CityFFD – City Fast Fluid Dynamics Model for Urban Microclimate Simulations

Mohammad Mortezazadeh Dorostkar

A Thesis

In the Department

of

Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Building Engineering) at

Concordia University

Montreal, Quebec, Canada

June 2019

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Mohammad Mortezazadeh

Entitled: CityFFD – City Fast Fluid Dynamics Model for Urban Microclimate Simulations

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

______________________________ Chair
Dr. Alex De Visscher

______________________________ External Examiner
Dr. Jianlei Niu

______________________________ External to Program
Dr. Ali Dolatabadi

______________________________ Examiner
Dr. Ted Stathopoulos

______________________________ Examiner
Dr. Attila Michael Zsaki

______________________________ Thesis Supervisor
Dr. Liangzhu (Leon) Wang

Approved by ______________________________
Dr. Michelle Nokken, Graduate Program Director

July 30, 2019 ______________________________
Dr. Amir Asif, Dean,
Gina Cody School of Engineering and Computer Science

Abstract

CityFFD – City Fast Fluid Dynamics Model for Urban Microclimate Simulations

Mohammad Mortezazadeh Dorostkar, Ph.D.

Concordia University, 2019

In recent years, due to the rapid population growth and the preference to live in urban areas, urbanization has intensely increased. Currently, based on a United Nation report, 55% of the population live in the cities and the number is expected to reach about 68% by 2050. Urban microclimate has significant impacts on human life and health, and building energy performance. Urban microclimate information, such as wind velocity, temperature, humidity, pollutant dispersion levels, and local precipitation, are often important for accurate evaluations of building energy performance, indoor and outdoor human comfort, extreme events, and emergency situations. For example, it was reported that indoor temperature estimated with the microclimate information could be at least 5 °C different from that without it, which could be significant for the evaluations of indoor thermal comfort. The study of urban microclimate includes both observational and numerical approaches. The observational study is often related to field measurements, satellite imagery, and laboratory tests, e.g. in wind tunnels. The numerical approach is often based on computer models, such as CFD (computational fluid dynamics), for high-resolution and relatively small computing domains, compared to larger scale regional climate models, such as WRF and GEM-SURF. The latter two models are mostly used to model the domain size of 1~10 km with the resolution more than 100 m so they are not developed for urban microclimate and building-level simulations. In comparison, CFD has been applied to the urban microclimate of less than 1 km with a resolution less than 10 m down to the building level.

However, conventional CFD solvers often perform unsatisfactorily for microscale and complicated problems because of numerical constraints such as stability issues associated with CFL condition, which is a necessary condition for convergence while solving certain partial differential equations (usually hyperbolic PDEs) numerically. Thus, conventional tools are often computationally expensive for modeling microclimates and consequently impractical for urban-scale problems. Recently, there are an increasing amount of efforts focusing on developing faster and accurate CFD techniques such as based on Fast Fluid Dynamics (FFD) methods. A FFD method relies on semi-Lagrangian and fractional step methods. FFD methods is fundamentally an explicit method without the CFL constraint so it is unconditionally stable even under large time steps and coarse grid resolutions, which are common for urban microclimate problems. In the meantime, the conventional FFD methods are often dependent on low-order interpolation schemes and thus with high numerical errors, which are the main drawbacks of this approach.

The main objective of this thesis is to develop a fast and accurate CFD solver with a series of new computing algorithms based on semi-Lagrangian approach for modeling urban/city scale microclimates. The new solver with the name of CityFFD (city Fast Fluid Dynamics), is designed for tackling the challenges of large domain, coarse grid, and/or large time step, which are typical for urban microclimate simulations, without a heavy reliance on computer resources, such as the possibility of running on personal computers. First, a novel high-order interpolation scheme is proposed to significantly reduce the numerical errors of conventional semi-Lagrangian solvers. The new interpolation scheme enables the possibility of obtaining fast and accurate results even on coarse grids. The second algorithm focuses on the simulation accuracy associated with the time step of the semi-Lagrangian method. A new scheme of an adaptive time step is developed to adjust the time step dynamically according to local truncation errors. To improve the estimation of the

characteristic curves, a new algorithm is proposed by considering the acceleration of the flow particles inside the computational domain which can provide highly accurate results and capture the complicated flow fields even by using a large time step. The fourth algorithm is to speed up the simulation by eliminating the need for solving the Poisson equation, which is often the most time-consuming operation of conventional semi-Lagrangian models. The new scheme is based on the concept of the artificial compressibility of solving incompressible flows and makes it easier to implement parallel computing techniques, such as the NVIDIA GPU CUDA and the OpenMP. The last feature of CityFFD is adding Large Eddy Simulation (LES) model to capture the turbulence behavior of the flow in urban environments. In this section, a parallel OpenMP geometry reader is developed to read the city scale geometries in a fast manner. At the end, the proposed CityFFD model is demonstrated by a case study: the modeling of an extreme weather event, the snow-storm of the century in Montreal, for evaluating building resilience during the storm, to show the importance of urban microclimate and its impact on human health and indoor environment.

Acknowledgements

I would like to express my special thanks of gratitude to my supervisor Dr. Liangzhu (Leon) Wang, for the continuous support of my Ph.D. and giving me the golden opportunity to do this wonderful project on the topic (development of a fast and accurate CFD model). Without his motivation, patient guidance, and persistent assistance, this thesis would not have been possible.

Some special words of gratitude go to my friend who has always been a major source of support in this project and has collaborated on the development of a powerful urban environment model: Ali Katal.

I would like to thank the members of examining committee, Dr. Jianlei Niu, Dr. Ted Stathopoulos, Dr. Ali Dolatabadi, and Dr. Attila Michael Zsaki, for their precious comments on my thesis work.

Last but not least, my deepest gratitude goes to my parents, Maryam Yaghoubzadeh and Ali Mortezazadeh, for their unconditional love and support; to my wife, Mitra Hassanzadeh, who helps me a lot and brings much happiness into my life.

Table of Contents

List of Figures

List of Tables

Nomenclature

A	Cell area (2-D)	$[m^2]$
A_{matrix}	Square matrix	-
a, b, c	Matrix components	-
B	Right hand side of the linear system of equations	-
b_1, b_2	Boundary points	$[m]$
C	Error coefficient functions	$[m^2]$
C_{PC}	Number of cores	-
c_{p_∞}	Reference specific heat capacity at constant pressure	$\left[\frac{K.m^2}{s^2}\right]$
c_s	Smagorinsky constant	-
E, E_b, E_f	Truncation error values	-
$E_{l_1}, E_{l_2}, E_{l_\infty}$	Error with first, second, and infinite norms, respectively	-
E_{cons}	Mass conservation error	-
f	Source term	-
g	Gravity	$\left[\frac{m}{s^2}\right]$
L	Length	$[m]$
p	Pressure	$\left[\frac{kg}{m.s^2}\right]$
P_{solver}	Portion of solver which is executed on multiple cores	-
Re	Reynolds number	-
$\bar{S}$	Large-scale strain rate	$\left[\frac{1}{s}\right]$
S	Characteristic curve	$[m]$
s, t, w	Weighting function coefficients for the interpolation scheme	-
S_{PC}	Speedup or the ratio of real computational speed for the parallel simulation to the serial simulation	-
t	time	$[s]$
T	Temperatures	$[K]$
x, y, z	Components of the position in the Cartesian coordinates	$[m]$
$x_{unknown}$	Vector of unknown variables	-
U	Velocity vector	$\left[\frac{m}{s}\right]$
u, v, w	Velocity components in the Cartesian coordinates	$\left[\frac{m}{s}\right]$
V	Local mean velocity	$\left[\frac{m}{s}\right]$

Greek and math symbols

α	thermal diffusivity	$\left[\frac{m^2}{s}\right]$
β	Artificial compressibility coefficient	$\left[\frac{m}{s}\right]$
β_∞	Reference thermal expansion	$\left[\frac{1}{K}\right]$
ρ	Density	$\left[\frac{kg}{m^3}\right]$
μ	Dynamic viscosity	$\left[\frac{kg}{m.s}\right]$
ν	Kinematic viscosity	$\left[\frac{m^2}{s}\right]$
$\emptyset$	General scalar variable	-
$\Delta t, \Delta x, \Delta y, \Delta z$	Time and space steps, respectively	$[m]$
$\Delta t_i, \Delta t_l, \Delta t_g$	user-defined initial, new local, and global time steps, respectively	$[s]$

Subscripts

a	Arrival point
d	Departure point
C	Cold reference temperature
H	Hot reference temperature
i	Index of Cartesian coordinates
n	Time step number
$n_{i,j}$	Departure point's neighbors
t	Turbulence properties
∞	Reference number
l_1	First norms
l_2	Second norms
l_∞	infinite norms

Abbreviations

ABL	Atmospheric Boundary Layer
CFD	Computational Fluid Dynamics
CityFFD	City Fast Fluid Dynamics
CityBEM	City Building Energy Model
CityBES	City Building Energy Saver
CitySIM	City Simulation
CPU	Central Processing Unit

FFD	Fast Fluid Dynamics
GA	Genetic Algorithm
GPU	Graphics Processing Unit
LES	Large Eddy Simulation
MMM	Mesoscale Meteorological Models
NWP	Numerical Weather Prediction
RANS	Reynolds-Averaged Navier-Stokes
SGS	Sub-Grid Scale model
SL	Semi-Lagrangian
SLAC	Semi-Lagrangian Artificial Compressibility method
SLPB	Semi-Lagrangian Pressure Based method
URANS	Unsteady Reynolds-Averaged Navier-Stokes
UBEM	Urban Building Energy Model

Preface

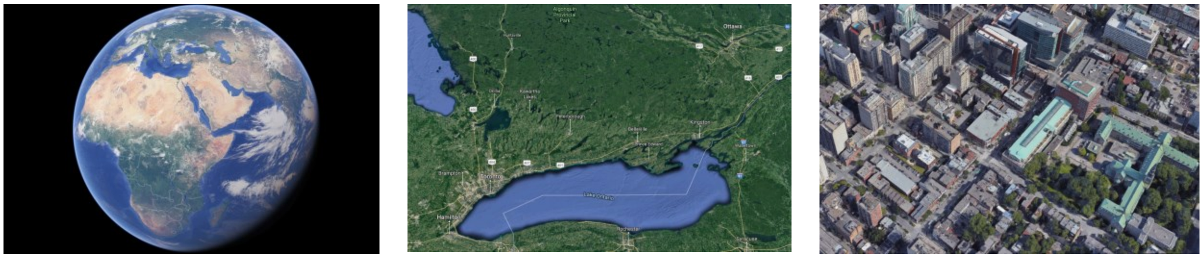
This is a manuscript-based thesis including a collection of four published journal papers and two manuscripts to be submitted. The six papers compose Chapter 3 ~ Chapter 8, and each manuscript is an independent chapter. These six manuscripts are modified from the original ones for the purpose of consistency and readability. The references of the manuscripts are combined at the end of the thesis.

Chapter 1 Introduction

The two most important topics among building engineers, city planners, and governments are building energy performance, and human health and comfort. Based on a United Nation report, 55% of the global population live in cities by 2050 and this number is expected to reach approximately 68% (United Nation, 2014). According to many studies and reports, people in cities are facing several serious problems, such as air pollution, high energy costs, urban heat island and thermal stress, and environmental hazards like flash flooding (Jacobi et al., 2010). It was reported that two main causes of these problems are urbanization and global climate change (Shashua-Bar and Hoffman, 2003). In recent years, due to rapid population growth and the preference to live in urban areas, urbanization has increased intensely. Urbanization, including both the expansion of the cities' boundaries and the increase in urban textures, is one of the most important reasons for urban microclimate change, which can directly affect building energy demand and human health and comfort. Additionally, global climate change, such as global warming and future temperature increase, can increase the extreme events and heat waves in the cities, and consequently, influence citizens' lives.

To create safer and healthier cities, it is necessary to study the urban environment to identify urban threats and risks for citizens and create solutions. The urban environment is a part of a larger domain known as the atmospheric boundary layer (ABL). This layer makes contact with the surface of the earth and covers the urban environment (Brown, G., 2001). An important characteristic of ABL is rapid variation of air characteristics, such as air wind speed, temperature, and humidity, all of which directly affect the urban environment. To study the atmospheric phenomena and ABL characteristics, researchers have highlighted three sub-domains in ABL characterized by size and time (Orlanski, 1975; Blocken, 2015):

1. Macroclimate: This sub-domain has a spatial scale of 100~2000 kilometers and includes weather phenomena such as global circulation and tropical cyclones. The temporal scale for macroclimate problems is a few days to a couple of years.
2. Mesoclimate: This sub-domain has a spatial scale from five to a few hundred kilometers and includes thunderstorms, precipitation, and sea breezes. The temporal scale for mesoclimate events is between one hour and one month.
3. Microclimate: This sub-domain includes any short-lived weather phenomena in a spatial scale of 2 km or less, including heat transfer, near-ground turbulence, vegetation, and surface water effects. The temporal scale of these problems can vary from a few seconds to some hours.



Macroclimate Mesoclimate Microclimate

Figure 1-1 Schematic of the special scales in ABL for three sub-domains.

Various approaches are used to investigate these sub-domains, such as satellite imagery, field measurements, and numerical simulations. In recent decades, due to the remarkable progress of the computer hardware resources, numerical simulations have gained popularity in the ABL airflow modeling and enabled researchers around the world to study the ABL. Because these sub-domains differ by phenomena, spatial scales, and time periods, different numerical approaches are used for each: numerical weather prediction (NWP) for macroclimates, mesoscale meteorological models (MMM) for mesoclimate, and computational fluid dynamics (CFD) for microclimates (Toparlar et al., 2017).

NWP is a mathematical approach to simulate atmosphere and ocean problems (Lynch, 2008). NWP models have been developed around the world, such as the HadCM3 by the Hadley Center in the United Kingdom and the ECMWF by the European Center for Medium-Range Weather Forecasts organization. NWP can generate both short- and long-term weather predictions and provide valuable information about climate change. Recently, NWP was significantly improved to model regional climates such as tracking tropical cyclones. However, it is not good for modeling some important problems that happen in a smaller area, such as wildfires and local microclimate events. For these types of problems, other forms of numerical models should be used, such as MMM and CFD (Coen et al., 2009).

MMM is a simulation model for mesoscale problems. This model can capture mesoclimate features, such as precipitation, cloud formation, sea breezes, and wildfires (Dudhia, 1989; Pielke, 1992). MMM can use NWP data as a boundary condition (Blocken, 2015). Two well-known MMM tools are the WRF (originally developed by Byrd Polar Research Center at Ohio State University) and the GEM-SURF (developed by the Meteorological Research Division, Numerical Weather Prediction-Environment Section, Environment Canada and Climate Change) (Skamarock and Klemp, 2008; Leroyer et al., 2011). WRF is one of the most popular mesoscale meteorological models with more than 30,000 users around the world. With this solver, researchers can model both real and idealized atmospheric conditions. MMM cannot model microscale flows and capture the effect of topography and building configuration on flow properties (Berge et al., 2006; Choi et al., 2012). Thus, it is not suitable for modeling urban microclimate at the building level and studying its impact on human health and building energy performance.

CFD is generally referred to as the microscale numerical model (Blocken, 2015). CFD is a branch of fluid mechanics and can provide high-resolution results for various spatial and temporal scales from nano- to deca-scales (Holland et al., 2015; Blocken, 2015). In recent decades, CFD has been widely used to simulate urban wind and thermal comfort, pollutant dispersion, wind-driven rain, and building energy performance, all of which are directly related to human life (Monnen et al., 2012). CFD can be integrated with MMM and use the data from the mesoscale software as boundary conditions for modeling urban microclimate (Choi et al., 2012; Kwak et al., 2015). For example, Kwak et al. (2015) simulated urban air quality in a high-rise building area of Seoul, Korea by coupling CFD and WRF. In their work, WRF provided the data of horizontal wind velocity components and temperatures as boundary conditions for CFD. They determined that building geometry and mobile emission had significant effects on urban air quality (Kwak et al., 2015). According to the literature, CFD should be used for modeling urban microclimate (Moonen et al., 2012; Blocken, 2014; Toparlar et al., 2017). In the section of the literature review, CFD is introduced with more details.

1.1. Statement of the problem

As mentioned previously, people in cities are exposed to various and challenging conditions, such as heatwaves and heat island effects, wind and storms, pollutant dispersion, and fire accidents (Mirzaei and Haghighat, 2010; Bonsal et al., 2000). Some of these problems happen in a short time and around the direct vicinity of people, and could severely affect human life (United States Fire Administration, 2006). Therefore, providing fast and accurate microclimate information can play an important role in better understanding, controlling and/or reducing the negative effects of these problems. As the examples of these problems, this thesis discusses them in terms of two topics: human health and comfort, and building energy performance.

1.1.1. Human health and comfort

People spend considerable time in the urban environment, for example, participating in outdoor activities or using transportation. Additionally, urban microclimate directly affects the indoor environment (Katal, Mortezazadeh, and Wang, 2019). Many researchers have studied the effect of urban microclimate—such as air wind speed, temperature, and pollutant dispersion—on human health and comfort.

