the ensemble Kalman filter, we need to generate a set of ensembles based on the
initial background covariance matrix. In the previous test, we make use of the former 24-hour simulation results to build the initial background covariance matrix. Here we further test by using the previous 6-hour, 12-hour, 18-hour and 24-hour to build background covariance matrices and then to evaluate the one-hour data assimilation performances ($N = 1$). We also use the tested best parameter values of $\lambda$ and $\gamma$ in the previous subsection for different pollutant species.

In Figures 4.12, 4.13 and 4.14, it can be seen that different pollutant species have different best tested numbers of hours for the generation of background covariance matrices based on the RMSEs calculation.
Figure 4.12: The changes on daily average sample number for single species in January.
Figure 4.13: The changes on daily average sample number for single species in February
From the Figures 4.12, 4.13 and 4.14, we can conclude that the best daily average numbers of samples are in the intervals $[15,20]$ for NO\(_2\), $[10,20]$ for O\(_3\), $[13,20]$ for SO\(_2\), $[14,20]$ for NO and $[5,15]$ for CO, respectively. The suitable numbers are not equal to the biggest number of samples always. And they show that different species have different suitable numbers of samples to accurately describe the model noises in Gaussian distribution. This parameter is depended on the species itself. And These characteristics are consistent among the months.
We remark that we repeat the evaluation of the data assimilation performance by using $\lambda$ and $\gamma$ for best tested number of hours for the generation of background covariance matrices. We find that the best tested values of $\lambda$ and $\gamma$ for the five pollutant species, are still valid as in the previous subsection.

**The Comparison of Localization Models**

Here we compare the performance of the space localization model in (3.42) and the observation space localization model in (3.43). We conduct 24-hour simulation for (3.42) and (3.43) and compare their prediction performance. The RMSE results of the two pollutants (CO and O$_3$) are the same for the two models. However, the RMSE results of the three pollutants (NO$_2$, SO$_2$ and NO) are different for the two models. In particular, the RMSEs of using (3.42) are 0.1007, 0.0013 and 0.0382 for NO$_2$, SO$_2$ and NO, and the RMSEs of using (3.43) are 0.1110, 0.0016 and 0.0505 for NO$_2$, SO$_2$ and NO. It shows that the performance of using (3.42) can be better than that of using (3.43). We figure out that the degree of decorrelation (localization) effect for CO and O$_3$ is small and therefore the RMSEs results of using (3.42) and (3.43) are about the same. The degree of decorrelation (localization) effect for NO$_2$, SO$_2$ and NO is large, and therefore the performance of using (3.42) can be better than that of using (3.43). These results are consistent with the numerical results in [59].

**Matrix Computation Results**

We first test the performance of the proposed sparse matrix computational method with respect to the number of ensembles. In Figure 4.15, we show the RMSEs of different pollutant species with respect to the number of ensembles. Here 24-hour simulation is used and the parameters setting is based on the best tested results in the previous subsection.
Figure 4.15: The RMSEs of five pollution species with respect to number of ensembles.

We can see from the figure that when the number of ensembles increases, the RMSE decreases, but it becomes stabilized after 9 ensembles are used. In Figure 4.16, we show their corresponding computational time required for the proposed data assimilation method.

Figure 4.16: The computational time required by data assimilation of five pollution species with respect to number of ensembles.

We see that the computational time depends linearly with respect to the number of ensembles. It is interesting to note that the sparse matrix computational method ($O(qrmn)$ per ensemble) can save more than 99% of computational time compared with the full matrix computational method ($O((m + q)n^2)$ per ensemble).
In Figures 4.17, 4.18 and 4.19, we show one-month data assimilation results by the proposed sparse computational method in January, February and March, respectively. We can see that the data assimilation can improve RMSE very well compared with the standard air quality simulation. It seems that the performance of the proposed method is about the same for the other month (February and March) in the test. Also the total computational time including data assimilation and simulation are required 1.28 hour and 358.4 hours for 1 month in January, July and March, respectively.

Figure 4.17: The RMSEs of five pollutant species for 1 month simulation in January.
Figure 4.18: The RMSEs of five pollutant species for 1 month simulation in February.
Figure 4.19: The RMSEs of five pollutant species for 1 month simulation in March.

These good performances increase our confidence to apply the proposed data assimilation to the other months. And since our workflow links the CMAQ system with data assimilation to become a high-precision and real-time system. These promising results can help us to employ our proposed method directly to the current forecast when the new observations are available.
4.4.2 Multiple Species

In this subsection, we test the effectiveness of the proposed method for multiple species. Here we focus on two sets of species, PM2.5 and PM10. We refer to the model in Section 3.1, where $c$ is equal to 40 and $o$ is equal to 2. Similar to the Section 4.1, we use the 1-hour RMSE performance to determine the four parameters used in the proposed data assimilation method, namely, $\lambda$, $\gamma$, the number of samples and ensembles used for the generation of initial background covariance matrices and ensemble covariance matrices. After checking RMSEs, we conclude that $\lambda$ and $\gamma$ are mainly involved in the intervals $[0.001,0.1]$ and $[3,9]$ respectively among the three months.

