

**A Heterogeneous Non-Overlapping Domain Decomposition Explicit
Finite Volume Method for Real-Time Estimation of 3D Advection-
Diffusion Fields with a Sensing Aerial Vehicle**

by

Xin Tian

A Dissertation

Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Aerospace Engineering

May 2021

APPROVED:

Nikolaos A. Gatsonis, Advisor
Professor and Department Head, Aerospace Engineering Department

Michael A. Demetriou, Committee Member
Professor, Aerospace Engineering Department

John Burns, Committee Member
Professor, Department of Mathematics, Virginia Tech

Jagannath Jayachandran, Graduate Committee Representative
Assistant Professor, Aerospace Engineering Department

This page intentionally left blank

Abstract

A heterogeneous, non-overlapping domain decomposition, explicit, finite volume method (HT-NODDE-FVM) is developed to solve numerically a 3D advection-diffusion process-state hybrid estimator. The HT-NODDE-FVM is applied to real-time estimation problems of 3D unsteady advection-diffusion plume fields produced by stationary and moving sources. The field measurements are taken by a sensor onboard a guided sensing aerial vehicle (SAV). The hybrid Luenberger-naive estimator uses those field measurements to compute the plume field in real-time and guide the SAV to locations that provide optimal information to the hybrid estimator. A structured and uniform grid is used to divide the computational domain of interest into multiple non-overlapping subdomains. In the subdomain where the SAV resides, a Luenberger estimator in the form of a 3D advection-diffusion partial differential equation is used to estimate the plume field. In the remainder of the subdomains, a naive observer of a similar form is used. The transmission conditions are used on the interfaces between adjacent subdomains for data communication. The spatial discretization of the hybrid Luenberger-naive estimator is conducted by the HT-NODDE-FVM with Total Variation Diminishing (TVD). Continuity and flux balance transmission conditions are enforced at the interfaces of adjacent subdomains when conducting the FVM-TVD discretization. The resulting semi-discrete equations are integrated by a 4th order Runge-Kutta method. OpenMP parallel paradigm is implemented to parallelize the HT-NODDE-FVM estimator. The verification and error analysis of the NODDE-FVM are performed with two benchmark tests. One is the 3D advection of different initial density distributions, the other is the 3D advection-diffusion of instantaneous gaseous releases under constant wind speed and eddy diffusivities for a range of Peclet numbers. The verification and error analysis of the HT-NODDE-FVM hybrid estimator are also conducted on an instantaneous release by a stationary source in a large domain with constant atmospheric properties. The impact of grid resolution, sensor model, estimation gain, and numerical data, on the L^1 , L^2 , and L^∞ norms of the estimation

error are examined by those test cases. Parallelization efficiency analysis of the OpenMP implementation of the hybrid estimator is also presented. Finally, the hybrid estimator and the HT-NODDE-FVM are applied to estimate the gaseous plumes released from stationary and moving sources in a km-scale computational domain under realistic atmospheric conditions and SAV parameters. Real-time estimation analysis is also conducted by comparing the wall clock time of completing an iteration over all the subdomains with the maximum allowable numerical time step for the temporal integration. The simulation results show that the hybrid estimator and the HT-NODDE-FVM can achieve real-time estimation of the advection-diffusion field in very fine grid settings.

Acknowledgements

I would like to gratefully acknowledge the direction and support from my advisor Professor Nikolaos Gatsonis and co-advisor Professor Michael Demetriou during my Ph.D. studies at Worcester Polytechnic Institute. I also would like to thank Professor John Burns and Professor Jagannath Jayachandran for serving as the committee members for my dissertation.

I am grateful to my family for supporting me on pursuing my Ph.D. degree, especially to my wife Tingting Li whose help and support made this work possible. I also would like to thank my daughter who brought to me a lot of laughter during my study in the U.S.

Finally, I am grateful for the financial support provided by the Aerospace Engineering Department at WPI through a Research Assistantship and a Teaching Assistantship.

Support by the DARPA Lagrange Program is greatly acknowledged.

Contents

Abstract	3
Acknowledgements	5
Contents	6
List of Figures	8
List of Tables	10
Chapter 1 Introduction	11
1.1 Background and Literature Review	12
1.2 Research Goals, Objectives, and Approach	16
Chapter 2 Process Model, Process-State Hybrid Estimator, and Sensing Aerial Vehicle Dynamics and Guidance Model	20
2.1 Process Model	20
2.2 Process-State Hybrid Estimator	33
2.3 SAV Guidance Model and Sensor Model	36
2.3.1 Modified Lyapunov Guidance Law	36
2.3.2 Lyapunov Guidance Law with SAV Dynamical Model	37
2.3.3 Sensor Model	41
Chapter 3 Heterogeneous Non-Overlapping Domain Decomposition Explicit Finite Volume Method, Verification and Error Analysis, Parallelization Efficiency	42
3.1 Computational Domain Decomposition and Parallelization	42
3.2 The HT-NODDE-FVM for the Hybrid Estimator	46
3.2.1 FVM-TVD Spatial Discretization	46
3.2.2 Runge-Kutta Temporal Integration	73
3.2.3 Numerical Stability	74
3.3 Verification and Error Analysis for the HT-NODDE-FVM	78
3.3.1 Non-Dimensional 3D Advection Equation	78
3.3.2 Non-Dimensional 3D Advection-Diffusion Equation	80
3.3.3 Verification of the Hybrid Estimator with HT-NODDE-FVM	82

3.3.4	Error Analysis of the Hybrid Estimator with HT-NODDE-FVM	85
3.4	Parallelization Efficiency Analysis	87
Chapter 4	Applications of the HT-NODDE-FVM to Large Scale Simulations	89
4.1	Instantaneous Gas Release	90
4.1.1	Modified Lyapunov Guidance Law Without SAV Dynamical Model, Slow and Fast SAVs	90
4.1.2	Lyapunov Guidance Law with SAV Dynamical Model, Slow SAV.....	92
4.2	Long-Pulse Gas Release.....	93
4.2.1	Modified Lyapunov Guidance Law Without SAV Dynamical Model, Slow and Fast SAVs	93
4.2.2	Lyapunov Guidance Law with SAV Dynamical Model, Slow SAV.....	95
4.3	Real-Time Estimation Analysis	95
Chapter 5	Summary, Conclusions, and Recommendations.....	97
5.1	Summary and Conclusions.....	97
5.2	Recommendations for Future Work.....	100
References	102

List of Figures

Figure 1.1 An SAV is estimating in real-time the gaseous plume released by an airplane (intruder) and also tracking it..... 11

Figure 1.2 The physical domain of the hybrid estimator model. The Luenberger estimator applies in the shaded region and the naive estimator elsewhere. 12

Figure 2.1 The process model describing a moving source releasing trace species with rate u (kg/s) in the lower atmosphere..... 33

Figure 2.2 Free-body diagram of the forces acting on the SAV in the climbing coordinate turn. 38

Figure 3.1 A flow chart showing the major elements of the HT-NODDE-FVM scheme. 43

Figure 3.2 (a) An example of the domain decomposition discretization (bold black lines indicate the locations of the subdomain interfaces). The Luenberger estimator models the subdomain with blue boundary lines where the SAV resides. All the remainder subdomains are modeled by the naive estimator. (b) The cells are shown in 2D for better visualization. The shaded cells are needed to calculate fluxes on subdomain interfaces when applying transmission conditions by using FVM-TVD scheme..... 46

Figure 3.3 An arbitrary finite volume Ω_{ijklm} in subdomain m used in evaluation of advective and diffusive fluxes..... 47

Figure 3.4 Finite volumes at west boundary with (a) Dirichlet boundary condition and (b) Neumann boundary condition (shown in 2D)..... 58

Figure 3.5 $L1$ norm of the error between NODDE-FVM and analytical solutions of the advection PDE as a function of grid resolution for conditions shown in Table 3.1. (a) $L1$ norm calculated by Eq. (3.135); (b) $L1$ norm calculated by Eq. (3.136)..... 80

Figure 3.6 $L1$ norm of the error between NODDE-FVM and analytical solutions of the advection-diffusion PDE as a function of grid resolution and Peclet number shown in Table 3.2. 82

Figure 3.7 Benchmark case used for verification and error analysis of the HT-NODDE-FVM estimator. A stationary source releases instantaneously a gas in a domain with uniform atmospheric parameters and constant wind in the X -direction. The SAV performs real-time estimation of the plume concentration. 83

Figure 3.8 Verification of the HT-NODDE-FVM estimator. $L2$ norm of the estimation error from Eq. (3.136) as a function of estimation gain for a coarse and fine grid. Sensor data are taken from the analytical solution Eq. (3.139). 85

Figure 3.9 Error analysis of the HT-NODDE-FVM estimator model. (a) $L1$, (b) $L2$, (c) $L\infty$ norm of error evaluated between estimated concentration and numerical sensor data (process-model concentration obtained with HT-NODDE-FVM)..... 86

Figure 3.10 Error analysis of the HT-NODDE-FVM estimator model. (a) $L1$, (b) $L2$, (c) $L\infty$ norm of the error evaluated between estimated concentration and true sensor data (analytical concentration). 87

Figure 3.11 Speedup (a) and parallel efficiency (b) of the NODDE-FVM-TVD-RK simulation of the instantaneous gas release in the atmosphere as a function of the number of OpenMP threads. 88

Figure 4.1 The simulation results of the instantaneous gas release for slow SAV (Aerosonde) guided by the modified Lyapunov guidance law without SAV dynamical model at time 480 s. 91

Figure 4.2 The simulation results of the instantaneous gas release for fast SAV (Sky Warrior) guided by the modified Lyapunov guidance law without SAV dynamical model at time 480 s. 91

Figure 4.3 The simulation results of the instantaneous gas release for slow SAV (Aerosonde) guided by the Lyapunov guidance law with SAV dynamical model at time 480 s. 92

Figure 4.4 An intruder is releasing contaminants over a city and an SAV is estimating the concentration of the contaminants and tracking the intruder..... 93

Figure 4.5 The simulation results of the long-pulse gas release for slow SAV (Aerosonde) guided by the modified Lyapunov guidance law without SAV dynamical model at time 1300 s. 94

Figure 4.6 The simulation results of the long-pulse gas release for fast SAV (Sky Warrior) guided by the modified Lyapunov guidance law without SAV dynamical model at time 1300 s. 94

Figure 4.7 The simulation results of the long-pulse gas release for slow SAV (Aerosonde) guided by the Lyapunov guidance law with SAV dynamical model at time 600 s. 95

Figure 4.8 Real-time estimation analysis as functions of number of subdomains (threads) and number of finite volumes (resolution). 96

List of Tables

Table 3.1 3D pure advection cases and results	78
Table 3.2 3D advection and diffusion cases and results	82
Table 3.3 Computational inputs for benchmark tests of the estimator	84
Table 3.4 SAV parameters and estimator inputs	84
Table 3.5 Grid resolutions and corresponding optimal gains used for error analysis for the estimator model in benchmark cases	86

Chapter 1 Introduction

The problem of interest in this work arises from the need to estimate in real-time the concentration profile of a plume generated from a stationary or moving aerial source (intruder). The estimation is provided by a sensing aerial vehicle (SAV) carrying a concentration sensor. The problem is depicted in Figure 1.1 and has a wide range of practical applications. Such dynamic sensing platforms with real-time, data-driven, state estimators can provide the spatial resolution of systems described by various partial differential equations (PDEs) including the unsteady advection-diffusion equation considered in this work. These state estimators use data from static or mobile sensors and, in order to be real-time implementable in large-scale applications, require fast and robust computational algorithms.



Credit of background city picture: britannica.com

Figure 1.1 An SAV is estimating in real-time the gaseous plume released by an airplane (intruder) and also tracking it.

The process-state in our case, i.e. the concentration field of the trace gaseous contaminant, is modeled by the unsteady advection-diffusion PDE. The hybrid estimator is implemented onboard a dynamic sensing platform, a sensing aerial vehicle in our case, which obtains a single pointwise concentration measurement in the field and provides in real-time the spatial resolution of the field in the entire domain, while repositions the sensing platform to locations that optimize the performance of the state estimation. The hybrid process-state estimator is modelled by a heterogeneous PDE of similar PDE form, and accounts for different observers in different regions of the spatial domain as shown in Figure 1.2.

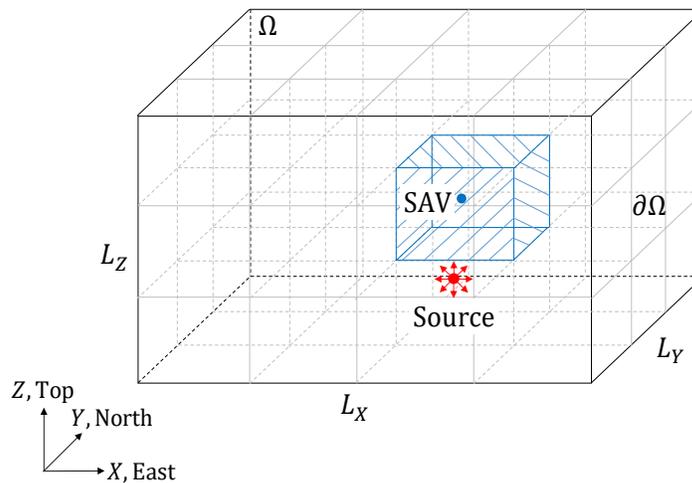


Figure 1.2 The physical domain of the hybrid estimator model. The Luenberger estimator applies in the shaded region and the naive estimator elsewhere.

The goal in this work is to develop a numerical method for the hybrid estimator based on a heterogeneous domain decomposition method (DDM). The new method must be executable in real-time and be applicable to km-scale computational domains with realistic SAV and atmospheric conditions.

1.1 Background and Literature Review

Previous work at WPI addresses various aspects of the real-time estimation problem. *Demetriou et al.* [2013 and references therein] modeled the process (plume) by a 2D advection-diffusion PDE. The process-state estimator is based on a Luenberger observer and its numerical solution is based on an adaptive, multigrid, Finite Volume Method (FVM) with upwinding. The SAV

dynamics is modeled in 2D and the guidance is derived with a Lyapunov redesign method which provides the control inputs (torques) to the SAV. *Egorova et al.* [2016 and references therein] modeled the process-state (plume) with the 3D advection-diffusion PDE and the process-state estimator is based on a Luenberger observer. The numerical solution of the 3D estimator is based on a FVM with TVD using locally adapted grids centered at the location of the SAV. The SAV is modeled as a point mass with a sixth degree of freedom nonlinear dynamic model. The Lyapunov-redesign method provides the control signal (velocities) to the SAV. *Hu and Demetriou* [2019] presented a hybrid process-state estimator for the 2D advection-diffusion equation along with a guidance law of a point-mass moving sensor. The 2D hybrid estimator applies a Luenberger model in the subdomain where the SAV resides and a naive estimator in the remainder of the subdomains. *Gatsonis et al.* [2020] expanded the process model and the process-state hybrid estimator from 2D to 3D. The numerical solution of the hybrid estimator is accomplished by a new heterogeneous non-overlapping domain decomposition explicit (HT-NODDE) method in order to achieve real-time computation.

Domain decomposition methods (DDMs) are used to solve linear or nonlinear systems of equations arising from the discretization of partial differential equations (PDEs). By decomposing the domain (or a data structure) into subdomains (or into data pieces) and sending them to different CPUs or threads, DDMs accomplish fast computation and ability to handle large scale problems. DDMs can also be used as preconditioners for Krylov subspace method based linear system solvers [*Smith et al.*, 2004]. In homogenous DDMs the entire domain and its subdomains are modeled by one type of PDE; in heterogeneous DDMs different PDEs apply to parts of the domain. DDMs are also characterized on whether the adjacent subdomains are overlapping (ODDM) or non-overlapping (NODDM). In the former case, the overlapped region is used to transfer data between the adjacent subdomains. Common ODDMs are classical Schwarz methods and multilevel Schwarz methods [*Smith et al.*, 2004]. In NODDMs, the data transfer or communication between the adjacent subdomains involves proper transmission conditions at the interface, which can be of Dirichlet-Neumann (D-N), Neumann-Neumann (N-N), and the Robin-Robin (R-R) type [*Rivera et al.*, 2003].

For homogeneous DDMs, one PDE is solved for all subdomains and most implementations are of the NODDM type [*Lube et al.*, 1998; *Achdou and Nataf*, 1997; *Rivera et al.*, 2003; *Wang et*

al., 2005; *Zheng et al.*, 2008; *Acebron et al.*, 2009; *Stolk*, 2013; *Kumar and Kumar*, 2017]. In heterogeneous DDMs, which is of interest in this work, different PDEs apply to various regions of the domain and their subdomains. The most common heterogeneous problems involve an advection-dominated region coupled with an advection-diffusion dominated region. *Gastaldi et al.* [1990] developed a NODDM for a steady advection and advection-diffusion coupled problem. Their iterative algorithm uses spectral Chebyshev collocation discretization. The transmission conditions enforce continuity on the inflow interface pointing to the advection subdomain and enforce flux balance on the entire interface. *Quarteroni et al.* [1992] implemented the method developed by *Gastaldi et al.* [1990] on a steady boundary layer problem. *Quarteroni et al.* [1998] developed a NODDM for unsteady inviscid/viscous coupled problems based on the Euler/Navier-Stokes equations, using FVM and a standard explicit four-stage integration scheme. The transmission conditions enforce equality of the inviscid advective flux with the combined viscous advective and diffusive fluxes; in addition, they enforce continuity of density on the inflow interface pointing to the inviscid subdomain. *Quarteroni et al.* [1998] applied their method to a quasi-one-dimensional nozzle flow, laminar flow over a flat plate, compressible flow over an airfoil, and turbulent flow over a RAE 2822 airfoil. *Gander et al.* [2007] developed a NODDM for an unsteady advection/advection-diffusion coupled problem. They implemented the same transmission conditions as in *Quarteroni et al.* [1998]. Their method involved a rapidly converging iteration algorithm which lead to an optimized Schwarz waveform relaxation method. *Halpern et al.* [2012] developed a heterogeneous NODDM based on Schwarz waveform relaxation algorithm for solving advection-diffusion-reaction problems with strong discontinuities. The Ventcell transmission conditions were used to achieve data communication between subdomains. Spatial discretization was performed with the generalized mortar finite elements and temporal discretization with the discontinuous Galerkin. This method was implemented on an advection-diffusion-reaction problem with different diffusivities in two non-overlapping subdomains and with the advection velocity normal to the interface in one subdomain and tangential to the interface in the other subdomain. *Li et al.* [2015] developed a NODDM to solve a coupled kinetic-heat system where a transport equation was solved in the kinetic region and a diffusion equation was solved in the fluid region. First the kinetic problem was solved with the boundary conditions on the interface given by a half-space problem with the Albedo operator. Then the results on the interface were used to solve the diffusion problem.

Birgler et al. [2018] developed a NODDM to solve a coupled problem of non-isothermal compositional Darcy flow and a RANS gas flow. The Darcy problem was solved by a two-point flux approximation scheme and the RANS model was solved by a staggered Marker and Cell scheme. Time was advanced by an implicit Euler scheme. The resulting nonlinear systems were solved by a Newton algorithm and a Quasi Newton algorithm, respectively. The R-R type transmission conditions were used on the interface to transfer data between the free flow and the flow in a porous medium.

In this work we present a new theoretical and numerical framework for the process-state hybrid estimator based on domain decomposition to achieve real-time implementation of the algorithms. The new theoretical framework involves a formal derivation of the process model equation for the contaminant concentration. The new process-state estimator is hybrid and involves a Luenberger estimator which applies in a region surrounding the SAV and a naive estimator in the rest of the domain. The theoretical issues of such hybrid estimators are addressed in detail by *Hu et al.* [2020] and which examine the sparsity of the filter kernels. Enforcing sparsity of the filter kernels ensures the nullity of the filter kernel in the outer subdomain, thus guaranteeing the estimator in the outer subdomain is rendered naive. Such manipulation of kernels was examined in *Akhtar et al.* [2015] in the context of sensor selection for the dual problem of control feedback kernels. The use of mobile sensors for filtering of partial differential equations is presented by *Burns and Rautenberg* [2015].

A new numerical framework is also presented in this work, based on a non-overlapping fully explicit heterogeneous domain decomposition (HT-NODDE) approach (Gatsonis et al, 2020). The spatial discretization in NODDE follows an FVM with TVD approach and the time integration of the semi-discrete equations is obtained with a Runge-Kutta algorithm, resulting in a method referred here and on as HT-NODDE-FVM. Transmission conditions are enforced through a barrier synchronization. The guidance of the SAV which is modeled as a point mass is based on a Lyapunov redesign and the control inputs to the SAV are based following recent work by *Tian et al.* [2020]. The resulting HT-NODDE-FVM is implemented using Open MP to allow use of readily available multi-core systems. An extensive set of validation and verification benchmark tests of the HT-NODDE-FVM and hybrid estimator are presented. Simulations using km-scale domains, SAVs with parameters from existing UAVs, and realistic ambient

atmospheric wind conditions demonstrate that the HT-NODDE-FVM estimator achieves real-time computation of the plume. The method can be applied to any application requiring heterogeneous domain decomposition.

1.2 Research Goals, Objectives, and Approach

The primary goal of this work is to estimate in real-time the concentration profile of a stationary or moving gas source by using a hybrid estimator with the heterogeneous non-overlapping domain decomposition explicit finite volume method (HT-NODDE-FVM). To achieve this goal, mathematical models and numerical models need to be derived and the developed method needs to be implemented on realistic problems.

The objectives and approaches used to accomplish the goal are as follows:

1. Derive the process model, the hybrid estimator model, and the sensor model.
 - a. Describe the atmosphere and the released gas with the Navier-Stokes equations of a multi-component system together with the continuity equation of single species; use the Boussinesq approximation for the lower atmospheric layer, use the Reynolds-averaged equations to account for turbulent effects, assume the released species is a single trace species and is in the diffusion regime, assume the molecular diffusion fluxes and their gradients are negligible compared with the turbulent ones, and assume the principal axes of eddy diffusivity tensor coincide with the coordinate axes, then we arrive at the 3D advection-diffusion equation of a single released trace species in the atmosphere.
 - b. The hybrid estimator uses a Luenberger observer in the domain where the SAV resides and a naive observer in the region outside.
 - c. Use a sensor model which is based on Lagrangian sensing technique.
 - d. Use two different approaches to guide the SAV. One is the modified Lyapunov guidance law and the other is the Lyapunov guidance law with six degree-of-freedom fixed-wing aircraft dynamical model.
2. Develop the HT-NODDE-FVM with TVD and Runge-Kutta (RK) method for the hybrid estimator model.

- a. Discretize the computational domain with a structured grid and divide the entire domain into multiple subdomains with each subdomain having same number of cells for optimal load balancing. The Luenberger estimator model is solved in the subdomain where the sensor resides while the naive estimator model is solved in the rest of subdomains.
 - b. For each subdomain, use a second order FVM with TVD scheme for the spatial discretization.
 - c. Impose transmission conditions on the interfaces between adjacent subdomains using second order discretization that results in a non-overlapping explicit DDM.
 - d. Use an explicit 4th order Runge-Kutta method for temporal integration of the semi-discrete equations.
 - e. Allocate each subdomain to a distinct CPU thread.
 - f. Solve the estimator model equations in all the subdomains using an OpenMP parallel implementation.
3. Verification and error analysis of the NODDE-FVM with TVD-RK algorithm.
- a. Apply the method on homogeneous non-dimensional 3D advection and advection-diffusion equations and compare the numerical solution with analytical solutions. Evaluate the order of accuracy by performing grid sensitivity analysis. For the non-dimensional 3D advection equation, four initial density configurations are used for the simulations including a continuous Gaussian density distribution, two piecewise Gaussian distributions, and a cubical density distribution. The order of accuracy is calculated by using the L^1 norm of the error between numerical and analytical solutions for each initial density distribution case. For the non-dimensional 3D advection-diffusion equation, three different Peclet numbers are used and cover diffusion dominated to advection dominated flows. The initial density distribution used in the simulations follows the analytical solution at certain time instance. The L^1 norm of the error between numerical and analytical solutions for each Peclet number case is used to calculate the order of accuracy.
 - b. Apply the method on the heterogenous hybrid estimator to study the effects of estimation gain and grid resolution on estimation results. The L^2 norm of the

estimation error as functions of estimation gain and grid resolution is plotted to study those effects.

4. Perform parallelization efficiency analysis for the OpenMP implementation of the NODDE-FVM with TVD-RK method.
 - a. Study the speedup and efficiency by recording the average elapsed wall-clock time of completing one single time level for different tests with different number of threads.
5. Implement the HT-NODDE-FVM with TVD-RK method on realistic estimation problems.
 - a. Use city-scale domains of km-size and realistic atmospheric conditions for the ambient wind profiles and diffusivities.
 - b. Generate sensor data by solving the process model in the same domain size used for the hybrid estimator.
 - c. Use conditions for SAVs that correspond to existing UAVs and aerial intruders.
 - d. Perform simulations for cases of instantaneous gaseous release and long-pulse gaseous release. Compare the estimated concentration profile with the process model concentration profile and track the source.
 - e. Evaluate the real-time computational ability of the HT-NODDE-FVM with TVD-RK method. The study of real-time computation or estimation is conducted by comparing the wall clock time of completing an iteration over all the subdomains Δt_{est} with the maximum numerical time step Δt_{max} used for the temporal integration. To achieve real-time estimation, it is required that the ratio of $\Delta t_{est}/\Delta t_{max}$ must be less than one. The real-time estimation analysis is conducted by varying the grid resolution as well as the number of subdomains (or threads) for both the process model and the hybrid estimator. The performance of the HT-NODDE-FVM with TVD-RK method is shown by plotting the ratio of $\Delta t_{est}/\Delta t_{max}$ as functions of the number of threads (or subdomains) and number of cells.

This thesis is organized as follows: Chapter 2 presents the mathematical models for process model, hybrid estimator model, SAV guidance model, and sensor model. Chapter 3 addresses the development of the HT-NODDE-FVM with TVD-RK algorithm including the computational

domain decomposition and parallelization, the HT-NODDE-FVM discretization for the hybrid estimator, the Runge-Kutta temporal integration, numerical stability analysis, verification and error analysis for the HT-NODDE-FVM, and the parallelization efficiency analysis. Chapter 4 shows the implementation of the hybrid estimator and the HT-NODDE-FVM method on realistic simulations and demonstrates the capability of achieving real-time estimation of the developed method. Finally, Chapter 5 presents the conclusion and recommendations for future work.

Chapter 2 Process Model, Process-State Hybrid Estimator, and Sensing Aerial Vehicle Dynamics and Guidance Model

This chapter presents the mathematical model for the process of a gas released by a stationary or moving point source at the lower atmosphere and the mathematical model for a 3D process-state hybrid estimator. It also briefly introduces the SAV guidance and dynamical model, as well as the sensor model.

2.1 Process Model

The process model used in this work is based on the unsteady advection-diffusion equation of a trace species in the atmosphere with source terms. It provides the basis for the estimator model as well as the numerical data used by the estimator model in simulations. In a real application these numerical data will be replaced by measurements. The detailed derivation of the unsteady advection-diffusion equation is provided here for completeness and follows *Bird et al.* [2006], *Seinfeld and Pandis* [2016], *Kundu et al.* [2015], and *Arya* [1999].

Assume there are s species in the atmosphere, the global continuity equation is

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v}) \quad (2.1)$$

where $\rho(\mathbf{r}, t)$ is the mass density of the multi-component system, kg/m^3 . It is a function of position and time,

$$\rho = \sum_{s=1}^N \rho_s \quad (2.2)$$

In the above, $\rho_s(\mathbf{r}, t)$ is the mass density of a single species s , kg/m^3 . It is also a function of position and time; $\mathbf{r} = X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}$ is the position vector in Cartesian coordinates; \mathbf{v} is the mass averaged velocity of the multi-component system, m/s

$$\mathbf{v} = \frac{\rho_1 \mathbf{v}_1 + \rho_2 \mathbf{v}_2 + \cdots + \rho_N \mathbf{v}_N}{\rho_1 + \rho_2 + \cdots + \rho_N} = \frac{\sum_{s=1}^N \rho_s \mathbf{v}_s}{\rho} \quad (2.3)$$

where \mathbf{v}_s is the flow velocity of a single species s , which is the average velocity of all the particles in the single species s , m/s .

The global momentum equation is

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) = -\nabla \cdot (\rho \mathbf{v} \mathbf{v}) - \nabla p - \nabla \cdot \bar{\boldsymbol{\tau}} + \rho \mathbf{g} \quad (2.4)$$

where p is the pressure on the multi-component system when the fluid is at rest, N/m^2 ; \mathbf{g} is the gravitational acceleration, m/s^2 ; $\bar{\boldsymbol{\tau}}$ is the stress tensor due to the viscosity when the fluid is moving, N/m^2 . It has the form of

$$\tau_{ij} = -\mu \left(\frac{\partial v_j}{\partial X_i} + \frac{\partial v_i}{\partial X_j} \right) + \left(\frac{2}{3} \mu - \mu_v \right) (\nabla \cdot \mathbf{v}) \delta_{ij} \quad (2.5)$$

where $i, j = X, Y, Z$; δ_{ij} is the Kronecker delta, $\delta_{ij} = 0$ when $i \neq j$, $\delta_{ij} = 1$ when $i = j$; μ is the dynamic viscosity, $\text{N} \cdot \text{s/m}^2$; μ_v is the dilatational viscosity, $\text{N} \cdot \text{s/m}^2$, it is known from the kinetic theory that μ_v is zero for monatomic gases at low density and also the Stokes assumption ($\mu_v = 0$) is found to be accurate in many situations because either the fluid's μ_v or the fluid's dilatation rate is small [Bird *et al.*, 2006], so Eq. (2.5) becomes

$$\tau_{ij} = -\mu \left(\frac{\partial v_j}{\partial X_i} + \frac{\partial v_i}{\partial X_j} \right) + \frac{2}{3} \mu (\nabla \cdot \mathbf{v}) \delta_{ij} \quad (2.6)$$

The global energy equation is

$$\begin{aligned} \frac{\partial}{\partial t} \left[\rho \left(\hat{U} + \frac{1}{2} |\mathbf{v}|^2 \right) \right] \\ = -\nabla \cdot \left[\rho \left(\hat{U} + \frac{1}{2} |\mathbf{v}|^2 \right) \mathbf{v} \right] - \nabla \cdot \mathbf{q} - \nabla \cdot (p\mathbf{v}) - \nabla \cdot (\bar{\mathbf{t}} \cdot \mathbf{v}) + \rho \mathbf{v} \cdot \mathbf{g} \end{aligned} \quad (2.7)$$

where \hat{U} is the internal energy per unit mass, J/kg; $\rho\hat{U}$ is the internal energy per unit volume; $\frac{1}{2}\rho|\mathbf{v}|^2$ is the kinetic energy per unit volume; \mathbf{q} is the heat flux vector, J/(m² · s).