Air wind speed is under the influence of urban texture. High wind speeds near high-rise buildings have been reported being uncomfortable or sometimes dangerous for the pedestrians (Penwarden and Wise, 1975; Blocken and Carmeliet, 2004). For example, two deaths were reported in England from skull injury from being blown over by high wind speed near tall buildings (Penwarden and Wise, 1975). Nowadays, many urban authorities request wind comfort studies before issuing permits for new high-rise buildings. Several wind standards and criteria have been proposed by wind engineers for pedestrian wind comfort. For example, in Montreal, Canada, the critical mean wind speed for winter and summer are 4 m/s and 6 m/s, respectively (Article 39 of the Règlements refondus de la ville de Montréal). A numerical tool that quickly generates accurate wind information in an urban area could help engineers in the preliminary design stage. Additionally, if the simulation could be done in real-time or faster, it could provide valuable information for different organizations and governments for risks assessment and warning, related to weather forecast.

Thermal comfort is another parameter of urban microclimate that is affected by solar radiation, air temperature, humidity, shading, heat and cold wave events etc. Two important parameters significantly affect thermal comfort (Harlan et al., 2006; Pantavou et al., 2011):

1. Urban heat island being intensified by the rapid urbanization

2. Heat/cold waves increasingly affected by the climate change

Hot/cold temperature could lead to thermal discomfort and even serious human health problems, including heat exhaustion, heat stroke, heart and lung diseases (Pantavou et al., 2011). Every year, there exist many reports on human mortality and disease regarding urban thermal stress. For example, in the summer of 2003, a severe heat wave in Europe caused almost 50,000 deaths (Mirzaei and Haghighat, 2010). Extreme temperature events, such as heat and cold waves, seem also becoming common in Canada, and an important concern among the researchers (Bonsal et al., 2000; Smoyer-Tomic et al., 2003; Martin et al., 2012). Smoyer-Tomic et al. (2003) mentioned that extreme heat wave events frequently happened in Montreal. The residents of Montreal are highly vulnerable to extreme heat events due to its low air conditioning rates and many outdated and high-density buildings (Smoyer-Tomic et al., 2003). 53 deaths were reported in Montreal because of a heatwave in the summer of 2018. Using NWP or MMM models, I can predict heat waves in the short-term, but this cannot capture the vulnerability at the building level. Although modern CFD tools may provide this type of information, conventional CFD models are limited in terms of computational speed and resources when modeling the urban microclimate at the building level for prediction purposes.

Urban air pollution, environmental hazards, and emergency situations such as fire accidents are other typical problems related to citizen health and safety in cities. Air pollution has been one of the major environmental problems, especially in megacities (Mage et al., 1996) and cities with low air conditioning rate (Brunekreef and Holgate, 2002). Exposure to pollutants can cause serious problems in the human body, such as respiratory and cardiovascular diseases. One of the worst pollutant events of the last few decades was the 1952 London smog event, which caused almost over 4000 deaths (Brunekreef and Holgate, 2002; Bell et al., 2004). The main causes of air pollution and its spreading in cities include vehicle exhausts, accidental toxic agents, anthropogenic heat, including human activities, the heating/cooling system of residential buildings, industrial heat producers and users. The urbanization may also cause other environmental hazards, such as flooding and fires. Nirupama and Simonovi (2007) studied the effect of urbanization on the increasing risks of river flooding in urban areas. They mentioned the increased urbanization leads to the changes in the earth's surface and river flow (Nirupama and Simonovi, 2007). Fire accidents are also among common problems in cities with fatalities, injuries and economic losses. These events occur in a short span of time, and their response time are usually less than 10 minutes (United States Fire Administration, 2006). Urban texture and wind airflow directly affect fire accidents so the urban microclimate plays an important role in controlling or predicting them. For these problems, a fast and accurate urban microclimate model is important and may provide critical information for their better understanding in an urban context, and identifying solutions for saving lives.

1.1.2. Building energy performance

The local temperature and wind variation near buildings change the energy consumption of the buildings by altering the conductive and convective heat transfer coefficients through the buildings' envelopes (Kolokotroni et al., 2006; Ghiaus et al., 2006). In the United States, the urban heat island causes the urban peak electric demand to increase by 5–10% (Akbari; 2005). Many researchers and engineers have investigated different scenarios to reduce the energy consumption of buildings. For example, Akbari (2005) estimated that approximately $5 billion per year could be saved in energy costs in the United States by using cool surfaces and shading trees.

Another example of the effect of urban microclimate on building energy performance is wind flow change due to urban texture and building configurations. Natural ventilation and passive cooling system are the traditional ways to benefit from the microclimate for cooling and provide fresh air into buildings (Bahadori, 1985). Orme (2001) mentioned that 68% of the total cooling/heating load of buildings is used through HVAC systems. Tong et al. (2016) studied the wind potential for natural ventilation and demonstrated that cooling energy usage can be reduced by 8–78% by using natural ventilation.

Developing a fast and accurate urban microclimate model that can calculate building energy consumption in real-time or even predict the short-term building energy usages at the city scale could help to adjust the supply of the energy systems and control the load on the grids by changing operation strategies (Su et al., 2017). This type of model could also help calculate the potential of wind and solar energy production at the local environment and help decision-makers, the government, and householders access to the data to make wise decisions for saving energy. For example, Amado et al. (2018) used a numerical model to predict the potential of solar energy on building roofs for PV system installation in the city of Oeiras, Portugal.

1.2. Objectives of this thesis

The previous section revealed the need for developing a fast and accurate urban microclimate solver. The main objective of this thesis was to develop a fast and accurate airflow simulation method for modeling urban/city scale microclimates. The new solver needs to tackle the challenges of large domain simulation including stability issues, accurate results on the coarse grid, and large time step, all of which are typical for urban microclimate simulations, without a heavy reliance on computer resources, such as the possibility of running on personal computers. A suitable turbulence model also needs to be selected to capture the turbulence behavior at urban microclimate scales.

Coarse grids and large time steps are inevitable for developing a fast urban microclimate model. However, conventional commercial/non-commercial CFD tools are not well equipped to tackle these two challenges, which are often associated with the problems of accuracy, and sometimes stability and convergence issues.

In recent years, previous studies have focused on developing fast and stable numerical solvers for fluid problems, such as fast fluid dynamics (FFD) methods. These methods rely on fractional step methods and a semi-Lagrangian approach, and are mostly used for indoor airflow problems (Zuo et al., 2012; Jin and Chen, 2015; Xue et al., 2016). As a result, their current algorithms are limited for solving the challenges of "large time step, coarse grids" from urban microclimate simulations. Thus, significant work must be done to apply FFD to urban microclimate flows.

Another concern about developing a suitable urban microclimate solver is the turbulence model. In conventional CFD models, using accurate turbulence models for airflow in the ABL and microclimate, such as large eddy simulation (LES), are usually too time consuming. Therefore, many researchers had to reply on other turbulence models that are less accurate, which is not an acceptable solution to modeling many urban aerodynamics problems (Tominaga and Stathopoulos, 2011). With the computing time saved from a fast and accurate numerical model, it then allows allow us to apply more sophisticated turbulence models, such as LES, for solving urban microclimate problems.

A fast microclimate simulation can be achieved by developing new mathematical algorithms and/or replying on new programming techniques, such as multi-core CPU computing (i.e.

OpenMP), and/or GPU computing. Thus, significant amount of research is thus needed to benefit from parallelized flow simulation techniques to access the maximum power of modern computing hardware.

1.3. Summary and thesis work introduction

This chapter introduces the current research gaps in the study of urban microclimate simulation and the objectives of this thesis. The difficulties of urban microclimate numerical modeling with the current numerical tools, most importantly computational speed and accuracy, are discussed. This research focused on the development of a fast and accurate CFD solver, which is so-called CityFFD (as City Fast Fluid Dynamics) to simulate large-scale problems, such as urban environment, without the need of supercomputers. CityFFD is a novel solver based on FFD methods and four newly proposed mathematical algorithms running on parallel programming techniques including OpenMP and GPU.

Chapter 2 presents a comprehensive literature review about urban microclimate simulation techniques, different turbulence models, and FFD.

Chapter 3–7 describe five features of CityFFD that make the model a powerful tool for modeling city scale problems on personal computers. In Chapter 3, a novel high-order interpolation scheme named backward-forward sweep interpolating algorithm is proposed, which can provide highly accurate results even on coarse grids. The model is a 4th-order interpolation scheme with the combination of 3rd-order backward and 3rd-order forward polynomial interpolation methods.

Chapter 4 investigates the impact of time steps on the simulation accuracy of the conventional linear and higher-order FFD with constant time steps by defining the error coefficient functions associated with numerical truncation errors. This chapter proposes an adaptive time-stepping method, which first calculates local adaptive time step based on truncation error coefficient functions, and then obtains global time step based on an averaging function for all grid points and minimizing truncation errors.

Chapter 5 describes the development of a new density-based (DB) semi-Lagrangian method to speed up the conventional pressure-based (PB) semi-Lagrangian methods. The semi-Lagrangian-based solvers are typically semi-Lagrangian PB solvers, where a Poisson equation is solved to obtain the pressure field and ensure a divergence-free flow field. As an elliptic-type equation, the Poisson equation often relies on an iterative solution, so it can create a challenge of parallel computing and a bottleneck of computing speed. This study proposes a new DB semi-Lagrangian method, the semi-Lagrangian artificial compressibility, which replaces the Poisson equation with a hyperbolic continuity equation and an added artificial compressibility term so a time-marching solution is possible.

In Chapter 6, the temporal accuracy of CityFFD is improved from the first-order to the second-order by considering the acceleration of the flow field and accurately estimating the characteristic curve of each fluid particle. A couple of CFD benchmarks are then used to validate the proposed algorithm. The transient behavior of complicated flow patterns can be captured even with large time steps and coarse grids.

Chapter 7 is devoted to the CityFFD turbulence model and generating urban 3-D geometry models. Based on the literature, the most accurate turbulence model for capturing the turbulence behavior of the flow in the ABL level is LES. CityFFD is equipped with the LES-SGS model to simulate the turbulence behavior of the flow in the urban microclimate.

Chapter 8 is to integrate CityFFD with a building energy model (CityBEM) to simulate an extreme event in Montreal (the snowstorm of the century in 1971) and evaluate building resilience during the storm. CityBEM provides boundary conditions, including the building surface temperatures for CityFFD, and then, CityFFD provides local wind and temperature around each individual building for CityBEM to calculate the energy loads of the buildings, and the indoor temperatures. In this chapter, the effect of local microclimate on the indoor environment and human health-related issues is shown.

Chapter 9 concludes the thesis with the proposed future work.

Chapter 2 Literature Review

2.1. Urban microclimate simulation

A numerical model of urban microclimate generally uses CFD (Blocken, 2015). CFD is a branch of fluid mechanics and includes different type numerical approaches for solving the conservation of equations, such as mass, momentum, and energy. Currently, CFD is widely used to model microscale problems up to approximately 2 km in scale (Blocken, 2014; Toparlar et al., 2017). In recent years, many researchers have used CFD to model microclimate and environmental problems in different generic and real urban settings, such as a group of building blocks, street canyons, and different part of cities around the world (Takahashi et al., 2004; Tominaga et al., 2015; Taleghani et al.; 2014). For example, Tominaga et al. (2015) simulated the cooling effect of a pond in the urban area of Central Hadano (Japan), and concluded that the pond's evaporation decreases the maximum temperature at the pedestrian level by 2°C when the wind velocity is approximately 3 m/s at the height of 10 m. Takahashi et al. (2004) developed a CFD model to simulate air temperature, humidity, and wind velocity and compared the results with the measurement data. They demonstrated the proposed model was good for simulating green roofs and green grounds to study urban heat island mitigation scenarios (Takahashi et al., 2004). Taleghani et al. (2014) used CFD for thermal mitigation strategies in Portland State University (Oregon, USA) and demonstrated the effectiveness of using natural elements, such as vegetation and water bodies, to reduce air temperatures.

ANSYS FLUENT (2011) and OpenFOAM (2014) are two well-known software programs for modeling airflow problems in different fields. Many researchers have used these tools for urban microclimate simulations (Tominaga et al., 2015; Gracik et al., 2015). These tools are very powerful to model several types of problems in the field of fluid mechanics. However, the main deficiency of these tools is that they need to use small time steps for convergence issues, so they are not practical to model the urban environment. These models are also complicated and require the setup of many variables before a simulation. In other words, they are not designed particularly for urban microclimate simulation and need special expertise. Therefore, other software has been developed to model urban scale problems in a simpler way, such as ENVI_MET, UrbaWind, PALM, SOLENE-microclimate, scFLOW, and KARALIT. ENVI_MET is one of the most well-known microclimate simulation software programs. It is a parallel CPU-based model and can model airflow, pollution, radiation, vegetation, etc. (Ozkeresteci et al., 2003; Bruse, 2004). Another program is UrbaWind, a finite volume and steady-state CFD solver equipped with the Reynolds-averaged Navier-Stokes (RANS) turbulence model (Kalmikov et al., 2010). UrbaWind uses a multigrid algebraic solver to speed up the simulation process. It also benefits from a mesh refinement procedure to generate high-quality grids near the walls. All these models are CPU parallel–based solvers and they have not accessed to the maximum power of personal computers, such as GPU computing. Two other problems with these models are stability issues and CFL constraints (Huttner, 2012). Some companies provide fast wind simulation services for users, such as Ingrid Cloud. Cloud computing seems a solution but at the moment it is still costly for modeling large scale problems and not all users could afford to these services.

According to the above discussion, the development of a fast, robust, and accurate urban microclimate model is necessary. To achieve this goal, different aspects of CFD models, including the details of numerical solvers and turbulence models, must be considered. In the following sections, these two issues are discussed.

2.1.1. Numerical solvers

The governing equations of mass, momentum, and energy (i.e. the Navier-Stokes equations) are often partial differential equations (PDE) with advection and diffusion terms. Diffusion terms are usually solved based on implicit methods such as Jacobi, Gauss-Seidel, and multigrid models. Advection terms are the nonlinear part of conservation equations and may cause stability problems. CFD has two main categories to solve these terms: Eulerian and Lagrangian methods. In the Eulerian algorithms, the fluid is considered as a continuum fluid. Then a fixed computational domain, Eulerian mesh, is defined within the real domain. Finally, all fluid flow variables, including pressure, velocity, and density, are considered as a function of space and time within the fixed computational domain. Here, the conservation equations are solved on a control volume basis. A fast CFD solver often relies on explicit algorithms, wherein one important constraint for Eulerian methods, the CFL (Courant-Friedrichs-Lewy) condition, requires that the time step be small enough to ensure numerical stability, and therefore potentially slows down the simulation significantly. In the Lagrangian algorithms, the fluid in the flow field is considered to be a large number of individual particles. Then each particle's path line is followed based on the characteristic curves, allowing the conservation of equations to be solved. Although the CFL condition may not be relevant in this algorithm, the Lagrangian mesh is not fixed. Thus, a new computational mesh needs to be generated at each time step, resulting in a more complicated and even slower solution. More details about these two approaches can be found throughout the literature (Shirolkar et al., 1996; Loth, 2000).

In recent years, an increasing amount of efforts have been spent on finding a fast and accurate approach known as FFD, which is a combination of both Eulerian and Lagrangian methods. In FFD, the advection term is solved based on the semi-Lagrangian method, which was proposed by (Courant et al., 1952). In fact, this method considers the fluid as a large number of discrete particles, then follows their path, and finally transfers them to the Eulerian mesh realized by an interpolation scheme. The FFD method is fundamentally explicit without the CFL constraint so it is unconditionally stable, allowing for large time steps and fast simulation for large domains. Additionally, it does not need to regenerate the computational mesh at each time step as the Lagrangian methods do. Therefore, FFD seems to be one of the appropriate models for urban microclimate simulations. FFD is discussed further in Section 2.2.

CFD solvers can also be investigated based on error and accuracy. Numerical errors include round-off and truncation errors. A round-off error occurs due to the computers' limit to represent the real value of the numbers (Ueberhuber, 2012). A truncation error is related to the accuracy of the numerical methods and has a significant impact on the accuracy (Hoffmann and Chiang, 2000; Patankar, 1980). Dissipation and dispersion errors are two common truncation errors in numerical simulation (Hoffmann and Chiang, 2000). Even-order truncation error derivatives cause dissipation errors and are characterized by a loss of wave amplitude or damping. Dispersion errors are caused by odd-order truncation error derivatives and generate oscillation especially near the sharp gradients. These two types of numerical errors are discussed in Chapter 3. There are various PDE solvers with different orders of accuracy (Hoffmann and Chiang, 2000; Patankar, 1980). Low-order methods are susceptible to high dissipation errors and high-order methods can lead to dispersion errors. The development of a high-order and robust CFD solver is an open topic among researchers (Mortezazadeh and Wang, 2017; Fei and Xiaohong, 2006; Zonglin, 2004).

2.1.2. Turbulence models

One important characteristic of the airflow is turbulence behavior in the ABL and especially in the urban microclimate. Thus, in the CFD simulation of urban environments, the use of an accurate turbulence model plays an important role in providing accurate results. There are three turbulence models that have been frequently used in the microclimate problems: (1) RANS; (2) Unsteady Reynolds-averaged Navier-Stokes (URANS); and (3) Large Eddy Simulation (LES). RANS is a steady-state model and refers to the time-averaging of variables, such as velocity components. This model has been used in the vast majority of urban microclimate and outdoor airflow simulations due to its low computational cost (Blocken et al., 2016). Blocken et al. (2015) demonstrated that a 3-D RANS model can calculate the mean wind speed at the pedestrian level with acceptable accuracy in a comparison with wind tunnel data. Despite its advantages, this model is not able to capture transient behavior of ABL problems (Blocken, 2015). URANS is the unsteady RANS model and can capture the transient behavior of the flows. Toparlar et al. (2015) studied the thermal comfort in Bergpolder Zuid, Rotterdam by a 3-D unsteady CFD model using URANS. They validated the CFD results with the satellite image data and showed a good agreement between the numerical results and the observational data, thus demonstrating the potential of CFD for accurate simulation of urban microclimate. The main drawback of URANS is the need for fine grids near walls; thus, LES is recommended for use instead of URANS (Franke et al., 2007).