![Graphs showing the changes on daily average of best combination of $\lambda$ and $\gamma$ for multiple species in January](image)

(a) The $\lambda$ of combination of PM2.5 and PM10
(b) The $\gamma$ of combination of PM2.5 and PM10

Figure 4.20: The changes on daily average of best combination of $\lambda$ and $\gamma$ for multiple species in January
Figure 4.21: The changes on daily average of best combination of $\lambda$ and $\gamma$ for multiple species in February.

Figure 4.22: The changes on daily average of best combination of $\lambda$ and $\gamma$ for multiple species in March.

the number of sampled hours for the initial background covariance matrices is in the interval $[8, 22]$. 
Figure 4.23: The changes on daily average sample number for multiple species in January

Figure 4.24: The changes on daily average sample number for multiple species in February
For PM2.5 and PM10, the RMSE results of using (3.42) and (3.43) can be very significant. We note that the degree of decorrelation (localization) effect of each pollutant in PM2.5 and PM10 is large. We conduct 24-hour simulation to find out that the RMSEs of using (3.42) are 44.85 and 134.97 for PM2.5 and PM10 and the RMSEs of using (3.43) are 643.36 and 1335.60 for PM2.5 and PM10. And the results show that the use of (3.42) can be two or three times smaller than the use of (3.43).

In Figure 4.26, we show the 24-hour RMSE performance with respect to the number of ensembles to be used. When the number of ensembles increases, the RMSE decreases. However, we see that when the number of ensembles is larger than or equal to 12, the RMSEs are about the same.
Figure 4.26: The RMSEs of multiple species with respect to number of ensembles.

In Figure 4.27, we further demonstrate that the computational time required by the data assimilation method increases linearly with respect to the number of ensembles. We remark that there is only one line as both PM2.5 and PM10 are estimated together in the proposed method.

Figure 4.27: The computational time required by data assimilation of multiple species with respect to the number of ensembles.

Finally, in Figures 4.28, 4.29 and 4.30, we show the RMSEs of the data assimilation results by using the proposed method with the above specific parameters for 1 month in January, February and March, respectively. We can see from the figures that the
The performance of the proposed method is better than that of the simulation. Also the results are consistent in the other months in the test.

Figure 4.28: The RMSEs of the combination of multiple pollutant species for 1 month simulation in January

Figure 4.29: The RMSEs of the combination of multiple pollutant species for 1 month simulation in February
Figure 4.30: The RMSEs of the combination of multiple pollutant species for 1 month simulation in March

The performance of the combination of PM2.5 and PM10 is not very similar like that of single species. There exist some intersections around the hour 450 of February and in hour 400-580 of March. After checking these special cases, we find that they are due to the conflict between the PM2.5 and PM10 with respect to a same set of $\lambda$, $\gamma$, the number of samples and the number of ensembles. This set of parameters for PM2.5 and PM10 has the lowest RMSE on the combination result, but not the minimum value for both of them always. When this set of parameters close to the set of parameters of minimum value for PM2.5 and that for PM10, the situation will be like the situation of single species, and the result presents very well. However, if this set is significantly different from the set of minimum value for PM2.5 and that for PM10, the conflict will occur and lead to the result performing very bad, even worse than the purely air quality simulation. The key issue is the same set of parameter for PM2.5 and PM10, so the solution is that we can use two sets of parameters for each of them so that the combination result will be the sum of two minimum value. That will guarantees more on the nice performance. But this action will bring more computational burdens and we will research it in the next step.
Chapter 5

Optimization Method

5.1 Motivation

I have mentioned in the previous chapter, the data assimilation is developed by two mainstream mathematical methods. The first method is Kalman filter method based on the estimation theory, and the second method is the variational method deriving from the optimization theory. And we have raised a proposed method in the last chapter about reducing the huge computational cost and increasing the efficiency of the localized EnKF by making use of the characteristic of sparsity in the linear mapping operator. Also, we bring in the strategy to choose the parameter dynamically. The numerical results show that both measures have significant improvements. The first one largely reduces the time consumption in the generation of the background covariance from $O(c^2omn^2)$ to $O(qc^2romn)$ and the storage of variables. This method can be applied on the multiple species whose observations are coupled with each other while the coupled relationship makes the data assimilation problem complex. The second one helps the proposed method guarantees that the error is reaching the smallest value dynamically. The good numerical performance provides the confidence in this mean either for single species or multiple species. However, some shortcomings are demonstrated even if the result is good, especially on the time of simulation for the ensembles. The data assimilation method is started from the numerical weather forecast. There are many partial differential equations to simulate the concept of chaos in meteorology at that time. Certainly, these model are very simple compared
to current time. Thus, there is no need to spend too much time on simulation. With the development of the models, scientists have begun to realize the time consumption of the simulation. So, they have employed the Monte Carlo method to decrease the number of samples needed by the Kalman filter changing from a large numbers to only a small number. That leads to the change in the generation of ensemble Kalman filter, but this improvement is not enough. Since the models have gradually become more and more complicated, the time cost for each ensemble has grown very rapidly. Thus, the ensemble Kalman filter has begun to lose its advantage of being efficient in the physical step. At the same time, the advantage of using the variational method has become more noticeable. The reason being that there is no need for more time to be spent on simulation in the physical step but only that the iterations need to be done in the analysis step and with effective numerical algorithm, this analysis step can be solved very efficiently bringing the data assimilation method to a new age. That is why there is need to improve further on the variational method. In this chapter, we will present a new method which is an improvement from the three dimensional method. This method makes use of the information of the gradient in the background state and forecast state. It also makes use of the information of the gradient to avoid the global uniform noises existing in those states and assures that the assimilation is running in the right direction even with an imperfect model.