The energy equation (2.7) contains both the mechanical and thermal energy terms [Bird *et al.*, 2006]. A separate equation for mechanical energy can be obtained from taking dot product of velocity vector \mathbf{v} with the momentum equation (2.4) and using continuity equation (2.1),

$$\begin{aligned} \frac{\partial}{\partial t} \left(\frac{1}{2} \rho |\mathbf{v}|^2 \right) = -\nabla \cdot \left(\frac{1}{2} \rho |\mathbf{v}|^2 \mathbf{v} \right) + \rho \mathbf{v} \cdot \mathbf{g} - \nabla \cdot (p\mathbf{v}) - p(-\nabla \cdot \mathbf{v}) - \nabla \cdot (\bar{\mathbf{t}} \cdot \mathbf{v}) \\ - (-\bar{\mathbf{t}} : \nabla \mathbf{v}) \end{aligned} \quad (2.8)$$

where the symbol “:” denotes the double dot product between tensors. Subtracting Eq. (2.8) from Eq. (2.7) yields the equation for thermal energy,

$$\frac{\partial}{\partial t} (\rho \hat{U}) = -\nabla \cdot (\rho \hat{U} \mathbf{v}) - \nabla \cdot \mathbf{q} - p \nabla \cdot \mathbf{v} - \bar{\mathbf{t}} : \nabla \mathbf{v} \quad (2.9)$$

For the internal energy per unit mass, \hat{U} , it is a function of pressure p and temperature T . If assume ideal gas, then \hat{U} is a function of temperature T only and has the form of $\hat{U} = \hat{C}_V T$, where \hat{C}_V is the specific heat capacity at constant volume, J/(kg · K). We use Fourier’s law for the heat flux, $\mathbf{q} = -k \nabla T$, where k is the thermal conductivity, J/(s · m · K). Then Eq. (2.9) becomes, for real gas,

$$\frac{\partial}{\partial t} (\rho \hat{U}) = -\nabla \cdot (\rho \hat{U} \mathbf{v}) + k \nabla^2 T - p \nabla \cdot \mathbf{v} - \bar{\mathbf{t}} : \nabla \mathbf{v} \quad (2.10)$$

and for ideal gas,

$$\frac{\partial}{\partial t} (\rho \hat{C}_V T) = -\nabla \cdot (\rho \hat{C}_V T \mathbf{v}) + k \nabla^2 T - p \nabla \cdot \mathbf{v} - \bar{\mathbf{t}} : \nabla \mathbf{v} \quad (2.11)$$

The term $p\nabla \cdot \mathbf{v}$ is the heating due to the volume expansion. The term $\bar{\boldsymbol{\tau}} : \nabla \mathbf{v}$ is the heating due to the viscous dissipation of energy, it has the form of

$$\bar{\boldsymbol{\tau}} : \nabla \mathbf{v} = -\mu \left[\frac{\partial v_i}{\partial X_j} + \frac{\partial v_j}{\partial X_i} - \frac{1}{3} (\nabla \cdot \mathbf{v}) \delta_{ij} \right]^2 - \mu_v (\nabla \cdot \mathbf{v})^2 \quad (2.12)$$

If the Stokes assumption ($\mu_v = 0$) applies, then Eq. (2.12) becomes

$$\bar{\boldsymbol{\tau}} : \nabla \mathbf{v} = -\mu \left[\frac{\partial v_i}{\partial X_j} + \frac{\partial v_j}{\partial X_i} - \frac{1}{3} (\nabla \cdot \mathbf{v}) \delta_{ij} \right]^2 \quad (2.13)$$

Equations (2.1), (2.4), and (2.10) are the global continuity equation, global momentum equation, and global energy equation, respectively. They represent five equations for the six unknowns v_X , v_Y , v_Z , p , ρ , and T . One more equation is needed for the closure of the system of equations. For ideal gas, the ideal-gas law can be used for the closure.

$$pM = \rho RT \quad (2.14)$$

where $R = 8.3144598 \text{ J}/(\text{K} \cdot \text{mol})$ is the gas constant; M is the molar mean molecular weight of the multi-component system, kg/mol

$$M = \frac{\rho}{c} = \frac{\sum_{s=1}^N \rho_s}{\sum_{s=1}^N c_s} = \frac{\sum_{s=1}^N \rho_s}{\sum_{s=1}^N \frac{\rho_s}{M_s}} \quad (2.15)$$

where c is the molar density of the multi-component system, mol/m³; c_s and M_s are the molar density and molar weight of a single species, respectively. Then equations (2.1), (2.4), (2.11), and (2.14) can be solved for velocity, pressure, density, and temperature for the multi-component system with proper boundary conditions and initial conditions.

For each single species, they must satisfy the continuity equation over a control volume at each instant

$$\frac{\partial \rho_s}{\partial t} = -\nabla \cdot (\rho_s \mathbf{v}_s) + S_s + R_s \quad (2.16)$$

where $s = 1, 2, \dots, N$ is the number of species; $S_s(\Theta_c, t)$ is the rate of addition of species s at location $\Theta_c = (X_c(t), Y_c(t), Z_c(t))$ and time t , $\text{kg}/(\text{m}^3 \cdot \text{s})$; $\Theta_c = (X_c(t), Y_c(t), Z_c(t))$ is the centroid of the source; $R_s(\rho_1, \dots, \rho_N, T)$ is the rate of generation of species s by chemical reaction, $\text{kg}/(\text{m}^3 \cdot \text{s})$. Equation (2.16) can be rewritten as

$$\frac{\partial \rho_s}{\partial t} = -\nabla \cdot [\rho_s(\mathbf{v}_s - \mathbf{v} + \mathbf{v})] + S_s + R_s = -\nabla \cdot [\rho_s(\mathbf{v}_s - \mathbf{v})] - \nabla \cdot (\rho_s \mathbf{v}) + S_s + R_s \quad (2.17)$$

where $\mathbf{v}_s - \mathbf{v}$ is the flow velocity of species s relative to the mass averaged velocity of the multi-component system, which is also called the diffusive velocity of species s in the multi-component system [Bird *et al.*, 2006]; $\rho_s \mathbf{v}$ is the advective flux of species s due to the flow of the multi-component system, $\text{kg}/(\text{m}^2 \cdot \text{s})$.

Define

$$\mathbf{j}_s = \rho_s(\mathbf{v}_s - \mathbf{v}) \quad (2.18)$$

as the diffusive flux of species s in the multi-component flow due to the molecular diffusion [Bird *et al.*, 2006], $\text{kg}/(\text{m}^2 \cdot \text{s})$. Then Eq. (2.17) becomes

$$\frac{\partial \rho_s}{\partial t} = -\nabla \cdot \mathbf{j}_s - \nabla \cdot (\rho_s \mathbf{v}) + S_s + R_s \quad (2.19)$$

Fick's law can be applied to calculate the diffusive flux due to the molecular diffusion,

$$\mathbf{j}_s = -D_s \nabla \rho_s \quad (2.20)$$

where D_s is the molecular diffusivity of species s in the carrier fluid namely the atmosphere, m^2/s . Substitute Eq. (2.20) into Eq. (2.19), yields

$$\frac{\partial \rho_s}{\partial t} = \nabla \cdot (D_s \nabla \rho_s) - \nabla \cdot (\rho_s \mathbf{v}) + S_s + R_s \quad (2.21)$$

which is the advection-diffusion equation of a single species generated by a point gas source in the atmosphere. With equations (2.1), (2.4), (2.11), (2.14), and (2.21), we can obtain the density or concentration distributions of each species releasing by point sources in the atmosphere.

Because we concentrate on solving the advection-diffusion problem in the lower atmosphere layer, equations (2.1), (2.4), and (2.11) can be simplified using the Boussinesq approximation [Seinfeld and Pandis, 2016]. The validity of such simplification has been proved by Spiegel and Veronis [1960]. The Boussinesq approximation suggests that the density changes in the fluid can be neglected except where the density ρ is multiplied by gravitational acceleration \mathbf{g} . The other properties of the fluid such as dynamic viscosity μ , thermal conductivity k , specific heat capacity at constant volume \hat{C}_v , and specific heat capacity at constant pressure \hat{C}_p are also constants in Boussinesq approximation. The Boussinesq approximation applies if the Mach number of the flow is small, propagation of sound or shock waves is not considered, the vertical scale of the flow is not too large, and the temperature differences in the fluid are small. It is commonly useful for analyzing oceanic and atmospheric flows [Seinfeld and Pandis, 2016].

Under the Boussinesq approximation, for perfect gases, equations (2.1), (2.4), and (2.11) become

$$\nabla \cdot \mathbf{v} = 0 \quad (2.22)$$

$$\frac{\partial \mathbf{v}}{\partial t} = -\nabla \cdot (\mathbf{v}\mathbf{v}) - \frac{1}{\rho_0} \nabla p + \frac{\mu}{\rho_0} \nabla^2 \mathbf{v} + [1 - \alpha(T - T_0)] \mathbf{g} \quad (2.23)$$

$$\rho_0 \hat{C}_p \frac{\partial T}{\partial t} = -\rho_0 \hat{C}_p \nabla \cdot (\mathbf{v}T) + k \nabla^2 T \quad (2.24)$$

where ρ_0 is a reference density corresponding to a reference temperature T_0 , which can be taken to be the mean temperature in the flow or the temperature at an appropriate boundary; $\alpha = -\frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p$ is the thermal expansion coefficient, K^{-1} , and $\alpha = \frac{1}{T}$ for ideal gas; \hat{C}_p is the specific heat capacity at constant pressure, $J/(kg \cdot K)$.

For the derivation of Eq. (2.23) from Eq. (2.4) underlying Boussinesq approximation, the density ρ multiplied by gravitational acceleration \mathbf{g} was replaced by $\rho = \rho_0 [1 - \alpha(T - T_0)]$. For the derivation of Eq. (2.24) from Eq. (2.11) underlying Boussinesq approximation, we assumed ideal gas, so that the ideal-gas law and other expressions related to ideal gas can be used. Also, the heating due to viscous dissipation of energy is negligible under Boussinesq approximation i.e. $\bar{\tau} : \nabla \mathbf{v} = 0$, because it is important only in flows with enormous velocity gradients.

The Boussinesq approximation was not applied to the single species continuity equation (2.21). When we focus on the system of multiple species mixing with the lower atmosphere, the changes of the density of the system can be neglected because the amounts of the atmospheric trace species are so small (in ppm) that they cannot affect the meteorology significantly [Seinfeld and Pandis, 2016]. But when we just focus on single species, the changes of density of a single species with location and time cannot be neglected anymore.

Equations (2.22), (2.23), (2.24), and (2.21) can be solved for the fluid velocities, pressure, and temperature at lower atmosphere and the density or concentration distributions of each species releasing by point sources in the lower atmosphere. But the flows of interest are always turbulent at the lower atmosphere [Seinfeld and Pandis, 2016]. Turbulent flows are random, so the variables in these equations are random functions. Thus, solving any of these equations exactly is impossible, we must resort to some statistical properties of the variables.

In particular, a turbulent field quantity, ϕ , is commonly separated into its first moment and its fluctuations, $\phi = \bar{\phi} + \phi'$, which is known as the Reynolds decomposition [Kundu *et al.*, 2015]. The m^{th} -moment or the m^{th} ensemble average is defined as

$$\overline{\phi^m(\mathbf{r}, t)} \equiv \frac{1}{\mathbb{N}} \sum_{n=1}^{\mathbb{N}} [\phi(\mathbf{r}, t : n)]^m \quad (2.25)$$

where \mathbb{N} is the number of realizations of random variable ϕ in the ensemble. A collection of independent realization of a random variable, which is obtained under identical conditions, is called an ensemble. An ensemble average is the ordinary arithmetic average over the collection and is denoted by an over bar.

The first moment is obtained when $m = 1$, which is the simplest statistical property of a random variable,

$$\overline{\phi(\mathbf{r}, t)} \equiv \frac{1}{\mathbb{N}} \sum_{n=1}^{\mathbb{N}} \phi(\mathbf{r}, t : n) \quad (2.26)$$

For oceanic and atmosphere field measurements, it is impossible to realize the identical conditions for each measurement. Thus, the ensemble average of the flow variable at \mathbf{r} is usually replaced by the time average of the flow variable at \mathbf{r} [Kundu *et al.*, 2015], which is defined as

$$\overline{\phi^m(\mathbf{r})} = \frac{1}{\Delta t} \int_{t_0 - \frac{\Delta t}{2}}^{t_0 + \frac{\Delta t}{2}} \phi^m(\mathbf{r}, t) dt \quad (2.27)$$

when Δt is large enough. Note that theoretically Eq. (2.27) is only valid for the case when the flow is stationary in time, which means its statistics are not dependent on time, but it is still accurate enough to use a finite-duration time average of a single realization to approximate the ensemble average of many realizations even when the flow changes with time [Kundu *et al.*, 2015]. In this case, Eq. (2.27) may be rewritten as

$$\overline{\phi^m(\mathbf{r}, t')}(t) = \frac{1}{\Delta t} \int_{t - \frac{\Delta t}{2}}^{t + \frac{\Delta t}{2}} \phi^m(\mathbf{r}, t') dt' \quad (2.28)$$

Using Reynolds decomposition, the averaged equations of global continuity, global momentum, global energy, and single species continuity with random flow variables can be derived. The starting point is the Boussinesq equation set (2.22), (2.23), and (2.24), and the single species continuity equation (2.21). It is convenient to rewrite these equations in summation convention for deriving the averaged equations

$$\frac{\partial v_i}{\partial X_i} = 0 \quad (2.29)$$

$$\frac{\partial v_i}{\partial t} + \frac{\partial}{\partial X_j} (v_j v_i) = -\frac{1}{\rho_0} \frac{\partial p}{\partial X_i} - g[1 - \alpha(T - T_0)]\delta_{i3} + \frac{\mu}{\rho_0} \frac{\partial^2 v_i}{\partial X_j^2} \quad (2.30)$$

$$\rho_0 \hat{C}_p \left[\frac{\partial T}{\partial t} + \frac{\partial}{\partial X_i} (v_i T) \right] = k \frac{\partial^2 T}{\partial X_i^2} \quad (2.31)$$

$$\frac{\partial \rho_s}{\partial t} = \frac{\partial}{\partial X_i} \left(D_s \frac{\partial \rho_s}{\partial X_i} \right) - \frac{\partial}{\partial X_i} (\rho_s v_i) + S_s + R_s \quad (2.32)$$

Under Reynolds decomposition, the random flow variables can be written as

$$v_i = \bar{v}_i + v'_i, \quad p = \bar{p} + p', \quad T = \bar{T} + T', \quad \rho_s = \bar{\rho}_s + \rho'_s \quad (2.33)$$

Substitute Eq. (2.33) into equations (2.29), (2.30), (2.31), and (2.32) accordingly, average each term, and use the fact that the average of variable fluctuation is zero, $\overline{\phi'} = 0$, yields

$$\frac{\partial \bar{v}_i}{\partial X_i} = 0 \quad (2.34)$$

$$\frac{\partial \bar{v}_i}{\partial t} + \frac{\partial}{\partial X_j} (\bar{v}_j \bar{v}_i) + \frac{\partial}{\partial X_j} (\overline{v'_j v'_i}) = -\frac{1}{\rho_0} \frac{\partial \bar{p}}{\partial X_i} - g[1 - \alpha(\bar{T} - T_0)] \delta_{i3} + \frac{\mu}{\rho_0} \frac{\partial^2 \bar{v}_i}{\partial X_j^2} \quad (2.35)$$

$$\rho_0 \hat{C}_p \left[\frac{\partial \bar{T}}{\partial t} + \bar{v}_i \frac{\partial \bar{T}}{\partial X_i} + \frac{\partial}{\partial X_i} (\overline{v'_i T'}) \right] = k \frac{\partial^2 \bar{T}}{\partial X_i^2} \quad (2.36)$$

$$\begin{aligned} \frac{\partial \bar{\rho}_s}{\partial t} = & \frac{\partial}{\partial X_i} \left(D_s \frac{\partial \bar{\rho}_s}{\partial X_i} \right) - \frac{\partial}{\partial X_i} (\overline{\rho_s v_i}) - \frac{\partial}{\partial X_i} (\overline{\rho'_s v'_i}) + S_s \\ & + R_s (\overline{\rho_1 + \rho'_1}, \dots, \overline{\rho_N + \rho'_N}, \overline{T} + T') \end{aligned} \quad (2.37)$$

Equations (2.34), (2.35), and (2.36) are the Reynolds averaged global continuity, momentum, and energy equations, while Eq. (2.37) is the Reynolds averaged single species continuity equation. This system of equations is not closed, since new unknowns of $\rho_0 \overline{v'_j v'_i}$ (when multiply both sides of Eq. (2.35) by ρ_0), $\rho_0 \hat{C}_p \overline{v'_i T'}$, and $\overline{\rho'_s v'_i}$ are introduced. The terms $\rho_0 \overline{v'_j v'_i}$ are the Reynolds stresses (turbulent momentum fluxes), the terms $\rho_0 \hat{C}_p \overline{v'_i T'}$ are turbulent heat fluxes, and the terms $\overline{\rho'_s v'_i}$ are turbulent mass fluxes of single species [Kundu *et al.*, 2015]. More equations regarding these new unknowns are needed to close the system of equations, but unfortunately even more new unknowns will be introduced if we do so, which means the Reynolds averaged system of equations is a closure problem. Thus, empirical models are preferred for the turbulent momentum, energy, and mass fluxes to close the system of equations. The most popular and widely used models for turbulent transport are the mixing-length models [Arya, 1999],

$$\overline{v'_j v'_i} = -K_{M,ji} \frac{\partial \overline{v_j}}{\partial X_i} \quad (2.38)$$

$$\overline{v'_i T'} = -K_{T,ik} \frac{\partial \overline{T}}{\partial X_i} \quad (2.39)$$

$$\overline{\rho'_s v'_i} = -K_{ik} \frac{\partial \overline{\rho_s}}{\partial X_i} \quad (2.40)$$

where $K_{M,ji}$ is the eddy viscosity or turbulent momentum diffusivity, m^2/s ; $K_{T,ik}$ is the eddy viscosity for heat transfer, m^2/s ; K_{ik} is the eddy diffusivity, m^2/s . In general, they are functions of location and direction, and their values or functional forms are usually determined empirically or experimentally.

Substitute equations (2.38), (2.39), and (2.40) into equations (2.35), (2.36), and (2.37), respectively, yields

$$\frac{\partial \overline{v_i}}{\partial t} + \frac{\partial}{\partial X_j} (\overline{v_j v_i}) - \frac{\partial}{\partial X_j} \left(K_{M,ji} \frac{\partial \overline{v_j}}{\partial X_i} \right) = -\frac{1}{\rho_0} \frac{\partial \overline{p}}{\partial X_i} - g[1 - \alpha(\overline{T} - T_0)] \delta_{i3} + \frac{\mu}{\rho_0} \frac{\partial^2 \overline{v_i}}{\partial X_j^2} \quad (2.41)$$

$$\rho_0 \hat{C}_p \left[\frac{\partial \overline{T}}{\partial t} + \overline{v_i} \frac{\partial \overline{T}}{\partial X_i} - \frac{\partial}{\partial X_i} \left(K_{T,ik} \frac{\partial \overline{T}}{\partial X_i} \right) \right] = k \frac{\partial^2 \overline{T}}{\partial X_i^2} \quad (2.42)$$

$$\begin{aligned} \frac{\partial \overline{\rho_s}}{\partial t} = & \frac{\partial}{\partial X_i} \left(D_s \frac{\partial \overline{\rho_s}}{\partial X_i} \right) - \frac{\partial}{\partial X_i} (\overline{\rho_s v_i}) + \frac{\partial}{\partial X_i} \left(K_{ik} \frac{\partial \overline{\rho_s}}{\partial X_i} \right) + S_s \\ & + R_s(\overline{\rho_1} + \rho'_1, \dots, \overline{\rho_N} + \rho'_N, \overline{T} + T') \end{aligned} \quad (2.43)$$

Equations (2.34), (2.41), (2.42), and (2.43) can be solved for averaged global velocities, averaged global pressure, averaged global temperature, and averaged density or concentration distributions of species releasing by point sources in the atmosphere, with proper boundary conditions, initial conditions, values or functional forms of the diffusivity terms, and functional forms of the reaction term. In general, the coupled equations (2.34), (2.41), (2.42), and (2.43) need to be solved simultaneously to obtain the changes in velocities \mathbf{v} , temperature T , and species concentration ρ_s and the effects of these changes on each other. As mentioned before, the amounts of the atmospheric trace species are so small (in ppm) that they cannot affect the meteorology significantly, thus the species continuity equation (2.43) can be solved

independently of the global equations (2.34), (2.41), and (2.42), which means the global equations can be dropped. With such assumption, the complete description of the behavior of the releasing species in the lower atmosphere can be obtained by Eq. (2.43) only.

Suppose there is only one chemically inert species (so that the reaction term in Eq. (2.43) can be dropped) releasing in the lower atmosphere, so the system fluid has two components – releasing gas species and atmosphere. Denote the density or mass concentration of the releasing gas and atmosphere as ρ_r and ρ_a , respectively. Then Eq. (2.43) becomes

$$\frac{\partial \overline{\rho_r}}{\partial t} = \frac{\partial}{\partial X_i} \left(D_r \frac{\partial \overline{\rho_r}}{\partial X_i} \right) - \frac{\partial}{\partial X_i} (\overline{\rho_r} \overline{v_i}) + \frac{\partial}{\partial X_i} \left(K_{ik} \frac{\partial \overline{\rho_r}}{\partial X_i} \right) + S_r \quad (2.44)$$

Rewrite Eq. (2.44) in vector form as

$$\frac{\partial \overline{\rho_r}}{\partial t} = \nabla \cdot (D_r \nabla \overline{\rho_r}) - \nabla \cdot (\overline{\rho_r} \overline{\mathbf{v}}) + \nabla \cdot (\overline{\mathbf{K}} \cdot \nabla \overline{\rho_r}) + S_r \quad (2.45)$$

where $\overline{\mathbf{K}}$ is the eddy diffusivity tensor. From Eq. (2.3), the mass averaged velocity of such binary mixture of a releasing gas and the atmosphere, \mathbf{v} is

$$\mathbf{v} = \frac{\rho_r \mathbf{v}_r + \rho_a \mathbf{v}_a}{\rho_r + \rho_a} \quad (2.46)$$

where \mathbf{v}_r is the velocity of the releasing gas; \mathbf{v}_a is the velocity of the atmosphere namely the wind velocity. Since the amount of the trace gas is so small, we can assume $\rho_r \ll \rho_a$. Then Eq. (2.46) becomes

$$\mathbf{v} \approx \frac{\rho_r}{\rho_a} \mathbf{v}_r + \mathbf{v}_a \quad (2.47)$$

Assume the transport and dilution of the releasing gas are controlled by the atmosphere i.e. at the diffusion regime [Court, 2012], then we have $\mathbf{v}_r \ll \mathbf{v}_a$, so Eq. (2.47) becomes

$$\mathbf{v} \approx \mathbf{v}_a \quad (2.48)$$

which means the mass averaged velocity of the mixture of a releasing gas and the atmosphere can be approximated by the atmosphere velocity itself with the assumptions mentioned above. Substitute Eq. (2.48) into Eq. (2.45), yields

$$\frac{\partial \overline{\rho_r}}{\partial t} = \nabla \cdot (D_r \nabla \overline{\rho_r}) - \nabla \cdot (\overline{\rho_r} \overline{\mathbf{v}_a}) + \nabla \cdot (\overline{\mathbf{K}} \cdot \nabla \overline{\rho_r}) + S_r \quad (2.49)$$

where $\overline{\mathbf{v}_a}$ is the averaged atmosphere velocity or the averaged wind velocity. In general, it is a function of position and time.

The eddy diffusivities are of order 1 to 100 m²/s for different atmospheric conditions of stable, neutral, and unstable [Arya, 1999; Seinfeld and Pandis, 2016], while the typical value of the molecular diffusivity of air is 1.5×10^{-5} m²/s [Arya, 1999]. So, we can assume the diffusive fluxes due to the molecular diffusion is negligible compared with the turbulent fluxes due to the turbulent atmospheric flow,

$$\nabla \cdot (D_r \nabla \overline{\rho_r}) \ll \nabla \cdot (\overline{\mathbf{K}} \cdot \nabla \overline{\rho_r}) \quad (2.50)$$

Denoting $\overline{\rho_r}$ as C and with the assumption of Eq. (2.50), Eq. (2.49) becomes

$$\frac{\partial C}{\partial t} = -\nabla \cdot (C \overline{\mathbf{v}_a}) + \nabla \cdot (\overline{\mathbf{K}} \cdot \nabla C) + S_r \quad (2.51)$$

For the eddy diffusivity tensor $\overline{\mathbf{K}}$, its principal axes coincide with the coordinate axes, so only three diagonal elements K_{XX} , K_{YY} , and K_{ZZ} are nonzero [Seinfeld and Pandis, 2016]. Then Eq. (2.51) can be rewritten in summation convention as

$$\frac{\partial C}{\partial t} = -\frac{\partial}{\partial X_i} (C \overline{v_{ai}}) + \frac{\partial}{\partial X_i} \left(K_{ii} \frac{\partial C}{\partial X_i} \right) + S_r \quad (2.52)$$

Expanding either Eq. (2.51) or Eq. (2.52) yields

$$\begin{aligned} \frac{\partial C}{\partial t} = & -\frac{\partial}{\partial X} (C \overline{v_{aX}}) - \frac{\partial}{\partial Y} (C \overline{v_{aY}}) - \frac{\partial}{\partial Z} (C \overline{v_{aZ}}) + \frac{\partial}{\partial X} \left(K_{XX} \frac{\partial C}{\partial X} \right) + \frac{\partial}{\partial Y} \left(K_{YY} \frac{\partial C}{\partial Y} \right) \\ & + \frac{\partial}{\partial Z} \left(K_{ZZ} \frac{\partial C}{\partial Z} \right) + S_r \end{aligned} \quad (2.53)$$

Denoting the averaged wind speed in X , Y , and Z directions $\overline{v_{aX}}$, $\overline{v_{aY}}$, and $\overline{v_{aZ}}$ as U , V , and W , Eq. (2.53) can be rewritten as

$$\begin{aligned} \frac{\partial C}{\partial t} = & -\frac{\partial}{\partial X}(CU) - \frac{\partial}{\partial Y}(CV) - \frac{\partial}{\partial Z}(CW) + \frac{\partial}{\partial X}\left(K_{XX}\frac{\partial C}{\partial X}\right) + \frac{\partial}{\partial Y}\left(K_{YY}\frac{\partial C}{\partial Y}\right) \\ & + \frac{\partial}{\partial Z}\left(K_{ZZ}\frac{\partial C}{\partial Z}\right) + S_r \end{aligned} \quad (2.54)$$

If the atmosphere is incompressible, $\nabla \cdot \bar{\mathbf{v}}_a = 0$, then equations (2.51), (2.52), and (2.54) become

$$\frac{\partial C}{\partial t} = -\bar{\mathbf{v}}_a \cdot \nabla C + \nabla \cdot (\bar{\mathbf{K}} \cdot \nabla C) + S_r \quad (2.55)$$

$$\frac{\partial C}{\partial t} = -\bar{v}_{ai} \frac{\partial C}{\partial X_i} + \frac{\partial}{\partial X_i} \left(K_{ii} \frac{\partial C}{\partial X_i} \right) + S_r \quad (2.56)$$

$$\frac{\partial C}{\partial t} = -U \frac{\partial C}{\partial X} - V \frac{\partial C}{\partial Y} - W \frac{\partial C}{\partial Z} + \frac{\partial}{\partial X} \left(K_{XX} \frac{\partial C}{\partial X} \right) + \frac{\partial}{\partial Y} \left(K_{YY} \frac{\partial C}{\partial Y} \right) + \frac{\partial}{\partial Z} \left(K_{ZZ} \frac{\partial C}{\partial Z} \right) + S_r \quad (2.57)$$

Equation (2.54) or (2.57) can be solved with proper boundary conditions, initial conditions, wind speed profiles, and values or functional forms of eddy diffusivities for the density or concentration distribution of a chemically inert gas species releasing in the lower atmosphere.

In summary, the constitution equation of the process model for the plume concentration in the domain depicted in Figure 2.1 is the well-known advection-diffusion equation

$$\frac{\partial C(\mathbf{r}, t)}{\partial t} = \mathcal{L}C(\mathbf{r}, t) + S(\Theta_c(t), t), \quad \forall \mathbf{r} \in \Omega \subset \mathbb{R}^3 \quad (2.58)$$

where \mathcal{L} is the advection-diffusion operator,

$$\begin{aligned} \mathcal{L}\varphi = & -\frac{\partial}{\partial X}(U\varphi) - \frac{\partial}{\partial Y}(V\varphi) - \frac{\partial}{\partial Z}(W\varphi) + \frac{\partial}{\partial X}\left(K_{XX}\frac{\partial \varphi}{\partial X}\right) + \frac{\partial}{\partial Y}\left(K_{YY}\frac{\partial \varphi}{\partial Y}\right) \\ & + \frac{\partial}{\partial Z}\left(K_{ZZ}\frac{\partial \varphi}{\partial Z}\right) \end{aligned} \quad (2.59)$$

φ is an arbitrary physical variable; $S(\Theta_c(t), t)$ is the moving (or stationary) source; $\Theta_c(t) = (X_c(t), Y_c(t), Z_c(t))$ is the source location at time t . Equation (2.58) is supplemented with boundary conditions depending on the type of the physical boundary (i.e. absorbing, reflecting) such as Dirichlet

$$C(\mathbf{r}, t) = 0, \quad \forall \mathbf{r} \in \partial\Omega_D \quad (2.60)$$

or Neumann

$$\frac{\partial C(\mathbf{r}, t)}{\partial \mathbf{n}} = 0, \quad \forall \mathbf{r} \in \partial\Omega_N \quad (2.61)$$

where \mathbf{n} is the outward normal on boundaries, together with the initial conditions

$$C(\mathbf{r}, 0) = 0, \quad \forall \mathbf{r} \in \Omega \subset \mathbb{R}^3 \quad (2.62)$$

The process model Eq. (2.58) can be rewritten in state-space form as

$$C_t = \mathcal{L}C + S(\Theta_c(t), t) \quad (2.63)$$

where the subscript t means the partial derivative with respect to time t .