LES resolves large scales of turbulence, which are dominant characteristics of the flows in the ABL (Smagorinsky, 1963; Sullivan, 1994). LES has been used to model various urban microclimate problems, such as natural ventilation simulation and pollutant dispersion in street canyons (Jiang et al., 2003; Tominaga and Stathopoulos, 2011; Gousseau et al., 2011). Some researchers compared LES and RANS models and concluded LES can provide more accurate results for outdoor airflow problems (Tominaga and Stathopoulos, 2011; Gousseau et al., 2011). For example, Tominaga and Stathopoulos (2011) showed LES can accurately model diffusion concentration due to its ability to model unsteady fluctuations. Gousseau et al. (2011) simulated pollutant dispersion in downtown Montreal. Studying the effect of two different turbulence models, they compared the accuracy of the results with the wind tunnel measurements and showed that LES can perform well to model pollutant dispersion. In general, LES is more complex and computationally expensive than RANS. Gousseau et al. (2011) mentioned that, for the same problem, LES can be seven times slower than RANS. In fact, LES needs high-order numerical schemes or fine grids near walls for the low-order schemes to accurately model the viscous sublayer, which is a thin layer near walls (Pope, 2000). It is possible to reduce grid resolutions by using LES wall functions (Wang and Moin; 2002). Based on these models, the value of velocity or eddy viscosity near walls, inside the viscous sublayer, is analytically or empirically estimated. Thus, there is no need to use very fine grids to capture turbulence behavior of the flows inside the sublayer.

2.2. Fast Fluid Dynamics (FFD)

Fast Fluid Dynamics is a CFD technique for solving the incompressible Navier-Stokes equations in a fast manner (Stam, 1999). Using fractional step method, FFD decomposes Navier-Stokes equations to four sub-equations, named 1) diffusion and source term; 2) advection term; 3) Poisson equation (projection); and 4) velocity correction (correction). Figure 2-1 shows the schematic of FFD used in the present research.

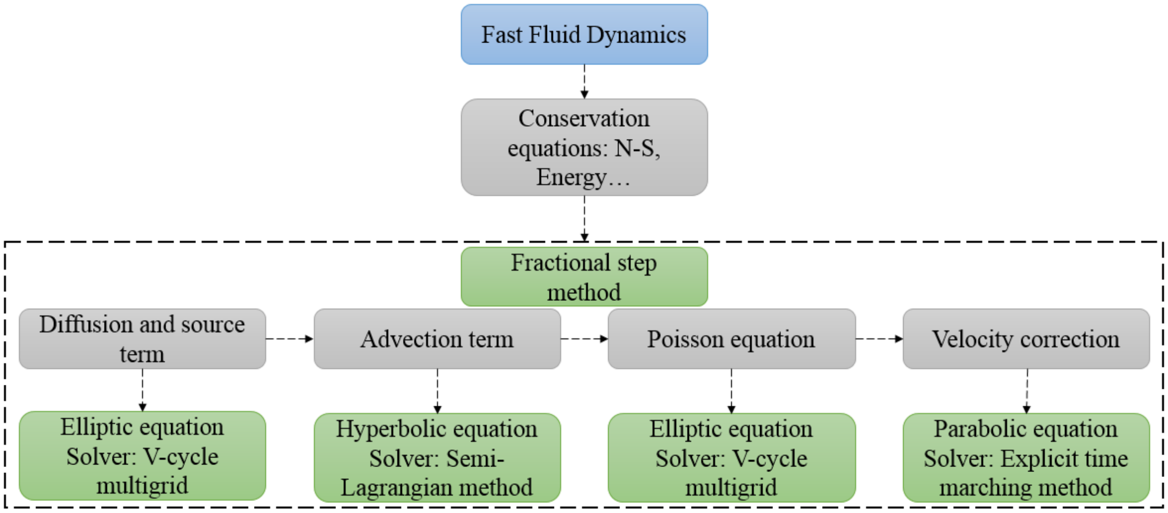

Fast Fluid Dynamics
Conservation equations: N-S, Energy…
Fractional step method
Diffusion and source term
Advection term
Poisson equation
Velocity correction
Elliptic equation Solver: V-cycle multigrid
Hyperbolic equation Solver: Semi-Lagrangian method
Elliptic equation Solver: V-cycle multigrid
Parabolic equation Solver: Explicit time marching method

Figure 2-1 Schematic of Fast Fluid Dynamics (FFD).

The reason for the distinction of FFD in comparison with the conventional methods is solving the advection term based on the semi-Lagrangian approach. Semi-Lagrangian was proposed by Courant et al. (1952) and has been widely used to solve advection terms of the Navier-Stokes equations for various problems. Firstly, it was developed for numerical weather forecasting, oceanic flows, and shallow water flows (Staniforth and Côté, 1991; Priestley, 1993; Garcia-Navarro and Priestley, 1994). Staniforth and Côté (1991) provided a comprehensive literature review on the semi-Lagrangian method and how it has been extended to different discretization approaches such as finite-difference and finite-element. Later the finite-difference form of semi-Lagrangian method has been widely used in other sciences such as building airflows (Zuo et al., 2012; Jin and Chen, 2015; Xue et al., 2016), multiphase simulations (Xiao and Ikebata, 2003; Gutiérrez and Bermejo, 2005; Lind and Phillips, 2012), electromagnetic applications (Dolean et al., 2010; Grote and Mitkova, 2012), and fluid flow simulations for game engines (Stam, 1999; Fedkiw et al., 2001). FFD uses finite-difference form of semi-Lagrangian method.

In the last decades, many researchers have used FFD in the field of building and environment (Zuo et al., 2012; Jin and Chen, 2015; Katal, Mortezazadeh, Wang; 2019). For example, Zuo and Chen

(2010(b)) simulated smoke distribution in a room and compared the results with CFD. They mentioned that FFD can be about 30 times faster than CFD. Zuo and Chen (2011) validated the accuracy of FFD for a problem with particle distribution inside a duct and showed the accuracy of FFD is in a good agreement with the experiment and CFD. Jin et al. (2013) simulated different types of natural ventilation around a single room with FFD and demonstrated the capability of FFD to capture main airflow features. Athanailidi et al. (2014) simulated airflow through two buildings and studied the wind funnel effect which is an unpleasant and sometimes dangerous phenomenon. Xue et al. (2016) developed a new semi-Lagrangian method based on PISO method, named SLPISO, and demonstrated this model is significantly faster than SIMPLE and PISO models and able to provide accurate results for the indoor.

Many researchers integrated FFD with other models and software, such as genetic algorithm, Rhino, OpenFOAM, multizone models, Modelica Buildings library, etc (Athanailidi et al., 2014; Tian et al., 2017; Katal, Mortezazadeh, Wang, 2019). For instance, Athanailidi et al. (2014) coupled Genetic Algorithm (GA) with FFD to find the best configuration for tensegrity structures between two buildings to solve the funnel effect. Karagkouni et al. (2014) evaluated the building performance by optimizing the wind-induced ventilation and using GA and FFD at the preliminary design phase. Tian et al. (2017) integrated multizone model for HVAC system with FFD by using Modelica Buildings library platform and simulated isothermal and thermal flow in a single room. Waibel et al. (2018) simulated airflow around the buildings by FFD which is implemented in Rhino and its visual programming platform Grasshopper.

The major advantage of using FFD method is its unconditional stability even for large time steps because of using semi-Lagrangian method for the advection terms, thus offering faster solution than other CFD solvers (Zuo and Chen, 2010(a); Mortezazadeh and Wang, 2017; Mortezazadeh and Wang, 2019). FFD has been also written based on parallel computing and GPU programming for further speedup (Zuo and Chen, 2010(a); Mortezazadeh and Wang, 2019, Katal, Mortezazadeh and Wang, 2019). Zuo and Chen (2010(a)) mentioned FFD on the GPU can be about 1000 times faster than CFD on the CPU. So, it makes this model a powerful solver for modeling large scale problems (Katal, Mortezazadeh and Wang, 2019).

2.2.1. Numerical procedure for FFD

In this section, the numerical procedure for solving Navier-Stokes equations based on FFD will be discussed. The proposed solver in the present work, named CityFFD, is based on the dimensionless equations. The governing equations, mass and momentum equations, in the dimensionless forms are:

$$\nabla \cdot U = 0 \tag{2-1}$$

$$\frac{\partial U}{\partial t} + (U \cdot \nabla)U = -\nabla p + \frac{1}{Re}\nabla^2 U + f \tag{2-2}$$

The conventional pressure-based semi-Lagrangian method comprises four main steps:

$$\frac{\partial U}{\partial t} = \frac{1}{Re}\nabla^2 U + f \tag{2-3}$$

$$\frac{\partial U}{\partial t} + (U \cdot \nabla)U = 0 \tag{2-4}$$

$$\nabla^2 p = \frac{1}{\Delta t}\nabla . U \tag{2-5}$$

$$\frac{\partial U}{\partial t} = -\nabla p \tag{2-6}$$

FFD numerical procedure is as follows:

1. Navier-Stokes equation, Eq. (2-2), is first reduced to diffusion and source terms, in the absence of pressure and advection terms, and is solved to calculate an intermediate velocity field for the diffusion and source terms only (Eq. 2-3).
2. Equation of the advection term (Eq. 2-4) is then solved by the semi-Lagrangian method to obtain the 2nd intermediate flow filed.
3. Poisson equation of pressure (i.e. pressure correction) is then solved for correcting the pressure field (Eq. 2-5).
4. Based on the pressure corrections, the velocity field is corrected to satisfy the condition of the divergence-free flows (Eq. 2-6).

In the above procedure, both the equations of the diffusion term (Eq. 2-3) and the pressure term (Eq. 2-5) are discretized based on the central 2nd-order of accuracy. These equations are elliptic, so an iterative solver is often needed. Here a three-level V-cycle multigrid method is used to speed up the calculation of the diffusion and Poisson equations (Appendix 1).

To explain the process of solving the Eulerian advection equation, Eq. (2-4), from the Lagrangian perspective by the semi-Lagrangian method, wIe start with the advection equation of a general scalar term, $\emptyset$, along its characteristic curve:

$$\frac{d\emptyset}{dS} = 0 \tag{2-7}$$

where $\emptyset = U$ for the linear momentum conservation equation and S can be written as follows:

$$dS = Udt \quad \rightarrow \quad S^n \approx S^{n+1} - U\Delta t \tag{2-8}$$

Here, $n + 1$ and n are the new and current time steps, respectively. Using a first-order temporal discretization, Eq. (2-7) becomes:

$$\emptyset_{S^{n+1}|X_a} \approx \emptyset_{S^n|X_d} \tag{2-9}$$

Because $\emptyset_{S^{n+1}|X_a}$ is always for the time $n+1$ and $\emptyset_{S^n|X_d}$ for n, for simplification and generalization, the time terms is dropped so Eq. (2-9) becomes:

$$\emptyset_a \approx \emptyset_d \tag{2-10}$$

Eq. (2-10) shows that ∅ is considered the same at both arrival and departure points within one time step. To calculate $\emptyset_d$, an interpolation scheme based on the neighbor Eulerian cells is needed (see Figure 2-2).

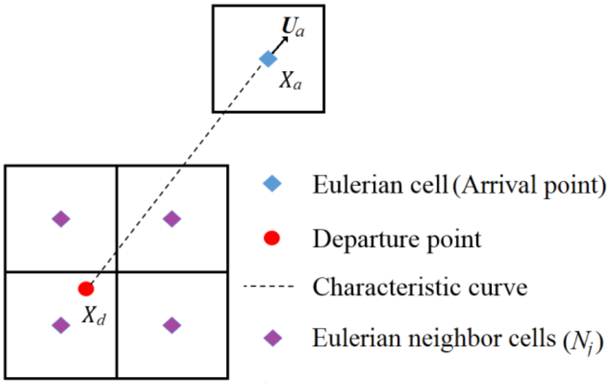

U_a
X_a
X_d
◆ Eulerian cell (Arrival point)
● Departure point
------ Characteristic curve
◆ Eulerian neighbor cells (N_j)

Fig. 1

Figure 2-2 Procedure of the semi-Lagrangian method with linear interpolation scheme.

Conventional FFD is based on a linear interpolation scheme which is low-order of accuracy. First, the departure point position is found. To find the position, we move backward to the location, X_d, at a distance of one time step times the current flow velocity, $U_a\Delta t$, from the arrival point location, X_a, along the characteristic curve:

$$X_d = X_a - U_a\Delta t \tag{2-11}$$

In the next step, the neighbor cells are found. In Figure 2-2, N_j (the Eulerian neighbor cells) shows the neighbor cells around the departure point. Using an interpolation scheme, the fluid property values at the departure point are calculated. Eq. (2-12) shows the linear interpolation scheme (2nd-order of accuracy) which is used to find the value of $\emptyset_d$ at the departure point:

$$\emptyset_a = \emptyset_d = \sum_{j=1}^{4} w_j \emptyset_{N_j} \tag{2-12}$$

where the w_j values are the weights of the interpolation scheme, e.g. for 2-D setup, bilinear interpolation can be applied.

Low-order interpolation schemes thus often cause inaccurate predictions of fluid properties, so, poor conservation of advected properties and/or high numerical dissipations (Zerroukat, 2010; Zuo et al., 2012). Another thing which affects the accuracy is the position of the departure point and its distance from its neighborhood Eulerian cells. My study in the next chapters shows that the magnitude of truncation error is smaller if the departure point is closer to the Eulerian grids.

On the other hand, conventional FFD is based on constant velocity assumption for constructing the characteristic curves. This assumption may provide accurate results by using small time step, but the huge numerical error, especially for the transient problems, can be seen for the large time steps. Using a large time step creates considerable deviation between the location of the estimated departure point and the accurate position of the point (see Figure 5-1).

The conventional semi-Lagrangian solver is pressure-based (PB) and by solving the Poisson equation, pressure domain is calculated and free divergence condition is satisfied. Pressure-based solver originally developed by Harlow and Welch (1965) for unsteady flow problems and then later extended by Patankar and Spalding (1972) for steady-state calculations. In this method, computations of the velocity and the pressure fields are decoupled (Blazek, 2005). Here, the pressure domain is calculated by solving the Poisson equation and then the velocity is obtained by the pressure domains. On irregular grids, solving the Poisson equation is not straightforward (Min and Gibou, 2006). In addition, solving the Poisson equation often needs iterative solvers, making it a real challenge for parallelizing FFD-based solvers on modern high-performance computing infrastructures, such as OpenMP and/or GPU, especially for large data cases (e.g. tens of millions of grids), which may only be solved in a parallel manner.

In this thesis, four novel ideas are proposed to overcome the drawbacks of conventional FFD models and a new FFD model, so-called CityFFD, is developed for modeling urban microclimates based on the coarse grids and large time steps. Chapter 3 is for solving the coarse grid problem by proposing a new high-order interpolation scheme which is capable to highly control the numerical errors. Chapter 4 is devoted to the time step size and its impact on the accuracy. Here it is shown different time steps can significantly affect the magnitude of truncation errors. In chapter 5, a new model is proposed to improve the accuracy of FFD by using a high-order temporal method. The proposed method can provide accurate results even by using large time steps. Chapter 6 investigates the computational cost of the different components of FFD. In this section, I show by removing the most time-consuming part of the conventional FFD method, the Poisson equation, it is possible to speed up the simulation at least three times. The proposed method is also better for parallelization techniques. In chapter 7, LES model is added to CityFFD for modeling the turbulence of microclimate problems. In the end, chapter 8 is dedicated for an application case. Here, a real city is simulated and the importance of urban microclimate on the indoor environment and building energy/thermal performance is demonstrated. It is achieved by integrating CityFFD with another newly developed model, CityBEM, which is a fast building energy model.

Chapter 3 A high-order backward forward sweep interpolating algorithm for semi-Lagrangian method

The contents of this chapter have been published in "Mortezazadeh, M. and Wang, L. L., 2017. A high-order backward forward sweep interpolating algorithm for semi-Lagrangian method. International Journal for Numerical Methods in Fluids, 84(10): 584-597". The contents are modified, and some parts of the original paper are removed to avoid repetition.

Conventional semi-Lagrangian methods often suffer from poor accuracy and imbalance problems of advected properties because of low-order interpolation schemes used and/or inability to reduce both dissipation and dispersion errors even with high-order schemes. In the current work, a 4th-order semi-Lagrangian method is proposed to solve the advection terms at a computing cost of 3rd-order interpolation scheme by applying backward and forward interpolations in an alternating sweep manner. The method was demonstrated for solving 1-D and 2-D advection problems, and 2-D and 3-D lid-driven cavity flows with a multi-level V-cycle multigrid solver. It shows that the proposed method can reduce both dissipation and dispersion errors in all regions, especially near sharp gradients, at the same accuracy as but less computing cost than the typical 4th-order interpolation because of fewer grids used. The proposed method is also shown able to achieve more accurate results on coarser grids than conventional linear and other high-order interpolation schemes in the literature.