5.2 Proposed Method

5.2.1 Use of Forecast Gradients

In [55], Lorenc proposed the variational data assimilation method by minimizing the cost function for finding \( a \) as (5.1):

\[
\min_a P(a) = (a - f_k)^T B^{-1} (a - f) + (y - H(a))^T V^{-1} (y - H(a)) .
\] (5.1)

When the mapping operator \( H \) is linear, the solution of (5.1) can be found by solving a linear system:

\[
(B^{-1} + H^T V^{-1} H)a = B^{-1} f + H^T V^{-1} y ,
\] (5.2)
where $B$ and $V$ are the background error covariance matrix and observation error covariance matrix respectively.

In this thesis, we make use of the forecast gradients to formulate a new objective function:

$$\min_a F(a) = \|\nabla a - \nabla f\|^2 + \omega(y - Ha)^T V^{-1} (y - Ha), \quad (5.3)$$

where $\| \cdot \|_2$ is the Euclidean norm, $\nabla$ is the gradient operator and $\omega$ is a positive regularization parameter which is used to control the balance between the two terms in (5.3). Here we make use of the Neumann boundary condition for the gradient operator. In air quality and weather prediction, $a$ represents unknown variables at the three-dimensional computational grid values. The corresponding $\nabla a$ refers to a $n$-by-$1$ matrix combined by the $x$-direction gradient, $y$-direction gradient and $z$-direction gradient values of $a$. The first term is based on the difference between the gradients of the forecast and analysis (unknown) of state variables. By matching the observations and the projected unknown state variables, the second term enables to characterize the solution of (5.3). More precisely, the solution of (5.3) can be obtained by solving the following linear system:

$$(M + \omega H^T V^{-1} H)a = M f + \omega H^T V^{-1} y, \quad (5.4)$$

where $M := \nabla^T \nabla$ is the second-order finite difference matrix with the Neumann boundary condition:

$$M = D \otimes I \otimes I + I \otimes D \otimes I + I \otimes I \otimes D$$

where $I$ is the identity matrix, and

$$D := \begin{bmatrix}
1 & -1 & 0 & \cdots & 0 \\
-1 & 2 & -1 & 0 & \vdots \\
0 & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & 0 \\
\vdots & 0 & -1 & 2 & -1 \\
0 & \cdots & 0 & -1 & 1
\end{bmatrix}. $$

It is clear that $(M + \omega H^T V^{-1} H)$ is positive semi-definite. When the null space of $M$ is not the same as the null space of $H^T V^{-1} H$, the coefficient matrix $(M + \omega H^T V^{-1} H)$
is positive definite. We note that $M$ is a positive semi-definite matrix, its eigenvector corresponding to its zero eigenvalue is a vector $1$ of all ones. The sufficient condition of non-singularity becomes $H^T V^{-1} H \neq 0$. We know that when the entries of $H$ and $V^{-1}$ are non-negative, the condition is valid. For example, when $H$ refers to the projection matrix of analysis state variables on the locations of observations, its entries are non-negative; when $V$ is a diagonal matrix where its diagonal entries are the variances of error observations, the inverses of these entries are positive.

The resulting linear system in (5.4) is positive definite, and it can be solved by the conjugate gradient method [35]. Assume that $V$ is a diagonal matrix and there are $r$ non-zero entries in each row of $H$, the cost of the conjugate gradient iteration in each iteration is dependent $O(n_x n_y n_z r)$ where $n_x$, $n_y$ and $n_z$ are the number of unknown variables in $x$-direction, $y$-direction and $z$-direction respectively. In Section 4, we demonstrate that the proposed method can be quite efficient compared with other testing methods.

Since $(M + \omega H^T V^{-1} H)$ is a symmetric positive definite matrix, we can use conjugate gradient method to find the numerical solution. And in each iteration, there is a matrix-vector multiplication between $(M + \omega H^T V^{-1} H)$ and the residual. The cost of the complete generation of $M$ and the matrix-vector multiplication would reach $O(4n^2)$ which is very expensive for one iteration when the number $n$ of variables is very large.

### 5.2.2 Multiple Species

In air quality forecast system, the concentrations of different species are predicted via the chemistry-transport model. In the pollutants recording system, sensors detect multiple species and accumulate them together to make observations at different time steps. For instance, in the air quality forecast system, there are two key observations: PM2.5 and PM10. There are a total of 40 species counted in these two observations. PM10 contains all 40 observations and PM2.5 contains 33 observations. The main contribution of this section is to propose a block matrix model to manage the proposed method for these accumulated multiple species together.