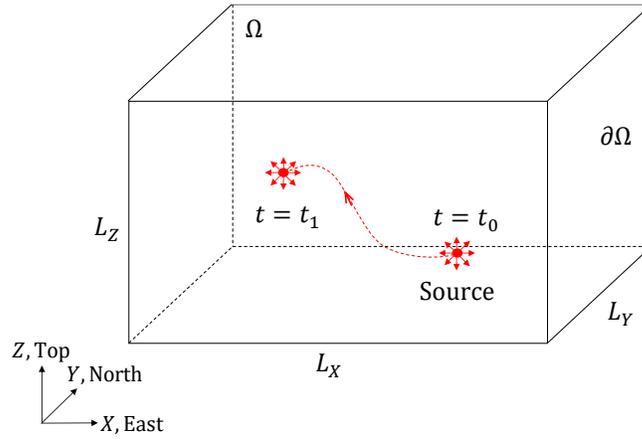


Figure 2.1 The process model describing a moving source releasing trace species with rate u (kg/s) in the lower atmosphere.

In Figure 2.1 Ω designates the entire domain of interest, $\partial\Omega$ the boundary of domain Ω and L_x , L_y , and L_z the domain lengths in X -, Y -, and Z -direction, respectively. The process model equations (2.58) – (2.62) can be solved numerically in domain Ω to provide the numerical data used by the estimator model in simulations.

2.2 Process-State Hybrid Estimator

The domain of interest is decomposed into N_{sd} non-overlapping subdomains with each designated by $\Omega_{(m)}$, $m = 1, 2, \dots, N_{sd}$ as shown in Figure 1.2. At time t , the SAV is at $\Theta_s(t) =$

$(X_s(t), Y_s(t), Z_s(t))$ in subdomain m_L , positioned there by the guidance model discussed in the next subsection 2.3. Figure 1.2 also shows the location of the source at time t . Our real-time estimator is *hybrid* and is based on a Luenberger estimator in the subdomain m_L where the SAV resides and a naive estimator in the rest of the subdomains, $m = 1, 2, \dots, N_{sd}$ ($m \neq m_L$), as indicated in Figure 1.2 [Hu and Demetriou, 2019]. The hybrid estimator uses a point measurement of the concentration $C_{(m_L)}(\mathbf{r}, t)$ taken at the location of the SAV and provides (once executed on the SAV's computer), in real-time, at time t the estimated concentration $\hat{C}_{(m)}(\mathbf{r}, t)$ at all locations \mathbf{r} in the domain of interest shown in Figure 1.2. The hybrid estimator [Hu and Demetriou, 2019] consists of the Luenberger estimator with the source term of the advection-diffusion equation replaced by an output injection term which depends on the sensor location,

$$\frac{\partial \hat{C}_{(m)}(\mathbf{r}, t)}{\partial t} = \mathcal{L} \hat{C}_{(m)}(\mathbf{r}, t) + R(\Theta_s(t), t), \quad \text{for } m = m_L, \forall \mathbf{r} \in \Omega_{(m)} \subset \Omega \subset \mathbb{R}^3 \quad (2.64)$$

and a naive observer

$$\begin{aligned} \frac{\partial \hat{C}_{(m)}(\mathbf{r}, t)}{\partial t} &= \mathcal{L} \hat{C}_{(m)}(\mathbf{r}, t), \\ &\text{for } m = 1, 2, \dots, N_{sd} \ (m \neq m_L), \forall \mathbf{r} \in \Omega_{(m)} \subset \Omega \subset \mathbb{R}^3 \end{aligned} \quad (2.65)$$

The above equations (2.64) and (2.65) are subject to transmission conditions that enforce continuity of concentration and conservation of fluxes on interfaces between subdomains [Blayo et al., 2007],

$$\begin{aligned} \hat{C}_{(p)\Gamma}(\mathbf{r}, t) &= \hat{C}_{(q)\Gamma}(\mathbf{r}, t), \\ \forall \mathbf{r} \in \Gamma; \Gamma &= \partial\Omega_{(p)} \cap \partial\Omega_{(q)}; p, q = 1, 2, \dots, N_{sd} \text{ and } p \neq q \end{aligned} \quad (2.66)$$

and

$$\begin{aligned} \mathbf{v}_{(p)} \cdot \mathbf{n}_{(p)} \hat{C}_{(p)\Gamma}(\mathbf{r}, t) - K_{(p)} \nabla \hat{C}_{(p)\Gamma}(\mathbf{r}, t) \cdot \mathbf{n}_{(p)} \\ &= \mathbf{v}_{(q)} \cdot \mathbf{n}_{(q)} \hat{C}_{(q)\Gamma}(\mathbf{r}, t) - K_{(q)} \nabla \hat{C}_{(q)\Gamma}(\mathbf{r}, t) \cdot \mathbf{n}_{(q)}, \\ \forall \mathbf{r} \in \Gamma; \Gamma &= \partial\Omega_{(p)} \cap \partial\Omega_{(q)}; p, q = 1, 2, \dots, N_{sd} \text{ and } p \neq q \end{aligned} \quad (2.67)$$

where $\mathbf{v}_{(p)}$ and $\mathbf{v}_{(q)}$ are the velocity fields, $K_{(p)}$ and $K_{(q)}$ are the eddy diffusivities, and $\mathbf{n}_{(p)}$ and $\mathbf{n}_{(q)}$ are the outward normal on the interface Γ of two adjacent subdomains p and q , respectively. The transmission conditions also serve as the consensus protocol between the Luenberger and naive observers and ensure that the naive observer receives process information [Hu and Demetriou, 2019].

The estimators are also supplemented with Dirichlet boundary conditions

$$\hat{C}_{(m)}(\mathbf{r}, t) = 0, \quad \forall \mathbf{r} \in \partial\Omega_{(m)} \cap \partial\Omega_D \quad (2.68)$$

or Neumann boundary conditions

$$\frac{\partial \hat{C}_{(m)}(\mathbf{r}, t)}{\partial \mathbf{n}} = 0, \quad \forall \mathbf{r} \in \partial\Omega_{(m)} \cap \partial\Omega_N \quad (2.69)$$

together with the initial conditions

$$\hat{C}_{(m)}(\mathbf{r}, 0) = 0, \quad \forall \mathbf{r} \in \Omega_{(m)} \subset \Omega \subset \mathbb{R}^3 \quad (2.70)$$

The output injection term in Eq. (2.64) is

$$R(\Theta_s(t), t) = \Lambda \cdot \delta(X - X_s(t))\delta(Y - Y_s(t))\delta(Z - Z_s(t)) \cdot [C(\Theta_s(t), t) - \hat{C}(\Theta_s(t), t)] \quad (2.71)$$

where $\Lambda > 0$ is the user-defined estimation gain; $\delta(\cdot)\delta(\cdot)\delta(\cdot)$ is the 3D Dirac delta function with the property of $\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(X, Y, Z)\delta(X - X_0)\delta(Y - Y_0)\delta(Z - Z_0) dXdYdZ = f(X_0, Y_0, Z_0)$; $C(\Theta_s(t), t)$ is the sensor measurement model which is based on the Lagrangian sensing technique [Bennett, 2006; Fiorelli et al., 2006],

$$C(\Theta_s(t), t) = \int_0^{L_X} \int_0^{L_Y} \int_0^{L_Z} C(X, Y, Z, t)\delta(X - X_s(t))\delta(Y - Y_s(t))\delta(Z - Z_s(t)) dZdYdX \quad (2.72)$$

$\hat{C}(\Theta_s(t), t)$ is the estimated concentration at the sensor location,

$$\begin{aligned} \hat{C}(\Theta_s(t), t) = & \int_0^{L_X} \int_0^{L_Y} \int_0^{L_Z} \hat{C}(X, Y, Z, t) \delta(X - X_s(t)) \delta(Y - Y_s(t)) \delta(Z \\ & - Z_s(t)) dZ dY dX \end{aligned} \quad (2.73)$$

If we denote the observation operator as

$$\begin{aligned} \mathcal{C}\varphi(X, Y, Z, t) = & \int_0^{L_X} \int_0^{L_Y} \int_0^{L_Z} \varphi(X, Y, Z, t) \delta(X - X_s(t)) \delta(Y - Y_s(t)) \delta(Z \\ & - Z_s(t)) dZ dY dX = \varphi(X_s(t), Y_s(t), Z_s(t), t) = \varphi(\Theta_s(t), t) \end{aligned} \quad (2.74)$$

and its adjoint operator as

$$\mathcal{C}^* = \delta(X - X_s(t)) \delta(Y - Y_s(t)) \delta(Z - Z_s(t)) \quad (2.75)$$

then the Luenberger estimator model can be written in state-space form as

$$\hat{C}_t = \mathcal{L}\hat{C} + \mathcal{C}^* \Lambda \mathcal{C} (C - \hat{C}) \quad (2.76)$$

and the naive estimator model can be written in state-space form as well

$$\hat{C}_t = \mathcal{L}\hat{C} \quad (2.77)$$

2.3 SAV Guidance Model and Sensor Model

Two different approaches are used to guide the SAV. One is the modified Lyapunov guidance law and the other is the Lyapunov guidance law with six degree-of-freedom fixed-wing aircraft dynamical model. Sensor model is also briefly introduced at the end of this subsection.

2.3.1 Modified Lyapunov Guidance Law

The SAV is approximated in this approach as a point mass and the guidance model is based on the modified Lyapunov guidance law proposed by *Hu and Demetriou* [2019]. The desired Cartesian velocity components of the SAV are given by

$$\begin{aligned}
U_s^d &\equiv \dot{X}_s^d = k_x \cdot \text{sign}(\varepsilon(t)) \cdot \text{sign}(\varepsilon_x(t)) \\
V_s^d &\equiv \dot{Y}_s^d = k_y \cdot \text{sign}(\varepsilon(t)) \cdot \text{sign}(\varepsilon_y(t)) \\
W_s^d &\equiv \dot{Z}_s^d = k_z \cdot \text{sign}(\varepsilon(t)) \cdot \text{sign}(\varepsilon_z(t))
\end{aligned} \tag{2.78}$$

where U_s^d or \dot{X}_s^d is the desired velocity in X -direction; V_s^d or \dot{Y}_s^d is the desired velocity in Y -direction; W_s^d or \dot{Z}_s^d is the desired velocity in Z -direction; $k_x > 0$, $k_y > 0$, and $k_z > 0$ are user-defined constant guidance gains; $\varepsilon(t)$ is the state-estimation error at the sensor location,

$$\varepsilon(t) \equiv e(\Theta_s(t), t) = C(\Theta_s(t), t) - \hat{C}(\Theta_s(t), t) \tag{2.79}$$

and $\varepsilon_i(t)$ is the gradient of the state-estimation error at the sensor location,

$$\varepsilon_i(t) \equiv \frac{\partial e(\Theta_s(t), t)}{\partial i}, \quad i = X, Y, Z \tag{2.80}$$

Integrating Eq. (2.78) the desired Cartesian velocities of the SAV with respect to time results in the desired location of the SAV,

$$X_s^d = \int_{t_1}^{t_2} U_s^d dt, \quad Y_s^d = \int_{t_1}^{t_2} V_s^d dt, \quad Z_s^d = \int_{t_1}^{t_2} W_s^d dt \tag{2.81}$$

2.3.2 Lyapunov Guidance Law with SAV Dynamical Model

In this approach we follow *Egorova et al.* [2016]. The SAV is considered as a fixed-wing aircraft with a point-mass M subjected. The thrust T , drag D , and lift force L are shown in Figure 2.2 [Beard and McLain, 2012]. The SAV is climbing at flight path angle γ and bank angle ϕ . χ is the course angle in an inertial reference frame with X , Y , and Z pointing to north, east, and the center of the earth, respectively.

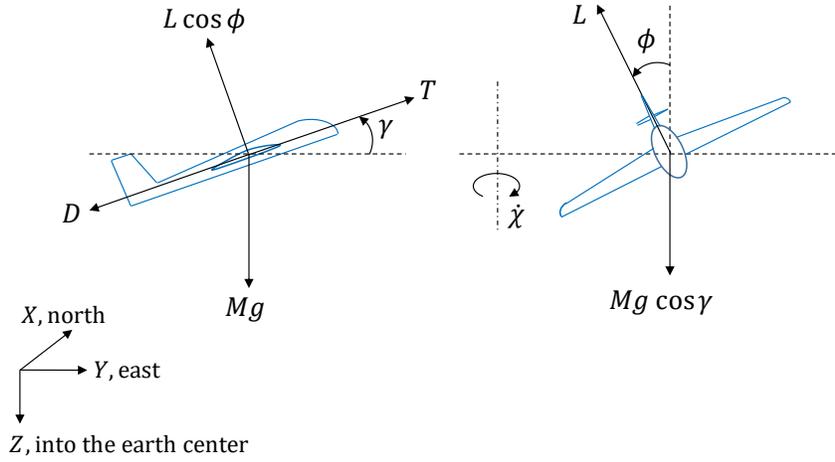


Figure 2.2 Free-body diagram of the forces acting on the SAV in the climbing coordinate turn.

The equations of motion of the SAV are

$$\begin{aligned}
 \dot{X}_s &= V_g \cos \gamma \cos \chi \\
 \dot{Y}_s &= V_g \cos \gamma \sin \chi \\
 \dot{Z}_s &= -V_g \sin \gamma \\
 \dot{V}_g &= \frac{1}{M} (T - D - Mg \sin \gamma) \\
 \dot{\gamma} &= \frac{1}{MV_g} (L \cos \phi - Mg \cos \gamma) \\
 \dot{\chi} &= \frac{L \sin \phi \cos(\chi - \psi)}{MV_g \cos \gamma}
 \end{aligned} \tag{2.82}$$

where \dot{X}_s , \dot{Y}_s , and \dot{Z}_s are the velocity components of the SAV relative to the inertial reference frame in X -, Y -, and Z -direction, respectively; ψ is the heading angle of the SAV; V_g is the magnitude of the ground velocity \mathbf{V}_g which is the SAV's velocity relative to the inertial reference frame. The ground velocity is related with the wind velocity \mathbf{V}_w which is relative to the inertial reference frame and the air velocity \mathbf{V}_a which is the SAV velocity relative to the wind velocity by the wind triangle defined as

$$\mathbf{V}_g = \mathbf{V}_a + \mathbf{V}_w \tag{2.83}$$

The components of the wind velocity \mathbf{V}_w are U , V , and W used in Eq. (2.59).

In Eq. (2.82) the drag and lift forces are expressed as

$$D = \frac{1}{2} \rho V_a^2 S C_D \quad (2.84)$$

$$L = \frac{1}{2} \rho V_a^2 S C_L \quad (2.85)$$

where ρ is the air density, V_a is the airspeed with is the magnitude of the air velocity \mathbf{V}_a , S is the planform area of the SAV wings. The relation between the drag coefficient C_D and the lift coefficient C_L is

$$C_D = C_{D_p} + \frac{C_L^2}{\pi e AR} \quad (2.86)$$

where C_{D_p} is the parasitic drag caused by the shear stress, $AR \equiv b^2/S$ is the aspect ratio of the wing, b is the wing span of the SAV, e is the Oswald efficiency factor [Beard and McLain, 2012; Anderson, 2005].

The control inputs to the SAV dynamical model Eq. (2.82) are the thrust T , bank angle ϕ , and the lift coefficient C_L . The SAV is guided by a Lyapunov redesign method based guidance law [Egorova et al., 2016],

$$\begin{aligned} U_s^d &\equiv \dot{X}_s^d = -k_X \cdot \varepsilon(t) \cdot \varepsilon_X(t) \\ V_s^d &\equiv \dot{Y}_s^d = -k_Y \cdot \varepsilon(t) \cdot \varepsilon_Y(t) \\ W_s^d &\equiv \dot{Z}_s^d = -k_Z \cdot \varepsilon(t) \cdot \varepsilon_Z(t) \end{aligned} \quad (2.87)$$

where U_s^d or \dot{X}_s^d , V_s^d or \dot{Y}_s^d , W_s^d or \dot{Z}_s^d , k_X , k_Y , k_Z , $\varepsilon(t)$, $\varepsilon_X(t)$, $\varepsilon_Y(t)$, and $\varepsilon_Z(t)$ are the same as in Eq. (2.78).

The control inputs are calculated starting from the desired Cartesian velocities as shown in Eq. (2.87). First, with the desired Cartesian velocities, the desired ground speed V_g^d , desired flight path angle γ^d , and the desired course angle χ^d are calculated using the first three equations of Eq. (2.82)

$$\begin{aligned}
V_g^d &= \sqrt{(U_s^d)^2 + (V_s^d)^2 + (W_s^d)^2} \\
\gamma^d &= \arcsin\left(-\frac{W_s^d}{V_g^d}\right) \\
\chi^d &= \text{atan2}(V_s^d, U_s^d)
\end{aligned} \tag{2.88}$$

Then the corresponding accelerations are obtained by the first-order model from *Beard and McLain* [2012]

$$\begin{aligned}
\dot{V}_g^d &= b_{V_g}(V_g^d - V_g) \\
\dot{\gamma}^d &= b_\gamma(\gamma^d - \gamma) \\
\dot{\chi}^d &= b_\chi(\chi^d - \chi)
\end{aligned} \tag{2.89}$$

where b_{V_g} , b_γ , and b_χ are user-defined positive constants. Lastly, combining equations (2.82), (2.84), (2.85), (2.86), (2.88) and (2.89) yields the control inputs for the SAV

$$\begin{aligned}
\phi &= \arctan2\left[\dot{\chi}^d \cos \gamma^d, \left(\dot{\gamma}^d + \frac{g \cos \gamma^d}{V_g^d}\right) \cos(\chi^d - \psi)\right] \\
C_L &= \frac{MV_g^d \left(\dot{\gamma}^d + \frac{g \cos \gamma^d}{V_g^d}\right)}{\frac{1}{2} \rho V_a^2 S \cos \phi}
\end{aligned} \tag{2.90}$$

$$T = M\dot{V}_g^d + \frac{1}{2} \rho V_a^2 S \left(C_{D_p} + \frac{C_L^2}{\pi e AR} \right) + Mg \sin \gamma^d$$

In the above, the control inputs are assumed to stay within certain ranges

$$0 \leq T \leq T_{\max}, \quad |\phi| \leq \phi_{\max}, \quad \dot{\phi} = b_\phi(\phi - \phi_{\text{previous}}) \leq \dot{\phi}_{\max}, \quad 0 \leq C_L \leq C_{L \max} \tag{2.91}$$

where T_{\max} , ϕ_{\max} , and $C_{L \max}$ are constants depending on a specific SAV. T_{\max} is usually found in the engine specifications. $C_{L \max}$ is the lift coefficient when the airplane is in the stall condition. ϕ_{\max} depends on the load factor defined by $n \equiv \frac{L}{Mg}$. The maximum allowable load

factor n_{\max} is typically equal to 1.5 [Anderson, 2005]. Then ϕ_{\max} is determined when $\gamma = \dot{\gamma} = 0$ in the fifth equation of Eq. (2.82)

$$\frac{1}{\cos \phi_{\max}} = \frac{L}{Mg} \leq n_{\max} \quad (2.92)$$

In Eq. (2.91) the rate of change of the bank angle is also constrained to ensure reasonable poses of the SAV. ϕ is the bank angle at the current time instance, ϕ_{previous} is the bank angle at the previous time instance, b_{ϕ} is a user-defined positive constant which is defined from *Beard and McLain* [2012], $\dot{\phi}_{\max}$ is the maximum rate of change of the bank angle.

2.3.3 Sensor Model

The sensor model in this work is assumed to be ideal and provides instantaneous readings (no delays) without errors [Demetriou et al., 2013]. The spatial distribution of the sensor is modeled as a 3D spatial Dirac delta function

$$y(\Theta_s(t), t) = \int_0^{L_X} \int_0^{L_Y} \int_0^{L_Z} C(X, Y, Z, t) \delta(X - X_s(t)) \delta(Y - Y_s(t)) \delta(Z - Z_s(t)) dZ dY dX \quad (2.93)$$

The sensor response is modeled as

$$y(\Theta_s(t), t) = \begin{cases} 0, & C(\Theta_s(t), t) < C_{\min} \\ C(\Theta_s(t), t), & C_{\min} \leq C(\Theta_s(t), t) \leq C_{\max} \\ C_{\max}, & C(\Theta_s(t), t) > C_{\max} \end{cases} \quad (2.94)$$

where C_{\min} is the sensor working threshold, C_{\max} is the sensor saturation.

Chapter 3 Heterogeneous Non-Overlapping Domain Decomposition Explicit Finite Volume Method, Verification and Error Analysis, Parallelization Efficiency

In this chapter, a heterogeneous non-overlapping domain decomposition explicit finite volume method (HT-NODDE-FVM) is developed to solve in parallel the hybrid estimator described in Chapter 2. This chapter presents the detailed algorithmic steps of the computational domain decomposition and parallelization, the detailed FVM-TVD spatial discretization in one of the subdomains, and the 4th order Runge-Kutta temporal integration. It also presents the numerical stability analysis. The verification and error analysis for the HT-NODDE-FVM with TVD-RK method and hybrid estimator is also presented. At the end of this chapter, the parallelization efficiency analysis of the NODDE-FVM-TVD-RK method is conducted. Results have been presented in *Gatsonis et al.* [2020] and *Tian et al.* [2020].

3.1 Computational Domain Decomposition and Parallelization

The hybrid estimator equations (2.64) and (2.65), transmission conditions equations (2.66) and (2.67), and the guidance model equations (2.78) – (2.81) and (2.82) – (2.92) for the SAV are solved numerically with the HT-NODDE-FVM. The overall computational procedure is shown in Figure 3.1.

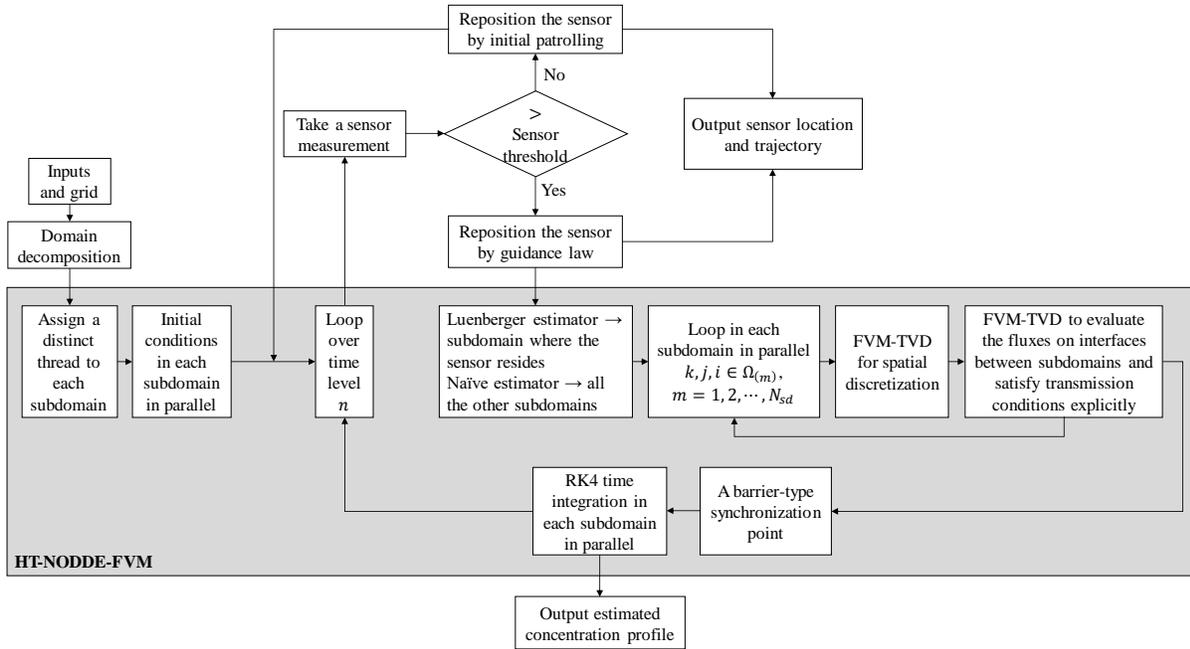


Figure 3.1 A flow chart showing the major elements of the HT-NODDE-FVM scheme.

Also, the overall computational cycle is implemented using OpenMP parallel paradigm [Chandra et al., 2001] and is listed as follows:

- S-1. Generate a grid for entire domain based on the inputs $L_X, L_Y, L_Z, N_X, N_Y, N_Z$ with cell sizes $\Delta X = L_X/N_X, \Delta Y = L_Y/N_Y$, and $\Delta Z = L_Z/N_Z$.
- S-2. Divide the entire domain into $N_{DDX} \times N_{DDY} \times N_{DDZ}$ subdomains with each subdomain having the same number of cells.
- S-3. Obtain a distinct thread ID for each subdomain using the OpenMP runtime routine $thread_{ID} = \text{OMP_GET_THREAD_NUM}()$.
- S-4. Set up the loop indices in each subdomain.
- S-5. Apply initial conditions Eq. (2.70) in each subdomain in parallel.
- S-6. For $t^n = n\Delta t < \text{total simulation time}$,
- S-7. SAV patrols the domain and sensor reads concentration data from the process model where the measured concentration is $C_S^n(\Theta_S^n, t^n)$.

- S-8. If $C_s^n(\Theta_s^n, t^n) > \text{sensor threshold}$, then go to steps S-9 – S-16, otherwise the SAV keeps its initial patrolling trajectory and go to step S-6.
- S-9. Calculate the error between measured concentration $C_s^n(\Theta_s^n, t^n)$ and estimated concentration $\hat{C}_s^n(\Theta_s^n, t^n)$ using Eq. (2.79) and the error gradients at the sensor location using Eq. (2.80), then reposition the sensor to Θ_s^{n+1} using Eq. (2.81) or equations (2.82) – (2.92).
- S-10. Activate the Luenberger estimator model Eq. (2.64) for the subdomain where the SAV resides and activate the naive estimator model Eq. (2.65) for the remainder of the subdomains.
- S-11. In each subdomain $\Omega_{(m)}$, loop for $k_{(m)}$, $j_{(m)}$, and $i_{(m)}$,
- S-12. Use the FVM-TVD (Total Variation Diminishing) scheme for spatial discretization and to evaluate the fluxes on interfaces between adjacent subdomains and satisfy the transmission conditions equations (2.66) and (2.67).
- S-13. Go to step S-11 until all the temporal iterations are completed.
- S-14. A barrier-type synchronization point ensures all the FVM-TVD related calculations are finished in each subdomain within current time level.
- S-15. Integrate the semi-discrete equations with respect to time by using 4th order Runge-Kutta method to get $\hat{C}_{(m)}^{n+1}$.
- S-16. Go to step S-6.

A structured grid is generated for the entire computational domain having lengths L_X , L_Y , and L_Z by using N_X , N_Y , and N_Z number of finite volumes (or cells) in each direction. The resulting cell sizes are $\Delta X = L_X/N_X$, $\Delta Y = L_Y/N_Y$, and $\Delta Z = L_Z/N_Z$. Then the entire domain is discretized using $N_{DDX} \times N_{DDY} \times N_{DDZ}$ cuboidal subdomains with each subdomain having the same number of cells $\frac{N_X}{N_{DDX}} \times \frac{N_Y}{N_{DDY}} \times \frac{N_Z}{N_{DDZ}}$. We use the same number of cells in each subdomain to assist in load balancing [Chandra *et al.*, 2001]. An example is shown in Figure 3.2 (a). Each subdomain is assigned to a distinct CPU thread with a distinct thread identity number obtained by using the

OpenMP runtime function `OMP_GET_THREAD_NUM()`. Then the loop indices in each subdomain are set up using the algorithm shown as follows

$$\begin{aligned}
 & thread_{ID} = \text{OMP_GET_THREAD_NUM}() \\
 & \begin{cases} k_{start} = \text{INT} \left(\frac{thread_{ID}}{N_{DDX} \cdot N_{DDY}} \right) \cdot \frac{N_Z}{N_{DDZ}} + 1 \\ k_{end} = \left[\text{INT} \left(\frac{thread_{ID}}{N_{DDX} \cdot N_{DDY}} \right) + 1 \right] \cdot \frac{N_Z}{N_{DDZ}} \end{cases} \\
 & \begin{cases} j_{start} = \text{MOD} \left[\text{INT} \left(\frac{thread_{ID}}{N_{DDX}} \right), N_{DDY} \right] \cdot \frac{N_Y}{N_{DDY}} + 1 \\ j_{end} = \left\{ \text{MOD} \left[\text{INT} \left(\frac{thread_{ID}}{N_{DDX}} \right), N_{DDY} \right] + 1 \right\} \cdot \frac{N_Y}{N_{DDY}} \end{cases} \quad (3.1) \\
 & \begin{cases} i_{start} = \text{MOD}(thread_{ID}, N_{DDX}) \cdot \frac{N_X}{N_{DDX}} + 1 \\ i_{end} = [\text{MOD}(thread_{ID}, N_{DDX}) + 1] \cdot \frac{N_X}{N_{DDX}} \end{cases}
 \end{aligned}$$

The initial conditions Eq. (2.70) are applied in each subdomain in parallel and then the computation follows the steps as mentioned above. All the calculations in the FVM-TVD implementation are conducted in parallel. A barrier-type synchronization point is used to make sure all the iterations in all the threads have completed before moving to the next time level. After the NODDE-FVM-TVD spatial discretization is performed, the semi-discrete equations are integrated over time by using 4th order Runge-Kutta method [Hirsh, 2007]. Then the output data for estimated concentrations are generated and the iteration continues until the final simulation time is reached. During the parallel implementation, to satisfy the transmission conditions equations (2.66) and (2.67) the data of the outermost two layers of cells in each subdomain (the region with solid gray hatched lines shown in Figure 3.2 (b)) are needed by the adjacent subdomains to calculate the fluxes on the subdomain interfaces when using the FVM-TVD scheme. Since OpenMP is a type of shared-memory parallel implementation, the whole memory can be accessed by each thread, no extra coding is needed to handle the data communication between different threads.

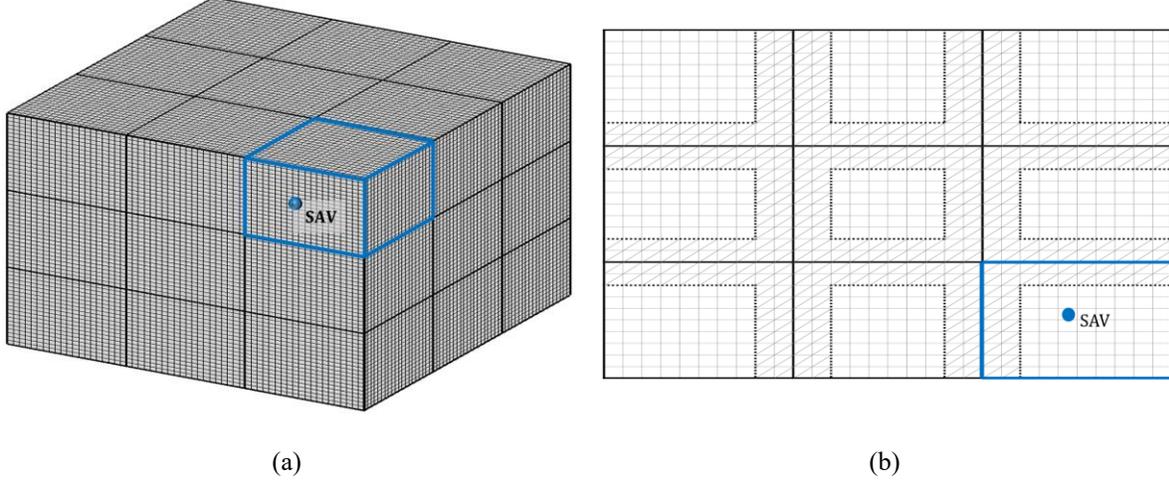


Figure 3.2 (a) An example of the domain decomposition discretization (bold black lines indicate the locations of the subdomain interfaces). The Luenberger estimator models the subdomain with blue boundary lines where the SAV resides. All the remainder subdomains are modeled by the naive estimator. (b) The cells are shown in 2D for better visualization. The shaded cells are needed to calculate fluxes on subdomain interfaces when applying transmission conditions by using FVM-TVD scheme.