3.1. Introduction

FFD relies on an interpolation scheme to calculate the unknown variables at the departure point (see Figure 2-1). Conventional FFD is based on the low-order interpolation scheme which often causes inaccurate predictions of fluid properties, which is consequently a major source for the well-known drawbacks of the method: e.g. poor conservation of advected properties and/or high numerical dissipations (Zerroukat, 2010; Zuo et al., 2012). During the past decades, many researchers have tried to solve the problems by developing better discretization schemes (Xiu and Karniadakis, 2001; Spiegelman and Katz, 2006), high-order interpolation schemes (Rosatti et al., 2005; Zun and Xiao, 2016), and hybrid methods combing interpolation schemes with different orders (Zuo et al., 2012). Zerroukat (2010) proposed a semi-Lagrangian method equipped with conservative remapping scheme. Sun and Xiao (2016) developed a high-order semi-Lagrangian scheme based on finite volume method and a non-oscillatory limiter to control the numerical oscillation, an open problem for high-order numerical methods. Xiu and Karniadakis (2001) developed higher-order temporal and spatial discretization schemes. Fedkiw et al. (2001) considered a monotonic cubic interpolation instead of polynomial interpolations to improve the accuracy. While these methods tried to improve the accuracy in different ways, they also add complexity and extra computational cost to the semi-Lagrangian method (Zuo et al., 2012). Therefore, Zuo et al. (2012) developed a relatively simple hybrid method by combining linear and 3rd-order backward interpolation schemes: using linear interpolation near the sharp gradient to reduce dispersion errors and using the 3rd-order interpolation in other regions to reduce numerical dissipations. However, combining lower- and higher-order interpolations cannot reduce dissipation or dispersion errors at the same time and the overall accuracy of a hybrid method is often less than expected.

In this study, an alternate sweep algorithm of a 3rd-order backward and a 3rd-order forward interpolation is proposed to improve the accuracy of the semi-Lagrangian method by reducing both the dissipation and dispersion errors for all flow regions, especially near regions with sharp gradients. The backward or forward interpolation itself is with 3rd-order but the overall accuracy of the proposed method after sweeping can achieve the 4th-order accuracy whereas using much fewer grids for interpolation than conventional 4th-order method. The proposed method and its order of accuracy are studied using Taylor series for 1-D semi-Lagrangian problem. Then the efficiency and accuracy are investigated by applying the method to solving 1-D and 2-D pure advection problems, and 2-D and 3-D lid-driven cavity problems governed by the full Navier-Stokes equations. A V-cycle multigrid method is used to speed up the solutions of the diffusion terms and the Poisson equation.

3.2. Methodology

In this section, first the general FFD method for incompressible flows, the semi-Lagrangian method, is briefly introduced and then the proposed backward and forward sweep interpolating scheme is explained.

3.2.1. Fast fluid dynamics method for incompressible flows

In this study, the FFD procedure for solving Eqs. (2-3) and (2-6) is as follows:

1. Eq. (2-3), diffusion and source terms, is solved by using three-level V-cycle multigrid method to obtain an intermediate velocity field.
2. Eq. (2-4), advection term, is solved by using the semi-Lagrangian method with the proposed 3rd-order backward forward sweep interpolation to get the 2nd intermediate velocity field.
3. Based on the 2nd intermediate velocity, Eq. (2-6), Poisson equation, is solved for the pressure field by the same multigrid method.
4. Finally, the velocity field is corrected by applying the new pressure field to Eq. (3-5). Then the calculation returns to step 1 and iterates till convergence.

3.2.2. Semi-Lagrangian method

As mentioned in section 2-2, an interpolation is needed to calculate $\emptyset_d$ in Eq. (2-10). First, I need to find the neighbor Eulerian cells about X_d (see Figure 3-1). The Eulerian cells inside the blue and red squares shown in Figure 3-1 belong to the 3rd-order forward and backward interpolation methods, respectively. In comparison, the green square region shows the neighbor cells about X_d for a typical 4th-order central interpolation. Here, $n_{i-1,j-1}$ to $n_{i+2,j+2}$ show the indices of Eulerian neighbor cells.

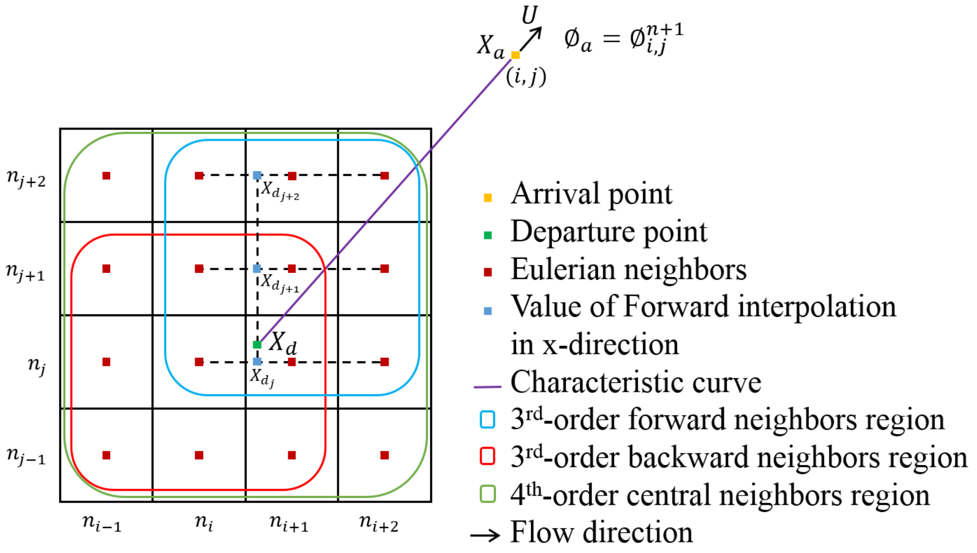
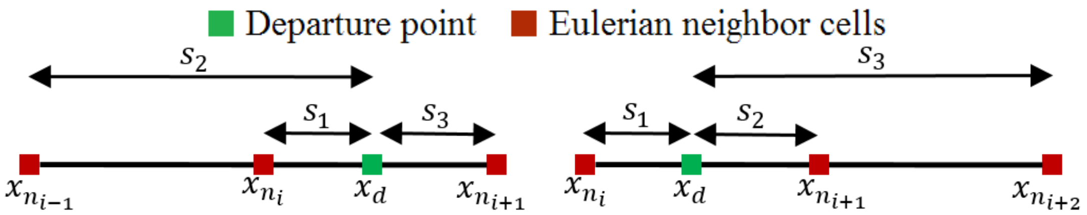


Figure 3-1 Schematic of the 2-D semi-Lagrangian method for 3rd-order backward and forward interpolations, and 4th-order central interpolation method.

3.2.3. The third-order backward forward sweep interpolation method

Here the proposed method for a 1-D semi-Lagrangian method on a uniform grid, i.e. $\Delta x =$ constant, is explained (see Figure 3-2).



(a) 3rd-order backward interpolation (b) 3rd-order forward interpolation

Figure 3-2 3rd-order interpolation schemes for 1-D advection problem.

For the 3rd-order backward interpolation scheme in Figure 3-2(a), Taylor series for the neighbor cells is applied about X_d as follows:

$$\begin{aligned}
\emptyset^n_{n_{i+1}} &= \emptyset_d + s_3 \frac{\partial \emptyset_d}{\partial x} + \frac{{s_3}^2}{2}\frac{\partial^2 \emptyset_d}{\partial x^2} + \frac{{s_3}^3}{6}\frac{\partial^3 \emptyset_d}{\partial x^3} + O(\Delta x^4) \\
\emptyset^n_{n_i} &= \emptyset_d - s_1 \frac{\partial \emptyset_d}{\partial x} + \frac{{s_1}^2}{2}\frac{\partial^2 \emptyset_d}{\partial x^2} - \frac{{s_1}^3}{6}\frac{\partial^3 \emptyset_d}{\partial x^3} + O(\Delta x^4) \\
\emptyset^n_{n_{i-1}} &= \emptyset_d - s_2 \frac{\partial \emptyset_d}{\partial x} + \frac{{s_2}^2}{2}\frac{\partial^2 \emptyset_d}{\partial x^2} - \frac{{s_2}^3}{6}\frac{\partial^3 \emptyset_d}{\partial x^3} + O(\Delta x^4)
\end{aligned} \tag{3-1}$$

where:

$$s_2 = \Delta x + s_1, s_3 = \Delta x - s_1 \tag{3-2}$$

The 3rd-order backward interpolation scheme for $\emptyset_d$ can then be obtained by eliminating the first and second derivatives in Eq. (3-1):

$$\emptyset_d = -\frac{s_1 s_3}{2\Delta x^2}\emptyset^n_{n_{i-1}} + \frac{s_2 s_3}{\Delta x^2}\emptyset^n_{n_i} + \frac{s_1 s_2}{2\Delta x^2}\emptyset^n_{n_{i+1}} + E_b \tag{3-3}$$

where the truncation error is

$$E_b = -\frac{s_1 s_2 s_3}{6}\frac{\partial^3 \emptyset_d}{\partial x^3} + O(\Delta x^4) \tag{3-4}$$

Following a similar procedure, the 3rd-order forward interpolation for $\emptyset_d$ can be obtained (Figure 3-2(b)):

$$\emptyset_d = \frac{s_2 s_3}{2\Delta x^2}\emptyset^n_{n_i} + \frac{s_1 s_3}{\Delta x^2}\emptyset^n_{n_{i+1}} - \frac{s_1 s_2}{2\Delta x^2}\emptyset^n_{n_{i+2}} + E_f \tag{3-5}$$

where:

$$E_f = \frac{s_1 s_2 s_3}{6}\frac{\partial^3 \emptyset_d}{\partial x^3} + O(\Delta x^4) \tag{3-6}$$

It is then possible to cancel the leading errors of Eqs. (3-4) and (3-6) because

$$E_b \approx -E_f \tag{3-7}$$

Therefore, by alternating the backward at one time-step followed by the forward interpolation at the next (i.e. sweeping), the leading errors of both interpolations can be canceled so the 4th-order of accuracy can be achieved at the end of $2\Delta t$ by using fewer grids than typical 4th-order methods as shown in Figure 3-1. Now the truncation error E after two time-step, i.e. $2\Delta t$, is:

$$E = E_b + E_f = O(\Delta x^4) \tag{3-8}$$

The alternative sweeping is necessary because if either of the two methods are used separately, the error for backward and forward methods is equal to:

$$E_{b,2\Delta t} \approx 2E_b = O(\Delta x^3) \tag{3-9}$$

$$E_{f,2\Delta t} \approx 2E_f = O(\Delta x^3)$$

So:

$$|E| < \left|E_{b,2\Delta t}\right| \text{ or } |E_{f,2\Delta t}| \tag{3-10}$$

Note that although Figure 3-2 uses a uniform mesh as an example, the previous analyses of truncation errors also apply to non-uniform/stretched meshes because the derivations actually do not need the information of the uniform mesh, i.e. Eq. (3-2), which is provided here only for easier representation of the orders of the truncation errors. To demonstrate how the proposed method can be applied to non-uniform meshes, this chapter investigated its performance for both a 1-D and a 3-D stretched mesh cases in later sections.

For the 2-D problem in Figure 3-1, the implementation of the sweeping method is as follows (the 3-D implementation can be done in the similar way):

1. Taking the 3rd-order forward interpolation as an example in the first time step, Eq. (3-5) is first applied to x-direction to obtain $\emptyset_{d_j}, \emptyset_{d_{j+1}}\ and\ \emptyset_{d_{j+2}}$ separately, e.g. for $\emptyset_{d_j}$

$$\emptyset_{d_j} \approx A\emptyset^n_{n_{i,j}} + B\emptyset^n_{n_{i+1,j}} + C\emptyset^n_{n_{i+2,j}} \tag{3-11}$$

where A, B, and C are the interpolation weights in Eq. (3-5) along the x-direction.

2. Then applying Eq. (3-11) in the y-direction to find $\emptyset_d$:

$$\emptyset_d \approx A^{'}\ \emptyset_{d_j} + B^{'}\ \emptyset_{d_{j+1}} + C^{'}\ \emptyset_{d_{j+2}} \tag{3-12}$$

where $A^{'}$, $B^{'}$, and $C^{'}$ are the interpolation weights for the y-direction.

3. In the next time step, a similar procedure is applied to the 3rd-order backward interpolation scheme to obtain $\emptyset_d$ at $2\Delta t$, which completes one “sweep”. The process is then repeated till convergence.

3.3. Case studies

In this section, the performance of the proposed method in terms of accuracy and computing time is compared to different interpolation schemes: linear, the hybrid of Zuo et al. (2012), and a 4th-order central interpolation for solving a 1-D and 2-D advection problem, and a 2-D and 3-D N-S equations. The following standard definitions of errors are used: l_1 norm, l_2 (or the second norm), and l_∞ (or the infinite norm) for a 2-D problem as an example (Lauritzen et al., 2010):

$$E_{l_1} = \frac{\sum_{i,j=1}^{imax,jmax} A_{i,j} \times |\emptyset_{i,j}^{Numeric} - \emptyset_{i,j}^{Analytic}|}{\sum_{i,j=1}^{imax,jmax} A_{i,j} \times \emptyset_{i,j}^{Analytic}} \tag{3-13}$$

$$E_{l_2} = \sqrt{\frac{\sum_{i,j=1}^{imax,jmax} \left[A_{i,j} \times \left(\emptyset_{i,j}^{Numeric} - \emptyset_{i,j}^{Analytic}\right)\right]^2}{\sum_{i,j=1}^{imax,jmax} \left(A_{i,j} \times \emptyset_{i,j}^{Analytic}\right)^2}} \tag{3-14}$$

$$E_{l_\infty} = \frac{\max|\emptyset_{i,j}^{Numeric} - \emptyset_{i,j}^{Analytic}|}{\max|\emptyset_{i,j}^{Analytic}|} \tag{3-15}$$

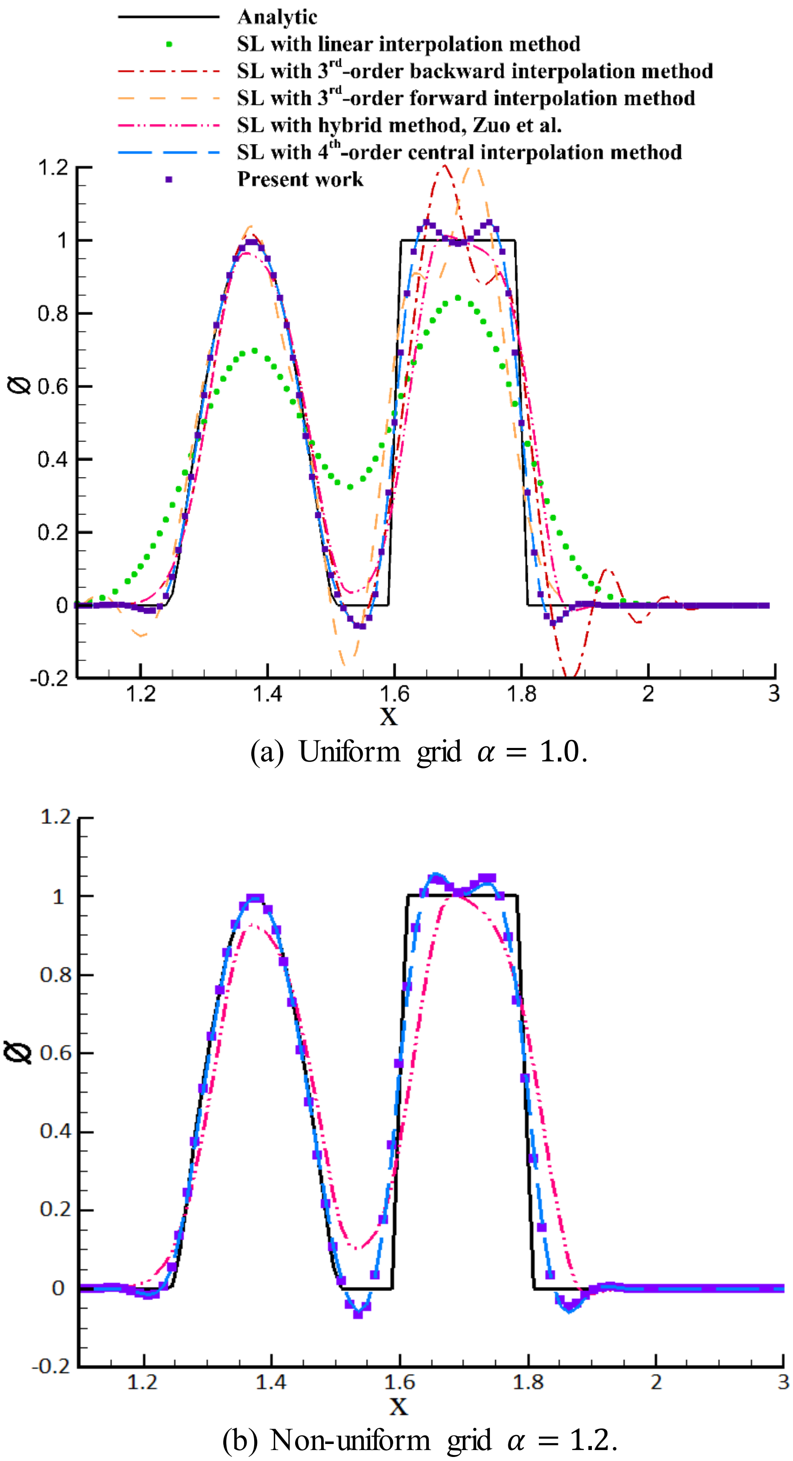
3.3.1. 1-D and 2-D advection problems

To demonstrate the capability of the proposed method for reducing both dissipation and dispersion errors, especially near regions with sharp gradients, first a 1-D advection problem of a general scalar term, $\emptyset_i$, is considered with two common shapes of waves, i.e. a sinusoidal and a step wave, as defined by the initial conditions and grids below:

$$\emptyset_i = \begin{cases} \sin(4\pi(x_i - 0.25)) & 0.25 \le x_i \le 0.5 \\ 1.0 & 0.6 \le x_i \le 0.8 \\ 0.0 & \text{otherwise} \end{cases} \tag{3-16a}$$

$$x_i = [\Delta x \times (i-1)]^{\alpha} \qquad \begin{cases} \alpha = 1 & \text{uniform grid} \\ \alpha = 1.2 & \text{non-uniform grid} \end{cases} \tag{3-16b}$$

where x_i is the ith position of the advected scalar, the velocity, $u = 1$ [m/s], the time step, $\Delta t = 0.005$ $[s]$, and the grid resolution is defined by Eq. (3-16b): $\alpha = 1.0$ for the uniform grid and $\alpha = 1.2$ for the non-uniform grid case, for both of which, $\Delta x = 0.01$ $[m]$.