With the loss of generality, we ignore the time index $t$ of the variables in the
following discussion. Let \( \mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(o)} \) be the observations of these \( o \) multiple observations. Let \( \mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(c)} \) be the model variables of these \( o \) multiple observations that are involved. Similar to (5.3), we propose and consider the following objective function (5.5):

\[
\min_{\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(o)}\}} \sum_{i=1}^{c} \left\| \nabla \mathbf{f}^{(i)} - \nabla \mathbf{a}^{(i)} \right\|^2 + \omega \sum_{j=1}^{o} \sum_{i=1}^{c} d_{i,j} \mathbf{Ha}^{(i)} - \mathbf{y}^{(j)} \right\|^2 \left( \mathbf{V}^{(j)} \right)^{-1} \\
(5.5)
\]

where \( f^{(i)} \) and \( a^{(i)} \) are the forecast and analysis of the \( i \)-th model variable \( c^{(i)} \),

\[
d_{i,j} = \begin{cases} 
1, & \text{if } c^{(i)} \text{ involves in the } j \text{-th observation;} \\
0, & \text{otherwise},
\end{cases} \quad 1 \leq i \leq c, \ 1 \leq j \leq o,
(5.6)
\]

\( V^{(j)} \) is the observation error covariance matrix of the \( j \)-th model variable, and \( \omega \) is a positive regularization parameter. The first term is based on the summation of the difference between the gradients of the forecast and the analysis of the \( i \)-th state variables. The second term is the data fitting of a group of analysis variables to the given observations. The solution of (5.5) can be determined by solving the following
system of equations:

\[
\begin{bmatrix}
M + \omega \sum_{j=1}^{o} d_{1,j} d_{1,j} H^T(V^{(j)})^{-1} H & \omega \sum_{j=1}^{o} d_{1,j} d_{2,j} H^T(V^{(j)})^{-1} H & \ldots & \omega \sum_{j=1}^{o} d_{1,j} d_{c,j} H^T(V^{(j)})^{-1} H \\
\omega \sum_{j=1}^{o} d_{2,j} d_{1,j} H^T(V^{(j)})^{-1} H & M + \omega \sum_{j=1}^{o} d_{2,j} d_{2,j} H^T(V^{(j)})^{-1} H & \ldots & \omega \sum_{j=1}^{o} d_{2,j} d_{c,j} H^T(V^{(j)})^{-1} H \\
\vdots & \vdots & \ddots & \vdots \\
\omega \sum_{j=1}^{o} d_{c,j} d_{1,j} H^T(V^{(j)})^{-1} H & \omega \sum_{j=1}^{o} d_{c,j} d_{2,j} H^T(V^{(j)})^{-1} H & \ldots & M + \omega \sum_{j=1}^{o} d_{c,j} d_{c,j} H^T(V^{(j)})^{-1} H
\end{bmatrix}
\]

\[
\begin{bmatrix}
a^{(1)} \\
a^{(2)} \\
\vdots \\
a^{(c)}
\end{bmatrix} =
\begin{bmatrix}
M_f^{(1)} + \omega \sum_{j=1}^{o} d_{1,j} H^T(V^{(j)})^{-1} y^{(j)} \\
M_f^{(2)} + \omega \sum_{j=1}^{o} d_{2,j} H^T(V^{(j)})^{-1} y^{(j)} \\
\vdots \\
M_f^{(c)} + \omega \sum_{j=1}^{o} d_{c,j} H^T(V^{(j)})^{-1} y^{(j)}
\end{bmatrix}
\]

(5.7)

It is interesting to note that the above coefficient matrix can be rewritten as the following structure:

\[
I \otimes M + \omega \sum_{j=1}^{o} d_{1,j} I \quad d_{2,j} I \quad \ldots \quad d_{c,j} I
\]

\[
H^T(V^{(j)})^{-1} H \quad d_{1,j} I \quad d_{2,j} I \quad \ldots \quad d_{c,j} I
\]

(5.8)

where \( \otimes \) is the Kronecker product. We note that \( I \otimes M \) is a positive semi-definite matrix, its eigenvector corresponding to its zero eigenvalue is a vector \( \mathbf{1} \) of all ones.

The sufficient condition of non-singularity becomes

\[
\omega \sum_{j=1}^{o} \begin{bmatrix}
d_{1,j} I \\
d_{2,j} I \\
\vdots \\
d_{c,j} I
\end{bmatrix} H^T(V^{(j)})^{-1} H \begin{bmatrix}
d_{1,j} I \\
d_{2,j} I \\
\vdots \\
d_{c,j} I
\end{bmatrix} \mathbf{1} \neq \mathbf{0}.
\]

(5.9)

We know that when \( H \) refers to the projection matrix of analysis state variables on the locations of observations, its entries are non-negative; when \( V \) is a diagonal
matrix where its diagonal entries are the variances of error observations, the inverse of these entries are positive. It follows that the condition is valid, and the coefficient matrix is positive definite. The conjugate gradient method can be applied to solve the resulting linear system.

In the next section, we will demonstrate that the performance of the proposed method is better than other testing methods.