3.2 The HT-NODDE-FVM for the Hybrid Estimator

3.2.1 FVM-TVD Spatial Discretization

The hybrid estimator equations (2.64) and (2.65) along with the transmission conditions equations (2.66) and (2.67) applied to a subdomain in Figure 1.2 can be expanded as

$$\begin{aligned}
\frac{\partial \hat{C}_{(m)}}{\partial t} = & -\frac{\partial(\hat{C}_{(m)}U)}{\partial X} - \frac{\partial(\hat{C}_{(m)}V)}{\partial Y} - \frac{\partial(\hat{C}_{(m)}W)}{\partial Z} + \frac{\partial}{\partial X} \left(K_{XX} \frac{\partial \hat{C}_{(m)}}{\partial X} \right) \\
& + \frac{\partial}{\partial Y} \left(K_{YY} \frac{\partial \hat{C}_{(m)}}{\partial Y} \right) + \frac{\partial}{\partial Z} \left(K_{ZZ} \frac{\partial \hat{C}_{(m)}}{\partial Z} \right) \\
& + \begin{cases} R_{(mL)}(\Theta_s(t), t), & \text{for Luenberger estimator} \\ 0, & \text{for naive estimator} \end{cases}
\end{aligned} \tag{3.2}$$

Integration of Eq. (3.2) follows the FVM implementation with TVD and Runge-Kutta Integration [Egorova *et al.*, 2016]. Integrating over a finite volume $\Omega_{ijk(m)}$ as shown in Figure 3.3 and using the Gauss theorem

$$\iiint_{\Omega} \nabla \cdot \mathbf{F} \, d\Omega = \iint_A \mathbf{F} \cdot \hat{\mathbf{n}} \, dA \quad (3.3)$$

yields

$$\begin{aligned} \frac{\partial}{\partial t} \iiint_{\Omega_{ijk(m)}} \hat{C}_{ijk(m)} \, d\Omega + \iint_{A_{ijk(m)}} \hat{\mathbf{F}}_{ijk(m)} \cdot d\mathbf{A} \\ = \begin{cases} \iiint_{\Omega_{ijk(m_L)}} R_{ijk(m_L)} \, d\Omega, & \text{for Luenberger estimator} \\ 0, & \text{for naive estimator} \end{cases} \end{aligned} \quad (3.4)$$

where the indices i, j, k in each subdomain m follow the global ordering.

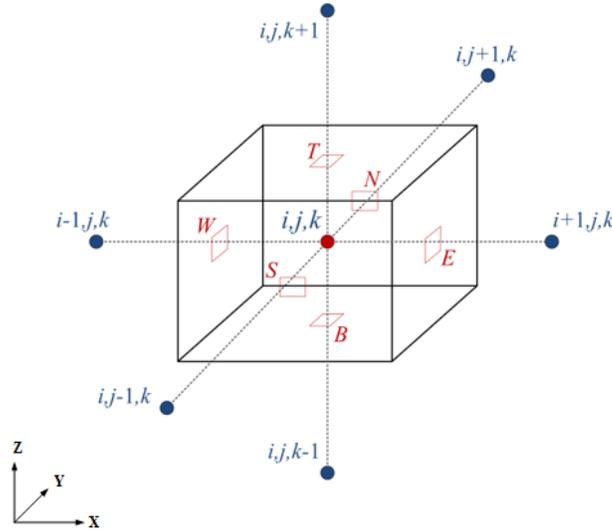


Figure 3.3 An arbitrary finite volume $\Omega_{ijk(m)}$ in subdomain m used in evaluation of advective and diffusive fluxes.

Apply the finite volume discretization to Eq. (3.4), the volume integrals will be expressed as the averaged values of the physical quantities over the finite volume and the surface integrals will be expressed as the sum of the physical quantities over all the boundary surfaces of the finite volume

$$\begin{aligned} \Omega_{ijk(m)} \frac{\partial \hat{C}_{ijk(m)}}{\partial t} + \sum_{l=1}^6 (\hat{\mathbf{F}}_{ijk(m)} \cdot \mathbf{n}) A_{ijk(m),l} \\ = \begin{cases} \Omega_{ijk(m_L)} R_{ijk(m_L)}, & \text{for Luenberger estimator} \\ 0, & \text{for naive estimator} \end{cases} \end{aligned} \quad (3.5)$$

In Eq. (3.5), $\hat{\mathbf{F}}_{ijk(m)}$ is the flux vector,

$$\begin{aligned} \hat{\mathbf{F}}_{ijk(m)} &= \hat{F}_{X,ijk(m)} \mathbf{i} + \hat{F}_{Y,ijk(m)} \mathbf{j} + \hat{F}_{Z,ijk(m)} \mathbf{k} \\ &= (\hat{f}_{X,ijk(m)}^A + \hat{f}_{X,ijk(m)}^D) \mathbf{i} + (\hat{f}_{Y,ijk(m)}^A + \hat{f}_{Y,ijk(m)}^D) \mathbf{j} \\ &\quad + (\hat{f}_{Z,ijk(m)}^A + \hat{f}_{Z,ijk(m)}^D) \mathbf{k} \end{aligned} \quad (3.6)$$

where the superscripts *A* and *D* mean the advective and diffusive fluxes, respectively. The advective flux components are

$$\hat{f}_{X,ijk(m)}^A = \hat{C}_{ijk(m)} U, \quad \hat{f}_{Y,ijk(m)}^A = \hat{C}_{ijk(m)} V, \quad \hat{f}_{Z,ijk(m)}^A = \hat{C}_{ijk(m)} W \quad (3.7)$$

and the diffusive flux components are

$$\begin{aligned} \hat{f}_{X,ijk(m)}^D &= -K_{XX} \frac{\partial \hat{C}_{ijk(m)}}{\partial X}, & \hat{f}_{Y,ijk(m)}^D &= -K_{YY} \frac{\partial \hat{C}_{ijk(m)}}{\partial Y}, \\ \hat{f}_{Z,ijk(m)}^D &= -K_{ZZ} \frac{\partial \hat{C}_{ijk(m)}}{\partial Z} \end{aligned} \quad (3.8)$$

Substitute Eq. (3.6) into Eq. (3.5), yields

$$\begin{aligned} \frac{\partial \hat{C}_{ijk(m)}}{\partial t} &= -\frac{1}{\Omega_{ijk(m)}} (\hat{F}_{ijk(m)}^E A_{ijk(m)}^E - \hat{F}_{ijk(m)}^W A_{ijk(m)}^W + \hat{F}_{ijk(m)}^N A_{ijk(m)}^N \\ &\quad - \hat{F}_{ijk(m)}^S A_{ijk(m)}^S + \hat{F}_{ijk(m)}^T A_{ijk(m)}^T - \hat{F}_{ijk(m)}^B A_{ijk(m)}^B) \\ &\quad + \begin{cases} R_{ijk(m_L)}, & \text{for Luenberger estimator} \\ 0, & \text{for naive estimator} \end{cases} \end{aligned} \quad (3.9)$$

where $\hat{F}_{ijk(m)}^E$ is the total flux on the east surface of the control volume $\Omega_{ijk(m)}$, and similarly $\hat{F}_{ijk(m)}^W$ is for the west surface, $\hat{F}_{ijk(m)}^N$ is for the north surface, $\hat{F}_{ijk(m)}^S$ is for the south surface, $\hat{F}_{ijk(m)}^T$ is for the top surface, and $\hat{F}_{ijk(m)}^B$ is for the bottom surfaces

$$\hat{F}_{ijk(m)}^l = \hat{f}_X^A{}^l|_{ijk(m)} + \hat{f}_X^D{}^l|_{ijk(m)}, \quad l = E, W, N, S, T, B \quad (3.10)$$

$A_{ijk(m)}^E$ is the area of the east surface of the control volume $\Omega_{ijk(m)}$, and similarly for $A_{ijk(m)}^W$, $A_{ijk(m)}^N$, $A_{ijk(m)}^S$, $A_{ijk(m)}^T$, and $A_{ijk(m)}^B$; $R_{ijk(m_L)}$ is the output injection term Eq. (2.71) in discretized form

$$R_{ijk(m_L)} = \begin{cases} 0, & \Theta_s(t) \notin (i, j, k) \\ \Omega_{ijk(m_L)} \Lambda [C(\Theta_s(t), t) - \hat{C}(\Theta_s(t), t)], & \Theta_s(t) \in (i, j, k) \end{cases} \quad (3.11)$$

The advective flux on a finite volume surface and on an interface between two subdomains is approximated using the total variation diminishing (TVD) scheme [Thuburn, 1997]. For a finite volume which is not located at any of the computational domain boundaries, the advective fluxes are approximated as follows:

In X -direction, if $U > 0$, then

$$\begin{aligned} \hat{f}_X^A{}^E|_{ijk(m)} &= \hat{C}_{i,j,k(m)} U_{i,j,k(m)} + \frac{1}{2} \xi(r_E^+) (\hat{C}_{i+1,j,k(m)} U_{i+1,j,k(m)} - \hat{C}_{i,j,k(m)} U_{i,j,k(m)}) \\ \hat{f}_X^A{}^W|_{ijk(m)} &= \hat{C}_{i-1,j,k(m)} U_{i-1,j,k(m)} \\ &\quad + \frac{1}{2} \xi(r_W^+) (\hat{C}_{i,j,k(m)} U_{i,j,k(m)} - \hat{C}_{i-1,j,k(m)} U_{i-1,j,k(m)}) \end{aligned} \quad (3.12)$$

and if $U < 0$, then

$$\begin{aligned} \hat{f}_X^A{}^E|_{ijk(m)} &= \hat{C}_{i+1,j,k(m)} U_{i+1,j,k(m)} \\ &\quad + \frac{1}{2} \xi(r_E^-) (\hat{C}_{i,j,k(m)} U_{i,j,k(m)} - \hat{C}_{i+1,j,k(m)} U_{i+1,j,k(m)}) \end{aligned} \quad (3.13)$$

$$\hat{f}_X^A{}^W|_{ijk(m)} = \hat{C}_{i,j,k(m)} U_{i,j,k(m)} + \frac{1}{2} \xi(r_W^-) (\hat{C}_{i-1,j,k(m)} U_{i-1,j,k(m)} - \hat{C}_{i,j,k(m)} U_{i,j,k(m)})$$

In equations (3.12) and (3.13), r is the ratio of local upstream concentration gradient to the local downstream concentration gradient, and the superscripts “+” and “-” denote the positive and negative wind speed directions respectively, while the subscripts “ E ” and “ W ” denote the wind speed across the east surface and west surface of the finite volume $\Omega_{ijk(m)}$ respectively.

$$\begin{aligned}
r_E^+ &= \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i-1,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}}, & r_W^+ &= \frac{\hat{C}_{i-1,j,k(m)} - \hat{C}_{i-2,j,k(m)}}{X_{i-1,j,k(m)} - X_{i-2,j,k(m)}} \\
&= \frac{\hat{C}_{i+1,j,k(m)} - \hat{C}_{i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}}, & &= \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i-1,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}} \\
r_E^- &= \frac{\hat{C}_{i+2,j,k(m)} - \hat{C}_{i+1,j,k(m)}}{X_{i+2,j,k(m)} - X_{i+1,j,k(m)}}, & r_W^- &= \frac{\hat{C}_{i+1,j,k(m)} - \hat{C}_{i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}} \\
&= \frac{\hat{C}_{i+1,j,k(m)} - \hat{C}_{i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}}, & &= \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i-1,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}}
\end{aligned} \tag{3.14}$$

$0 \leq \xi(r) \leq 1$ is the limiter function of the ratio r associated with the TVD scheme. When $\xi(r) = 0$,

$$\begin{aligned}
\hat{f}_X^A|_{ijk(m)}^E &= \begin{cases} \hat{C}_{i,j,k(m)} U_{i,j,k(m)}, & U > 0 \\ \hat{C}_{i+1,j,k(m)} U_{i+1,j,k(m)}, & U < 0 \end{cases} \\
\hat{f}_X^A|_{ijk(m)}^W &= \begin{cases} \hat{C}_{i-1,j,k(m)} U_{i-1,j,k(m)}, & U > 0 \\ \hat{C}_{i,j,k(m)} U_{i,j,k(m)}, & U < 0 \end{cases}
\end{aligned} \tag{3.15}$$

which is the first order upwind scheme. The first order scheme has the dissipation effect at discontinuities, which will deviate the numerical solution from the analytical solution. When $\xi(r) = 1$,

$$\begin{aligned}
\hat{f}_X^A|_{ijk(m)}^E &= \frac{1}{2} (\hat{C}_{i,j,k(m)} U_{i,j,k(m)} + \hat{C}_{i+1,j,k(m)} U_{i+1,j,k(m)}), & U > 0 \text{ or } U < 0 \\
\hat{f}_X^A|_{ijk(m)}^W &= \frac{1}{2} (\hat{C}_{i-1,j,k(m)} U_{i-1,j,k(m)} + \hat{C}_{i,j,k(m)} U_{i,j,k(m)}), & U > 0 \text{ or } U < 0
\end{aligned} \tag{3.16}$$

which is the second order central difference scheme. The second order scheme has the dispersion effect which will cause ripples at discontinuities. This will reduce the accuracy of the numerical solution. So, there could exist discontinuities in the numerical solutions obtained from the TVD scheme. To mitigate the negative effects of the dissipation stemming from the first order scheme and the dispersion stemming from the second order scheme, the limiter function $\xi(r)$ is employed. For the numerical implementation in this thesis, the limiter function is chosen as the Min-Mod [Kurganov and Tadmor, 2000]

$$\xi(r) = \begin{cases} \min(r, 1), & r > 0 \\ 0, & r \leq 0 \end{cases} \quad (3.17)$$

Similarly, in Y -direction, if $V > 0$, then

$$\begin{aligned} \hat{f}_Y^A \Big|_{ijk(m)}^N &= \hat{C}_{i,j,k(m)} V_{i,j,k(m)} + \frac{1}{2} \xi(r_N^+) (\hat{C}_{i,j+1,k(m)} V_{i,j+1,k(m)} - \hat{C}_{i,j,k(m)} V_{i,j,k(m)}) \\ \hat{f}_Y^A \Big|_{ijk(m)}^S &= \hat{C}_{i,j-1,k(m)} V_{i,j-1,k(m)} \\ &\quad + \frac{1}{2} \xi(r_S^+) (\hat{C}_{i,j,k(m)} V_{i,j,k(m)} - \hat{C}_{i,j-1,k(m)} V_{i,j-1,k(m)}) \end{aligned} \quad (3.18)$$

and if $V < 0$, then

$$\begin{aligned} \hat{f}_Y^A \Big|_{ijk(m)}^N &= \hat{C}_{i,j+1,k(m)} V_{i,j+1,k(m)} \\ &\quad + \frac{1}{2} \xi(r_N^-) (\hat{C}_{i,j,k(m)} V_{i,j,k(m)} - \hat{C}_{i,j+1,k(m)} V_{i,j+1,k(m)}) \\ \hat{f}_Y^A \Big|_{ijk(m)}^S &= \hat{C}_{i,j,k(m)} V_{i,j,k(m)} + \frac{1}{2} \xi(r_S^-) (\hat{C}_{i,j-1,k(m)} V_{i,j-1,k(m)} - \hat{C}_{i,j,k(m)} V_{i,j,k(m)}) \end{aligned} \quad (3.19)$$

where

$$\begin{aligned} r_N^+ &= \frac{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j-1,k(m)}}{Y_{i,j,k(m)} - Y_{i-1,j,k(m)}}}{\frac{\hat{C}_{i,j+1,k(m)} - \hat{C}_{i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}}}, & r_S^+ &= \frac{\frac{\hat{C}_{i,j-1,k(m)} - \hat{C}_{i,j-2,k(m)}}{Y_{i,j-1,k(m)} - Y_{i,j-2,k(m)}}}{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j-1,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}}} \\ r_N^- &= \frac{\frac{\hat{C}_{i,j+2,k(m)} - \hat{C}_{i,j+1,k(m)}}{Y_{i,j+2,k(m)} - Y_{i,j+1,k(m)}}}{\frac{\hat{C}_{i,j+1,k(m)} - \hat{C}_{i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}}}, & r_S^- &= \frac{\frac{\hat{C}_{i,j+1,k(m)} - \hat{C}_{i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}}}{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j-1,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}}} \end{aligned} \quad (3.20)$$

The limiter function is the same as Eq. (3.17).

In Z -direction, if $W > 0$, then

$$\hat{f}_Z^A \Big|_{ijk(m)}^T = \hat{C}_{i,j,k(m)} W_{i,j,k(m)} + \frac{1}{2} \xi(r_T^+) (\hat{C}_{i,j,k+1(m)} W_{i,j,k+1(m)} - \hat{C}_{i,j,k(m)} W_{i,j,k(m)}) \quad (3.21)$$

$$\begin{aligned}\hat{f}_Z^A|_{ijk(m)}^B &= \hat{C}_{i,j,k-1(m)}W_{i,j,k-1(m)} \\ &\quad + \frac{1}{2}\xi(r_B^+)(\hat{C}_{i,j,k(m)}W_{i,j,k(m)} - \hat{C}_{i,j,k-1(m)}W_{i,j,k-1(m)})\end{aligned}$$

and if $W < 0$, then

$$\begin{aligned}\hat{f}_Z^A|_{ijk(m)}^T &= \hat{C}_{i,j,k+1(m)}W_{i,j,k+1(m)} \\ &\quad + \frac{1}{2}\xi(r_T^-)(\hat{C}_{i,j,k(m)}W_{i,j,k(m)} - \hat{C}_{i,j,k+1(m)}W_{i,j,k+1(m)})\end{aligned}\quad (3.22)$$

$$\hat{f}_Z^A|_{ijk(m)}^B = \hat{C}_{i,j,k(m)}W_{i,j,k(m)} + \frac{1}{2}\xi(r_B^-)(\hat{C}_{i,j,k-1(m)}W_{i,j,k-1(m)} - \hat{C}_{i,j,k(m)}W_{i,j,k(m)})$$

where

$$\begin{aligned}r_T^+ &= \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j,k-1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}}, & r_B^+ &= \frac{\hat{C}_{i,j,k-1(m)} - \hat{C}_{i,j,k-2(m)}}{Z_{i,j,k-1(m)} - Z_{i,j,k-2(m)}} \\ & \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j,k+1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k+1(m)}}, & & \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j,k-1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}}\end{aligned}\quad (3.23)$$

$$\begin{aligned}r_T^- &= \frac{\hat{C}_{i,j,k+2(m)} - \hat{C}_{i,j,k+1(m)}}{Z_{i,j,k+2(m)} - Z_{i,j,k+1(m)}}, & r_B^- &= \frac{\hat{C}_{i,j,k+1(m)} - \hat{C}_{i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}} \\ & \frac{\hat{C}_{i,j,k+1(m)} - \hat{C}_{i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}}, & & \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j,k-1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}}\end{aligned}$$

The limiter function is also the same as Eq. (3.17).

The diffusive flux on a finite volume surface and on an interface between two subdomains is approximated using the central differencing scheme. For a finite volume which is not located at any of the computational domain boundaries, the diffusive fluxes are approximated as follows:

In X -direction,

$$\begin{aligned}\hat{f}_X^D|_{ijk(m)}^E &= -K_{XX}|_{ijk(m)}^E \frac{\hat{C}_{i+1,j,k(m)} - \hat{C}_{i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}} \\ \hat{f}_X^D|_{ijk(m)}^W &= -K_{XX}|_{ijk(m)}^W \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i-1,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}}\end{aligned}\quad (3.24)$$

where $K_{XX}|_{ijk(m)}^E$ and $K_{XX}|_{ijk(m)}^W$ are the eddy diffusivities on the east and west surfaces of the finite volume, respectively. They are approximated by

$$\begin{aligned}
K_{XX}|_{ijk(m)}^E &= \frac{X_{i+1,j,k(m)} - X_{\text{surf},i+1,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}} K_{XX,i,j,k(m)} \\
&\quad + \frac{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}} K_{XX,i+1,j,k(m)} \\
K_{XX}|_{ijk(m)}^W &= \frac{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}} K_{XX,i-1,j,k(m)} \\
&\quad + \frac{X_{\text{surf},i,j,k(m)} - X_{i-1,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}} K_{XX,i,j,k(m)}
\end{aligned} \tag{3.25}$$

where the subscript ‘‘surf’’ means the coordinate of the finite volume surfaces.

In Y-direction,

$$\begin{aligned}
\hat{f}_Y^D|_{ijk(m)}^N &= -K_{YY}|_{ijk(m)}^N \frac{\hat{C}_{i,j+1,k(m)} - \hat{C}_{i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}} \\
\hat{f}_Y^D|_{ijk(m)}^S &= -K_{YY}|_{ijk(m)}^S \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j-1,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}}
\end{aligned} \tag{3.26}$$

where $K_{YY}|_{ijk(m)}^N$ and $K_{YY}|_{ijk(m)}^S$ are the eddy diffusivities on the north and south surfaces of the finite volume, respectively. They are approximated by

$$\begin{aligned}
K_{YY}|_{ijk(m)}^N &= \frac{Y_{i,j+1,k(m)} - Y_{\text{surf},i,j+1,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}} K_{YY,i,j,k(m)} \\
&\quad + \frac{Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}} K_{YY,i,j+1,k(m)} \\
K_{YY}|_{ijk(m)}^S &= \frac{Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}} K_{YY,i,j-1,k(m)} \\
&\quad + \frac{Y_{\text{surf},i,j,k(m)} - Y_{i,j-1,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}} K_{YY,i,j,k(m)}
\end{aligned} \tag{3.27}$$

In Z-direction,

$$\begin{aligned}\hat{f}_Z^D|_{ijk(m)}^T &= -K_{ZZ}|_{ijk(m)}^T \frac{\hat{C}_{i,j,k+1(m)} - \hat{C}_{i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}} \\ \hat{f}_Z^D|_{ijk(m)}^B &= -K_{ZZ}|_{ijk(m)}^B \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j,k-1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}}\end{aligned}\quad (3.28)$$

where $K_{ZZ}|_{ijk(m)}^T$ and $K_{ZZ}|_{ijk(m)}^B$ are the eddy diffusivities on the top and bottom surfaces of the finite volume, respectively. They are approximated by

$$\begin{aligned}K_{ZZ}|_{ijk(m)}^T &= \frac{Z_{i,j,k+1(m)} - Z_{\text{surf},i,j,k+1(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}} K_{ZZ,i,j,k(m)} \\ &\quad + \frac{Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}} K_{ZZ,i,j,k+1(m)} \\ K_{ZZ}|_{ijk(m)}^B &= \frac{Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}} K_{ZZ,i,j,k-1(m)} \\ &\quad + \frac{Z_{\text{surf},i,j,k(m)} - Z_{i,j,k-1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}} K_{ZZ,i,j,k(m)}\end{aligned}\quad (3.29)$$

Substitute equations (3.10) – (3.29) into Eq. (3.9), yields

$$\begin{aligned}\frac{\partial \hat{C}_{ijk(m)}}{\partial t} &= -\frac{1}{\Omega_{ijk(m)}} (a_{ijk(m)}^E \hat{C}_{i+1,j,k(m)} + a_{ijk(m)}^W \hat{C}_{i-1,j,k(m)} + a_{ijk(m)}^N \hat{C}_{i,j+1,k(m)} \\ &\quad + a_{ijk(m)}^S \hat{C}_{i,j-1,k(m)} + a_{ijk(m)}^T \hat{C}_{i,j,k+1(m)} + a_{ijk(m)}^B \hat{C}_{i,j,k-1(m)} \\ &\quad + a_{ijk(m)}^C \hat{C}_{i,j,k(m)}) + S_{ijk(m)}^{DC} \\ &\quad + \begin{cases} R_{ijk(m)_L}, & \text{for Luenberger estimator} \\ 0, & \text{for naive estimator} \end{cases}\end{aligned}\quad (3.30)$$

where $a_{ijk(m)}^E$, $a_{ijk(m)}^W$, $a_{ijk(m)}^N$, $a_{ijk(m)}^S$, $a_{ijk(m)}^T$, $a_{ijk(m)}^B$, and $a_{ijk(m)}^C$ are the weights generated from the FVM discretization,

$$a_{ijk(m)}^E = \left[\min(U_{i+1,j,k(m)}, 0) - \frac{K_{XX}|_{ijk(m)}^E}{X_{i+1,j,k(m)} - X_{i,j,k(m)}} \right] A_{ijk(m)}^E \quad (3.31)$$

$$a_{ijk(m)}^W = \left[-\max(U_{i-1,j,k(m)}, 0) - \frac{K_{XX}|_{ijk(m)}^W}{X_{i,j,k(m)} - X_{i-1,j,k(m)}} \right] A_{ijk(m)}^W$$

$$a_{ijk(m)}^N = \left[\min(V_{i,j+1,k(m)}, 0) - \frac{K_{YY}|_{ijk(m)}^N}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}} \right] A_{ijk(m)}^N \quad (3.32)$$

$$a_{ijk(m)}^S = \left[-\max(V_{i,j-1,k(m)}, 0) - \frac{K_{YY}|_{ijk(m)}^S}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}} \right] A_{ijk(m)}^S$$

$$a_{ijk(m)}^T = \left[\min(W_{i,j,k+1(m)}, 0) - \frac{K_{ZZ}|_{ijk(m)}^T}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}} \right] A_{ijk(m)}^T \quad (3.33)$$

$$a_{ijk(m)}^B = \left[-\max(W_{i,j,k-1(m)}, 0) - \frac{K_{ZZ}|_{ijk(m)}^B}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}} \right] A_{ijk(m)}^B$$

$$a_{ijk(m)}^C = -a_{ijk(m)}^E - a_{ijk(m)}^W - a_{ijk(m)}^N - a_{ijk(m)}^S - a_{ijk(m)}^T - a_{ijk(m)}^B$$

$$+ [\max(U_{i,j,k(m)}, 0) + \min(U_{i+1,j,k(m)}, 0)] A_{ijk(m)}^E$$

$$- [\max(U_{i-1,j,k(m)}, 0) + \min(U_{i,j,k(m)}, 0)] A_{ijk(m)}^W$$

$$+ [\max(V_{i,j,k(m)}, 0) + \min(V_{i,j+1,k(m)}, 0)] A_{ijk(m)}^N$$

$$- [\max(V_{i,j-1,k(m)}, 0) + \min(V_{i,j,k(m)}, 0)] A_{ijk(m)}^S \quad (3.34)$$

$$+ [\max(W_{i,j,k(m)}, 0) + \min(W_{i,j,k+1(m)}, 0)] A_{ijk(m)}^T$$

$$- [\max(W_{i,j,k-1(m)}, 0) + \min(W_{i,j,k(m)}, 0)] A_{ijk(m)}^B$$

$S_{ijk(m)}^{DC}$ is the deferred correction source term generated by the TVD scheme,

$$\begin{aligned}
S_{ijk(m)}^{DC} = & -\frac{1}{\Omega_{ijk(m)}} \left\{ \frac{1}{2} [\alpha_E \xi(r_E^+) - (1 - \alpha_E) \xi(r_E^-)] (\hat{C}_{i+1,j,k(m)} U_{i+1,j,k(m)} \right. \\
& - \hat{C}_{i,j,k(m)} U_{i,j,k(m)}) A_{ijk(m)}^E \\
& - \frac{1}{2} [\alpha_W \xi(r_W^+) - (1 - \alpha_W) \xi(r_W^-)] (\hat{C}_{i,j,k(m)} U_{i,j,k(m)} \\
& - \hat{C}_{i-1,j,k(m)} U_{i-1,j,k(m)}) A_{ijk(m)}^W \\
& + \frac{1}{2} [\alpha_N \xi(r_N^+) - (1 - \alpha_N) \xi(r_N^-)] (\hat{C}_{i,j+1,k(m)} V_{i,j+1,k(m)} \\
& - \hat{C}_{i,j,k(m)} V_{i,j,k(m)}) A_{ijk(m)}^N \\
& - \frac{1}{2} [\alpha_S \xi(r_S^+) - (1 - \alpha_S) \xi(r_S^-)] (\hat{C}_{i,j,k(m)} V_{i,j,k(m)} \\
& - \hat{C}_{i,j-1,k(m)} V_{i,j-1,k(m)}) A_{ijk(m)}^S \\
& + \frac{1}{2} [\alpha_T \xi(r_T^+) - (1 - \alpha_T) \xi(r_T^-)] (\hat{C}_{i,j,k+1(m)} W_{i,j,k+1(m)} \\
& - \hat{C}_{i,j,k(m)} W_{i,j,k(m)}) A_{ijk(m)}^T \\
& - \frac{1}{2} [\alpha_B \xi(r_B^+) - (1 - \alpha_B) \xi(r_B^-)] (\hat{C}_{i,j,k(m)} W_{i,j,k(m)} \\
& \left. - \hat{C}_{i,j,k-1(m)} W_{i,j,k-1(m)}) A_{ijk(m)}^B \right\} \tag{3.35}
\end{aligned}$$

where

$$\begin{aligned}
\alpha_E = \begin{cases} 1, & U_{i+1,j,k(m)} > 0 \\ 0, & U_{i+1,j,k(m)} < 0 \end{cases}; \quad \alpha_W = \begin{cases} 1, & U_{i-1,j,k(m)} > 0 \\ 0, & U_{i-1,j,k(m)} < 0 \end{cases} \\
\alpha_N = \begin{cases} 1, & V_{i,j+1,k(m)} > 0 \\ 0, & V_{i,j+1,k(m)} < 0 \end{cases}; \quad \alpha_S = \begin{cases} 1, & V_{i,j-1,k(m)} > 0 \\ 0, & V_{i,j-1,k(m)} < 0 \end{cases} \\
\alpha_T = \begin{cases} 1, & W_{i,j,k+1(m)} > 0 \\ 0, & W_{i,j,k+1(m)} < 0 \end{cases}; \quad \alpha_B = \begin{cases} 1, & W_{i,j,k-1(m)} > 0 \\ 0, & W_{i,j,k-1(m)} < 0 \end{cases} \tag{3.36}
\end{aligned}$$

For a finite volume which is located at one of the computational domain boundaries, the boundaries are treated by using Leonard mirror node extrapolation [Versteeg and Malalasekera, 2007]. For a finite volume which is located on the west boundary of the computational domain,

the advective fluxes on the west surface of the finite volume $\Omega_{ijk(m)}$ in equations (3.12) and (3.13) become, for Dirichlet boundary conditions as shown in Figure 3.4 (a),

$$\hat{f}_X^A \Big|_{ijk(m)}^W = U_{WB,j,k(m)} \hat{C}_{WB}, \quad \text{with } i = 1 \quad (3.37)$$

And for Neumann boundary conditions as shown in Figure 3.4 (b), \hat{C}_{WB} is not given directly. Instead, \hat{C}_{WB} is approximated from the given Neumann boundary condition as follows,

$$\frac{\partial \hat{C}}{\partial X} \Big|_{WB} = \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{WB}}{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}} = N_{WB}, \quad \text{with } i = 1 \quad (3.38)$$

Equation (3.38) can be rewritten as

$$\hat{C}_{WB} = \hat{C}_{i,j,k(m)} - N_{WB}(X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}), \quad \text{with } i = 1 \quad (3.39)$$

Substitute Eq. (3.39) into Eq. (3.37), yields

$$\hat{f}_X^A \Big|_{ijk(m)}^W = U_{WB,j,k(m)} [\hat{C}_{i,j,k(m)} - N_{WB}(X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)})], \quad \text{with } i = 1 \quad (3.40)$$

The wind speed at the west boundary $U_{WB,j,k(m)}$ in equations (3.37) and (3.40) is approximated by

$$U_{WB,j,k(m)} = U_{i,j,k(m)} - \frac{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}} (U_{i+1,j,k(m)} - U_{i,j,k(m)}), \quad (3.41)$$

with $i = 1$

The ratio r_E^+ in Eq. (3.14) becomes, for Dirichlet boundary conditions,

$$r_E^+ = \frac{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{WB}}{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}}}{\frac{\hat{C}_{i+1,j,k(m)} - \hat{C}_{i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}}}, \quad \text{with } i = 1 \quad (3.42)$$

and for Neumann boundary conditions,

$$r_E^+ = \frac{N_{WB}}{\frac{\hat{C}_{i+1,j,k(m)} - \hat{C}_{i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}}}, \quad \text{with } i = 1 \quad (3.43)$$

r_E^- keeps the same equation but with $i = 1$. The r_W^+ and r_W^- are no longer needed because the advective fluxes on the west surface have been approximated by Eq. (3.37) or Eq. (3.40).