(a) Uniform grid $\alpha = 1.0$.

(b) Non-uniform grid $\alpha = 1.2$.

Figure 3-3 Semi-Lagrangian results for the 1-D advection problem with two waves at t = 1 [s] for (a) the uniform grid and (b) the non-uniform grid case.

Figure 3-3 shows the results for different interpolation schemes at $t = 1\ [s]$ for both the uniform and non-uniform cases. As shown by Figure 3-3(a) for the uniform grid, the dominant error for the linear interpolation scheme is the numerical dissipation, while both 3rd-order interpolation methods show oscillations, especially near the sharp gradients, caused by dispersion errors as also observed by Zuo et al. (2012). Note that the oscillation behavior for the 3rd-order backward and forward methods seems opposite, e.g. especially for the step wave. Figure 3-3 also compares the present work to the hybrid method (Zuo et al., 2012), and a 4th-order central interpolation method. Table 3-1 compares quantitatively the errors for different schemes. For the 1-D problem, compared to the hybrid method, the proposed method can reduce all types of error criteria by about 50%. Additionally, it shows that the proposed method is as accurate as the 4th-order central interpolation scheme. Figure 3-3(b) shows the performance of the proposed method for the non-uniform grid when $\alpha = 1.2$. As expected, the method also applies well to the stretched mesh case with a similar performance as the uniform grid case when compared to the hybrid method (Zuo et al., 2012) and the 4th-order central method.

To further verify the conclusion, the study is extended to a 2-D advection problem with initial conditions defined by Eq. (3-17) and shown in Figure 3-4:

$$\emptyset = \begin{cases} 1.0 & 0.1 \le x \le 0.5, 0.23 \le y \le 0.37 \\ 1.0 & 0.23 \le x \le 0.37, 0.1 \le y \le 0.5 \\ 0.0 & \text{otherwise} \end{cases} \tag{3-17}$$

where x and y represent the position of the cell centers on 2-D coordinates. Here $\Delta t = 0.005$ [s] and $\Delta x = \Delta y = 0.01\ [m]$ with a constant velocity field defined by Eq. (3-18). The total grid number is 400×400.

$$u = 1.0\ [m/s] \text{ and } v = 1.0\ [m/s] \tag{3-18}$$

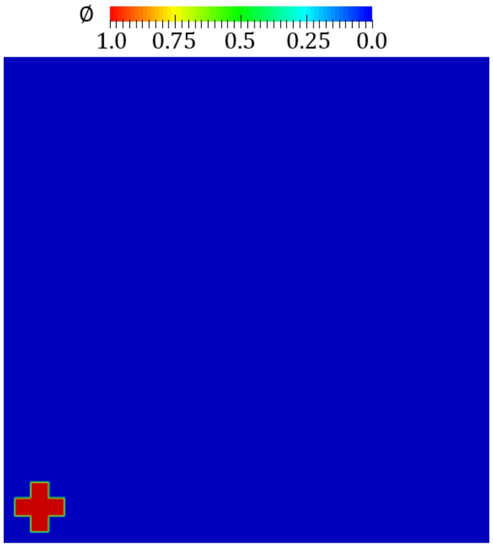
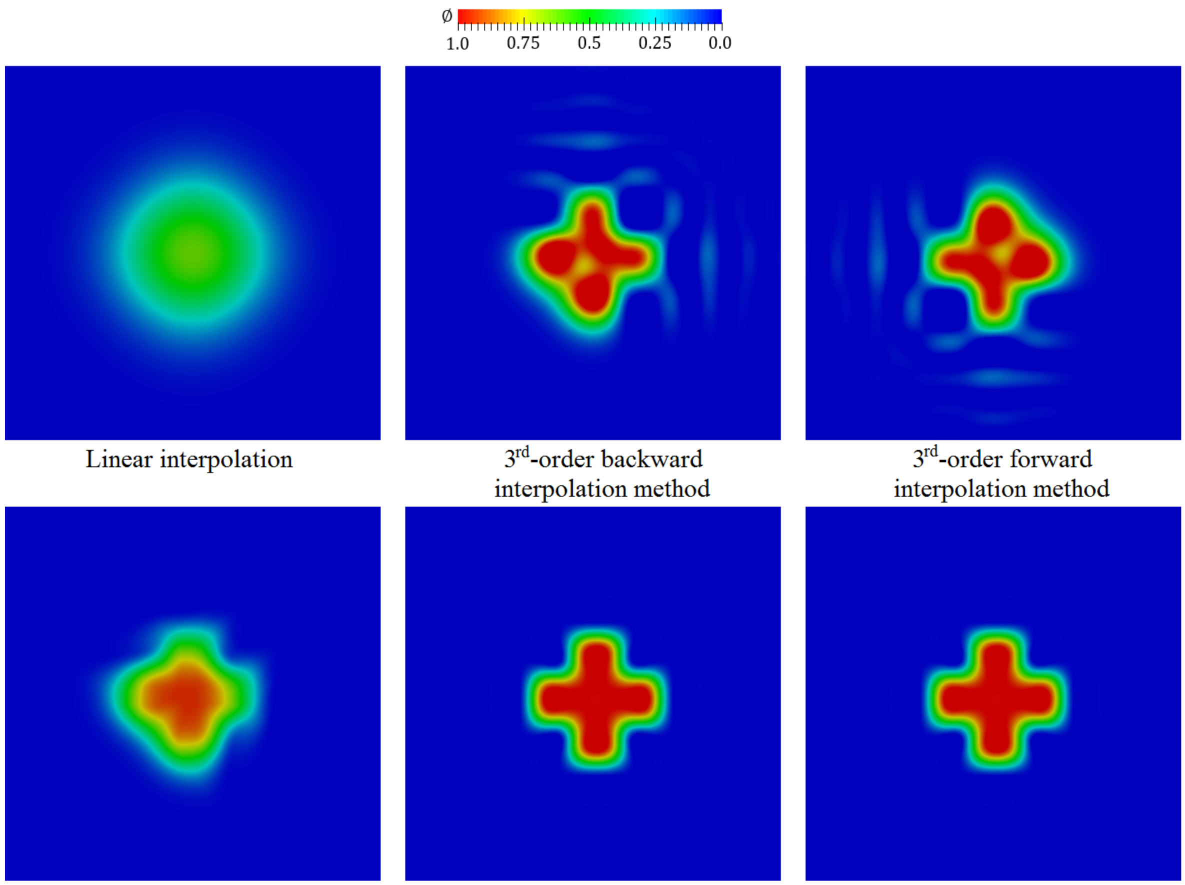


Figure 3-4 Initial condition for the 2-D advection problem.

Figure 3-5 and Table 3-1 compare the results at $t = 3\ [s]$. As expected, it is obvious that the accuracy of the proposed method is significantly better than the 3rd-order and hybrid methods. Figure 3-5 shows that both the backward and forward interpolation have strong dispersion errors when used separately. The hybrid method reduces the dispersion errors because of the use of linear interpolation, which however increases the dissipation error near the sharp gradient regions. The present work is shown to be able to reduce both dispersion and dissipation errors at a similar order of accuracy as the 4th-order interpolation method with fewer computing time: 167 s for the present work compared to 172 s for the 4th-order method as illustrated in Table 3-1. The difference is not noteworthy for this problem but can be significant for other cases as shown later, e.g. 3-D cases. Therefore, the present work is able to achieve 4th-order accuracy at the computing cost of the 3rd-order method.



Linear interpolation

3rd-order backward interpolation method

3rd-order forward interpolation method

Hybrid method, (Zuo et al. 2012)

4th-order central interpolation method

Present work

Figure 3-5 Contours of scalar values for the 2-D advection problem solved by semi-Lagrangian equipped with different interpolation methods.

Table 3-1 Comparison of different interpolation schemes for 1-D and 2-D advection problem at t=1 [s] for 1-D and t=3 [s] for 2-D.

Scheme		E_{l_1}	E_{l_2}	E_{l_∞}	Time $[s]$
Linear	1-D	0.4758	0.3884	0.4809	-
	2-D	1.0925	0.6434	0.7305	161
3rd-order Backward	1-D	0.2194	0.2118	0.5281	-
	2-D	0.6675	0.4802	0.9305	167
3rd-order Forward	1-D	0.2079	0.2118	0.5291	-
	2-D	0.6213	0.4276	0.9302	166
Hybrid (Zuo et al., 2012)	1-D	0.1913	0.2393	0.5874	-
	2-D	0.6136	0.5035	0.8762	163
4th-order Central	1-D	0.0842	0.1267	0.3112	-

	2-D	0.3297	0.3162	0.6687	172
Present Work	1-D	0.0828	0.1256	0.3093	-
	2-D	0.3277	0.3151	0.6692	167

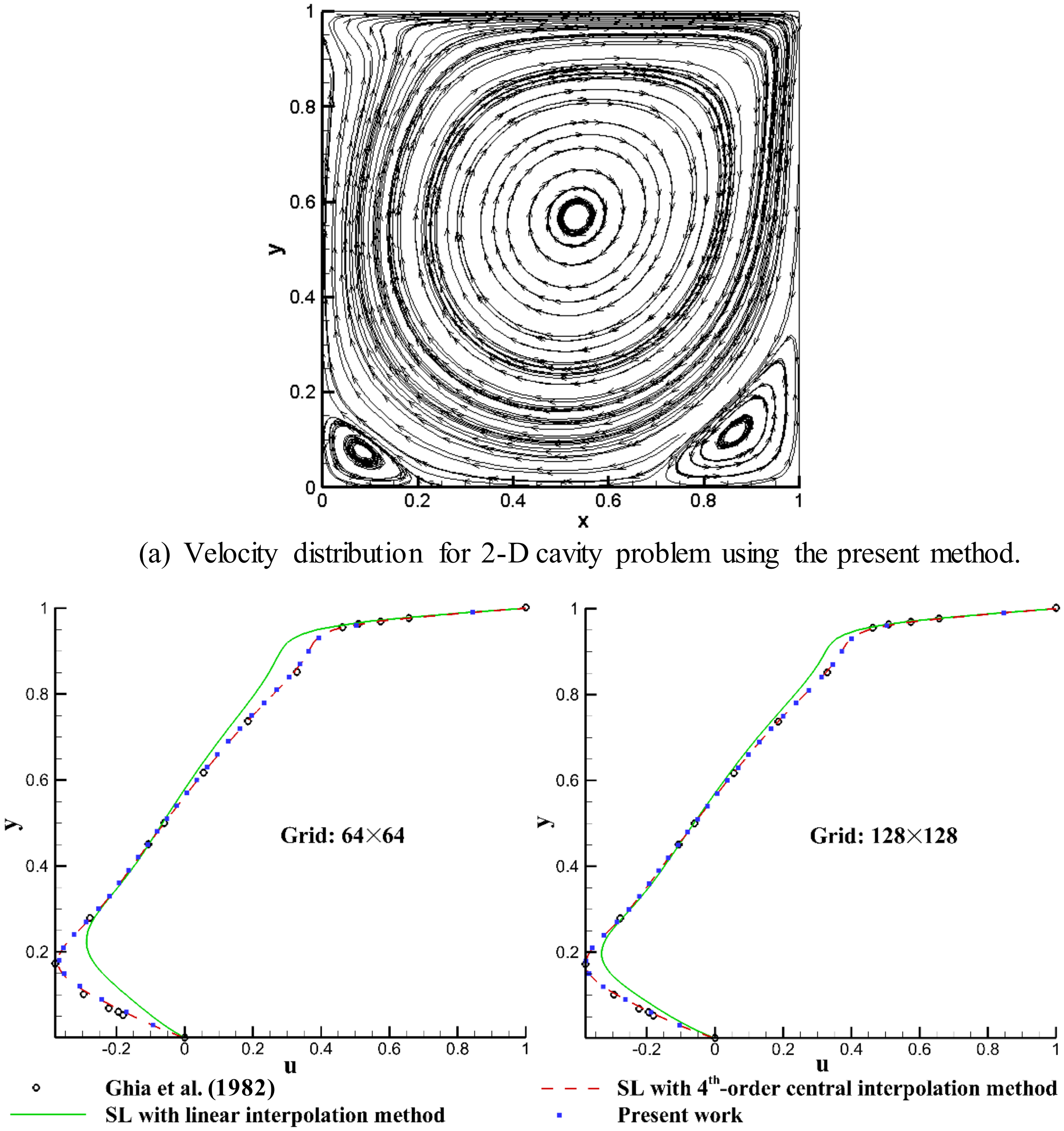
Note that for a problem with uniform grids, e.g. the current problem, E_{l_1} also indicates the error of mass conservation as defined by Eq. (3-19):

$$m_{i,j} = A_{i,j}\emptyset_{i,j} \quad \rightarrow \quad E_{cons} = E_{l_1} = \frac{\sum_{i,j=1}^{imax,jmax} |m_{i,j}^{Numeric} - m_{i,j}^{Analytic}|}{\sum_{i,j=1}^{imax,jmax} m_{i,j}^{Analytic}} \qquad (3\text{-}19)$$

Using the proposed method, 70% of the mass imbalance error could be reduced in comparison with the linear semi-Lagrangian method, and about 50% in comparison with the hybrid method (Zuo et al., 2012). So, by reducing the truncation errors, the proposed method is able to reduce the mass imbalance problem which is a well-known drawback of semi-Lagrangian methods.

3.3.2. 2-D Navier–Stokes equations

Here, the FFD procedure of solving the full Navier-Stokes equations is implemented with the proposed backward and forward sweep semi-Lagrangian method for the advection terms. A typical benchmark problem for incompressible flows is used: the 2-D lid-driven cavity flow problem. The convergence criterion is 10^{-6}, which is based on the maximum value of the difference between the current and previous time step values. In this problem, the fluid moves inside a 2-D square. The boundary of the domains are walls with the non-slip condition. The upper wall has a velocity in x-direction, $u = 1.0\ [m/s]$. The domain is $1 \times 1\ [m^2]$ with structured and uniform grids.



(a) Velocity distribution for 2-D cavity problem using the present method.

(b) Comparison of velocity component in the x-direction for different grids.

Figure 3-6 2-D lid-driven cavity problem solved by semi-Lagrangian method, Re=1000, $\Delta t = 0.005$ [s].

Figure 3-6(a) shows the streamline for the cavity flow with $Re = 1000$. Here it is clear that the proposed method is able to capture small vortexes even in the corner of the computational domain. Figure 3-6(b) indicates u velocity values in the y-direction and it shows that the current method can achieve the same accuracy as the 4^{th}-order central interpolation method.

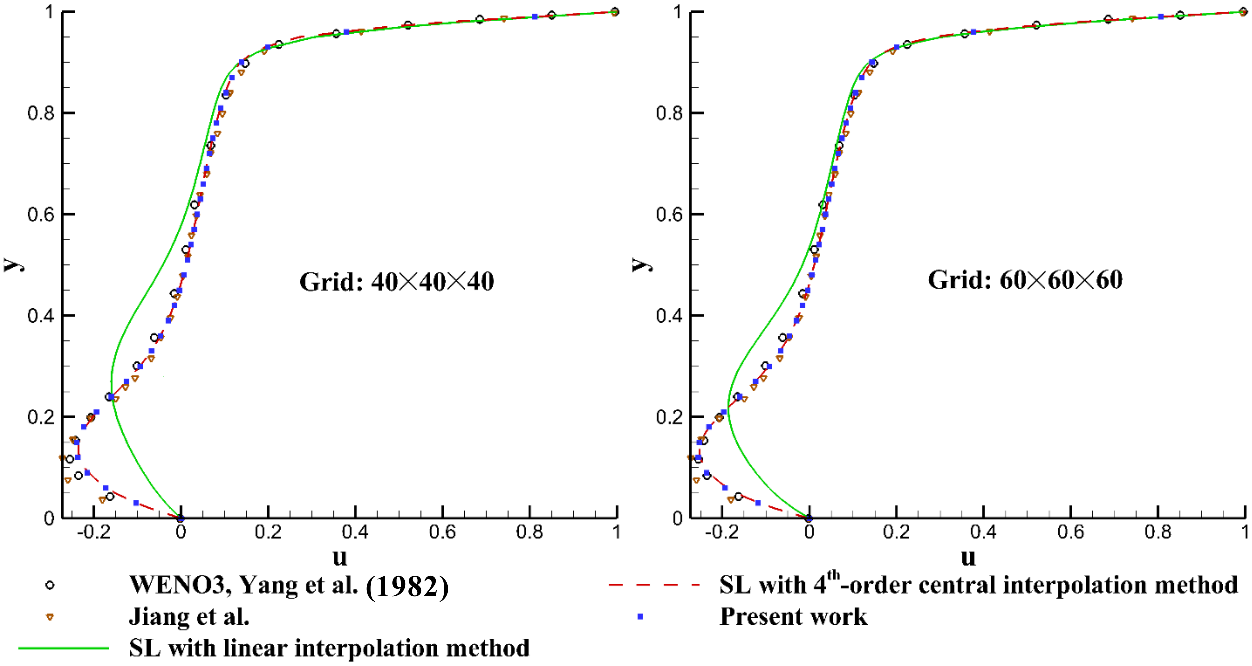
Note that the proposed method is faster than the typical 4^{th}-order method because it needs fewer grid points and thus fewer floating point operations: the number of arithmetic operations (Saad, 2003) for the 1-D problem is around $26 \times i_{max}$ for the present method while $51 \times i_{max}$ for the 4^{th}-order method (note i_{max} is the total number of grids). The advantage of the proposed method in comparison with 4^{th}-order central interpolation method is that it reduces the computing time approximately 15% as shown by Table 3-2 because a fewer number of nodes are engaged in the interpolation. Meanwhile, the comparison between the proposed method and the linear interpolation shows the computing time did not increase much while the accuracy is significantly improved. Additionally, Figure 3-6(b) shows the present work can give an acceptable accuracy even on the coarse grid, i.e. 64×64, whereas the linear semi-Lagrangian method is not accurate enough even on the finer grids, i.e. 128×128. Figure 3-6(b) also shows that for the cavity problem, using 128×128 grids are enough to approximately get the results similar to Ghia et al.'s work, which is based on a vorticity-stream function formulation equipped with an implicit multigrid method (Ghia et al., 1982).