5.3 Implementation

5.3.1 Reduction for the Matrix-Vector Multiplication with Sparse Characteristics

Now let us discuss a case when the number of arithmetic operation can be reduced by considering the structure of $\mathbf{M}$. We consider that the $\nabla_T \nabla x$, $\nabla_T \nabla y$, $\nabla_T \nabla z$ are very sparse, i.e., there are a lot of zeros in these matrices. In the meanwhile, $\mathbf{I}_x$, $\mathbf{I}_y$ and $\mathbf{I}_z$ are identity matrices so that they are sparse as well. As a result, the kronecker products, $\mathbf{M}_x$, $\mathbf{M}_y$, $\mathbf{M}_z$ are implied as sparse matrices. For instance, we employ first-order difference to form the gradient matrices $\nabla x$, $\nabla y$ and $\nabla z$. The number of arithmetic operation required to generate $\mathbf{M}$ and do the matrix-vector multiplication is of $O(18n)$. The cost is summarized in the following steps.

1. From the setting of gradient matrices, $\nabla x$, $\nabla y$, $\nabla z$ have the sparse structures like Figure 5.1(a), and $\nabla_T \nabla x$, $\nabla_T \nabla y$, $\nabla_T \nabla z$ are similar to Figure 5.1(b). The number of nonzero entrance on the first and last lines is 2 while the number is 3 on the other lines for $\nabla_{(x,y,z)} \nabla_{(x,y,z)}$. The differences among these three ones are the sizes of matrices and the total numbers of nonzero entrances.
2. After constructing the $\nabla_{\{x,y,z\}} \nabla_{\{x,y,z\}}$, do the kronecker product of $\nabla_{x} \nabla_{x} \otimes I_{y} \otimes I_{z}$, $I_{x} \otimes \nabla_{y} \nabla_{y} \otimes I_{z}$ and $I_{x} \otimes I_{y} \otimes \nabla_{z} \nabla_{z}$ to get $M_{\{x,y,z\}}$. In this process, $I_{\{x,y,z\}}$ are identity matrices, so the entrances in $M_{\{x,y,z\}}$ are mapped from $\nabla_{\{x,y,z\}} \nabla_{\{x,y,z\}}$. Hence, the values of entrances in $M_{\{x,y,z\}}$ could be generated point by point and still keep the sparse characteristics like Fig. (5.2). The generation cost of $M_{\{x,y,z\}}$ is reduced to $O(9n)$.

As shown in Fig. (5.2) for $M_{x}$, the first and last $n yn z$ lines only have 2 nonzero entrances and the other lines have 3 instead. This also happens in $M_{y}$ and $M_{z}$. The first and last $n z$ lines in each $(n y n z \times n y n z)$ block in $M_{y}$ have 2 nonzero entrances while other lines in this block have 3 and there are $n x$ same blocks in $M_{y}$. And each $(n z \times n z)$ block in $M_{z}$ is same as $\nabla_{z} \nabla_{z}$ and there are $n x n y$ same blocks in it.

3. The cost on matrix-vector multiplication between $(M + \omega H^{T} V^{-1} H)$ and the residual mainly concentrates on the multiplications of $M_{x}$, $M_{y}$ and $M_{z}$ with
the residual respectively. For $M_{\{x,y,z\}}$, they have 2 nonzero entrances in $n_y n_z$, $n_x n_z$, $n_x n_y$ lines and 3 in other lines respectively. The matrix-vector can be calculated with limited nonzero points and the cost is reduced to $O(9n)$.

4. Therefore, the whole operations on generating $M$ and matrix-vector multiplication in one iteration is reduced from the complete calculation $O(4n^2)$ to $O(18n)$. There would be a quite decrease when $n$ is large which is normally appeared in high dimensional cases and big data problems.

We called the above method as the reduction for the matrix-vector multiplication with sparse characteristics on the computation of gradient method.

\subsection*{5.3.2 Reduction for the Matrix-Vector Multiplication with Geometrical Feature}

After concluding the reduction for the matrix-vector multiplication with sparse characteristics, we figure out another angle to understand the matrix-vector multiplication based on the structure and elements of $M_{\{x,y,z\}}$. Additionally, it provides a further reduction on the matrix-vector and with no need to construct the $M$ to complete the process between $(M + \omega H^T V^{-1} H)$ and the residual. This would decreases the operation amount from $O(18n)$ of previous reduction to $O(6n)$. The cost is summarized in the following steps:

1. We noted that when applying the first order difference to form the gradient matrix, the $\nabla^T_{\{x,y,z\}} \nabla_{\{x,y,z\}}$ will be tridiagonal matrices and the entrances on main diagonal are 2 while the entrances on the line below the main diagonal or above the main diagonal will be equal to $-1$. This feature is also appeared on the $M_{\{x,y,z\}}$ in step 2 of reduction by using sparse characteristics. From this, we are aware that multiplying the $M_{\{x,y,z\}}$ on a vector(residual) is actually calculating the differences between the two layers or a middle layer with its neighboring two layers.

2. Here we take the multiplication of applying $M_x$ on a vector as an example to present the technique. Since the vector are compressed from $\mathbb{R}^3$, first, we will reconstruct its 3D state as in Figure (5.3).
Base on the principle of kronecker product, we put the first set of \( n_z \) variables in vector on \((0, 0, 0) \sim (0, 0, n_z)\) of the model in Figure 5.3 as the initial step. Then the start point moves to \((0, 1, 0)\) to put the second set of \( n_z \) variables in vector on \((0, 1, 0) \sim (0, 1, n_z)\). In a similar way, employing the first set of \( n_z n_y \) variables of vector on the 1st layer as Figure 5.4, then the second set of \( n_z n_y \) variables of vector on the 2nd layer. Continue to put the rest of sets of \( n_z n_y \) variables of vector until the \( n_x \)th layer is completed.