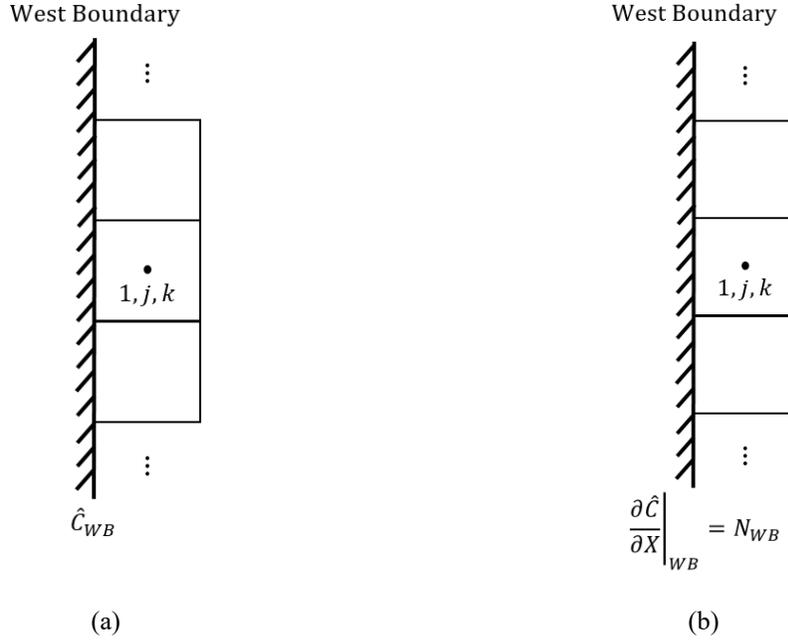


Figure 3.4 Finite volumes at west boundary with (a) Dirichlet boundary condition and (b) Neumann boundary condition (shown in 2D).

The diffusive flux on the west surface of the finite volume $\Omega_{ijk(m)}$ in Eq. (3.24) becomes, for Dirichlet boundary conditions,

$$\hat{f}_X^D \Big|_{ijk(m)}^W = -K_{XX} \Big|_{ijk(m)}^W \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{WB}}{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}}, \quad \text{with } i = 1 \quad (3.44)$$

and for Neumann boundary conditions,

$$\hat{f}_X^D \Big|_{ijk(m)}^W = -K_{XX} \Big|_{ijk(m)}^W N_{WB}, \quad \text{with } i = 1 \quad (3.45)$$

The $K_{XX} \Big|_{ijk(m)}^W$ in Eq. (3.25) becomes

$$K_{XX}|_{ijk(m)}^W = K_{XX,i,j,k(m)} - \frac{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}}{X_{i+1,j,k(m)} - X_{i,j,k(m)}} (K_{XX,i+1,j,k(m)} - K_{XX,i,j,k(m)}), \quad (3.46)$$

with $i = 1$

Then the corresponding results for the finite volumes on west boundary are the same as Eq. (3.30) except that $a_{ijk(m)}^W = 0$ and a new term $S_{ijk(m)}^{BC}$ appears on the right-hand side of the equation,

$$\begin{aligned} \frac{\partial \hat{C}_{ijk(m)}}{\partial t} = & -\frac{1}{\Omega_{ijk(m)}} (a_{ijk(m)}^E \hat{C}_{i+1,j,k(m)} + a_{ijk(m)}^N \hat{C}_{i,j+1,k(m)} + a_{ijk(m)}^S \hat{C}_{i,j-1,k(m)} \\ & + a_{ijk(m)}^T \hat{C}_{i,j,k+1(m)} + a_{ijk(m)}^B \hat{C}_{i,j,k-1(m)} + a_{ijk(m)}^C \hat{C}_{i,j,k(m)}) + S_{ijk(m)}^{DC} \quad (3.47) \\ & + S_{ijk(m)}^{BC} + \begin{cases} R_{ijk(m)_L}, & \text{for Luenberger estimator} \\ 0, & \text{for naive estimator} \end{cases} \end{aligned}$$

where $S_{ijk(m)}^{BC}$ is the boundary condition source term resulting from the treatment for the boundary finite volumes, for Dirichlet boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} \left[- \left(U_{WB,j,k(m)} + \frac{K_{XX}|_{ijk(m)}^W}{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}} \right) A_{ijk(m)}^W \hat{C}_{WB} \right], \quad (3.48)$$

with $i = 1$

and for Neumann boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} [U_{WB,j,k(m)} (X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}) + K_{XX}|_{ijk(m)}^W] A_{ijk(m)}^W N_{WB}, \quad (3.49)$$

with $i = 1$

The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_W^+) = 0$ and $\xi(r_W^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^W = 0$ and the term multiplied by $A_{ijk(m)}^W$ is replaced, for Dirichlet boundary conditions, by

$$\frac{K_{XX}|_{ijk(m)}^W}{X_{i,j,k(m)} - X_{\text{surf},i,j,k(m)}}, \quad \text{with } i = 1 \quad (3.50)$$

and for Neumann boundary conditions, by

$$-U_{WB,j,k(m)} \quad (3.51)$$

Similarly, we can derive the equations for the finite volumes located at east, north, south, top, and bottom boundaries and at the corners formed by south-west, north-west, north-east, south-east, south-top, north-top, north-bottom, south-bottom, south-west-top, north-west-top, north-east-top, south-east-top, south-west-bottom, north-west-bottom, north-east-bottom, and south-east-bottom surfaces after proper boundary treatments.

For a finite volume at the east boundary of the computational domain, the advective fluxes on the east surface of the finite volume $\Omega_{ijk(m)}$ in equations (3.12) and (3.13) become, for Dirichlet boundary conditions,

$$\hat{f}_X^A|_{ijk(m)}^E = U_{EB,j,k(m)} \hat{C}_{EB}, \quad \text{with } i = N_X \quad (3.52)$$

And for Neumann boundary conditions, \hat{C}_{EB} is not given directly. Instead, \hat{C}_{EB} is approximated from the given Neumann boundary condition as follows,

$$\frac{\partial \hat{C}}{\partial X}|_{EB} = \frac{\hat{C}_{EB} - \hat{C}_{i,j,k(m)}}{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}} = N_{EB}, \quad \text{with } i = N_X \quad (3.53)$$

Equation (3.53) can be rewritten as

$$\hat{C}_{EB} = \hat{C}_{i,j,k(m)} + N_{EB} (X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}), \quad \text{with } i = N_X \quad (3.54)$$

Substitute Eq. (3.54) into Eq. (3.52), yields

$$\hat{f}_X^A|_{ijk(m)}^E = U_{EB,j,k(m)} [\hat{C}_{i,j,k(m)} + N_{EB} (X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)})], \quad \text{with } i = N_X \quad (3.55)$$

The wind speed at the east boundary $U_{EB,j,k(m)}$ in equations (3.52) and (3.55) is approximated by

$$U_{EB,j,k(m)} = U_{i,j,k(m)} + \frac{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}} (U_{i,j,k(m)} - U_{i-1,j,k(m)}), \quad (3.56)$$

with $i = N_X$

The ratio r_W^- in Eq. (3.14) becomes, for Dirichlet boundary conditions,

$$r_W^- = \frac{\hat{C}_{EB} - \hat{C}_{i,j,k(m)}}{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}}, \quad \text{with } i = N_X \quad (3.57)$$

$$\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i-1,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}}$$

and for Neumann boundary conditions,

$$r_W^- = \frac{N_{EB}}{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i-1,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}}}, \quad \text{with } i = N_X \quad (3.58)$$

r_W^+ keeps the same equation but with $i = N_X$. The r_E^+ and r_E^- are no longer needed because the advective fluxes on the east surface have been approximated by Eq. (3.52) or Eq. (3.55).

The diffusive flux on the east surface of the finite volume $\Omega_{ijk(m)}$ in Eq. (3.24) becomes, for Dirichlet boundary conditions,

$$\hat{f}_X^D|_{ijk(m)}^E = -K_{XX}|_{ijk(m)}^E \frac{\hat{C}_{EB} - \hat{C}_{i,j,k(m)}}{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}}, \quad \text{with } i = N_X \quad (3.59)$$

and for Neumann boundary conditions,

$$\hat{f}_X^D|_{ijk(m)}^E = -K_{XX}|_{ijk(m)}^E N_{EB}, \quad \text{with } i = N_X \quad (3.60)$$

The $K_{XX}|_{ijk(m)}^E$ in Eq. (3.25) becomes

$$K_{XX}|_{ijk(m)}^E = K_{XX,i,j,k(m)} + \frac{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}}{X_{i,j,k(m)} - X_{i-1,j,k(m)}} (K_{XX,i,j,k(m)} - K_{XX,i-1,j,k(m)}), \quad (3.61)$$

with $i = N_X$

Then the corresponding result for the finite volumes on east boundary is the same as Eq. (3.30) except that $a_{ijk(m)}^E = 0$ and a boundary condition source term $S_{ijk(m)}^{BC}$ appears on the right-hand side of the equation, for Dirichlet boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} \left[\left(U_{EB,j,k(m)} - \frac{K_{XX}|_{ijk(m)}^E}{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}} \right) A_{ijk(m)}^E \hat{C}_{EB} \right], \quad (3.62)$$

with $i = N_X$

and for Neumann boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} \left[U_{EB,j,k(m)} (X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}) - K_{XX}|_{ijk(m)}^E \right] A_{ijk(m)}^E N_{EB}, \quad (3.63)$$

with $i = N_X$

The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_E^+) = 0$ and $\xi(r_E^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^E = 0$ and the term multiplied by $A_{ijk(m)}^E$ is replaced, for Dirichlet boundary conditions, by

$$\frac{K_{XX}|_{ijk(m)}^E}{X_{\text{surf},i+1,j,k(m)} - X_{i,j,k(m)}}, \quad \text{with } i = N_X \quad (3.64)$$

and for Neumann boundary conditions, by

$$U_{EB,j,k(m)} \quad (3.65)$$

For a finite volume at the south boundary of the computational domain, the advective fluxes on the south surface of the finite volume $\Omega_{ijk(m)}$ in equations (3.18) and (3.19) become, for Dirichlet boundary conditions,

$$\hat{f}_Y^A|_{ijk(m)}^S = V_{i,SB,k(m)} \hat{C}_{SB}, \quad \text{with } j = 1 \quad (3.66)$$

and for Neumann boundary conditions,

$$\hat{f}_Y^A|_{ijk(m)}^S = V_{i,SB,k(m)} \left[\hat{C}_{i,j,k(m)} - N_{SB} (Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}) \right], \quad \text{with } j = 1 \quad (3.67)$$

The wind speed at the west boundary $V_{i,SB,k(m)}$ in equations (3.66) and (3.67) is approximated by

$$V_{i,SB,k(m)} = V_{i,j,k(m)} - \frac{Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}} (V_{i,j+1,k(m)} - V_{i,j,k(m)}), \quad (3.68)$$

with $j = 1$

The ratio r_N^+ in Eq. (3.20) becomes, for Dirichlet boundary conditions,

$$r_N^+ = \frac{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{SB}}{Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}}}{\frac{\hat{C}_{i,j+1,k(m)} - \hat{C}_{i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}}}, \quad \text{with } j = 1 \quad (3.69)$$

and for Neumann boundary conditions,

$$r_N^+ = \frac{N_{SB}}{\frac{\hat{C}_{i,j+1,k(m)} - \hat{C}_{i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}}}, \quad \text{with } j = 1 \quad (3.70)$$

r_N^- keeps the same equation but with $j = 1$. The r_S^+ and r_S^- are no longer needed because the advective fluxes on the west surface have been approximated by Eq. (3.66) or Eq. (3.67).

The diffusive flux on the south surface of the finite volume $\Omega_{ijk(m)}$ in Eq. (3.26) becomes, for Dirichlet boundary conditions,

$$\hat{f}_Y^D|_{ijk(m)}^S = -K_{YY}|_{ijk(m)}^S \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{SB}}{Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}}, \quad \text{with } j = 1 \quad (3.71)$$

and for Neumann boundary conditions,

$$\hat{f}_Y^D|_{ijk(m)}^S = -K_{YY}|_{ijk(m)}^S N_{SB}, \quad \text{with } j = 1 \quad (3.72)$$

The $K_{YY}|_{ijk(m)}^S$ in Eq. (3.27) becomes

$$K_{YY}|_{ijk(m)}^S = K_{YY,i,j,k(m)} - \frac{Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}}{Y_{i,j+1,k(m)} - Y_{i,j,k(m)}} (K_{YY,i,j+1,k(m)} - K_{YY,i,j,k(m)}), \quad (3.73)$$

with $j = 1$

Then the corresponding result for the finite volumes on south boundary is the same as Eq. (3.30) except that $a_{ijk(m)}^S = 0$ and a boundary condition source term $S_{ijk(m)}^{BC}$ appears on the right-hand side of the equation, for Dirichlet boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} \left[-\left(V_{i,SB,k(m)} + \frac{K_{YY}|_{ijk(m)}^S}{Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}} \right) A_{ijk(m)}^S \hat{C}_{SB} \right], \quad (3.74)$$

with $j = 1$

and for Neumann boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} [V_{i,SB,k(m)}(Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}) + K_{YY}|_{ijk(m)}^S] A_{ijk(m)}^S N_{SB}, \quad (3.75)$$

with $j = 1$

The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_S^+) = 0$ and $\xi(r_S^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^S = 0$ and the term multiplied by $A_{ijk(m)}^S$ is replaced, for Dirichlet boundary conditions, by

$$\frac{K_{YY}|_{ijk(m)}^S}{Y_{i,j,k(m)} - Y_{\text{surf},i,j,k(m)}}, \quad \text{with } j = 1 \quad (3.76)$$

and for Neumann boundary conditions, by

$$-V_{i,SB,k(m)} \quad (3.77)$$

For a finite volume at the north boundary of the computational domain, the advective fluxes on the north surface of the finite volume $\Omega_{ijk(m)}$ in equations (3.18) and (3.19) become, for Dirichlet boundary conditions,

$$\hat{f}_Y^A|_{ijk(m)}^N = V_{i,NB,k(m)} \hat{C}_{NB}, \quad \text{with } j = N_Y \quad (3.78)$$

and for Neumann boundary conditions,

$$\hat{f}_Y^A|_{ijk(m)}^N = V_{i,NB,k(m)} [\hat{C}_{i,j,k(m)} + N_{NB}(Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)})], \quad \text{with } j = N_Y \quad (3.79)$$

The wind speed at the north boundary $V_{i,NB,k(m)}$ in equations (3.78) and (3.79) is approximated by

$$V_{i,NB,k(m)} = V_{i,j,k(m)} + \frac{Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}} (V_{i,j,k(m)} - V_{i,j-1,k(m)}), \quad (3.80)$$

with $j = N_Y$

The ratio r_S^- in Eq. (3.20) becomes, for Dirichlet boundary conditions,

$$r_S^- = \frac{\frac{\hat{C}_{NB} - \hat{C}_{i,j,k(m)}}{Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}}}{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j-1,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}}}, \quad \text{with } j = N_Y \quad (3.81)$$

and for Neumann boundary conditions,

$$r_S^- = \frac{N_{NB}}{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j-1,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}}}, \quad \text{with } j = N_Y \quad (3.82)$$

r_S^+ keeps the same equation but with $j = N_Y$. The r_N^+ and r_N^- are no longer needed because the advective fluxes on the north surface have been approximated by Eq. (3.78) or Eq. (3.79).

The diffusive flux on the north surface of the finite volume $\Omega_{ijk(m)}$ in Eq. (3.26) becomes, for Dirichlet boundary conditions,

$$\hat{f}_Y^D \Big|_{ijk(m)}^N = -K_{YY} \Big|_{ijk(m)}^N \frac{\hat{C}_{NB} - \hat{C}_{i,j,k(m)}}{Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}}, \quad \text{with } j = N_Y \quad (3.83)$$

and for Neumann boundary conditions,

$$\hat{f}_Y^D \Big|_{ijk(m)}^N = -K_{YY} \Big|_{ijk(m)}^N N_{NB}, \quad \text{with } j = N_Y \quad (3.84)$$

The $K_{YY} \Big|_{ijk(m)}^N$ in Eq. (3.27) becomes

$$K_{YY}|_{ijk(m)}^N = K_{YY,i,j,k(m)} + \frac{Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}}{Y_{i,j,k(m)} - Y_{i,j-1,k(m)}} (K_{YY,i,j,k(m)} - K_{YY,i,j-1,k(m)}), \quad (3.85)$$

with $j = N_Y$

Then the corresponding result for the finite volumes on north boundary is the same as Eq. (3.30) except that $a_{ijk(m)}^N = 0$ and a boundary condition source term $S_{ijk(m)}^{BC}$ appears on the right-hand side of the equation, for Dirichlet boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} \left[\left(V_{i,NB,k(m)} - \frac{K_{YY}|_{ijk(m)}^N}{Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}} \right) A_{ijk(m)}^N \hat{C}_{NB} \right], \quad (3.86)$$

with $j = N_Y$

and for Neumann boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} [V_{i,NB,k(m)} (Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}) - K_{YY}|_{ijk(m)}^N] A_{ijk(m)}^N N_{NB}, \quad (3.87)$$

with $j = N_Y$

The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_N^+) = 0$ and $\xi(r_N^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^N = 0$ and the term multiplied by $A_{ijk(m)}^N$ is replaced, for Dirichlet boundary conditions, by

$$\frac{K_{YY}|_{ijk(m)}^N}{Y_{\text{surf},i,j+1,k(m)} - Y_{i,j,k(m)}}, \quad \text{with } j = N_Y \quad (3.88)$$

and for Neumann boundary conditions, by

$$V_{i,NB,k(m)} \quad (3.89)$$

For a finite volume at the bottom boundary of the computational domain, the advective fluxes on the bottom surface of the finite volume $\Omega_{ijk(m)}$ in equations (3.21) and (3.22) become, for Dirichlet boundary conditions,

$$\hat{f}_Z^A|_{ijk(m)}^B = W_{i,j,BB(m)} \hat{C}_{BB}, \quad \text{with } k = 1 \quad (3.90)$$

and for Neumann boundary conditions,

$$\hat{f}_Z^A|_{ijk(m)}^B = W_{i,j,BB(m)} [\hat{C}_{i,j,k(m)} - N_{BB}(Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)})], \quad \text{with } k = 1 \quad (3.91)$$

The wind speed at the bottom boundary $W_{i,j,BB(m)}$ in equations (3.90) and (3.91) is approximated by

$$W_{i,j,BB(m)} = W_{i,j,k(m)} - \frac{Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}} (W_{i,j,k+1(m)} - W_{i,j,k(m)}), \quad (3.92)$$

with $k = 1$

The ratio r_T^+ in Eq. (3.23) becomes, for Dirichlet boundary conditions,

$$r_T^+ = \frac{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{BB}}{Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}}}{\frac{\hat{C}_{i,j,k+1(m)} - \hat{C}_{i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}}}, \quad \text{with } k = 1 \quad (3.93)$$

and for Neumann boundary conditions,

$$r_T^+ = \frac{N_{BB}}{\frac{\hat{C}_{i,j,k+1(m)} - \hat{C}_{i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}}}, \quad \text{with } k = 1 \quad (3.94)$$

r_T^- keeps the same equation but with $k = 1$. The r_B^+ and r_B^- are no longer needed because the advective fluxes on the west surface have been approximated by Eq. (3.90) or Eq. (3.91).

The diffusive flux on the bottom surface of the finite volume $\Omega_{ijk(m)}$ in Eq. (3.28) becomes, for Dirichlet boundary conditions,

$$\hat{f}_Z^D|_{ijk(m)}^B = -K_{ZZ}|_{ijk(m)}^B \frac{\hat{C}_{i,j,k(m)} - \hat{C}_{BB}}{Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}}, \quad \text{with } k = 1 \quad (3.95)$$

and for Neumann boundary conditions,

$$\hat{f}_Z^D|_{ijk(m)}^B = -K_{ZZ}|_{ijk(m)}^B N_{BB}, \quad \text{with } k = 1 \quad (3.96)$$

The $K_{ZZ}|_{ijk(m)}^B$ in Eq. (3.29) becomes

$$K_{ZZ}|_{ijk(m)}^B = K_{ZZ,i,j,k(m)} - \frac{Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}}{Z_{i,j,k+1(m)} - Z_{i,j,k(m)}} (K_{ZZ,i,j,k+1(m)} - K_{ZZ,i,j,k(m)}), \quad (3.97)$$

with $k = 1$

Then the corresponding result for the finite volumes on bottom boundary is the same as Eq. (3.30) except that $a_{ijk(m)}^B = 0$ and a boundary condition source term $S_{ijk(m)}^{BC}$ appears on the right-hand side of the equation, for Dirichlet boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} \left[-\left(W_{i,j,BB(m)} + \frac{K_{ZZ}|_{ijk(m)}^B}{Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}} \right) A_{ijk(m)}^B \hat{C}_{BB} \right], \quad (3.98)$$

with $k = 1$

and for Neumann boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} [W_{i,j,BB(m)} (Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}) + K_{ZZ}|_{ijk(m)}^B] A_{ijk(m)}^B N_{BB}, \quad (3.99)$$

with $k = 1$

The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_B^+) = 0$ and $\xi(r_B^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^B = 0$ and the term multiplied by $A_{ijk(m)}^B$ is replaced, for Dirichlet boundary conditions, by

$$\frac{K_{ZZ}|_{ijk(m)}^B}{Z_{i,j,k(m)} - Z_{\text{surf},i,j,k(m)}}, \quad \text{with } k = 1 \quad (3.100)$$

and for Neumann boundary conditions, by

$$-W_{i,j,BB(m)} \quad (3.101)$$

For a finite volume at the top boundary of the computational domain, the advective fluxes on the top surface of the finite volume $\Omega_{ijk(m)}$ in equations (3.21) and (3.22) become, for Dirichlet boundary conditions,

$$\hat{f}_Z^A|_{ijk(m)}^T = W_{i,j,TB(m)} \hat{C}_{TB}, \quad \text{with } k = N_Z \quad (3.102)$$

and for Neumann boundary conditions,

$$\hat{f}_Z^A \Big|_{ijk(m)}^T = W_{i,j,TB(m)} [\hat{C}_{i,j,k(m)} + N_{TB} (Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)})], \quad (3.103)$$

with $k = N_Z$

The wind speed at the north boundary $W_{i,j,TB(m)}$ in equations (3.102) and (3.103) is approximated by

$$W_{i,j,TB(m)} = W_{i,j,k(m)} + \frac{Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}} (W_{i,j,k(m)} - W_{i,j,k-1(m)}), \quad (3.104)$$

with $k = N_Z$

The ratio r_B^- in Eq. (3.23) becomes, for Dirichlet boundary conditions,

$$r_B^- = \frac{\frac{\hat{C}_{TB} - \hat{C}_{i,j,k(m)}}{Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}}}{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j,k-1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}}}, \quad \text{with } k = N_Z \quad (3.105)$$

and for Neumann boundary conditions,

$$r_B^- = \frac{N_{TB}}{\frac{\hat{C}_{i,j,k(m)} - \hat{C}_{i,j,k-1(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}}}, \quad \text{with } k = N_Z \quad (3.106)$$

r_B^+ keeps the same equation but with $k = N_Z$. The r_T^+ and r_T^- are no longer needed because the advective fluxes on the top surface have been approximated by Eq. (3.102) or Eq. (3.103).

The diffusive flux on the top surface of the finite volume $\Omega_{ijk(m)}$ in Eq. (3.28) becomes, for Dirichlet boundary conditions,

$$\hat{f}_Z^D \Big|_{ijk(m)}^T = -K_{ZZ} \Big|_{ijk(m)}^T \frac{\hat{C}_{TB} - \hat{C}_{i,j,k(m)}}{Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}}, \quad \text{with } k = N_Z \quad (3.107)$$

and for Neumann boundary conditions,

$$\hat{f}_Z^D \Big|_{ijk(m)}^T = -K_{ZZ} \Big|_{ijk(m)}^T N_{TB}, \quad \text{with } k = N_Z \quad (3.108)$$

The $K_{ZZ}|_{ijk(m)}^T$ in Eq. (3.29) becomes

$$K_{ZZ}|_{ijk(m)}^T = K_{ZZ,i,j,k(m)} + \frac{Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}}{Z_{i,j,k(m)} - Z_{i,j,k-1(m)}} (K_{ZZ,i,j,k(m)} - K_{ZZ,i,j,k-1(m)}), \quad (3.109)$$

with $k = N_Z$

Then the corresponding result for the finite volumes on top boundary is the same as Eq. (3.30) except that $a_{ijk(m)}^T = 0$ and a boundary condition source term $S_{ijk(m)}^{BC}$ appears on the right-hand side of the equation, for Dirichlet boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} \left[\left(W_{i,j,TB(m)} - \frac{K_{ZZ}|_{ijk(m)}^T}{Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}} \right) A_{ijk(m)}^T \hat{C}_{TB} \right], \quad (3.110)$$

with $k = N_Z$

and for Neumann boundary conditions,

$$S_{ijk(m)}^{BC} = -\frac{1}{\Omega_{ijk(m)}} [W_{i,j,TB(m)} (Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}) - K_{ZZ}|_{ijk(m)}^T] A_{ijk(m)}^T N_{TB}, \quad \text{with } k = N_Z \quad (3.111)$$

The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_T^+) = 0$ and $\xi(r_T^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^T = 0$ and the term multiplied by $A_{ijk(m)}^T$ is replaced, for Dirichlet boundary conditions, by

$$\frac{K_{ZZ}|_{ijk(m)}^T}{Z_{\text{surf},i,j,k+1(m)} - Z_{i,j,k(m)}}, \quad \text{with } k = N_Z \quad (3.112)$$

and for Neumann boundary conditions, by

$$W_{i,j,TB(m)} \quad (3.113)$$

For the finite volumes located at the corners of the computational domain, the results are the same as Eq. (3.30) except that the weights are zeros on the corresponding boundary surfaces. The additional boundary condition source term $S_{ijk(m)}^{BC}$ on the right-hand side of the equation is the combination of two or three of equations (3.48), (3.62), (3.74), (3.86), (3.98), and (3.110) for

Dirichlet boundary conditions, or equations (3.49), (3.63), (3.75), (3.87), (3.99), and (3.111) for Neumann boundary conditions, according to which boundaries the control volume is located on. The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with zero limiter functions on the domain boundary surfaces where the finite volume resides. The weight $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that the weights are zeros on the corresponding boundary surfaces and the terms multiplied by the areas of the finite volume surfaces that are on the domain boundaries are replaced by equations (3.50), (3.64), (3.76), (3.88), (3.100), or (3.112) for Dirichlet boundary conditions, or equations (3.51), (3.65), (3.77), (3.89), (3.101), or (3.113) for Neumann boundary conditions, accordingly.

For example, if a finite volume is located at the corner formed by south-west boundaries, the result is the same as Eq. (3.30) except that $a_{ijk(m)}^W = 0$ and $a_{ijk(m)}^S = 0$, and the additional boundary condition source term $S_{ijk(m)}^{BC}$ on the right-hand side of the equation is the combination of Eq. (3.48) and Eq. (3.74) for Dirichlet boundary conditions, or the combination of Eq. (3.49) and Eq. (3.75) for Neumann boundary conditions. The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_W^+) = 0$, $\xi(r_W^-) = 0$, $\xi(r_S^+) = 0$, and $\xi(r_S^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^W = 0$ and $a_{ijk(m)}^S = 0$ and the terms multiplied by $A_{ijk(m)}^W$ and $A_{ijk(m)}^S$ are replaced by Eq. (3.50) and Eq. (3.76) respectively for Dirichlet boundary conditions, or replaced by Eq. (3.51) and Eq. (3.77) respectively for Neumann boundary conditions.