Table 3-2 Computational time in seconds for the lid-driven cavity problems.

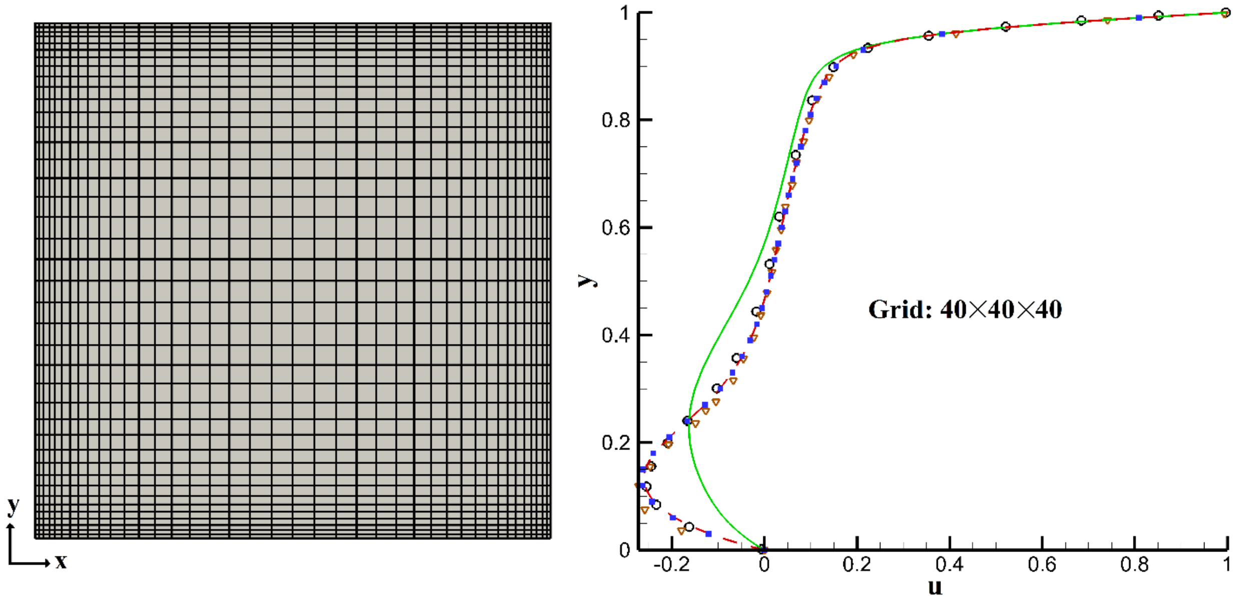
Scheme	2-D		3-D	
	Grids	Time	Grids	Time $[s]$
Linear	64×64	21	40×40×40	307
	128×128	67	60×60×60	1,197
4^{th}-order central	64×64	26	40×40×40	637
	128×128	82	60×60×60	2,391
Present work	64×64	22	40×40×40	403
	128×128	73	60×60×60	1,575

3.3.3. 3-D Navier–Stokes equations

For high-order interpolation schemes (3rd or 4th-order central schemes), more grid points will be involved and the computing cost may rise accordingly for all methods. Therefore, it is worthwhile to check the performance of the proposed method for 3-D Navier-Stokes problems. Here, the work is extended to solve the 3-D lid-driven cavity problem with $Re = 1000$ and compare the results with the WENO method, a weighted non-oscillatory (ENO) scheme based on a class of lower–upper approximate-factorization implicit weighted (Yang et al., 1998), and Jiang et al. (1994) results, which are based on a least-square finite element method (LSFEM). Figure 3-7 shows that the proposed method is significantly more accurate in comparison with the conventional linear semi-Lagrangian method. Therefore, I can use a much fewer number of computational cells than linear interpolation method but notably achieve higher accuracy. For example, the accuracy for $40 \times 40 \times 40$ grids using the present work is much better than $60 \times 60 \times 60$ grid by linear interpolation scheme as shown in Figure 3-7(a). The running time for the present work for $40 \times 40 \times 40$ grids is 403 [s], while for the linear semi-Lagrangian method and $60 \times 60 \times 60$ grids, the computational time is almost tripled: 1,197 [s].



(a) Uniform grids.



(b) Non-uniform grid of total $40 \times 40 \times 40$ with $\alpha = 0.5$ and $\beta = 1.1$.

Figure 3-7 3-D lid-driven cavity flow problem solved by semi-Lagrangian method for (a) two uniform grids and (b) one non-uniform grid case.

In the real applications, stretching grids are often used especially near boundaries such as walls. To investigate the accuracy of the proposed method for non-uniform grids, this study used a case with non-uniform grids defined by Eq. (3-20) (Hoffmann and Chiang, 2000(a)) and shown by Figure 3-7(b).

$$x_i = \frac{(2\alpha + \beta)(\frac{\beta + 1}{\beta - 1})^{(\frac{\gamma_i - \alpha}{1-\alpha})} + 2\alpha - \beta}{(2\alpha + 1)[1 + (\frac{\beta + 1}{\beta - 1})^{(\frac{\gamma_i - \alpha}{1-\alpha})}]} \qquad \text{(3-20)}$$

where $\gamma_i = (i-1)\Delta x$ and $\Delta x = 0.025\ [m]$.

Eq. (3-20) is the stretching function for the x-direction only. For the y- and z-directions, a similar stretching function is used for the 3-D case here. As shown in Figure 3-7(b), the present work performs well for the non-uniform grid case (total grid = 40×40×40) in comparison with the linear semi-Lagrangian and the 4th-order interpolation methods in a similar trend as observed in the uniform grid cases in Figure 3-7(a). This confirms the performance of the present work for both uniform and stretching grids as expected.

3.4. Summary and Conclusions

Because of its unconditional stability even for large time steps, the semi-Lagrangian method has been popular among researchers for solving many challenging problems. However, one of its well-known drawbacks is the inherent high dissipation or dispersion errors and associated mass imbalance and/or non-conservations of advected properties due to the interpolation schemes used. In this chapter, a high-order semi-Lagrangian method was proposed by using a 3rd-order backward and forward sweeping interpolation scheme to improve the interpolation accuracy and reduce the non-conservation problems. It has been shown that the alternating sweep of the 3rd-order backward and forward method can achieve an overall accuracy of 4th-order with lower computing cost than a typical 4th-order method. The present method is also able to reduce both dissipation and dispersion errors in all regions: applying the sweeping interpolation scheme can prevent high oscillations near sharp gradient regions. The proposed method was verified by 1-D and 2-D advection problems, and 2-D and 3-D lid-driven cavity flow problems governed by the full Navier-Stokes equation. The results showed that the proposed method can reduce the mass imbalance problem 70% in comparison with the conventional linear semi-Lagrangian method and 50% in comparison with the hybrid method in the literature. The cost to reach 4th-order accuracy is also much less than the typical 4th-order method because fewer grid cells are required and thus fewer floating point operations are involved. Meanwhile, because of its high-order of accuracy, the method was demonstrated for the 3-D lid-driven problem with more accurate but faster results even on coarse grids than the conventional linear semi-Lagrangian method for both uniform and non-uniform meshes.

Chapter 4 An adaptive time-stepping semi-Lagrangian method for incompressible flows

The contents of this chapter have been published in "Mortezazadeh, M. and Wang, L., 2019. An adaptive time-stepping semi-Lagrangian method for incompressible flows. Numerical Heat Transfer, Part B: Fundamentals, 1-18." . The contents are modified, and some parts of the original paper are removed to avoid repetition.

The semi-Lagrangian method is widely applied to solving the advection term of the Navier–Stokes (N–S) equations whereas the role of time step is often unclear. This article proposed an adaptive time-stepping method, which first calculates local adaptive time step based on truncation error coefficient functions, and then to obtain global time step based on an averaging function for all grid points. The new method was tested for solving 1-D and 2-D advections with different initial time steps and grid resolutions, and the transient incompressible N–S equations. Better simulation accuracy can be achieved than the cases with constant time steps.

4.1. Introduction

In the FFD method, interpolation scheme plays an important role: the accuracy of a semi-Lagrangian method depends much on the interpolation scheme used. In chapter 3, I proposed a 4th-order backward and forward sweep interpolating algorithm to reduce both dissipation and dispersion errors in all regions, especially near sharp gradients at a computing cost of 3rd-order scheme. Here I showed that the accuracy of the semi-Lagrangian method may not be improved at all by simply increasing grid resolutions, and it was however more effective to apply higher interpolation schemes. Additionally, some researchers developed new techniques to maintain the conservation of properties. Other researchers also proposed other ideas to satisfy mass conservation. For example, Jin and Chen (2015) calculated the surplus or deficit values of mass conservation at each time step, based which a distribution function was developed to be either removed or added to the fluid domain, so the conservation of mass was satisfied.

On the other hand, for a semi-Lagrangian method with any interpolation schemes, the interpolating accuracy depends directly on the relative location of the departure point about its neighbor cells, i.e. the relative distances (or the weights): s_0, s_1, t_0 and t_1 as shown in Figure 4-1. A special case may help to understand such a dependence: when the departure point is at the same location as that of the Eulerian grid (e.g. cell center) and an interpolation is thus not needed. In other words, the reason that an interpolation scheme is necessary is that the departure points are often different from the Eulerian grids. Although most of the previous studies focused on high-order interpolation schemes, the literature search shows that no study has been conducted on the dependency of the semi-Lagrangian method on the locations of departure points about their neighbor cells. Since a departure point is determined by tracing the corresponding arrival point backward based on one-time step multiplied by local velocity along the characteristic curve, its location is thus a function of the time step, local fluid velocity, and grid resolution, involving multiple independent variables. Therefore, to avoid overcomplicating the problem, this study focuses on the time step and its impact on the semi-Lagrangian method, and consequently the full Navier-Stokes (N-S) equations. A few key questions need to be answered: 1) For a given interpolation method, what is the actual role of the time step for the semi-Lagrangian method in terms of simulation accuracy? 2) Is a smaller time step associated with a better simulation accuracy or vice versa? 3) How can the time step be adapted (i.e. adaptive time-stepping) to achieve more accurate results?

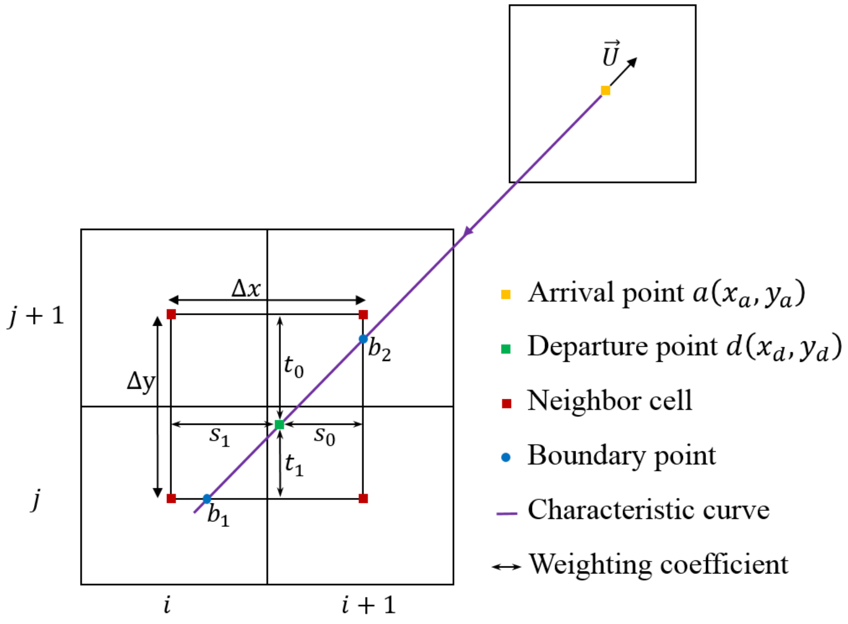


Figure 4-1 Schematic of the semi-Lagrangian method in 2-D coordinates.

Previous studies on time steps and their adaptive techniques are mostly developed for Eulerian methods, especially explicit ones, to overcome stability concerns due to the CFL constraints. A constant time step may alleviate the CFL constraints, but it is often computationally expensive. An alternative is the use of local adaptive time-stepping methods (Collino, 2003; Carlini et al., 2006; Minkoff and Kridler; 2006; Ghough et al., 2011; Grote and Mitkova, 2012; Gander and Halpern, 2013; Boscheri et al., 2015). Arnone et al. (1993) considered a local time-stepping method to achieve the maximum allowable time step for solving steady-state incompressible N-S equations. They calculated the time step for each computational cell by defining two different time steps, namely, the convective and diffusive time steps. Later, Ghough et al. (2011) extended their work to unsteady problems by using a dual time-stepping-method and a local time-stepping for the pseudo-time. Rice (1960) proposed a Runge-Kutta method to solve ODE problems to improve the convergence rate by splitting and solving the conservation equations by a small and large time step. Some other researchers proposed space-time adaptive schemes (Berger and Oliger, 1984; Berger,

1985) based on domain decomposition techniques: solving each subdomain by different grid resolutions and time steps. Another category of adaptive time-stepping is space-time finite element methods: time is considered as another direction like spatial directions and discretized by a finite element method (Eriksson et al., 2004).

Previous studies on adaptive time-stepping for the semi-Lagrangian method are found fairly limited in the literature. Among them, the most relevant one from the literature search is the study by Carlini et al. (2006), who used two different time steps, i.e. Δt and $\Delta t/2$, for solving the level-set equations of Mean Curvature Motion (MCM) using the semi-Lagrangian. The large time step is first used, and the local truncation error is calculated, and the process is repeated for the half time step. A combined truncation error is then obtained and evaluated to determine a new time step. Although their study managed to adjust time steps by evaluating local truncation errors, there is still a lack of a better understanding on the relations among time step, departure point location, and more importantly, the associated simulation accuracy.

In this study, a new adaptive time-stepping semi-Lagrangian method is proposed to modify the position of the departure point and improve the accuracy of the method by reducing cumulative local truncation errors based on a mathematical evaluation of the extrema of the error bounded by neighbor Eulerian cells surrounding the departure point. For any given initial time steps (e.g. from user inputs), the initial departure point and its neighbor cells are first identified. Functions of truncation errors for individual departure point are then constructed about its neighbor cells. This chapter then shows that the minima of the error functions can be achieved at two new locations of the departure points along the characteristic curve. Both new departure points correspond to the local time steps, of which the larger time step is chosen to be the local adaptive time step because of its fewer time step iterations required and consequently faster calculation. Using an averaging function, a unique adaptive global time step (i.e. simulation results independent of the initial time steps from the user's inputs) is then defined for all grids. In the present work, the new method is applied to both linear and 3rd-order interpolation schemes as an example, but the general methodology can be applied to other high-order interpolation schemes. This chapter starts with a general introduction to the semi-Lagrangian method followed by a detailed explanation of the proposed algorithm. The performance of the new method is then demonstrated by three cases: a 1-D and 2-D pure advection problems with both uniform and non-uniform grids, and a transient flow around a cylinder governed by the 2-D incompressible N-S equations, in terms of simulation accuracy as compared to the conventional method with constant time step. Based on the results of the adaptive time steps, further discussions were also conducted regarding what is a good estimate of an initial time step when using the semi-Lagrangian method for solving the N-S equations.