3. According to the sparse structure of \( M_x \), the first line of it is actually subtracting the value on \( (0, 0, 0) \) from the value on \( (1, 0, 0) \). And the second line
similarly subtracts the value on \((0, 0, 1)\) from the value on \((1, 0, 1)\). The same behavior is used to process the whole 1st layer from the first set of \(n_z n_y\) lines in \(M_x\). This means that the performance of the first set of \(n_z n_y\) lines multiplying the vector is the same as the performance of subtract the values from the 1st layer to the 2nd layer of the model in Figure 5.4. The same technique can be employed for the last set of \(n_z n_y\) variables in vector to subtract the \(n_x\) th layer from \((n_x - 1)\)th layer. For the rest of the sets of \(n_z n_y\) variables in vector, they are subtracting the values on the targeted layer from the values on the neighboring two layers. So the cost of getting the result on each layer is \(O(2n)\).

4. The same strategy is applied for the matrix-vector multiplication on \(M_y\) and \(M_z\) with the vector by distributing the values in the vector to the z-y-x order of the 3D model. The geometry meaning of the multiplication for \(M_y\) is calculating the differences between y layers while \(M_z\) is counting the differences between z layers. In the end, the multiplication of \(M\) with the residual combines all the differences from \(x\), \(y\) and \(z\) directions of the 3D model of the residual and the cost is reduced to \(O(6n)\) for the matrix-vector multiplication. Furthermore, we can skip the construction of the 3D model if we understand the principle of this technique and directly do the subtraction on the vector.

We call this method as the reduction for the matrix-vector multiplication with geometrical feature on the computation of gradient method. It reduces the operation from \(O(4n^2)\) of complete calculation, the \(O(18n)\) of making use of sparse characteristic to \(O(6n)\) now. As a summary, by making use of the sparsity of \(M\) (\(9n \ll 4n^2\), if \(n\) is large) and the meaning of the kronecker product, the cost of the matrix-vector multiplication is \(O(6n)\) per iteration.

5.4 Numerical Results

In this section, we will test the chemistry transport model to demonstrate the effectiveness of the proposed method in data assimilation. We also compare the performance of the proposed method with the 3DVar method.
To evaluate the prediction performance of different methods, we study the error of the analysis of state variables referring to the $k$-th observation station by the testing method without using the $k$-th observation station in data assimilation. The average error is then given by

$$\text{Error} = \frac{1}{N} \sum_{k=1}^{N} \left[ \frac{1}{N_k} \sum_{j=1}^{N_k} (a_{k,j} - y_{k,j})^2 \right],$$

(5.10)

where $N$ is the total number of observational stations, and $N_k$ is the total number of observations referring to the $k$-th observational station, $a_{k,j}$ is the $j$-th analysis state relevant to the $k$-th observation station and $y_{k,j}$ is the $j$-th observation relevant to the $k$-th observational station. The stopping criterion of the conjugate gradient method for solving linear systems in (5.2), (5.4) and (5.7) is that their residuals of the iterates computed by the conjugate gradient method must be less than $1e^{-7}$.

The Chemistry Transport Model solves mass balance equations in a three-dimensional domain for air quality forecast, see CMAQ [6]. This kind of model has been used in testing for data assimilation methods, see for instance [69, 19, 25]. In our experiment, we study air pollutant concentrations in the Hong Kong area. The computational domain involves $74 \times 74 \times 26$ (142,376) grid points. The species NO$_2$, O$_3$, SO$_2$, NO, CO, PM2.5 and PM10 are employed to test the performance of the proposed method and the 3DVar method. PM2.5 and PM10 are coupled observations. There are 33 pollutant species to be accumulated together to form PM2.5. For PM10, it includes PM2.5 and seven more pollutant species coupled together. In the Hong Kong air quality forecast environment, there are four, eleven, eight, eleven, eleven, nine and seven observational stations for PM2.5, NO$_2$, O$_3$, PM10, SO$_2$, NO and CO respectively.

In the proposed method, we need to set the values of $\omega$ in (5.3) and (5.5). Here we make use of the 1-hour chemistry transport simulation to test several sets of values: the first set is $\{0.001, 0.002, \ldots, 0.009\}$; the second set is 10 times of the first set $\{0.01, 0.02, \ldots, 0.09\}$; the third set is 100 times of the first set $\{0.1, 0.2, \ldots, 0.9\}$; by this analogy, the fourth set is $\{1, 2, \ldots, 9\}$, the fifth set is $\{10, 20, \ldots, 90\}$ and the sixth set is $\{100, 200, \ldots, 900\}$. With the value $\{1000\}$, there are totally 55 values and choosing the value of $\omega$ such that the ratio between $\text{Error}_{\text{proposed}}$ and $\text{Error}_{\text{cmaq}}$ is the smallest among the testing values. The ratio can be interpreted
as the largest improvement on the prediction by the proposed method compared with that by CMAQ. We note that when $\omega$ is small (or large), the observation data is less (or more) importantly employed. For the 3DVar method, the background and observation covariance matrices are estimated by the former 24-hour chemistry transport simulation and 1-week observation data respectively. Since the size of the background covariance matrix is very large, the estimated covariance matrix would be singular. Here we employ the estimate of the variance of each state only to construct a diagonal background covariance matrix in (5.2).