Another example, if a finite volume is located at the corner formed by south-west-top boundaries, the result is the same as Eq. (3.30) except that $a_{ijk(m)}^W = 0$, $a_{ijk(m)}^S = 0$, and $a_{ijk(m)}^T = 0$ and the additional boundary condition source term $S_{ijk(m)}^{BC}$ on the right-hand side of the equation is the combination of equations (3.48), (3.74), and (3.110) for Dirichlet boundary conditions, or the combination of equations (3.49), (3.75), and (3.111) for Neumann boundary conditions. The $S_{ijk(m)}^{DC}$ is the same as Eq. (3.35) but with $\xi(r_W^+) = 0$, $\xi(r_W^-) = 0$, $\xi(r_S^+) = 0$, $\xi(r_S^-) = 0$, $(r_T^+) = 0$, and $\xi(r_T^-) = 0$. The $a_{ijk(m)}^C$ is the same as Eq. (3.34) except that $a_{ijk(m)}^W = 0$, $a_{ijk(m)}^S = 0$, and $a_{ijk(m)}^T = 0$ and the terms multiplied by $A_{ijk(m)}^W$, $A_{ijk(m)}^S$, and $a_{ijk(m)}^T$ are replaced by Eq. (3.50), Eq. (3.76), and Eq. (3.112) respectively for Dirichlet boundary conditions, or replaced by Eq. (3.51), Eq. (3.77), and Eq. (3.113) respectively for Neumann boundary conditions.

The transmission conditions equations (2.66) and (2.67) are satisfied explicitly when conducting the FVM-TVD spatial discretization. As an example, consider the east, north, and top finite volume surfaces as shown in Figure 3.3 are the interfaces between adjacent subdomains, assume positive wind speeds in each direction and diffusion out from the finite volume $\Omega_{ijk(m)}$. The transmission conditions are satisfied on the interfaces on volume $\Omega_{ijk(m)}$,

$$\begin{aligned}
\hat{f}_X^A|_{ijk(m)}^E + \hat{f}_X^D|_{ijk(m)}^E &= -\hat{f}_X^A|_{i+1,j,k,(m^E)}^W - \hat{f}_X^D|_{i+1,j,k,(m^E)}^W \\
\hat{f}_Y^A|_{ijk(m)}^N + \hat{f}_Y^D|_{ijk(m)}^N &= -\hat{f}_Y^A|_{i,j+1,k,(m^N)}^S - \hat{f}_Y^D|_{i,j+1,k,(m^N)}^S \\
\hat{f}_Z^A|_{ijk(m)}^T + \hat{f}_Z^D|_{ijk(m)}^T &= -\hat{f}_Z^A|_{i,j,k+1,(m^T)}^B - \hat{f}_Z^D|_{i,j,k+1,(m^T)}^B
\end{aligned} \tag{3.114}$$

where m^E , m^N , and m^T are the subdomain numbers which are adjacent to the east, north, and top of the interfaces on finite volume $\Omega_{ijk(m)}$.

The entire set of $N_{X(m)} \times N_{Y(m)} \times N_{Z(m)}$ finite volumes in each subdomain $i, j, k \in \Omega_{(m)}$ are mapped to a 1D vector as follows,

$$\begin{aligned}
np_{(m)} &= np_{(m)}(i, j, k) = i + (j - 1)N_{X(m)} + (k - 1)N_{X(m)}N_{Y(m)} \\
i &= 1, 2, \dots, N_{X(m)}; \quad j = 1, 2, \dots, N_{Y(m)}; \quad k = 1, 2, \dots, N_{Z(m)}
\end{aligned} \tag{3.115}$$

The estimated concentration in a finite volume is $\hat{C}_{ijk(m)} \equiv \hat{C}_{np(m)}$ and the semi-discrete Eq. (3.9) can be written in state space form as

$$\begin{aligned}
&\hat{\mathbf{x}}_{(m)}(t) \\
&= P_{FVM(m)}\hat{\mathbf{x}}_{(m)}(t) + \mathbf{S}^{BC} + P_{TVD(m)}\hat{\mathbf{x}}_{(m)}(t) \\
&+ \begin{cases} \mathbf{Q}_{(m_L)}(t)\Lambda[C(\Theta_s(t), t) - \hat{C}(\Theta_s(t), t)], & \text{for Luenberger estimator} \\ 0, & \text{for naive estimator} \end{cases} \\
&= \mathbf{P}_{RHS(m)}(t)
\end{aligned} \tag{3.116}$$

where $\hat{\mathbf{x}}_{(m)}(t) = \hat{\mathbf{C}}_{(m)}(X, Y, Z, t)$ is the estimated concentration state,

$$\hat{\mathbf{C}}_{(m)} = \left[\hat{C}_{1(m)}, \hat{C}_{2(m)}, \dots, \hat{C}_{np(m)}, \dots, \hat{C}_{N(m)} \right]^T, \quad N_{(m)} = N_{X(m)} \times N_{Y(m)} \times N_{Z(m)} \tag{3.117}$$

$P_{FVM(m)}$ is a 7-diagonal sparse matrix resulting from the FVM-TVD spatial discretization of the advection-diffusion operator; \mathbf{S}^{BC} is an $N_{(m)} \times 1$ vector arising from the boundary conditions; $P_{TVD(m)}$ is a 7-diagonal sparse matrix resulting from the multiplication by the limiter functions of TVD scheme. The output injection term of the Luenberger estimator model is expressed as $\mathbf{Q}_{(m_L)}(t)\Lambda[C(\Theta_s(t), t) - \hat{C}(\Theta_s(t), t)]$ which is the product of sensor location vector $\mathbf{Q}_{(m_L)}(t)$, the user-defined estimation gain Λ , and the state-estimation error at the sensor location, $C(\Theta_s(t), t) - \hat{C}(\Theta_s(t), t)$. The elements $q_{np(m_L)}(t)$ of sensor location vector $\mathbf{Q}_{(m_L)}(t)$ is, $q_{np(m_L)}(t) = \begin{cases} \Omega_{np(m_L)}, & \Theta_s(t) \in np(m_L) \\ 0, & \Theta_s(t) \notin np(m_L) \end{cases}$ where $\Omega_{np(m_L)}$ is the volume of the finite volume (or cell) where the sensor resides.

3.2.2 Runge-Kutta Temporal Integration

The state-space form Eq. (3.116) is integrated with respect to time by using 4th order Runge-Kutta method [Hirsh, 2007] as follows,

$$\hat{\mathbf{x}}_{(m)}^{n+1} = \hat{\mathbf{x}}_{(m)}^n + \alpha_4 \Delta t \sum_{l=0}^3 \alpha_l \mathbf{P}_{RHS(m)}^l \quad (3.118)$$

where n is the time level, $\mathbf{P}_{RHS(m)}$ is the right hand side of Eq. (3.116), and the coefficients are $\alpha_0 = 1$, $\alpha_1 = 2$, $\alpha_2 = 2$, $\alpha_3 = 1$, $\alpha_4 = \frac{1}{6}$.

For the modified Lyapunov guidance law, Eq. (2.81) is also implemented by using the 4th order Runge-Kutta method,

$$\mathbf{r}_s^{n+1} = \mathbf{r}_s^n + \alpha_4 \Delta t_s \sum_{l=0}^3 \alpha_l \mathbf{N}_{RHS}^l \quad (3.119)$$

where $\mathbf{r}_s(t) = [X_s^d \quad Y_s^d \quad Z_s^d]^T$, Δt_s is the integration time step of the SAV and in most cases $\Delta t_s \geq \Delta t$, $\mathbf{N}_{RHS} = [U_s^d \quad V_s^d \quad W_s^d]^T$ is the right hand side of Eq. (2.81), and the coefficients α_i are the same as appearing in Eq. (3.118).

For the Lyapunov guidance law with SAV dynamical model, Eq. (2.82) is also integrated by using the 4th order Runge-Kutta method,

$$\mathbf{N}^{n+1} = \mathbf{N}^n + \alpha_4 \Delta t_s \sum_{l=0}^3 \alpha_l \mathbf{R}_{RHS}^l \quad (3.120)$$

where $\mathbf{N} = [X_s \ Y_s \ Z_s \ V_g \ \gamma \ \chi]^T$, $\mathbf{R}_{RHS} = [\dot{X}_s \ \dot{Y}_s \ \dot{Z}_s \ \dot{V}_g \ \dot{\gamma} \ \dot{\chi}]^T$ is the right hand side of Eq. (2.82), Δt_s and α_l keep the same values as in Eq. (3.119).

3.2.3 Numerical Stability

The time step Δt in Eq. (3.118) must be smaller than the maximum allowable time step Δt_{max} that satisfies the numerical stability condition. The Δt_{max} can be determined by Von Neumann method for stability analysis [Hirsh, 2007].

Substitute equations (3.31) – (3.36) into (3.30), assume constant wind speeds U , V , and W and constant eddy diffusivities K_{XX} , K_{YY} , and K_{ZZ} , assume uniform grid such that $X_{i+1,j,k(m)} - X_{i,j,k(m)} = X_{i,j,k(m)} - X_{i-1,j,k(m)} = \Delta X(m)$, $Y_{i,j+1,k(m)} - Y_{i,j,k(m)} = Y_{i,j,k(m)} - Y_{i,j-1,k(m)} = \Delta Y(m)$, $Z_{i,j,k+1(m)} - Z_{i,j,k(m)} = Z_{i,j,k(m)} - Z_{i,j,k-1(m)} = \Delta Z(m)$, and $\Omega_{ijk(m)} = \Delta X(m)\Delta Y(m)\Delta Z(m)$, $A_{ijk(m)}^E = A_{ijk(m)}^W = \Delta Y(m)\Delta Z(m)$, $A_{ijk(m)}^N = A_{ijk(m)}^S = \Delta X(m)\Delta Z(m)$, $A_{ijk(m)}^T = A_{ijk(m)}^B = \Delta X(m)\Delta Y(m)$, ignore the output injection term for the purpose of numerical stability analysis, and use the first order differencing for the temporal derivative $\frac{\partial \hat{c}_{ijk(m)}}{\partial t} = \frac{\hat{c}_{ijk(m)}^{n+1} - \hat{c}_{ijk(m)}^n}{\Delta t}$. For limiter function $\xi(r) = 0$, which is the first order upwind scheme for advective flux, if $U > 0$, $V > 0$, and $W > 0$, then yields

$$\begin{aligned}
\hat{C}_{ijk(m)}^{n+1} = & \hat{C}_{ijk(m)}^n - \sigma_{X(m)}(\hat{C}_{i,j,k(m)}^n - \hat{C}_{i-1,j,k(m)}^n) - \sigma_{Y(m)}(\hat{C}_{i,j,k(m)}^n - \hat{C}_{i,j-1,k(m)}^n) \\
& - \sigma_{Z(m)}(\hat{C}_{i,j,k(m)}^n - \hat{C}_{i,j,k-1(m)}^n) \\
& + \beta_{X(m)}(\hat{C}_{i+1,j,k(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i-1,j,k(m)}^n) \\
& + \beta_{Y(m)}(\hat{C}_{i,j+1,k(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i,j-1,k(m)}^n) \\
& + \beta_{Z(m)}(\hat{C}_{i,j,k+1(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i,j,k-1(m)}^n)
\end{aligned} \tag{3.121}$$

where $\sigma_{X(m)} = \frac{U\Delta t}{\Delta X(m)}$, $\sigma_{Y(m)} = \frac{V\Delta t}{\Delta Y(m)}$, $\sigma_{Z(m)} = \frac{W\Delta t}{\Delta Z(m)}$, $\beta_{X(m)} = \frac{K_{XX}\Delta t}{(\Delta X(m))^2}$, $\beta_{Y(m)} = \frac{K_{YY}\Delta t}{(\Delta Y(m))^2}$, and $\beta_{Z(m)} = \frac{K_{ZZ}\Delta t}{(\Delta Z(m))^2}$.

Substitute into Eq. (3.121) a Fourier mode in the form of $\hat{C}_{ijk(m)}^n = \mathbb{V}^n e^{i\varphi} e^{i\psi} e^{ik\theta}$ where φ , ψ , and θ are phase angles and use $e^{i\alpha} = \cos \alpha + I \sin \alpha$ and $e^{-i\alpha} = \cos \alpha - I \sin \alpha$, then the amplification factor is

$$\begin{aligned}
G = \frac{\mathbb{V}^{n+1}}{\mathbb{V}^n} = & 1 - \sigma_{X(m)}(1 - \cos \varphi + I \sin \varphi) - \sigma_{Y(m)}(1 - \cos \psi + I \sin \psi) \\
& - \sigma_{Z(m)}(1 - \cos \theta + I \sin \theta) + 2\beta_{X(m)} \cos \varphi + 2\beta_{Y(m)} \cos \psi \\
& + 2\beta_{Z(m)} \cos \theta - 2(\beta_{X(m)} + \beta_{Y(m)} + \beta_{Z(m)})
\end{aligned} \tag{3.122}$$

The Von Neumann stability condition is

$$|G| \leq 1 \tag{3.123}$$

For $\varphi = \psi = \theta = \pi$,

$$G = 1 - 2(\sigma_{X(m)} + \sigma_{Y(m)} + \sigma_{Z(m)}) - 4(\beta_{X(m)} + \beta_{Y(m)} + \beta_{Z(m)}) \tag{3.124}$$

Substitute Eq. (3.124) into Eq. (3.123), yields

$$\Delta t_1 \leq \frac{1}{\frac{U}{\Delta X(m)} + \frac{V}{\Delta Y(m)} + \frac{W}{\Delta Z(m)} + 2 \left[\frac{K_{XX}}{(\Delta X(m))^2} + \frac{K_{YY}}{(\Delta Y(m))^2} + \frac{K_{ZZ}}{(\Delta Z(m))^2} \right]} \tag{3.125}$$

For $\varphi = \psi = \theta = 0$, the Von Neumann stability condition Eq. (3.123) is satisfied automatically.

If $U < 0$, $V < 0$, and $W < 0$, then we have

$$\begin{aligned}
\hat{C}_{ijk(m)}^{n+1} = & \hat{C}_{ijk(m)}^n - \sigma_{X(m)}(\hat{C}_{i+1,j,k(m)}^n - \hat{C}_{i,j,k(m)}^n) - \sigma_{Y(m)}(\hat{C}_{i,j+1,k(m)}^n - \hat{C}_{i,j,k(m)}^n) \\
& - \sigma_{Z(m)}(\hat{C}_{i,j,k+1(m)}^n - \hat{C}_{i,j,k(m)}^n) \\
& + \beta_{X(m)}(\hat{C}_{i+1,j,k(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i-1,j,k(m)}^n) \\
& + \beta_{Y(m)}(\hat{C}_{i,j+1,k(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i,j-1,k(m)}^n) \\
& + \beta_{Z(m)}(\hat{C}_{i,j,k+1(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i,j,k-1(m)}^n)
\end{aligned} \tag{3.126}$$

then the amplification factor is

$$\begin{aligned}
G = \frac{\mathbb{V}^{n+1}}{\mathbb{V}^n} = & 1 - \sigma_{X(m)}(\cos \varphi + I \sin \varphi - 1) - \sigma_{Y(m)}(\cos \psi + I \sin \psi - 1) \\
& - \sigma_{Z(m)}(\cos \theta + I \sin \theta - 1) + 2\beta_{X(m)} \cos \varphi + 2\beta_{Y(m)} \cos \psi \\
& + 2\beta_{Z(m)} \cos \theta - 2(\beta_{X(m)} + \beta_{Y(m)} + \beta_{Z(m)})
\end{aligned} \tag{3.127}$$

For $\varphi = \psi = \theta = \pi$, substitute Eq. (3.127) into Eq. (3.123), we have the same time step condition as Eq. (3.125) where the U , V , and W are all absolute values for the case of $U < 0$, $V < 0$, and $W < 0$. For $\varphi = \psi = \theta = \pi$, we have $G = 1$ which satisfies the Von Neumann stability condition Eq. (3.123) automatically.

For limiter function $\xi(r) = 1$, which is the second order central difference scheme for advective flux, if $U > 0$, $V > 0$, and $W > 0$, then we have

$$\begin{aligned}
\hat{C}_{ijk(m)}^{n+1} = & \hat{C}_{ijk(m)}^n - \frac{\sigma_{X(m)}}{2}(\hat{C}_{i+1,j,k(m)}^n - \hat{C}_{i-1,j,k(m)}^n) \\
& - \frac{\sigma_{Y(m)}}{2}(\hat{C}_{i,j+1,k(m)}^n - \hat{C}_{i,j-1,k(m)}^n) - \frac{\sigma_{Z(m)}}{2}(\hat{C}_{i,j,k+1(m)}^n - \hat{C}_{i,j,k-1(m)}^n) \\
& + \beta_{X(m)}(\hat{C}_{i+1,j,k(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i-1,j,k(m)}^n) \\
& + \beta_{Y(m)}(\hat{C}_{i,j+1,k(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i,j-1,k(m)}^n) \\
& + \beta_{Z(m)}(\hat{C}_{i,j,k+1(m)}^n - 2\hat{C}_{i,j,k(m)}^n + \hat{C}_{i,j,k-1(m)}^n)
\end{aligned} \tag{3.128}$$

The amplification factor is

$$G = \frac{\mathbb{V}^{n+1}}{\mathbb{V}^n} = 1 - \sigma_{X(m)} I \sin \varphi - \sigma_{Y(m)} I \sin \psi - \sigma_{Z(m)} I \sin \theta + 2\beta_{X(m)} \cos \varphi + 2\beta_{Y(m)} \cos \psi + 2\beta_{Z(m)} \cos \theta - 2(\beta_{X(m)} + \beta_{Y(m)} + \beta_{Z(m)}) \quad (3.129)$$

For $\varphi = \psi = \theta = \pi$,

$$G = 1 - 4(\beta_{X(m)} + \beta_{Y(m)} + \beta_{Z(m)}) \quad (3.130)$$

Substitute Eq. (3.130) into Eq. (3.123) and use $2(\beta_{X(m)} + \beta_{Y(m)} + \beta_{Z(m)}) - \sigma_{X(m)}^2 - \sigma_{Y(m)}^2 - \sigma_{Z(m)}^2 \geq 0$ [Hirsh, 2007], yields

$$\Delta t_2 \leq \frac{1}{2 \left[\frac{K_{XX}}{(\Delta X(m))^2} + \frac{K_{YY}}{(\Delta Y(m))^2} + \frac{K_{ZZ}}{(\Delta Z(m))^2} \right]} \quad (3.131)$$

and

$$\Delta t_3 \leq \frac{2 \left[\frac{K_{XX}}{(\Delta X(m))^2} + \frac{K_{YY}}{(\Delta Y(m))^2} + \frac{K_{ZZ}}{(\Delta Z(m))^2} \right]}{\frac{U^2}{(\Delta X(m))^2} + \frac{V^2}{(\Delta Y(m))^2} + \frac{W^2}{(\Delta Z(m))^2}} \quad (3.132)$$

For $\varphi = \psi = \theta = 0$, we have $G = 1$ which satisfies the Von Neumann stability condition Eq. (3.123) automatically.

If $U < 0$, $V < 0$, and $W < 0$, we have exactly the same results as those for the case of $U > 0$, $V > 0$, and $W > 0$.

In summary, the time step Δt in Eq. (3.118) must satisfy

$$\Delta t \leq \Delta t_{max} = \min(\max \Delta t_1, \max \Delta t_2, \max \Delta t_3) \quad (3.133)$$

3.3 Verification and Error Analysis for the HT-NODDE-FVM

3.3.1 Non-Dimensional 3D Advection Equation

The first test provides the verification of the NODDE-FVM with TVD-RK applied to the 3D pure advection equation. Equation (2.58) with zero diffusivities and zero source term is non-dimensionalized using $C = C^*C_0$, $t = t^*L_0/V_0$, $U = U^*V_0$, $V = V^*V_0$, $W = W^*V_0$, $X = X^*L_0$, $Y = Y^*L_0$, and $Z = Z^*L_0$, where C_0 , L_0 , and V_0 are reference concentration, length, and wind speed, respectively, and becomes

$$\frac{\partial C^*}{\partial t^*} + U^* \frac{\partial C^*}{\partial X^*} + V^* \frac{\partial C^*}{\partial Y^*} + W^* \frac{\partial C^*}{\partial Z^*} = 0 \quad (3.134)$$

Four initial density configurations are used for the simulations. Case 1a corresponds to a continuous Gaussian density distribution with $C^* = \exp\left(-\frac{r^{*2}}{R^{*2}}\right)$, $r^{*2} = (X^* - X_r^*)^2 + (Y^* - Y_r^*)^2 + (Z^* - Z_r^*)^2$ where X_r^* , Y_r^* , and Z_r^* are the coordinates of the center, and $R = R^*L_0$. Cases 1b and 1c correspond to a piecewise Gaussian distribution $C^* = \exp\left(-\frac{r^{*2}}{R^{*2}}\right)$ for $r^{*2} \leq R^{*2}$ and $C^* = \frac{1}{e}$ for $r^{*2} > R^{*2}$. Case 1d corresponds to a cubical density distribution $C^* = 1$ for $r_X^{*2} \leq R^{*2}$ and $r_Y^{*2} \leq R^{*2}$ and $r_Z^{*2} \leq R^{*2}$, and $C^* = 0$ for $r_X^{*2} > R^{*2}$ and $r_Y^{*2} > R^{*2}$ and $r_Z^{*2} > R^{*2}$, where $r_X^{*2} = (X^* - X_r^*)^2$, $r_Y^{*2} = (Y^* - Y_r^*)^2$, and $r_Z^{*2} = (Z^* - Z_r^*)^2$. The widths of the distributions are shown in Table 3.1.

Table 3.1 3D pure advection cases and results

Case	Initial Condition	Order p L^1 from Eq. (3.135)	Order p L^1 form Eq. (3.136)
1a	Gaussian, $R^{*2} = 0.005$	1.5340	1.5284
1b	Piecewise Gaussian, $R^{*2} = 0.005$	1.1515	1.1374
1c	Piecewise Gaussian, $R^{*2} = 0.04$	1.2930	1.2914
1d	Cubical, $R^{*2} = 0.005$	0.7886	0.7886

For the simulations, we use a domain with $L_X^* = L_Y^* = L_Z^* = 1$, $U^* = 1$, $V^* = W^* = 0$, $X_r^* = 0.25$, $Y_r^* = Z_r^* = 0.5$. Dirichlet boundary conditions are enforced with $C^* = 0$ for Case 1a and 1d, and $C^* = 1/e$ for Case 1b and 1c. The computational domain is divided into 40 subdomains with $N_{DDX} = N_{DDY} = 2$ and $N_{DDZ} = 10$. The simulations were run with grid resolutions between $50 \times 50 \times 50 \leq \frac{L_X^*}{\Delta X^*} \times \frac{L_Y^*}{\Delta Y^*} \times \frac{L_Z^*}{\Delta Z^*} \leq 400 \times 400 \times 400$ and a constant $\Delta t^* = 0.002$ for up to $t^* = 0.5$. Results are used to evaluate the L^1 norm of the error following *LeVeque* [2007]. For uniform finite volumes and using the cell-centered values for both the numerical and the analytical solutions the q -norm is

$$L^q(C^*) = \left(\Delta X^* \Delta Y^* \Delta Z^* \sum_{np=1}^N |C_{np}^*{}_{\text{numerical}} - C_{np}^*{}_{\text{analytical}}|^q \right)^{\frac{1}{q}} \quad (3.135)$$

Using the continuous function for the analytical solution within each finite volume np , the q -norm becomes

$$L^q(C^*) = \left[\sum_{np=1}^N \left| (\Delta X^* \Delta Y^* \Delta Z^*)^{\frac{1}{q}} C_{np}^*{}_{\text{numerical}} - (\Delta X^* \Delta Y^* \Delta Z^*)^{\frac{1-q}{q}} \iiint_{\Omega_{np}} C_{np}^*{}_{\text{analytical}} d\Omega \right|^q \right]^{\frac{1}{q}} \quad (3.136)$$

The order of accuracy is evaluated by

$$p = \frac{\log[L^q(C_{\Delta_1}^*)] - \log[L^q(C_{\Delta_2}^*)]}{\log(\Delta_1^*) - \log(\Delta_2^*)} \quad (3.137)$$

The L^1 norm of the error is shown in Figure 3.5 and the order of accuracy is shown in Table 3.1 for all the cases considered. The results show that for smoothly varying functions the NODDE-FVM becomes between order 1 and 2, which is consistent with the results of single-domain FVM-TVD method [*Prabhakaran and Doss, 2015*].

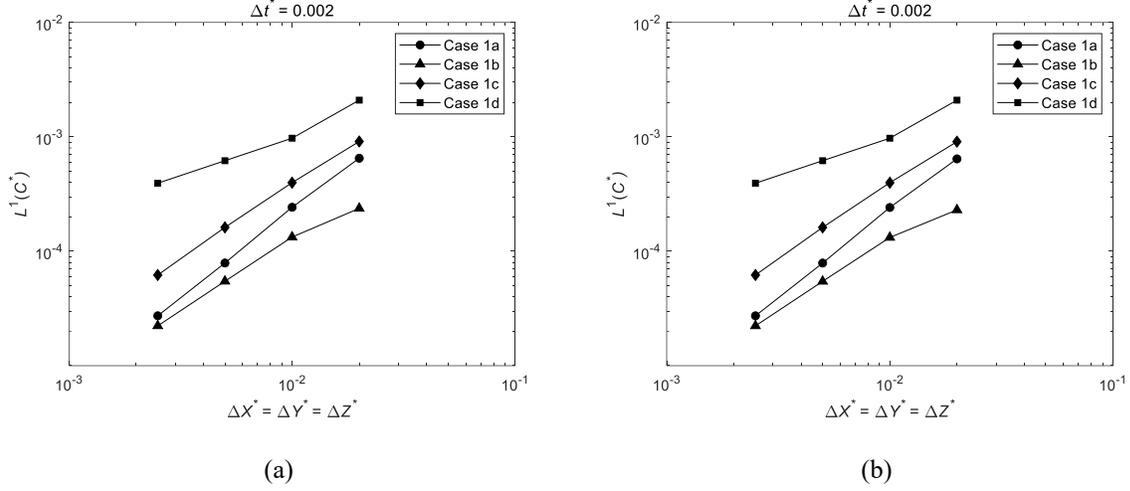


Figure 3.5 L^1 norm of the error between NODDE-FVM and analytical solutions of the advection PDE as a function of grid resolution for conditions shown in Table 3.1. (a) L^1 norm calculated by Eq. (3.135); (b) L^1 norm calculated by Eq. (3.136).

3.3.2 Non-Dimensional 3D Advection-Diffusion Equation

The second test provides the verification of the NODDE-FVM with TVD-RK applied to the 3D advection-diffusion equation. The process model Eq. (2.58) assuming constant winds and uniform diffusivities, is non-dimensionalized using $C = C^*C_0$, $t = t^*L_0/V_0$, $U = U^*V_0$, $V = V^*V_0$, $W = W^*V_0$, $X = X^*L_0$, $Y = Y^*L_0$, $Z = Z^*L_0$, $K_{XX} = K_{XX}^*K_0$, $K_{YY} = K_{YY}^*K_0$, $K_{ZZ} = K_{ZZ}^*K_0$, and $Pe = \frac{V_0L_0}{K_0}$. The non-dimensional advection-diffusion equation becomes

$$\frac{\partial C^*}{\partial t^*} + U^* \frac{\partial C^*}{\partial X^*} + V^* \frac{\partial C^*}{\partial Y^*} + W^* \frac{\partial C^*}{\partial Z^*} - \frac{1}{Pe} K_{XX}^* \frac{\partial^2 C^*}{\partial X^{*2}} - \frac{1}{Pe} K_{YY}^* \frac{\partial^2 C^*}{\partial Y^{*2}} - \frac{1}{Pe} K_{ZZ}^* \frac{\partial^2 C^*}{\partial Z^{*2}} = 0 \quad (3.138)$$

An analytical solution of Eq. (2.58) for an instantaneous point gas release at location X_r , Y_r , Z_r and time t_r with wind speeds $U \neq 0$ and $V = W = 0$, and boundary conditions $C(X, Y, Z, t) = 0$ for $X, Y, Z \rightarrow \pm\infty$ is given as [Arya, 1999]

$$C(X, Y, Z, t) = \frac{Q}{(2\pi)^{\frac{3}{2}}\sigma_X\sigma_Y\sigma_Z} \cdot \exp\left\{-\frac{\{X - [X_r + U(t - t_r)]\}^2}{2\sigma_X^2} - \frac{(Y - Y_r)^2}{2\sigma_Y^2} - \frac{(Z - Z_r)^2}{2\sigma_Z^2}\right\} \quad (3.139)$$

In the above $\sigma_X = \sqrt{2K_{XX}(t - t_r)}$, $\sigma_Y = \sqrt{2K_{YY}(t - t_r)}$, and $\sigma_Z = \sqrt{2K_{ZZ}(t - t_r)}$; Q is the released mass in kg. The non-dimensional analytical solution Eq. (3.139) for $K_{XX} = K_{YY} = K_{ZZ} = K$ can be written as

$$C^*(r^*, t^*) = \frac{A^*}{(\pi R^{*2})^{\frac{3}{2}}} \exp\left(-\frac{r^{*2}}{R^{*2}}\right) \quad (3.140)$$

where $A^* = \frac{Q^*Q_0}{c_0\left(\frac{K_0L_0}{V_0}\right)^{\frac{3}{2}}} \exp\left(\frac{V_0L_0}{K_0}\right)$, $r^{*2} = \{X^* - [X_r^* + U^*(t^* - t_r^*)]\}^2 + (Y^* - Y_r^*)^2 + (Z^* - Z_r^*)^2$,

$R^{*2} = 4K^*(t^* - t_r^*)$, and $t_r^* = 0$. For the simulations, we use $A^* = 10^{-9}$, $L_X^* = L_Y^* = L_Z^* = 1$, $U^* = 1$, $V^* = W^* = 0$, $X_r^* = 0.1$, $Y_r^* = Z_r^* = 0.5$, and $K^* = 0.0025$. The initial density used in the simulations follows Eq. (3.140) with $t^* = 0.1$. Three different Peclet numbers are used and cover diffusion dominated (Case 2a) to advection dominated flow (Case 2c) as shown in Table 3.2. Dirichlet boundary conditions with $C^* = 0$ are applied to all boundaries. The simulations were run with grid resolutions between $50 \times 50 \times 50 \leq \frac{L_X^*}{\Delta X^*} \times \frac{L_Y^*}{\Delta Y^*} \times \frac{L_Z^*}{\Delta Z^*} \leq 400 \times 400 \times 400$ and time steps are shown in Table 3.2. The simulations were run for up to $t^* = 0.6$. Results are used to evaluate the L^1 norm of the error defined by Eq. (3.136) and the order of accuracy defined by Eq. (3.137). Figure 3.6 shows the L^1 norm of the error as a function of grid resolution. Table 3.2 shows that the NODDE-FVM with TVD achieves near second order of accuracy for all the cases considered.