4.2. Methodology

4.2.1. Semi-Lagrangian method

In the conventional FFD and 2-D coordinates (Figure 4-1), according to Eq. (2-8) the relation of locations between the arrival point and the departure point can be shown as the scalar form, and then the vector form in terms of $\vec{x}_d$ and $\vec{x}_a$:

$$\begin{matrix} x_d = x_a - u_a\Delta t \\ y_d = y_a - v_a\Delta t \end{matrix} \rightarrow X_d = X_a - U_a\Delta t \tag{4-1}$$

Then the characteristic curve can be approximated by:

$$y - y_a = \frac{v_a}{u_a}(x - x_a) \tag{4-2}$$

The semi-Lagrangian method calculates the scalar quantity of the arrival point at the time $t = n + 1$ from that of the departure point at the time $t = n$ by using an interpolation scheme about the neighbor Eulerian cells of the departure point:

$$\emptyset_a = \emptyset_d = \sum_{i=1}^{nb} w_i \emptyset_i \tag{4-3}$$

where nb is the number of neighbor cells about the departure point, *d*. For each of the Eulerian neighbor cells, i, the location is defined by $X_{i,j}, X_{i+1,j}, X_{i,j+1}$, or $X_{i+1,j+1}$, for the 2-D example in Figure 4-1. w_i is the interpolation weight for each neighbor cell.

4.2.2. Truncation errors of interpolations

For any interpolating schemes, the method to find the scalar value at an arbitrary location, x_d, in one direction can be generalized by the following polynomial (Stewart, 1996):

$$\emptyset_{x_d} \approx \sum_{i=1}^{nb} \left[\left(\prod_{\substack{1 \le j \le nb \\ j \ne i}}^{nb} \frac{(x_d - x_j)}{(x_i - x_j)} \right) \emptyset_{x_i} \right] \tag{4-4}$$

where $x_d \in [x_1, x_{nb}]$. Here, i shows the neighbor Eulerian cells around the departure point. If nb neighbor cells are used, the local truncation error (Stewart, 1996) can be shown to be

$$E_{Truncation} = \left| \frac{1}{nb!} \prod_{j=1}^{nb} (x_d - x_j) \frac{\partial^{nb} \emptyset}{\partial x^{nb}} \right| \tag{4-5}$$

Here, I define the error coefficient, C,

$$C = | \prod_{j=1}^{nb} (x_d - x_j) | \tag{4-6}$$

In the following analysis, I investigate how the errors and error growths are associated with the locations of departure points, and accordingly time steps. I start with the 1-D advection problem and then extends the analysis to the 2-D case.

4.2.3. Error analysis for 1-D problems

4.2.3.1. Linear interpolation

First, a common linear interpolation scheme is used here to calculate $\emptyset_d$ for the 1-D problem in Figure 4-2 (for brevity $\emptyset_d$ is used instead of $\emptyset_{x_d}$). Based on a user-defined initial time step, the initial location of the departure point is assumed to be located at the point d ($x = x_d$ in Figure 4-2), at which the scalar, $\emptyset_d$, can be interpolated.

First, I apply the linear interpolation to Eq. (4-4):

$$\emptyset_d = \frac{(\Delta x - s_1)\emptyset_{b1} + s_1\emptyset_{b2}}{\Delta x} - (\Delta x - s_1)s_1\frac{\partial^2\emptyset_d}{2!\,\partial x^2} \tag{4-7}$$

The truncation error then is:

$$E_{\text{Truncation}}^{\text{linear}} = |-(\Delta x - s_1)s_1\frac{\partial^2\emptyset_d}{2!\,\partial \mathrm{x}^2}| = C \cdot |\frac{\partial^2\emptyset_d}{2!\,\partial \mathrm{x}^2}| = O(\Delta x^2) \tag{4-8}$$

where $\emptyset_{b1}$ and $\emptyset_{b2}$ denote the scalar values at the neighbor cells, i.e. x_{b1} and x_{b2}, respectively in Figure 4-2(a). The truncation error shows the linear interpolation is $O(\Delta x^2)$ in space, indicating numerical dissipation error. Here, the error coefficient is:

$$C^{\text{linear}} = (\Delta x - s_1)s_1 = s_1 s_0 \tag{4-9}$$

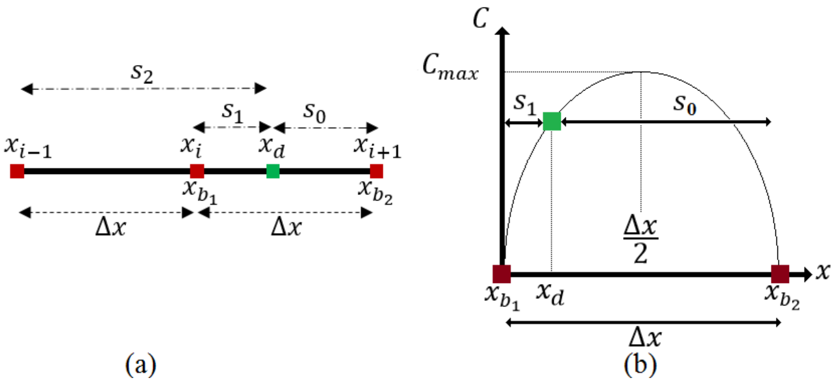


Figure 4-2 (a) Schematic of 1-D interpolation method and (b) error coefficient function for 1-D problem: quadratic polynomial curve.

Eq. (4-9) is a quadratic polynomial curve and can be plotted in Figure 4-2(b). It is clear that the minimum C can be achieved at either left or right bound of Δx, at which the truncation error would become zero. In other words, no interpolation is needed when the departure point is located at the left or right boundary points: $s_1 = 0$ or $s_1 = \Delta x$. Note here for 1-D advections, the boundary points are the same as the neighbor cells about the departure point, which is not the case for higher-dimensional problems (e.g. 2-D) as discussed in later sections. Then, the adapted new local time step can be found for the boundary points b_1 and b_2, respectively:

$$\Delta t_{b_1} = \frac{x_a - x_{b_1}}{u_a} \tag{4-10a}$$

$$\Delta t_{b_2} = \frac{x_a - x_{b_2}}{u_a} \tag{4-10b}$$

Although both adaptive time steps in Eq. (4-10) ensure zero truncation error, it is always preferred to choose the one with a larger time step to reduce the total number of time step iterations. Note the adapted time step here is the local time for each grid, which may be the same for a uniform grid for a given and uniform flow field, but not for non-uniform grids and/or flows. To allow all the grids to march at the same time step, an averaging function is needed to ensure a global time step, which will be discussed with details in a later section. As a result, the departure points may not all be adjusted to be the boundary points of a cell. However, Figure 4-2(b) also shows that the local truncation error of a cell can always be reduced when the location of the departure point, x_d, is as close as possible to either boundary points of the cell of the departure point.

Here, by plotting the error coefficients and finding their extrema, the impact of departure point locations on magnitude of truncation errors is revealed so that they can be minimized by choosing an adaptive new time step for a given flow field on a given grid. The above analysis method is therefore not limited to linear interpolation but can be extended to higher order schemes.

4.2.3.2. Higher-order interpolation

Here, the previous analysis was extended to a 3rd-order backward interpolation for the 1-D problem in Figure 4-2(a). Applying Eqs. (4-5) and (4-6), I obtain the following truncation error in Eq. (4-11), and the error coefficient in Eq. (4-12) for the 3rd-order interpolation.

$$E_{\text{Truncation}}^{\text{3rd}} = \left|\frac{1}{6}(x_d - x_{i-1})(x_d - x_i)(x_d - x_{i+1})\frac{\partial^3 \emptyset_d}{\partial x^3}\right| = O(\Delta x^3) \quad (4\text{-}11)$$

$$C^{\text{3rd}} = |(x_d - x_{i-1})(x_d - x_i)(x_d - x_{i+1})| = s_2 s_1 s_0 \quad (4\text{-}12)$$

Eq. (4-11) shows that the truncation error of the 3rd-order interpolation is dispersive, producing oscillations near regions with sharp gradients. From Figure 4-2(a), when $x_d = x_i$ (i.e. $s_1 = 0$), or $x_d = x_{i+1}$ (i.e. $s_0 = 0$), C^{3rd} equals to zero so the numerical dispersion error becomes zero. Therefore, choosing the new departure point at b_1 or b_2 locally will eliminate numerical dispersion errors for the 3rd-order interpolation schemes. This is similar to the linear interpolation case, where the numerical dissipation error can be eliminated.

4.2.4. Error analysis for multidimensional problems

Different from 1-D problems, where the characteristic curve follows the direction of the 1-D coordinate, the interpolation and its truncation error are more complicated for 2-D or 3-D problems. For linear interpolations, a common interpolating method for the 2-D case in Figure 4-1 is the bilinear scheme:

$$\emptyset_d = \frac{t_0\left(\frac{s_0\emptyset_{i,j} + s_1\emptyset_{i+1,j}}{s_0 + s_1}\right) + t_1\left(\frac{s_0\emptyset_{i,j+1} + s_1\emptyset_{i+1,j+1}}{s_0 + s_1}\right)}{t_0 + t_1} + E_{\text{Truncation}}^{\text{linear}} \tag{4-13}$$

where $s_0 = \Delta x - s_1$, $t_0 = \Delta y - t_1$, and the truncation error

$$E_{\text{Truncation}}^{\text{linear}} = |\frac{s_1 s_0}{2!}\frac{\partial^2 \emptyset_d}{\partial \mathrm{x}^2} + \frac{t_1 t_0}{2!}\frac{\partial^2 \emptyset_d}{\partial \mathrm{y}^2}| = C_x \cdot |\frac{\partial^2 \emptyset_d}{2!\,\partial \mathrm{x}^2}| + C_y \cdot |\frac{\partial^2 \emptyset_d}{2!\,\partial \mathrm{y}^2}| \tag{4-14}$$

where C_x and C_y are the error coefficient functions along x and y-directions, respectively. They can be further expressed in terms of time steps and the neighbor cell, $\vec{x}_{i,j}(x_i, x_j)$ based on Eqs. (4-1) and (4-6).

$$C_x^{\text{linear}} = (x_a - u_a\Delta t - x_i)(\Delta x - x_a + u_a\Delta t + x_i) = s_1 s_0 \tag{4-15a}$$

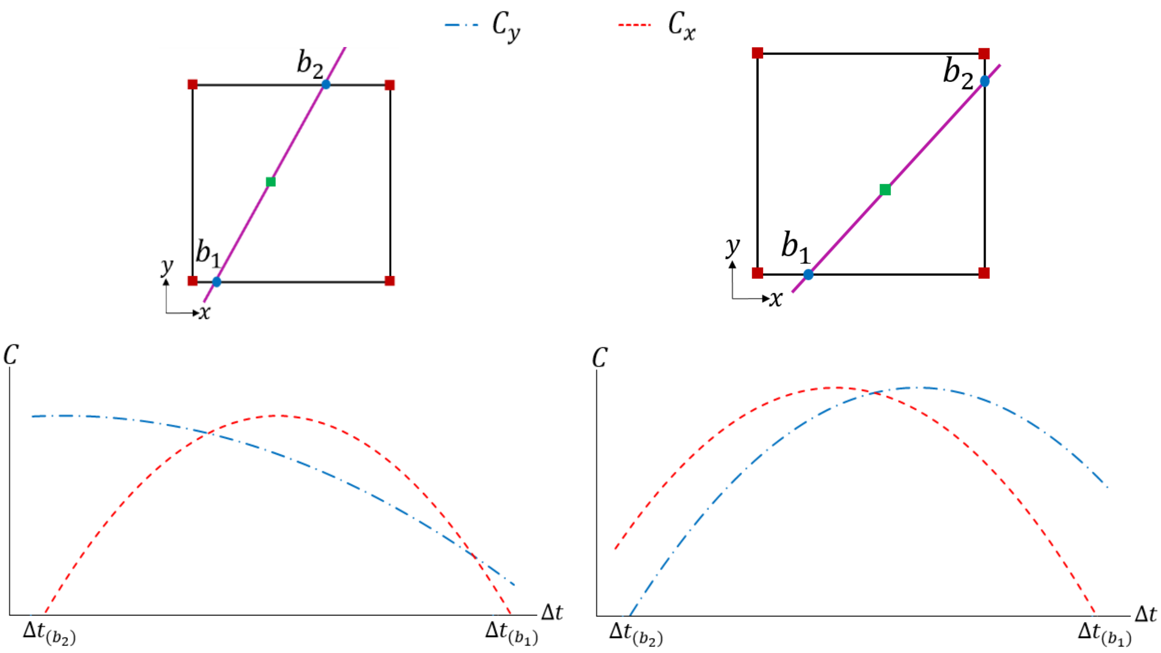
$$C_y^{\text{linear}} = (y_a - v_a\Delta t - y_j)(\Delta y - y_a + v_a\Delta t + y_j) = t_1 t_0 \tag{4-15b}$$

A similar analysis can be applied to the 3$^{\text{rd}}$-order backward interpolation scheme, so I obtain the following error coefficient functions.

$$C_x^{\text{3rd}} = s_2 s_1 s_0 \tag{4-16a}$$

$$C_y^{\text{3rd}} = t_2 t_1 t_0 \tag{4-16b}$$

Both Eqs. (4-15) and (4-16) show that the error coefficient functions (either x or y-direction of the 2-D problem) are the same as that of the 1-D problem in the corresponding direction, indicating the previous 1-D analysis can be extended to each direction of a multi-dimensional problem separately. It is also observed that they are functions of the time step, Δt, and local velocities and grids, and thus the error could grow differently in x and y-directions. It is possible to plot them in terms of the time step. Without losing generality, the curves of $C_x \sim \Delta t$ and $C_y \sim \Delta t$ for the linear interpolation are discussed for two cases in terms of the locations of the boundary points (i.e. b_1 and b_2 in Figure 4-1), depending on how the characteristic curve intercepts the cell of the departure point, as shown in Figure 4-3. A similar analysis can be applied to the higher-order interpolations.



(a) $C_{y,b_1} = C_{y,b_2} = 0$ (b) $C_{y,b_1} = C_{x,b_2} = 0$

Figure 4-3 Local error coefficient functions versus time steps for x and y-direction for 2-D problem.

For 2-D problems, it is possible to generalize to different scenarios how the characteristic curve passes through a 2-D cell as shown in Figure 4-3. Note that here the 2-D cell is comprised of all neighbor points surrounding the departure point as shown in Figure 4-1. It can be shown from Eq. (4-15) that in all cases, either C_x or C_y becomes zero at b_1 or b_2, depending on the boundary point locations. For example, in Figure 4-3(a), the characteristic curve passes through the two horizontal bounds of the cell so that $C_{y,b_1} = C_{y,b_2} = 0$ at both boundary points, because $t_1 = 0$ for C_{y,b_1}, and $t_0 = 0$ for C_{y,b_2}. Therefore, choosing b_1 as the new departure point to find the new adaptive time step will ensure the truncation error coefficients to be minimal in both x and y directions. In comparison, the truncation errors may behave differently as observed in Figure 4-3(b), $C_{y,b_1} = C_{x,b_2} = 0$. As a result, choosing either b_1 or b_2 as the new departure point may minimize the error in one direction but cause it to grow in the other direction. In fact, different strategies may be developed on how to choose b_1 and b_2 to find an adaptive local time step. In this chapter, I suggest choosing either b_1 or b_2 to achieve a larger time step to reduce the number of time step iterations as explained for the 1-D problem. Other strategies may also be possible, for example, choosing the adaptive time step to eliminate/reduce numerical dissipation/dispersion errors in specific flow directions or specific flow regions, which I leave to the readers.

In summary, for multi-dimensional problems, the following local adaptive time stepping scheme to reduce the local truncation error growth is proposed as: the new departure point should be chosen as either b_1 or b_2, based on whichever the associated new adaptive time step is larger. Once the new departure point is chosen, the new time step can be calculated by using Eq. (4-10).

4.2.5. Calculation procedure for global adaptive time stepping

The previous section focuses on the local errors and adaptive time step. For all grids in a flow domain, each cell may suggest a different local time step. Therefore, to march all grids at the same time step, this study proposes an averaging function for the global time step based on the distribution of the advected scalars.

$$\Delta t_g = \frac{\sum_{l=1}^{l_{max}} A_l \emptyset_l \Delta t_l}{\sum_{l=1}^{l_{max}} A_l \emptyset_l} \tag{4-17}$$

where Δt_g is the new global time step, A_l is the area of the l^{th} local cell for a 2-D problem, l_{max} is the number of cells, and Δt_l is the local time step for each cell found by the local time stepping scheme in the previous section. The overall adaptive time-stepping semi-Lagrangian method is then proposed as follows and as illustrated by Figure 4-4:

1. Based on the user-defined initial time step, Δt_i, the locations of the initial departure points are calculated using Eq. (4-1).
2. The local error growth functions are constructed for all grids using Eq. (4-15) or (4-16) so that new local time steps, i.e. Δt_l, are determined following the local time stepping scheme based on Eq. (4-10).
3. The new global time step, Δt_g, is then calculated using the averaging function, Eq. (4-17). Using the new global time step, the new locations of all departure points are then updated by Eq. (4-1).

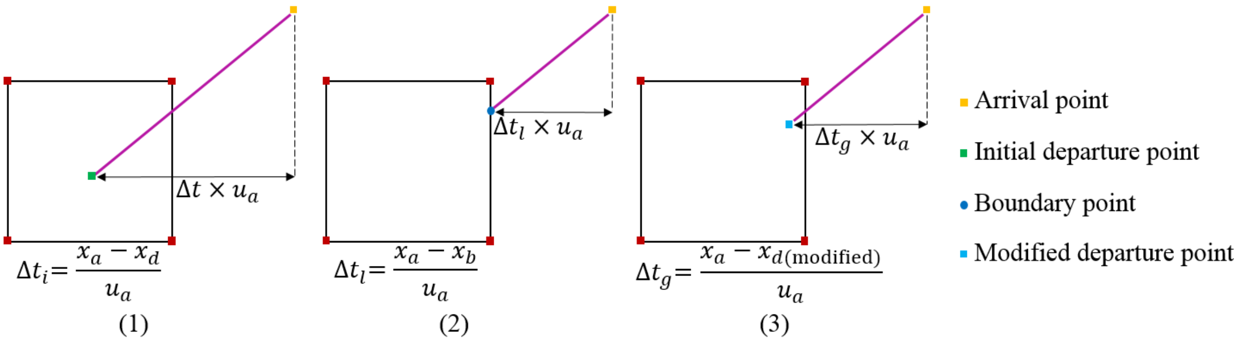


Figure 4-4 Adaptive time-stepping steps: (1) step 1, (2) step 2, and (3) step 3, using a local cell as an example.