5.4.1 Single Species

The regularization coefficient $\omega$

The regularization coefficient $\omega$ presents the importance of the difference between the observations and the projected analysis state with respect to the difference between the forecast state and analysis state. Large $\omega$ implies that the truth is more closer to the observations compared to the forecast state so that the analysis state will try to approach these reliable observations. On the contrary, the small $\omega$ indicates the less confidence shown on those observations. In Figure 5.5, we demonstrate the changes on daily average of $\omega$ for the single species - NO$_2$, O$_3$, SO$_2$, NO and CO.
Figure 5.5: The changes on daily average of $\omega$ for single species in January

Concluded from the figure, first of all, the parameter $\omega$ for 5 single species are all bigger than 100 and presents as a large number. Which means the observations for these 5 species are more reliable compared to the their concentrations within the forecast. And, the parameter $\omega$ are in the value interval $[300,450]$ for NO$_2$, $[300,700]$ for O$_3$, $[200,500]$ for SO$_2$, $[200,400]$ for NO and $[100,500]$ for CO, respectively.

Data assimilation results with proposed method

In Figure 5.6, we show the computed errors in (5.10) for 1 month for the species
NO$_2$, O$_3$, SO$_2$, NO and CO, respectively. The chosen values of $\omega$ are selected from the previous part for the proposed method. In the figures, the chemistry transport model forecast is used for the baseline comparison.

Figure 5.6: The daily average errors for the prediction of single species by different methods.

We can see that the errors of the proposed method are smaller than those of the baseline method and the 3DVar method in the most of cases, the case of NO$_2$, O$_3$, SO$_2$ and CO. And there are two special situations occurred in the case of SO$_2$,
NO and CO which we will explain in the follow. The first question is why the performance of 3DVar is better than that of our proposed method on NO. It is due to the accuracy of description on the background covariance, at this time, the estimation of using a diagonal matrix on the background covariance is better than the estimation of using a full description matrix on the background covariance. Thus, our proposed method doesn’t reach its best performance because of the limitation on the estimation of background covariance within our method. It could be fixed by replacing the background covariance in our method with a diagonal one or developing a more accurate description on the background covariance of our method. The second problem is also caused by the background covariance. The diagonalized and biased background covariance leads to a bad result on the 3DVar method of the SO\textsubscript{2} and CO. Further, from the performance of CO, there exists a large and coincident improvement which means the uniform noises are contained in the global model state or within the model. And this is one of advantages of our proposed method - reducing the global uniform noises within the model state.

The average errors and improvements over 1 month are show in the Table 5.1 for CMAQ, 3DVar and the proposed method respectively.

Table 5.1: The average errors and improvements of CCTM with proposed method and other existing methods for single species in January

<table>
<thead>
<tr>
<th></th>
<th>CCTM</th>
<th>CCTM + 3DVar</th>
<th>Improvement</th>
<th>CCTM + Proposed Method</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO\textsubscript{2}</td>
<td>0.0233</td>
<td>0.0202</td>
<td>13.30%</td>
<td>0.0122</td>
<td>47.64%</td>
</tr>
<tr>
<td>O\textsubscript{3}</td>
<td>0.0148</td>
<td>0.0128</td>
<td>13.51%</td>
<td>0.0075</td>
<td>49.32%</td>
</tr>
<tr>
<td>SO\textsubscript{2}</td>
<td>0.0030</td>
<td>0.0036</td>
<td>-20.00%</td>
<td>0.0024</td>
<td>20.00%</td>
</tr>
<tr>
<td>NO</td>
<td>0.0396</td>
<td>0.0157</td>
<td>60.35%</td>
<td>0.0260</td>
<td>34.34%</td>
</tr>
<tr>
<td>CO</td>
<td>0.3623</td>
<td>0.5368</td>
<td>-48.16%</td>
<td>0.0410</td>
<td>88.68%</td>
</tr>
</tbody>
</table>

We can see that our proposed method has large improvements for singles species compared to the 3DVar in the most of cases.

We also find that the computational time required by the 3DVar method is about 0.03 second for an 1-hour data assimilation calculation and the computational time required by the proposed method is about 3 seconds for an 1-hour analysis calculation. The proposed method obtains quite a good prediction accuracy subject to reasonable
5.4.2 Multiple Species

The experimental setting is similarly set up for the coupled observations of PM2.5 and PM10. We make use of the 1-hour chemistry transport simulation to choose $\omega$ from the 55 values such that the sum of two ratios ($\text{Error}_{\text{proposed}}$ and $\text{Error}_{\text{cmaq}}$) for PM2.5 and PM10 is the smallest (since PM2.5 contains 33 subspecies and PM10 has 40, we use weight to evaluate the performance and the weight for PM2.5 is $\frac{1}{33}$ while the weight for PM10 is $\frac{1}{40}$).