Table 3.2 3D advection and diffusion cases and results

Case	Peclet Number	Time Step	Order p L^1 form Eq. (3.136)
2a	$Pe = 0.5$	$\Delta t^* = 0.00016$	1.9850
2b	$Pe = 1$	$\Delta t^* = 0.00025$	1.8475
2c	$Pe = 2$	$\Delta t^* = 0.000625$	1.5993

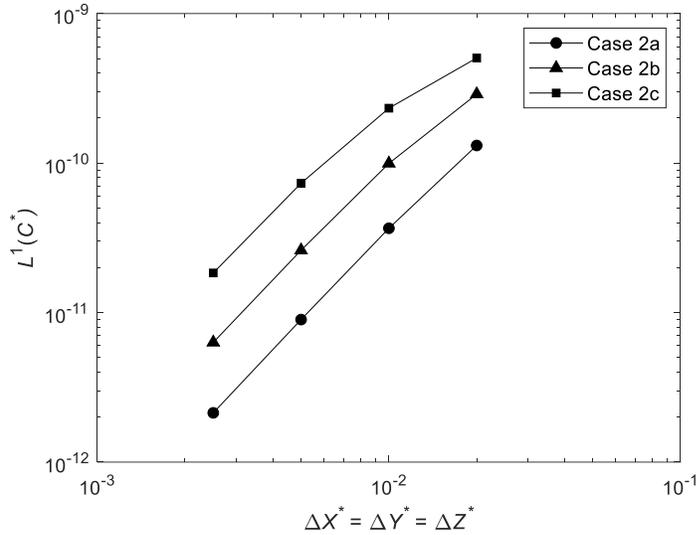


Figure 3.6 L^1 norm of the error between NODDE-FVM and analytical solutions of the advection-diffusion PDE as a function of grid resolution and Peclet number shown in Table 3.2.

3.3.3 Verification of the Hybrid Estimator with HT-NODDE-FVM

The third benchmark test provides verification of the HT-NODDE-FVM estimator equations (2.64) – (2.67) and examines the impact of grid resolution, sensor model, and estimation gain Λ in Eq. (2.71) on the L^2 norm of the estimation error evaluated from Eq. (3.136). The case involves an instantaneous release by a stationary source in a large domain with constant atmospheric properties, which is depicted in Figure 3.7 and has an analytical solution given by Eq. (3.139).

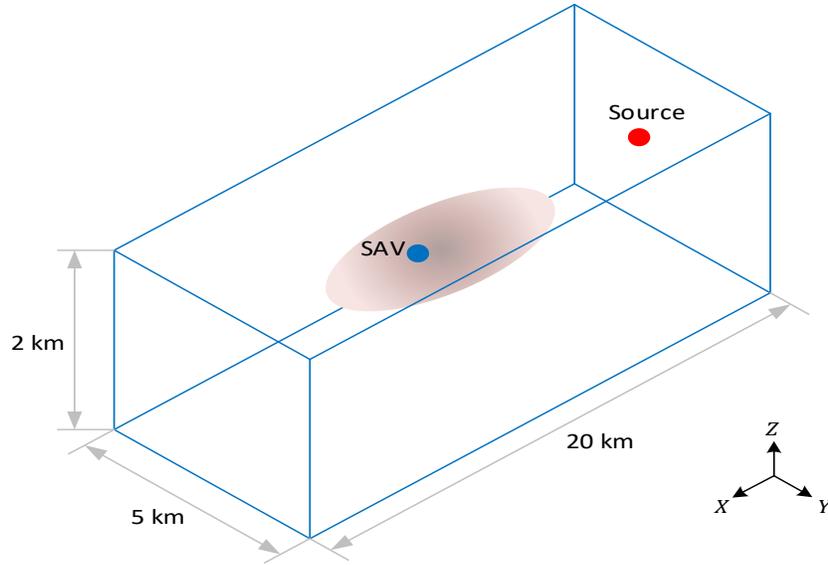


Figure 3.7 Benchmark case used for verification and error analysis of the HT-NODDE-FVM estimator. A stationary source releases instantaneously a gas in a domain with uniform atmospheric parameters and constant wind in the X -direction. The SAV performs real-time estimation of the plume concentration.

The computational inputs are listed in Table 3.3. The SAV parameters and estimator inputs are listed in Table 3.4. The SAV starts patrolling from its initial location on a helical trajectory. When the SAV sensor detects a concentration above the threshold, the HT-NODDE-FVM estimator equations (2.64) – (2.67) provides the estimated concentration and control inputs to the guidance law equations (2.78) – (2.81) which reposition the SAV. Two different approaches are used to generate the numerical sensor data at the location of the SAV. In the first approach, the cell value of the concentration is calculated by the analytical solution Eq. (3.139) at the cell center and assigned to the SAV location. In the second approach, the concentration at the SAV location is calculated directly from the analytical solution Eq. (3.139). A coarse and fine grid resolution is used with $N_X = 300$, $N_Y = 75$, $N_Z = 30$ cells and $N_X = 600$, $N_Y = 150$, $N_Z = 60$ cells respectively. The total number of subdomains is 27 with $N_{DDX} = N_{DDY} = N_{DDZ} = 3$ for both grid resolutions. A series of simulations were carried out by varying the estimation gain Λ . The time step used for both grid resolutions is 1 s which satisfies the numerical stability condition Eq. (3.133). The simulation time is 1300 s.

Table 3.3 Computational inputs for benchmark tests of the estimator

Parameter	Value
Domain length (L_X, L_Y, L_Z), km	20, 5, 2
Wind speed (U, V, W), m/s	10, 0, 0
Eddy diffusivity (K_{XX}, K_{YY}, K_{ZZ}), m^2/s	100, 100, 40
Mass of release at initial time (Q), kg	100
Location of the initial release (X_0, Y_0, Z_0), km	4.1, 2.5, 1

Table 3.4 SAV parameters and estimator inputs

Parameter	Value
SAV initial location (X_{s0}, Y_{s0}, Z_{s0}), km	10.4, 2.5, 0.55
SAV patrolling linear velocity (v_s), m/s	70 [<i>Writer</i> , 2019]
Helix center of SAV patrolling ($X_{scc}, Y_{scc}, Z_{scc}$), km	8, 2.5, 0.55
Helix radius of SAV patrolling (r), km	2.4
Pitch of the helix, rad	120π
Sensor threshold (C_{\min}), kg/m^3 (or ppb)	1×10^{-9} (or 1 ppm)
Guidance gains (k_X, k_Y, k_Z)	50, 50, 4

Figure 3.8 plots the L^2 norm of the estimation error evaluated from Eq. (3.136) at $t = 1300$ s as a function of the estimation gain Λ . It is shown that for both the coarse and fine grid resolutions there is an optimum gain that minimizes the estimation error. Figure 3.8 shows that the numerical sensor data approach does not impact the estimation error. Figure 3.8 also shows that the coarse grid has better estimation error than the fine grid due to the larger sensor volume of the former.

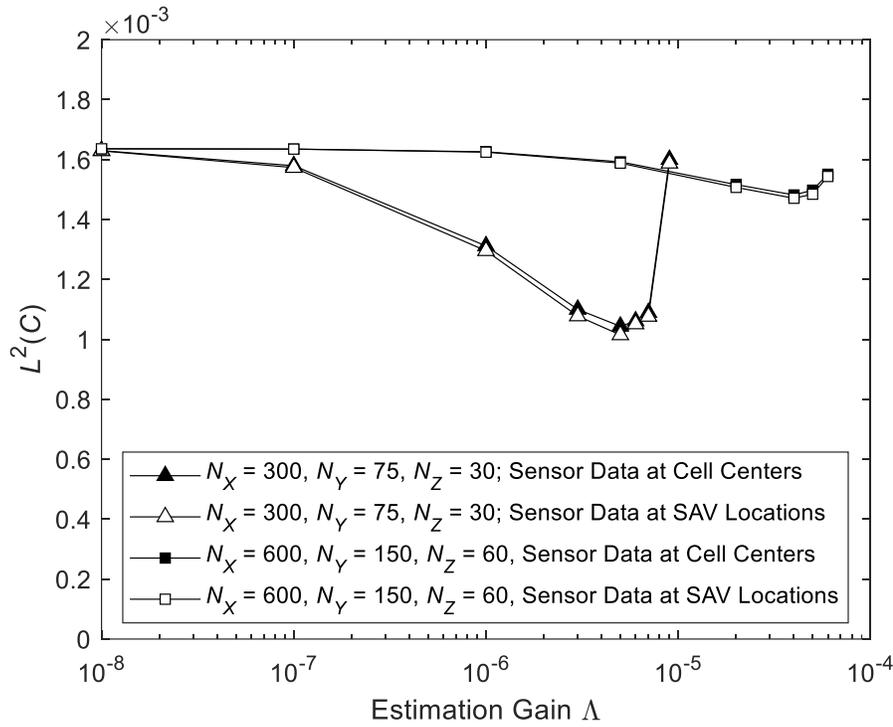


Figure 3.8 Verification of the HT-NODDE-FVM estimator. L^2 norm of the estimation error from Eq. (3.136) as a function of estimation gain for a coarse and fine grid. Sensor data are taken from the analytical solution Eq. (3.139).

3.3.4 Error Analysis of the Hybrid Estimator with HT-NODDE-FVM

The fourth benchmark test provides further error analysis of the HT-NODDE-FVM estimator equations (2.64) – (2.67) and examines the impact of grid resolution, estimation gain and numerical data. All the input simulation conditions for the gas release and ambient conditions are the same as the third benchmark case as shown in Table 3.3 and Table 3.4. The grid resolutions and optimal estimation gains used in the simulations are listed in Table 3.5. The estimator error norms L^1 , L^2 , and L^∞ are evaluated using Eq. (3.136) and plotted in Figure 3.9 and Figure 3.10. In Figure 3.9 the errors are evaluated between estimated concentration and numerical sensor data obtained from the NODDE-FVM of the process model Eq. (2.58). In Figure 3.10 the errors are evaluated between estimated concentration and the analytical solution Eq. (3.139).

Table 3.5 Grid resolutions and corresponding optimal gains used for error analysis for the estimator model in benchmark cases

Number of cells in each direction (N_X, N_Y, N_Z)	Total number of cells	Cell size ($\Delta X = \Delta Y = \Delta Z, \text{m}$)	Optimal estimation gain (Λ)
600, 150, 60	5.4×10^6	33.33	4×10^{-5}
300, 75, 30	6.75×10^5	66.67	5×10^{-6}
240, 60, 24	3.456×10^5	83.33	3×10^{-6}
180, 45, 18	1.458×10^5	111.11	1×10^{-6}
120, 30, 12	4.32×10^4	166.67	4×10^{-7}
60, 15, 6	5.4×10^3	333.33	4×10^{-9}

Figure 3.9 shows that as grid resolution coarsens the process state knowledge increases due to the increase in sensor volume and as a result the estimator error decreases. In the limit of a single-cell domain the error would be identically zero because the sensor would have knowledge of the entire state. Figure 3.10 shows that the errors decrease with increasing grid size, reach a minimum and then increase in contrast with the monotonic behavior of Figure 3.9. This behavior is due to the compound effects of increased state knowledge and reduced numerical fidelity as cell size increases. This benchmark test, therefore, decouples the impact of these two competing effects.

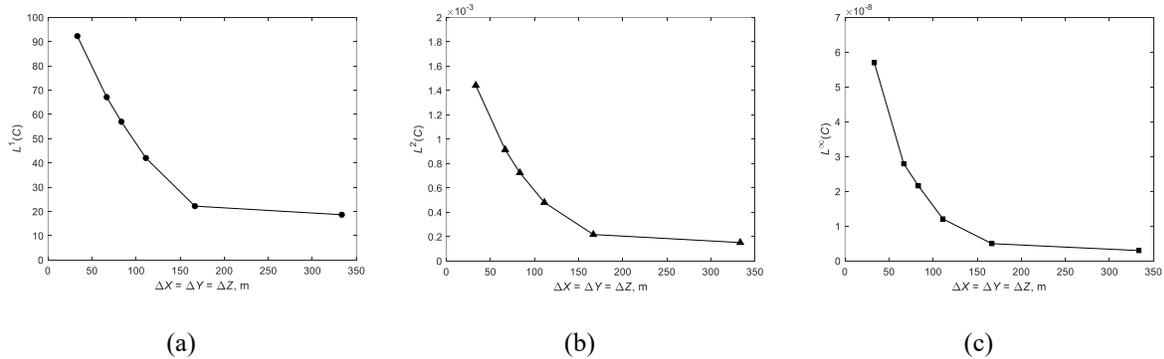


Figure 3.9 Error analysis of the HT-NODDE-FVM estimator model. (a) L^1 , (b) L^2 , (c) L^∞ norm of error evaluated between estimated concentration and numerical sensor data (process-model concentration obtained with HT-NODDE-FVM).

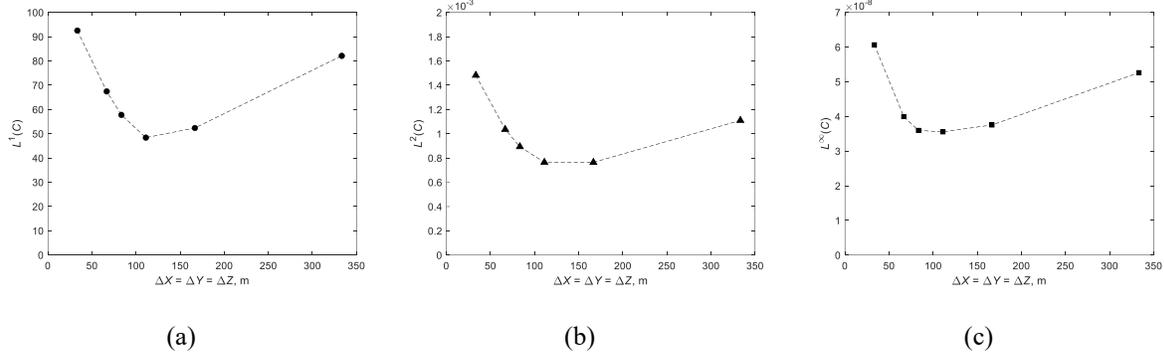


Figure 3.10 Error analysis of the HT-NODDE-FVM estimator model. (a) L^1 , (b) L^2 , (c) L^∞ norm of the error evaluated between estimated concentration and true sensor data (analytical concentration).

3.4 Parallelization Efficiency Analysis

In this section the parallelization efficiency analysis of the OpenMP implementation of NODDE-FVM method is performed. The NODDE-FVM simulations were performed for an instantaneous gas release described in Eq. (2.58). The input conditions are domain lengths $L_X = L_Y = L_Z = 30$ m, wind speeds $U = 1$ m/s and $V = W = 0$, eddy diffusivities $K_{XX} = K_{YY} = K_{ZZ} = 0.5$ m²/s, and an instantaneous gaseous release of 1 kg at time $t = 0$ s and at location $X_0 = 6$ m, $Y_0 = Z_0 = 15$ m. The domain was divided with $N_X = N_Y = N_Z = 240$ for a total of 1.3824×10^7 cells. The number of subdomains in X - and Y -direction was fixed as $N_{DDX} = N_{DDY} = 2$, and the number of subdomains in Z -direction was varied from 1 to 10, $N_{DDZ} = 1, 2, \dots, 10$. The time step was set $\Delta t = 0.001$ s. The simulations were run on a dual socket node with two Intel Xeon Silver 4114 CPUs with total number of 20 physical cores (10 cores per CPU). The hyper-threading, a technology that allows two streams of operations to be executed on the same core, was enabled so that there are 40 threads (logical cores) in total. The NODDE-FVM was compiled using the Intel Fortran Compiler 11.1 on Red Hat Enterprise Linux Server release 7.6 (Maipo). The average elapsed wall-clock time of completing one single time level were recorded for all the tests with different number of threads. We use two metrics to evaluate parallel performance as number of threads increases: first is the speedup, defined as the ratio of the average elapsed wall-clock time of completing one single time level for one thread test to that for multiple threads test; second is the parallel efficiency, defined as the ratio of the actual speedup to the theoretical speedup. Figure 3.11 shows the speedup and parallel efficiency as a function of

the number of OpenMP threads. The theoretical speedup is a linear function of the number of threads as shown in Figure 3.11(a).

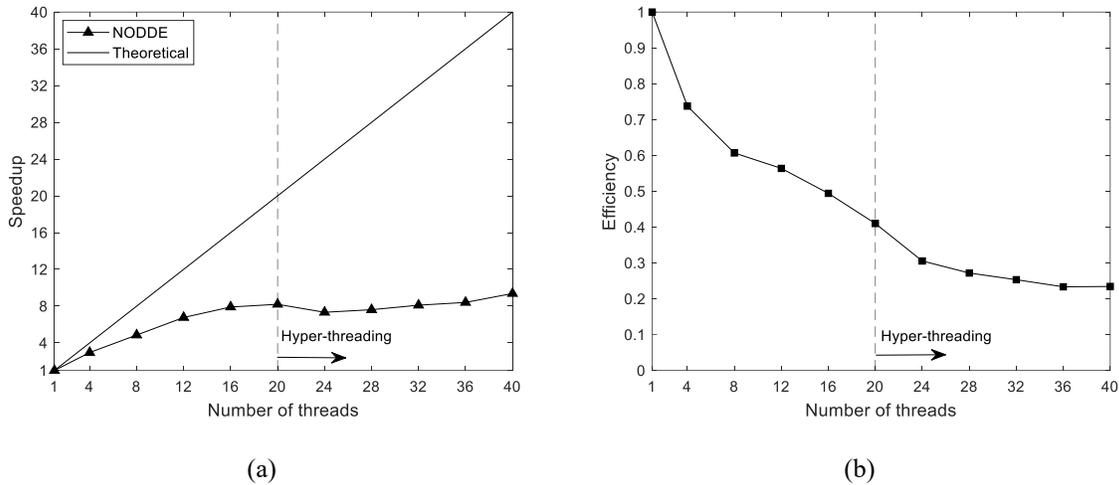


Figure 3.11 Speedup (a) and parallel efficiency (b) of the NODDE-FVM-TVD-RK simulation of the instantaneous gas release in the atmosphere as a function of the number of OpenMP threads.

As the number of threads increases to 20, the speedup keeps increasing but not as greatly as the theoretical speedup and the speedup keeps getting away from the line of the theoretical speedup. This could be due to the non-uniform memory access (NUMA) architecture in which the operation system must coordinate the accesses to the memory from different CPUs. Such coordination workload keeps increasing as the number of threads increases, which causes the slowdown of the speedup and the decreasing of the parallel efficiency. When the number of threads is greater than 20, the hyper-threading is activated resulting in some of the cores executing two streams of operations which leads to the drop of the speedup and parallel efficiency as shown in Figure 3.11. As the number of threads keeps increasing to the fully loaded case of 40 threads, the increasing of the speedup shows up again and the maximum speedup happens when the node is fully loaded because all the resources are used for computing and the full load on all cores eliminates the negative effects of the NUMA architecture.

Chapter 4 Applications of the HT-NODDE-FVM to Large Scale Simulations

This chapter presents the applications of the hybrid estimator and HT-NODDE-FVM on real-time estimations of the advection-diffusion fields in a city-scale domain with realistic atmospheric conditions and SAV parameters. Instantaneous and long-pulse gas release cases are considered to demonstrate the capability of the hybrid estimator and HT-NODDE-FVM for real-time estimations. In these simulations, we also study the effects of different guidance laws and different SAVs on the estimation results. We also conduct the real-time estimation analysis at the end of this chapter. Results have been presented in *Gatsonis et al. [2020]* and *Tian et al. [2020]*.

The hybrid estimator and the developed HT-NODDE-FVM method are used for simulations in a domain that is city-scale as shown in Figure 3.7. The computational domain has $L_X = 20$ km, $L_Y = 5$ km, $L_Z = 2$ km and is discretized with $N_X = 300$, $N_Y = 75$, $N_Z = 30$ finite volumes. The number of subdomains is 27 with $N_{DDX} = 3$, $N_{DDY} = 3$, and $N_{DDZ} = 3$ in X -, Y -, and Z -direction respectively. The wind speed model in X -direction follows the power-law profile [*Seinfeld and Pandis, 2016*]

$$U = \begin{cases} U_r \left(\frac{Z}{Z_r} \right)^p, & Z < Z_r \\ U_r, & Z \geq Z_r \end{cases} \quad (4.1)$$

where Z_r is the reference height, $Z_r = 500$ m, and U_r is the corresponding reference wind speed, $U_r = 10$ m/s or 5 m/s for different simulations; the exponent p depends on the surface roughness and stability conditions, and for well-developed urban areas, $p = 0.4$. The eddy diffusivities with the assumption of unstable atmospheric condition are $K_{XX} = K_{YY} = 100$ m²/s, $K_{ZZ} = 40$ m²/s. Numerical sensor data are generated by solving the process model Eq. (2.58)

with the NODDE-FVM using a grid $N_x = 300$, $N_y = 75$, and $N_z = 30$. The aerial intruder is releasing a contaminant gas and is assumed to represent a helicopter or a light aircraft. The SAV parameters are representative of the Aerosonde UAV (slow) and Sky Warrior UAV (fast). The SAV is positioned at $t = 0$ at $X_{s0} = 10.4$ km, $Y_{s0} = 2.5$ km, $Z_{s0} = 0.55$ km, and is patrolling on a helical trajectory along the Z -direction with center $X_{scc} = 8$ km, $Y_{scc} = 2.5$ km, $Z_{scc} = 0.55$ km, radius $r = 2.4$ km, and pitch 120π . The SAV's patrolling linear velocity (counterclockwise) is $v_s = 30$ m/s for Aerosonde [Maurer, 2002] and 70 m/s for Sky Warrior [Writer, 2019]. The sensor threshold is set to 1×10^{-9} kg/m³ (1 ppb). The estimation gain is $\Lambda = 5 \times 10^{-6}$ and the guidance gains are $k_x = k_y = 21$, $k_z = 4$ for Aerosonde and $k_x = k_y = 50$, $k_z = 4$ for Sky Warrior. The time step used in all the simulations is 1 s.

4.1 Instantaneous Gas Release

4.1.1 Modified Lyapunov Guidance Law Without SAV Dynamical Model, Slow and Fast SAVs

We consider first an instantaneous gas release from an intruder in a city-scale domain and examine the effects of fast and slow SAVs when using modified Lyapunov guidance law without SAV dynamical model in a configuration depicted in Figure 3.7. The intruder releases 100 kg of contaminants at $t = 0$ s and at $X_c = 4.1$ km, $Y_c = 2.5$ km, $Z_c = 1$ km, then it keeps hovering. The reference wind speed is $U_r = 10$ m/s. The simulation time is 480 s.

The simulation results at the final simulation time are shown in Figure 4.1 and Figure 4.2. The red and blue dots denote the intruder and SAV, respectively. The blue lines denote the SAV trajectory. The intruder (red dot) releases 100 kg of contaminants at $t = 0$ s and keeps hovering at its initial location. The SAV keeps patrolling on the helical trajectory until it detects a concentration value which is greater than or equal to the sensor threshold, then the hybrid estimator starts working to estimate the concentration profile and guide the SAV. The random blue line shows the SAV trajectory after it is guided by the hybrid estimator. The SAV trajectory (blue line) is also shown in the process model profile to aid in the comparison between the process model concentration profile and estimated concentration profile and show where it is in

the “true” plume. The results show that the estimation result of the fast SAV is better. Because the fast SAV can patrol more region in the plume and get more information about the concentration, which is critical to the estimation result.

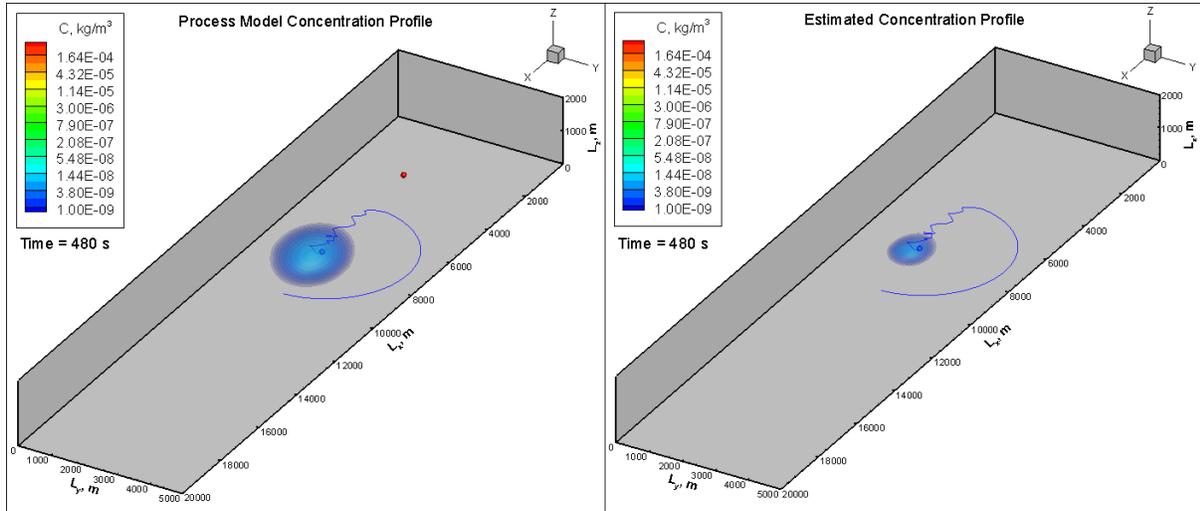


Figure 4.1 The simulation results of the instantaneous gas release for slow SAV (Aerosonde) guided by the modified Lyapunov guidance law without SAV dynamical model at time 480 s.

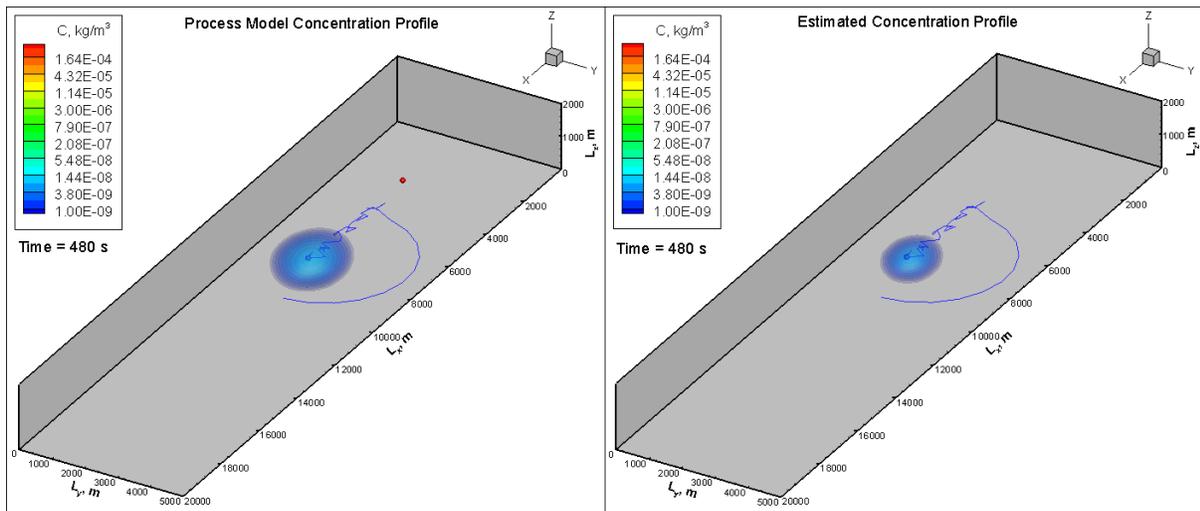


Figure 4.2 The simulation results of the instantaneous gas release for fast SAV (Sky Warrior) guided by the modified Lyapunov guidance law without SAV dynamical model at time 480 s.

4.1.2 Lyapunov Guidance Law with SAV Dynamical Model, Slow SAV

Next, we consider using the Lyapunov guidance law with the SAV dynamical model included. The simulation conditions are the same as in Section 4.1.1 except for the additional slow SAV (Aerosonde) parameters: the mass of the SAV is 13.5 kg, the wing planform area is 0.55 m², the wing span is 2.9 m, the ground speed range is 27-33 m/s, the maximum thrust is 50 N, the maximum bank angle is 0.5 rad, the lift coefficient in stall condition is 1.5, the Oswald efficiency factor is 0.9, the parasitic drag coefficient is 0.0437, the initial patrolling flight path angle and course angle of the SAV are $\gamma_0 = 0.0524$ rad and $\chi_0 = 1.5708$ rad. The simulation result is shown in Figure 4.3.

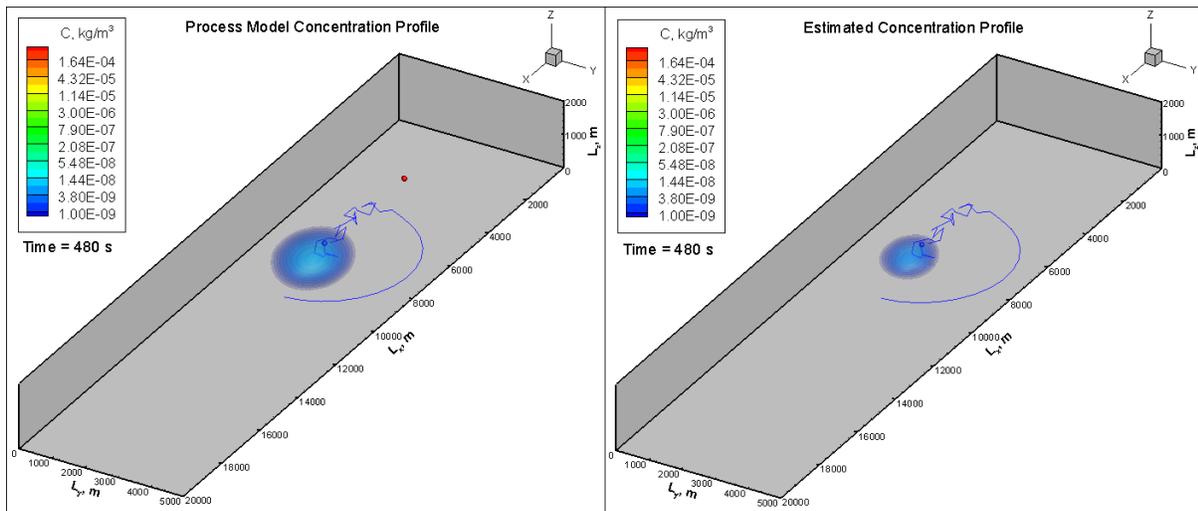


Figure 4.3 The simulation results of the instantaneous gas release for slow SAV (Aerosonde) guided by the Lyapunov guidance law with SAV dynamical model at time 480 s.

By comparing Figure 4.3 with Figure 4.1, we can see that the result of the Lyapunov guidance law with SAV dynamical model is better than that of the modified Lyapunov guidance law without SAV dynamical model, which means with the Lyapunov guidance law and SAV dynamical model included even the slow SAV (Aerosonde) can patrol more region in the plume. It can also be seen from Figure 4.3 and Figure 4.2 that the result is very close to the result of fast SAV (Sky Warrior) guided by the modified Lyapunov guidance law without SAV dynamical model.