![Figure 5.7: The changes on daily average of $\omega$ for multiple species in January](image)

From Figure 5.7, the parameter $\omega$ for multiple species are in the value interval $[100,500]$ and also presents as a larger number. It has the similar feature as single species that the observations are more reliable compared to the forecast.
Figure 5.8: The daily average errors for the prediction of single species by different methods

In Figures 5.8, we show that the computational results for the errors of the weighted combination of PM2.5 and PM10. We can see that the computed errors of the proposed method are smaller than those of the baseline method and the 3DVar method. The average errors of PM2.5 and PM10, and the improvements for their weighted combination over one month are shown in Table 5.2.

Table 5.2: The average errors and improvements of CCTM with proposed method and other existing methods for multiple species in January

<table>
<thead>
<tr>
<th></th>
<th>CCTM</th>
<th>CCTM + 3DVar</th>
<th>Improvement</th>
<th>CCTM + Proposed Method</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM2.5</td>
<td>23.5017</td>
<td>8.4110</td>
<td>34.23%</td>
<td>10.897</td>
<td>49.83%</td>
</tr>
<tr>
<td>PM10</td>
<td>33.5837</td>
<td>30.6298</td>
<td></td>
<td>17.935</td>
<td></td>
</tr>
</tbody>
</table>

It implies that our proposed method has larger improvement compared to the other existing data assimilation methods. However, the computational time required by the proposed method for one-hour analysis is 348.0 seconds which is higher (1.0 seconds) by using the 3DVar method.
Chapter 6

Summary and Future Research

6.1 Summary

We have introduced and developed several numerical algorithms for data assimilation.

In Chapter 2 and Chapter 3, we have introduced the concept of air quality simulation and the principle of data assimilation method. The development on air quality simulation is caused by the need for protection. With more accurate air quality model simulation, people are able to protect the environment, animals and human lives and health. The establishment of the concept of data assimilation is due to the impossibility and expensiveness of setting up the observations in large numbers. That demands those techniques using limited number of observations to estimate the change on the global state with high accuracy and efficiency.

According to the concept of air quality simulation and data assimilation, we can find that the air quality model holds vast number of variables and are not easy to increase the accuracy from the chemical and physical perspectives while the data assimilation is an effective mathematic method for the “Big Data” problem and can largely increase the accuracy. In Chapter 4, we assume that linking them together will improve the performance of our forecast and in the meanwhile, it is not too time consuming. The workflow which we have built up shows that the combination is very successful and the data assimilation can be incorporated with the chemical transport model in an automatic way. Besides, we have do some improvement on the localized ensemble Kalman filter and present the matrix computational methods for
sparse observations. Consequently, the computational cost of LEnKF can be reduced very significantly. This method also shows its superiority to handling the LEnKF data assimilation for multiple species. Our experimental results for real air quality forecast data have shown that the efficiency and the effectiveness of the proposed sparse matrix computation method.

The improvement on localized ensemble Kalman filter is not enough. So, in Chapter 5, we have studied and developed an optimization method for solving data assimilation problem by using the gradients of the forecast of state variables. The resulting method is required to solve a linear system where the coefficient matrix involves the sum of the second-order difference matrix and the sparse matrix which refers to the observation locations. We apply the conjugate gradient method to solve this kind of linear systems and use the fast matrix-vector multiplication to guarantee the process is working in an efficient way. Numerical examples in air quality have shown that the performance of the proposed method is better than that of other testing methods and can very promingly remove the uniform noises within the model state.

6.2 Future Research

Our research opens up numerous avenues for future work. Some of the major ones are as follows:

**Parameter estimation**
In the future research work, we will consider how to estimate the parameters from the background state and observation. In this way, first, the parameters will be estimated very accurately, not like the rough parameter pairs we have set in the thesis. Secondly, much time can be saved for searching the best parameter pairs. Thirdly, the estimation based on the forecast and observation themselves will lead to the improvement on the theory of data assimilation.

**Covariance studying**
An accurate description of the background and observation covariances means a lot to the improvements of data assimilation. Actually, the background covariance is not precise because the matrix is rank deficiency. In some sense, the covariance
inflation and localization are the way to pursue a more accurate approximation on
the covariance. For the next step, we can maintain the part of covariance with stable
variables and improve the part of covariance with unstable variables and use the
parameter to weight so that, the approximation is assumed to be closer to the truth.

**Fast solving on linear system**

We could present a fast solution method for solving large linear systems, especially for
linear systems arising from coupled observations. For example, we may consider using
the preconditioning strategy to solve large linear systems. As for the preconditioning
strategy, there will be many methods to test and much and more research work to
do, like the Jacobi preconditioning, Incomplete Cholesky factorization, Incomplete
LU factorization and so on.

**Nonuniform noise**

The optimization approach uses the information of the gradient in the concentra-
tion of pollutant species, as the gradient presents the differences between the layers.
The approach tries to make ensure that the analysis solution has similar differences
between the layers as the forecast while keeping the value of the projected analysis
solution the same as the observation. This principle will succeed in removing the
uniform noises because it considers the information of first order. Further, we may
consider the information of second order and are able to remove the nonuniform noises
within the forecast state.
Bibliography


128


Academic qualifications of the thesis author, Mr. ZHU Zhaochen:

- Received the degree of Bachelor of Engineering Mechanics from Zhejiang University, July 2013.

August 2017