4.2 Long-Pulse Gas Release

4.2.1 Modified Lyapunov Guidance Law Without SAV Dynamical Model, Slow and Fast SAVs

Next, we consider a long-pulse gas release and examine the effects of slow and fast SAVs when using modified Lyapunov guidance law without SAV dynamical model. This problem is depicted in Figure 4.4. The reference wind speed is $U_r = 5$ m/s. The intruder starts releasing contaminants at the initial location $X_c = 0.1$ km, $Y_c = 2.5$ km, $Z_c = 1$ km and at time $t = 80$ s and lasts for 200 s with a release rate of 0.5 kg/s. The intruder has velocity components of $v_x = 50$ m/s, $v_y = 0$ m/s, $v_z = 0$ m/s. The simulation time is 1300 s.

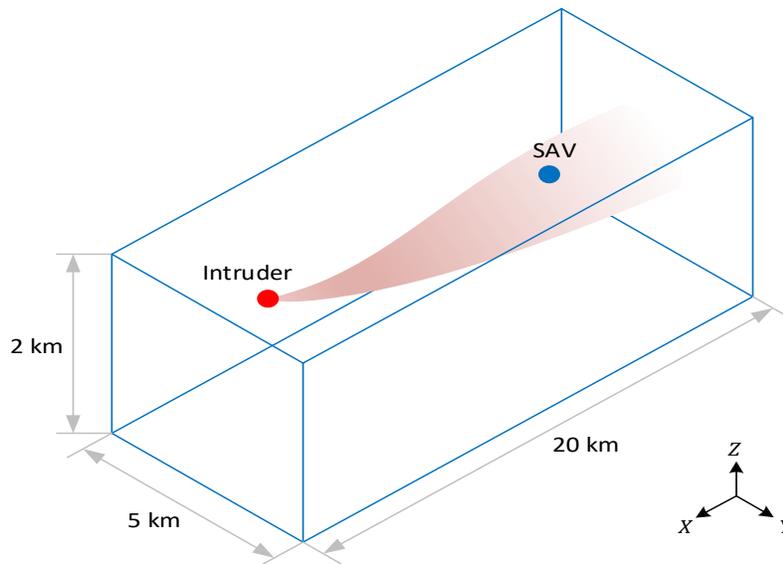


Figure 4.4 An intruder is releasing contaminants over a city and an SAV is estimating the concentration of the contaminants and tracking the intruder.

The simulation results at the final time are shown in Figure 4.5 and Figure 4.6. In this simulation the intruder (red dot) starts the release at $t = 80$ s and then keeps moving until it reaches the boundary of the domain. The gas release forms a cylindrical plume. The SAV keeps the initial helical patrolling trajectory and starts activating the hybrid estimator until it detects a concentration value greater than or equal to the sensor threshold. Then the hybrid estimator starts estimating the concentrations of the plume and the SAV is guided by the modified Lyapunov

guidance law. The comparison between Figure 4.5 and Figure 4.6 shows clearly that the fast SAV has the better estimation results. This is due to the fact that the faster SAV catches up the region with large concentration values and performs a better estimation.

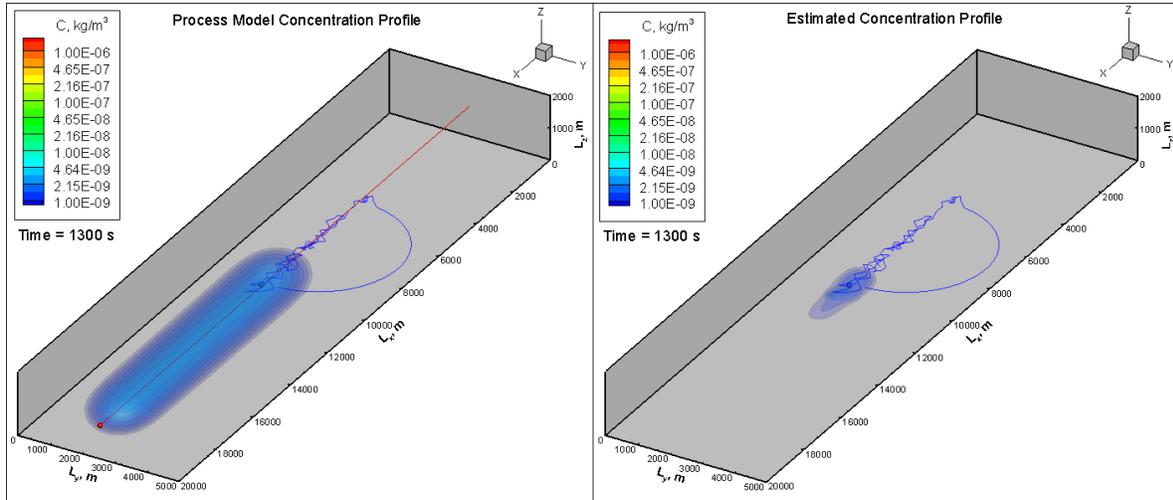


Figure 4.5 The simulation results of the long-pulse gas release for slow SAV (Aerosonde) guided by the modified Lyapunov guidance law without SAV dynamical model at time 1300 s.

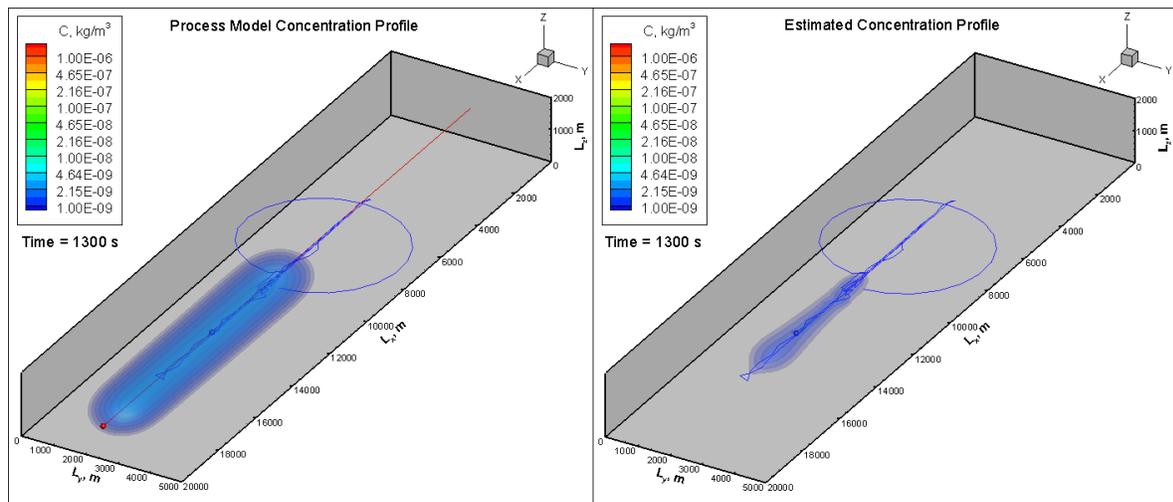


Figure 4.6 The simulation results of the long-pulse gas release for fast SAV (Sky Warrior) guided by the modified Lyapunov guidance law without SAV dynamical model at time 1300 s.

4.2.2 Lyapunov Guidance Law with SAV Dynamical Model, Slow SAV

Finally, we consider using the Lyapunov guidance law with the SAV dynamical model included for the long-pulse gas release case. The simulation conditions are the same as in Section 4.2.1 and the SAV parameters are the same as in Section 4.1.2. The simulation results at the final time $t = 600$ s are shown in Figure 4.7. In this simulation, the SAV is guided by the Lyapunov guidance law with SAV dynamical model. We can see from Figure 4.7 that the SAV can track the plume well and the hybrid estimator is trying to reproduce the shape of the cylindrical plume.

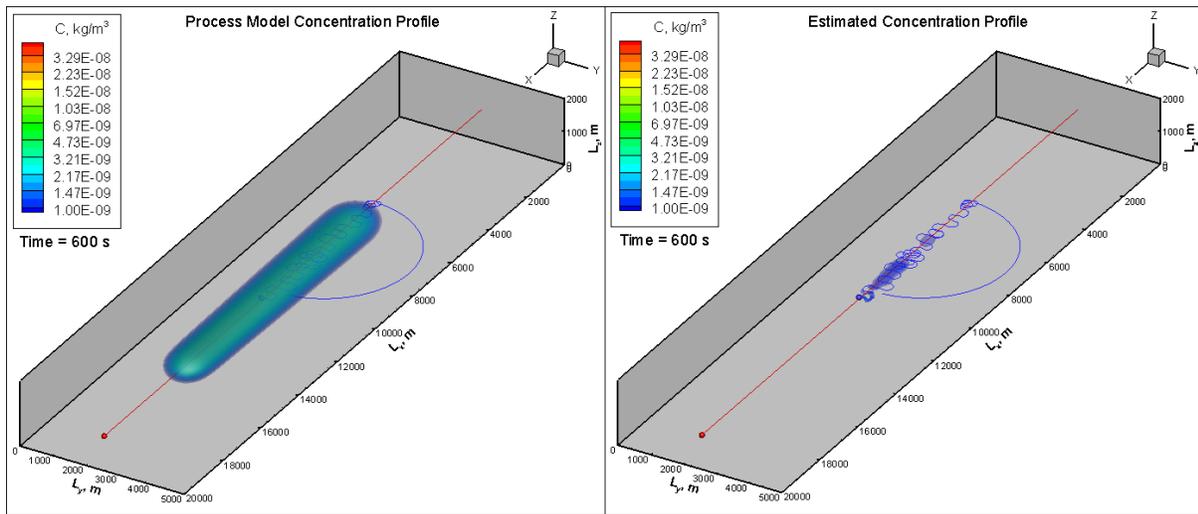


Figure 4.7 The simulation results of the long-pulse gas release for slow SAV (Aerosonde) guided by the Lyapunov guidance law with SAV dynamical model at time 600 s.

4.3 Real-Time Estimation Analysis

To achieve real-time estimation by the SAV using the HT-NODDE-FVM hybrid estimator, it is required that the wall clock time of completing an iteration over the entire domain (all subdomains) Δt_{est} is smaller than the maximum numerical time step Δt_{max} calculated by Eq. (3.133). A series of simulations are conducted using the scenario of instantaneous gas release with the wind profile and the computational conditions and guidance law as in Section 4.1.1. Real-time estimation analysis is conducted by varying the grid resolution as well as the number of subdomains (or threads) for both the process model and the hybrid estimator model. The results in Figure 4.8 show the performance of the HT-NODDE-FVM hybrid estimator by

plotting the ratio of $\Delta t_{est}/\Delta t_{max}$ as a function of the number of subdomains (or threads) and number of finite volumes. To achieve real-time estimation the ratio of $\Delta t_{est}/\Delta t_{max}$ must be less than 1. Figure 4.8 shows that even with 6 threads the HT-NODDE-FVM can accomplish real-time implementation for all discretization cases considered. The real-time hybrid estimator has been implemented without sensor delays and SAV dynamics. Inclusion of these processes will provide additional time margins (longer time scales) that can improve the real-time estimation performance and allow use of even more refined grids for improved accuracy.

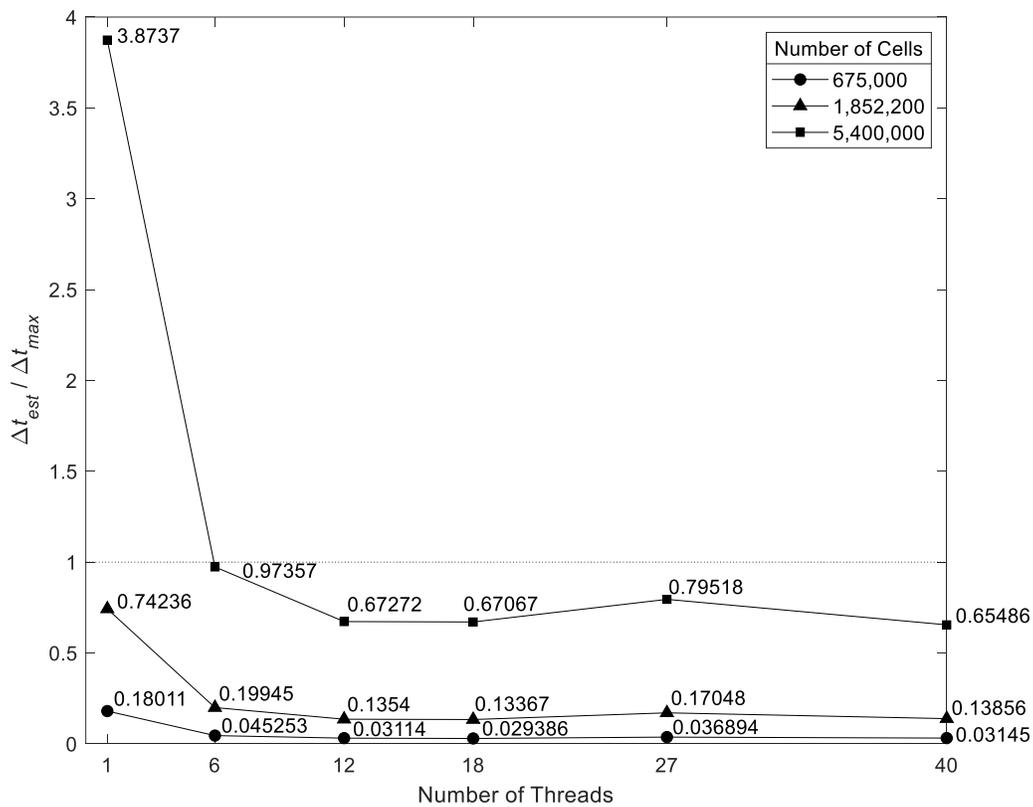


Figure 4.8 Real-time estimation analysis as functions of number of subdomains (threads) and number of finite volumes (resolution).

Chapter 5 Summary, Conclusions, and Recommendations

5.1 Summary and Conclusions

This work is devoted in the development of a hybrid estimator and its numerical solutions via a new heterogeneous nonoverlapping domain decomposition explicit finite volume method with total variation diminishing (TVD) and a Runge-Kutta integrator (HT-NODDE-FVM). The HT-NODDE-FVM for a is developed to perform real-time estimation of a contaminant plume released by a stationary or moving gaseous source in three-dimensional domains under prescribed atmospheric conditions and guides a sensing aerial vehicle (SAV) to positions in the plume that optimize the performance of the estimator.

The process model was derived starting from the Navier-Stokes equations of multi-component system together with the continuity equation of single species. By using the Boussinesq approximation for the lower atmosphere layer, using the Reynolds averaged equations to account for turbulent effects, assuming the released species is a single trace species and is in the diffusion regime, assuming the molecular diffusion fluxes and their gradients are negligible compared with the turbulent ones, and assuming the principal axes of eddy diffusivity tensor coincide with the coordinate axes, the process model arrived at the well-known advection-diffusion equation of a single species in the atmosphere

$$\begin{aligned} \frac{\partial C}{\partial t} = & -\frac{\partial}{\partial X}(CU) - \frac{\partial}{\partial Y}(CV) - \frac{\partial}{\partial Z}(CW) + \frac{\partial}{\partial X}\left(K_{XX}\frac{\partial C}{\partial X}\right) + \frac{\partial}{\partial Y}\left(K_{YY}\frac{\partial C}{\partial Y}\right) \\ & + \frac{\partial}{\partial Z}\left(K_{ZZ}\frac{\partial C}{\partial Z}\right) + S_r \end{aligned} \quad (5.1)$$

The estimator model used in this work is *hybrid*, it is based on the advection-diffusion equation and involves a Luenberger observer in the vicinity of the SAV and a naive observer in the rest of the region, resulting in a hybrid advection-diffusion type of particle differential equation with transmission conditions between the two regions.

The spatial distribution of the sensor is modeled as a 3D spatial Dirac delta function based on Lagrangian sensing technique. The sensor is assumed to be ideal and provides instantaneous readings (no delays) without errors.

Two different guidance methods for the SAV were implemented. One is the modified Lyapunov guidance law without SAV dynamical model, the other is the Lyapunov guidance law with six degree of freedom fixed wing aircraft dynamical model. It was assumed that the SAV has the knowledge of concentration and the concentration gradient at its location and it moves following the guidance law to minimize the estimation error.

In this work we developed a new HT-NODDE-FVM for the numerical solution of the hybrid estimator and was implemented with total variation diminishing (TVD) filter and a Runge-Kutta integrator, in parallel. The entire computational domain is discretized with a structured grid and is divided into multiple subdomains with each subdomain having the same number of finite volume cells. The transmission conditions are imposed explicitly when performing the FVM-TVD spatial discretization without the need for iterations as common in implicit domain decomposition methods. The HT-NODDE-FVM is parallelized by using the OpenMP paradigm such that each subdomain is handled by a distinct CPU thread. All estimator model equations in all the subdomains are solved in parallel. A barrier-type synchronization point is applied to ensure all the FVM-TVD related calculations are finished in each subdomain within current time level before moving to the next time level.

The verification and error analysis of the NODDE-FVM were conducted by solving non-dimensional 3D advection and advection-diffusion equations. The numerical solutions were compared with the corresponding analytical solutions. When applied to advection or advection-diffusion equations, the method becomes homogenous non-overlapping domain decomposition method. For the non-dimensional 3D advection equation, four initial density configurations were used for the simulations including a continuous Gaussian density distribution, two piecewise

Gaussian distributions, and a cubical density distribution. The order of accuracy was calculated by using the L^1 norm of the error between numerical and analytical solutions for each initial density distribution case. The results showed that for smoothly varying functions the NODDE-FVM becomes between order 1 and 2, which is consistent with the results of single-domain FVM-TVD method. For the non-dimensional 3D advection-diffusion equation, three different Peclet numbers were used and covered diffusion dominated to advection dominated flows. The initial density distribution used in the simulations followed the analytical solution at certain time instance. The L^1 norm of the error between numerical and analytical solutions for each Peclet number case was used to calculate the order of accuracy. The results showed that the NODDE-FVM with TVD achieves near second order of accuracy for all the cases considered.

The method was also applied on the hybrid estimator to study the effects of estimation gain, grid resolution, and sensor model on estimation results by studying the L^2 norm of the estimation error. This case involved an instantaneous release by a stationary source in a large domain with constant atmospheric properties. Two different grid resolutions were used for the simulations. Results showed that for both the coarse and fine grid resolutions there is an optimum gain that minimizes the estimation error. Results also showed that the coarse grid has better estimation error than the fine grid due to the larger sensor volume of the former. Another simulation was conducted to study further the impact of grid resolution, estimation gain, and numerical data on the estimation error. Different grid resolutions and different optimal estimation gains were used for this simulation. The estimator error norms L^1 , L^2 , and L^∞ were evaluated and plotted. The estimation error was evaluated in two ways – one was evaluated between estimated concentration and numerical sensor data obtained from the NODDE-FVM of the process model, the other was evaluated between estimated concentration and the analytical solution. Results of the first way showed that as grid resolution coarsens the process state knowledge increases due to the increase in sensor volume and as a result the estimator error decreases. In the limit of a single-cell domain the error would be identically zero because the sensor would have knowledge of the entire state. Results of the second way showed that the errors decrease with increasing grid size, reach a minimum and then increase. This behavior is due to the compound effects of increased state knowledge and reduced numerical fidelity as cell size increases.

Parallelization efficiency analysis for the OpenMP implementation of the NODDE-FVM-TVD-RK method was conducted to study the speedup and efficiency of the parallelization by recording the average elapsed wall-clock time of completing one single time iteration for different tests with different number of threads. Results show the expected behaviors of the non-uniform memory access (NUMA) architecture and the full load of all the logical computer threads.

The HT-NODDE-FVM with TVD-RK method was implemented on realistic estimation problems for instantaneous and long-pulse gaseous release cases in a km-scale domain with realistic atmospheric conditions and SAV parameters. These simulations were used to study the impacts of different guidance methods and different SAVs on the estimation results. Results show that fast SAV can provide better estimation results than slow SAV because the fast SAV can patrol more plume region. Both of the two guidance methods can provide good estimation and tracking results, although the Lyapunov guidance law with SAV dynamical model shows better estimation results in the instantaneous gaseous release case.

The real-time estimation analysis was conducted by comparing the wall-clock time of completing a single time iteration over all the subdomains (Δt_{est}) with the maximum allowable time step derived from the numerical stability analysis (Δt_{max}). The relation between the ratio of $\Delta t_{est}/\Delta t_{max}$ and the number of subdomains (or threads) for different grid resolutions for both the process model and the hybrid estimator model was used to evaluate the capability of the real-time estimation. All the simulations using km-scale domain under realistic ambient atmospheric conditions demonstrate that the hybrid estimator with the HT-NODDE-FVM method achieves real-time computation of the advection-diffusion field.

5.2 Recommendations for Future Work

The focus of this work was on the development of the HT-NODDE-FVM with TVD-RK method to solve the hybrid estimator to achieve real-time estimation of the advection-diffusion fields. In the future, the estimation results could be improved, and the experimental work may be needed to validate the numerical results.

- In this work, we used one single sensor to measure the concentration. Using multiple sensors on board multiple SAVs with a fixed flying formation to measure the concentrations and evaluate the concentration gradients could improve the estimation results and relax the assumption of gradient knowledge imposed in the current derivation.
- Another way to improve the concentration gradient evaluations could be to implement a grid adaptation algorithm such that the subdomain where the sensor resides has higher grid resolution than the other subdomains. Such grid adaptation algorithms have been developed and implemented in a 2D hybrid estimator simulation [Hu *et al.*, 2021] and in 3D homogenous estimators [Egorova *et al.*, 2016].
- In order to improve the accuracy of the Luenberger-naïve estimator, a more computationally expensive estimator based on the Riccati equations can be used in the subdomain where the sensor resides.
- To validate the hybrid estimator and the HT-NODDE-FVM, set up of experiments is needed. One example can employ a sensing ground vehicle and execute the estimator on a computer that communicates with it. The set up requires a harmless gaseous source which can be stationary or mobile. Such setups have been under construction [Barney and Rivard, 2016; McGrath *et al.*, 2018]. These ground tests can be expanded using small UAVs.

References

- [1] Acebrón JA, Rodríguez-Rozas Á, Spigler R. Domain decomposition solution of nonlinear two-dimensional parabolic problems by random trees. *Journal of Computational Physics*. 2009 Aug 20; 228(15):5574-91.
- [2] Achdou Y, Nataf F. A Robin-Robin preconditioner for an advection-diffusion problem. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*. 1997 Dec 1; 325(11):1211-6.
- [3] Akhtar I, Borggaard J, Burns JA, Imtiaz H, Zietsman L. Using functional gains for effective sensor location in flow control: a reduced-order modelling approach. *Journal of Fluid Mechanics*. 2015 Oct 25; 781:622.
- [4] Anderson JD. *Introduction to flight*. Vol. 199, McGraw-Hill, Boston; 2005.
- [5] Arya SP. *Air pollution meteorology and dispersion*. New York: Oxford University Press; 1999.
- [6] Barney MR, Rivard SM. *Design and Integration of an Indoor Plume Experimental Setup*. Major Qualifying Project. Worcester Polytechnic Institute. 2016 Mar 23; E-project-032316-113717.
- [7] Beard RW, McLain TW. *Small unmanned aircraft: Theory and practice*. Princeton university press; 2012 Feb 26.
- [8] Bennett A. *Lagrangian fluid dynamics*. Cambridge University Press; 2006 Mar 9.
- [9] Bird RB, Stewart WE, Lightfoot EN. *Transport phenomena*. John Wiley & Sons; 2006 Dec 11.
- [10] Birgler N, Masson R, Trety L. A domain decomposition method to couple nonisothermal compositional gas liquid Darcy and free gas flows. *Journal of Computational Physics*. 2018 Sep 1; 368:210-35.

- [11] Blayo E, Halpern L, Japhet C. Optimized Schwarz waveform relaxation algorithms with nonconforming time discretization for coupling convection-diffusion problems with discontinuous coefficients. In Domain decomposition methods in science and engineering XVI 2007 (pp. 267-274). Springer, Berlin, Heidelberg.
- [12] Burns JA, Rautenberg CN. The infinite-dimensional optimal filtering problem with mobile and stationary sensor networks. Numerical Functional Analysis and Optimization. 2015 Feb 1; 36(2):181-224.
- [13] Chandra R, Dagum L, Kohr D, Menon R, Maydan D, McDonald J. Parallel programming in OpenMP. Morgan Kaufmann; 2001.
- [14] Court JR. Estimation of the Concentration from a Moving Gaseous Source in the Atmosphere Using a Guided Sensing Aerial Vehicle (Doctoral dissertation, Worcester Polytechnic Institute); 2012.
- [15] Demetriou MA, Gatsonis NA, Court JR. Coupled controls-computational fluids approach for the estimation of the concentration from a moving gaseous source in a 2-D domain with a Lyapunov-guided sensing aerial vehicle. IEEE Transactions on Control Systems Technology. 2013 Jul 3; 22(3):853-67.
- [16] Egorova T, Gatsonis NA, Demetriou MA. Estimation of gaseous plume concentration with an unmanned aerial vehicle. Journal of guidance, control, and dynamics. 2016 Jun; 39(6):1314-24.
- [17] Fiorelli E, Leonard NE, Bhatta P, Paley DA, Bachmayer R, Fratantoni DM. Multi-AUV control and adaptive sampling in Monterey Bay. IEEE journal of oceanic engineering. 2006 Oct; 31(4):935-48.
- [18] Gander MJ, Halpern L, Japhet C, Martin V. Advection diffusion problems with pure advection approximation in subregions. In Domain decomposition methods in science and engineering XVI 2007 (pp. 239-246). Springer, Berlin, Heidelberg.
- [19] Gastaldi F, Quarteroni AL, Landriani GS. Coupling of two-dimensional hyperbolic and elliptic equations. Computer methods in applied mechanics and engineering. 1990 Jun 1; 80(1-3):347-54.
- [20] Gatsonis NA, Tian X, Demetriou MA, Burns J. A heterogeneous non-overlapping domain decomposition explicit finite volume method for real-time estimation of 3D advection-diffusion fields via a moving sensor. Journal of Computational Physics. 2020 (submitted).

- [21] Halpern L, Japhet C, Szeftel J. Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problems. *SIAM Journal on Numerical Analysis*. 2012; 50(5):2588-611.
- [22] Hirsch C. *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Elsevier; 2007 Jul 18.
- [23] Hu W, Demetriou MA. Domain decomposition methods for the state estimation of parabolic PDEs in 2D rectangular domains: well-posedness and convergence. In 2019 18th European Control Conference (ECC) 2019 Jun 25 (pp. 1920-1925). IEEE.
- [24] Hu W, Demetriou MA, Tian X, Gatsonis NA. Hybrid domain decomposition filters for advection-diffusion PDEs with mobile sensors. *Automatica*. 2021 (submitted).
- [25] Kumar S, Kumar BR. A domain decomposition Taylor Galerkin finite element approximation of a parabolic singularly perturbed differential equation. *Applied Mathematics and Computation*. 2017 Jan 15; 293:508-22.
- [26] Kundu PK, Cohen IM, Dowling DR. *Fluid mechanics*. 6th ed. San Diego, CA: Academic Press; 2015.
- [27] Kurganov A, Tadmor E. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *Journal of Computational Physics*. 2000 May 1; 160(1):241-82.
- [28] LeVeque RJ. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Society for Industrial and Applied Mathematics; 2007 Jan 1.
- [29] Li Q, Lu J, Sun W. Diffusion approximations and domain decomposition method of linear transport equations: Asymptotics and numerics. *Journal of Computational Physics*. 2015 Jul 1; 292:141-67.
- [30] Lube G, Müller L, Otto FC. A non-overlapping DDM of Robin-Robin type for parabolic problems. In *Proceedings of the Eleventh International Conference on Domain Decomposition Methods*, Greenwich, UK 1998.
- [31] Maurer J. Polar remote sensing using an unpiloted aerial vehicle (UAV). <http://www2.hawaii.edu/~jmaurer/uav/>; 2002.

- [32] McGrath OM, Kontopyrgos M, Kasapis S, Halama S, Preston BP. Estimation of a Plume with an Unmanned Terrestrial Vehicle. Major Qualifying Project. Worcester Polytechnic Institute. 2018 Mar 20; E-project-032018-145855.
- [33] Prabhakaran S, Doss LJ. Total variation diminishing finite volume schemes for one dimensional advection-diffusion equation and the relationship between flux limiter and mesh parameters. *International Journal of Pure and Applied Mathematics*. 2015; 101(2):233-50.
- [34] Quarteroni A, Pasquarelli F, Valli A. Heterogeneous domain decomposition: principles, algorithms, applications. In *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations 1992* (pp. 129-150). SIAM, Philadelphia.
- [35] Quarteroni A, Stolic L. Homogeneous and heterogeneous domain decomposition for compressible fluid flows at high Reynolds numbers. In *Computational Fluid Dynamics Review 1998: (In 2 Volumes) 1998* (pp. 1064-1078).
- [36] Rivera W, Zhu J, Huddleston D. An efficient parallel algorithm with application to computational fluid dynamics. *Computers & Mathematics with Applications*. 2003 Jan 1; 45(1-3):165-88.
- [37] Seinfeld JH, Pandis SN. *Atmospheric chemistry and physics: from air pollution to climate change*. 3rd ed. Hoboken, NJ: Wiley-Blackwell; 2016.
- [38] Spiegel EA, Veronis G. On the Boussinesq approximation for a compressible fluid. *The Astrophysical Journal*. 1960 Mar; 131:442.
- [39] Stolk CC. A rapidly converging domain decomposition method for the Helmholtz equation. *Journal of Computational Physics*. 2013 May 15; 241:240-52.
- [40] Thuburn J. TVD schemes, positive schemes, and the universal limiter. *Monthly weather review*. 1997 Aug; 125(8):1990-3.
- [41] Tian X, Demetriou MA, Gatsonis NA. Domain Decomposition for a Hybrid State Estimation of a Plume Field with a Moving Sensor. AIAA 2020-0394, AIAA SciTech 2020 Forum, Orlando, FL, USA, January 6-10, 2020, doi:10.2514/6.2020-0394.
- [42] Versteeg HK, Malalasekera W. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education; 2007.
- [43] Wang H, Liu J, Espedal MS, Ewing RE. A characteristic nonoverlapping domain decomposition method for multidimensional convection-diffusion equations. *Numerical*

Methods for Partial Differential Equations: An International Journal. 2005 Jan; 21(1):89-103.

[44] Writer S. General Atomics MQ-1C Gray Eagle (Sky Warrior). https://www.militaryfactory.com/aircraft/detail.asp?aircraft_id=785; 2019.

[45] Zheng Z, Simeon B, Petzold L. A stabilized explicit Lagrange multiplier based domain decomposition method for parabolic problems. Journal of Computational Physics. 2008 May 1; 227(10):5272-85.