4.3. Case studies

In this section, both 1-D and 2-D pure advection problems, and the 2-D N-S equations are solved for the demonstration of the new method by comparison to the conventional semi-Lagrangian method. The comparison of simulation accuracy is determined by a l_2 and l_∞ norm (Takacs, 1985; Ziao and Yabe, 2001).

$$E_{l_2} = \frac{1}{N}\sum_{i,j}\left(\emptyset_{i,j}^{Analytical} - \emptyset_{i,j}^{Numerical}\right)^2 \tag{4-18}$$

$$E_{l_\infty} = \frac{\max|\emptyset_{i,j}^{Analytical} - \emptyset_{i,j}^{Numerical}|}{\max|\emptyset_{i,j}^{Analytical}|} \tag{4-19}$$

where N is the total number of grids.

4.3.1. One-dimensional step wave advection

The first test case is a 1-D step wave advection problem. The wave moves at a constant velocity of $u = 1.0$ m/s with an initial time step of 0.01 [s]. The advected scalar with an initial step shape is defined by

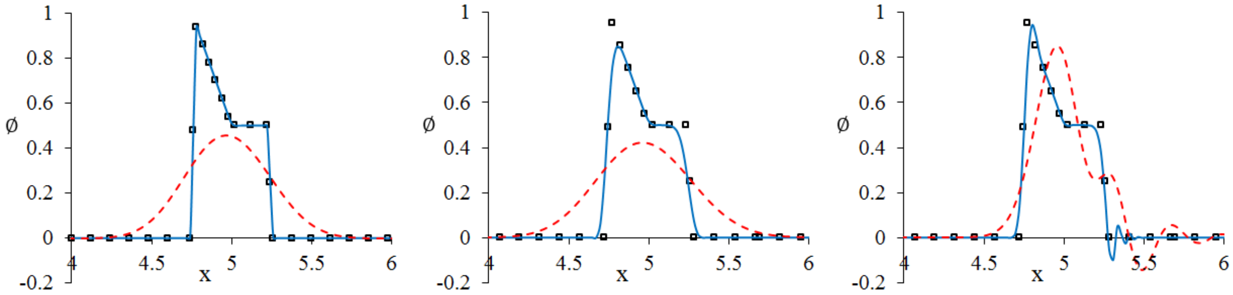
$$\emptyset_{x_i}^{t=0} = \begin{cases} 1.0 - 2.0 \times (x_i - 0.25) & 0.25 \leq x_i \leq 0.5 \\ 0.5 & 0.5 < x_i \leq 0.75 \\ 0.0 & \text{otherwise} \end{cases} \tag{4-20}$$

Different grid resolutions are tested here:

$$x_i = [0.02 \times (i-1)]^{\alpha} \qquad \begin{cases} \alpha = 1 & uniform\ grid \\ \alpha = 1.1 & non - uniform\ grid \\ \alpha = 1.5 & non - uniform\ grid \end{cases} \tag{4-21}$$

Where α is the grid stretching factor, of which a greater value indicates a higher non-uniformity of the grid. Here $\alpha = 1$ indicates the uniform grid of 450 cells for the case; $\alpha = 1.1$ is for 370 cells with $\Delta x_{\min} = 0.0135$ and $\Delta x_{\max} = 0.02658$: the highest stretching ratio is about 2; $\alpha = 1.5$ is for 220 cells with $\Delta x_{\min} = 0.0028$ and $\Delta x_{\max} = 0.06$ so the highest stretching ratio is about 22. For the non-uniform grids, the grid gets coarser when the wave moves from the left to the right along x-direction.

▫ Analytic - - - SL —— Present work



(Linear interpolation) (Linear interpolation) (3rd-order interpolation)

$\alpha = 1$ (450 cells) $\alpha = 1.1$ (370 cells) $\alpha = 1.1$ (370 cells)

(a) (b) (c)

Figure 4-5 Comparison between the conventional semi-Lagrangian (SL) and the present work for both uniform (α=1) and non-uniform grid (α=1.1) for the 1-D step wave advection at t=4.5 s.

Figure 4-5 compares the results for the conventional semi-Lagrangian method (SL), and the present adaptive time stepping method with the same interpolation schemes for both uniform and non-uniform grids at the time $t = 4.5\ s$. The results show that the conventional linear SL method suffers strong numerical dissipation errors, and the situation worsens when the non-uniform grids are used. In comparison, the adaptive time-stepping semi-Lagrangian method can improve the simulation accuracy considerably. For the uniform grid in Figure 4-5(a), because of the uniform grid and constant velocity, the adaptive local time step is the same for each grid point, which modifies all the departure points to be exactly on the Eulerian grids. The truncation errors are consequently zero at all grid points. Thus, the present work can achieve the exact solution after using the adaptive time step. When a non-uniform grid is used, $\alpha = 1.1$ in Figure 4-5(b), the present work is shown to be able to maintain the level of accuracy. Figure 4-5(c) also shows that because of the dispersion error of the 3rd-order interpolation, the conventional SL method is strongly dispersive near both sharp gradient regions. By applying the proposed adaptive time stepping method, the oscillations are significantly reduced. Therefore, the present adaptive time-stepping method is shown to be able to reduce numerical truncation errors, either dissipation or dispersion, for both uniform and non-uniform grids.

To evaluate the time accuracy of the present work, the performance of the adaptive time-stepping method can be further illustrated with the details of total error, E_{l_2}, growth with time in Figure 4-6. Among all three grid resolutions, I only compare the results for the most non-uniform grid of $\alpha = 1.5$. The total error of the SL increases with the time significantly and continuously till $E_{l_2} > 0.008$ at $t = 5.5$ [s] whereas the present study maintains a constant error of $E_{l_2} < 0.001$ for most of the time duration because of the use of the adaptive global time step, which varies from 0.025 s to 0.054 s. This is because the non-uniform grid gets coarser along the +x-axis so a larger adaptive time step is needed to modify all the departure points to get closer to the Eulerian grids, when the wave moves from the left to the right with time. Here, the present work is able to dynamically adjust the time step at each semi-Lagrangian iteration with the variations of grid sizes for better simulation accuracy, which cannot be achieved when the time step is constant. Therefore, the present work is shown to produce more time-accurate results than the constant time step SL method by limiting the growth of numerical errors through adaptive time stepping.

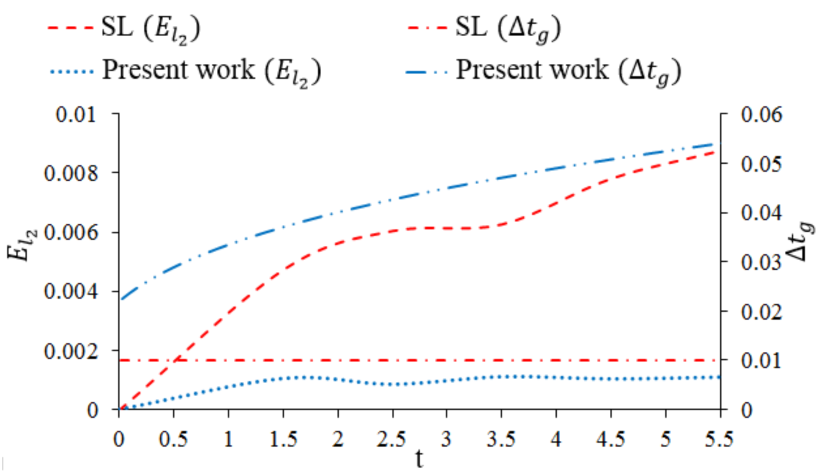


Figure 4-6 Comparison between SL and present work for the error growth with the time for the 1-D step wave with the non-uniform grid $\alpha = 1.5$.

In the present study, the time step is adapted when a user defines an initial time step as illustrated by the procedure in Figure 4-4. To check the performance of the present work for different user-defined initial time steps, I compare the results for two initial time steps of 0.01 s and 0.043 s between the present work and the conventional SL as shown in Figure 4-7. As expected, the SL method produces different results for different time steps. But interestingly, when compared to the analytical solution, the larger time step, $\Delta t = 0.043\ s$, causes fewer numerical dissipations and thus generates better results than that of $\Delta t = 0.01\ s$. Therefore, it confirms that the time step plays an important role in the conventional SL method and a smaller time step seems unassociated with better accuracy. By calculating different local time steps (between 0.01 s and 0.062 s) at different grid points in this problem as shown by Figure 4-7, the present work is able to improve the results significantly. Figure 4-7 also shows that despite different initial time steps, the adaptive time step method is able to achieve the same global adaptive time step and thus the same results for both initial time steps.

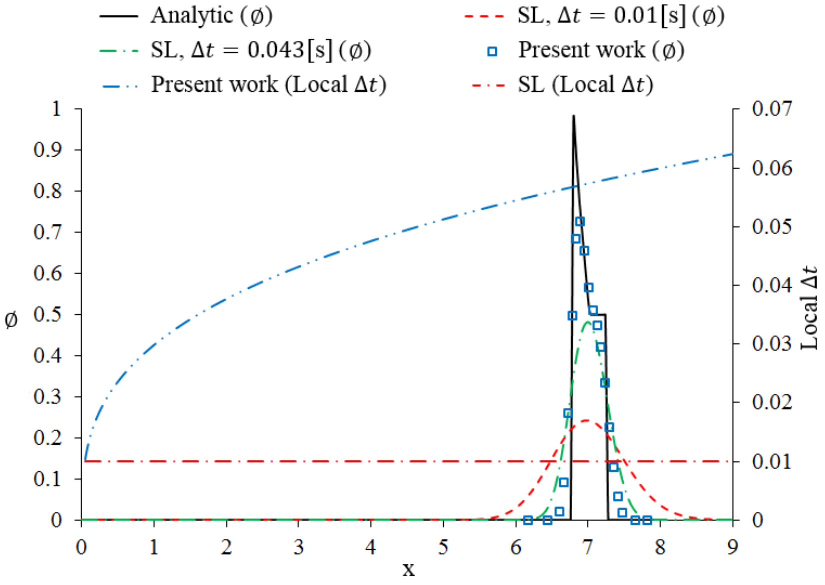


Figure 4-7 Comparison of the present work and the SL for different initial time steps and local time steps for the 1-D step wave advection $\boldsymbol{at\ t} =$ $\mathbf{6.5}\ \boldsymbol{s}$ with the non-uniform grid $\boldsymbol{\alpha} = \mathbf{1.5}$.

4.3.2. Two-dimensional square wave advection

Here, a 2-D square wave advection problem is investigated with the initial conditions defined by Eq. (4-22) and shown by Figure 4-8.

$$\emptyset_{(t=0,x_i,y_j)} = \begin{cases} 1.0 & (20,20) \leq (i,j) \leq (30,30) \\ 0.0 & \text{otherwise} \end{cases} \qquad (4\text{-}22)$$

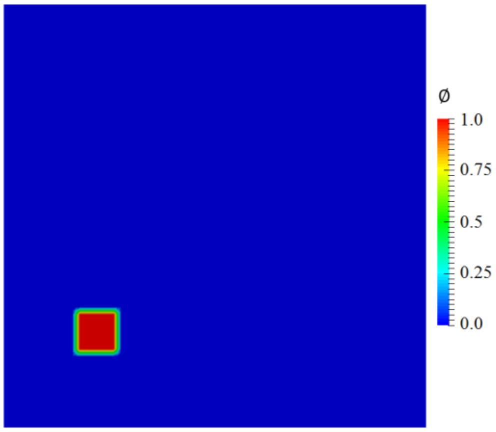
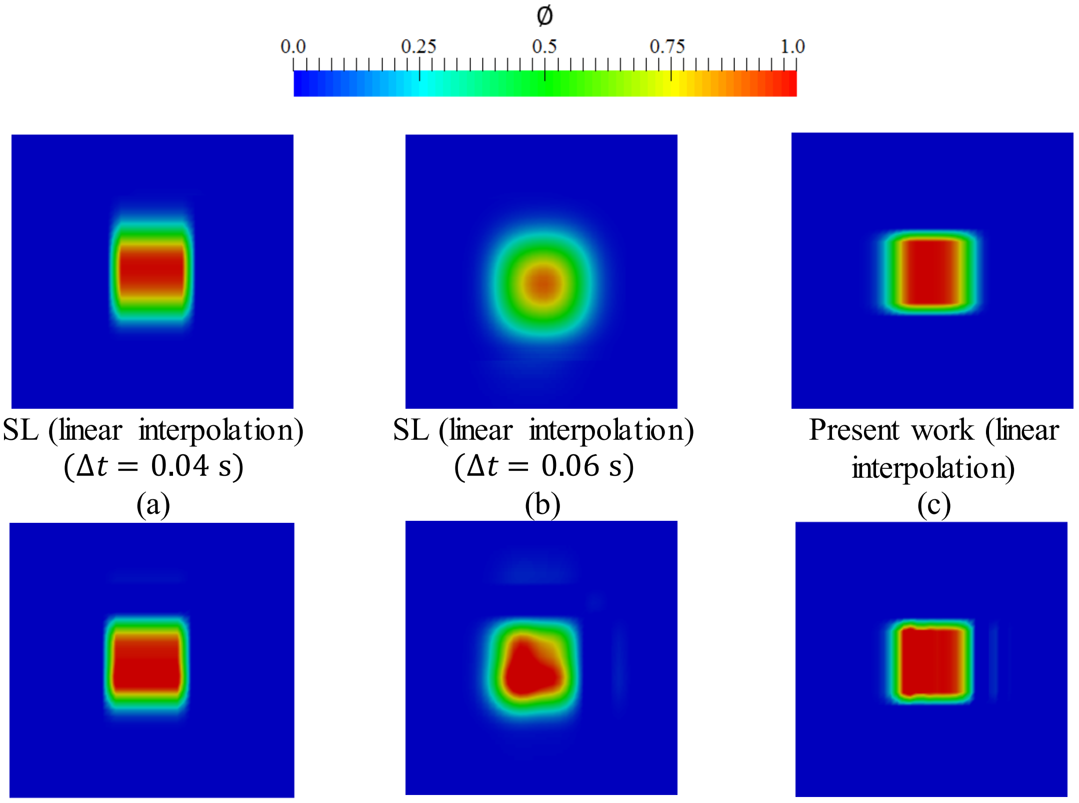


Figure 4-8 The initial condition of the 2-D square wave advection problem.

A uniform grid (100×100) with $\Delta x = \Delta y = 0.04$ m is used, and the flow field is defined by Eq. (4-23).

$$u = 1.0 \text{ m/s}, \;\; v = 1.1 \text{ m/s} \qquad 0 \leq t \leq 2\ s \tag{4-23}$$



SL (linear interpolation) $(\Delta t = 0.04\ \text{s})$ (a)

SL (linear interpolation) $(\Delta t = 0.06\ \text{s})$ (b)

Present work (linear interpolation) (c)

SL (3rd-order interpolation) $(\Delta t = 0.04\ s)$ (d)

SL (3rd-order interpolation) $(\Delta t = 0.06\ s)$ (e)

Present work (3rd-order interpolation) (f)

Figure 4-9 The 2-D advection problem for a uniform grid at $t = 2.0\ s$.

Here, two different user-defined initial time steps are considered, 0.04 [s], and 0.06 [s]. Figure 4-9 shows the results at $t = 2.0$ [s] for the conventional SL method and the present work with linear and 3rd-order interpolations. I have the following observations:

- The results of the SL method depend on user-defined initial time steps. A smaller time step does not ensure better accuracy or vice versa.
- For the SL method, the truncation error may grow differently in x and y-directions. Figure 4-9(a) shows that it is more in the y-direction than the x for $\Delta t = 0.04$ s. In comparison, when $\Delta t = 0.06$ s in Figure 4-9(b), the error growth seems similar for both directions.
- When the 3rd-order backward interpolation is used for $\Delta t = 0.04$ s in Figure 4-9(d), the SL result is better than that of the linear interpolation but with a significant difference of errors in x and y-directions. The similar pattern was also observed by the previous study (Mortezazadeh and Wang, 2017) for high-order backward interpolation schemes of the SL method with constant time step due to the use of the backward scheme, where uneven number of neighbor cells were used (see Figure 4-2(a)).

Figures 4-9(c) and 4-9(f) show the results for the present work with the proposed adaptive time stepping method. Both results show significant improvement of accuracy over the constant time step SL, although the 3rd-order interpolation is more accurate than the linear interpolation. The adapted global time step is 0.0727 s at the time step of $t = 2.0$ [s]: a slight adjustment of the time step from the original $\Delta t = 0.06$ s to 0.0727 s contributes to a huge improvement of the accuracy. Once again, this case demonstrates how critical the time step of the semi-Lagrangian method can become for simulation accuracy.

4.3.3. Two-dimensional Navier–Stokes equations

To apply the proposed adaptive time semi-Lagrangian method to solving the N-S equations, an unsteady airflow around a cylinder with $Re = 100$ is used as shown in Figure 4-10 based on the previous study (Malan et al., 2002). Using the fractional step method (Mortezazadeh and Wang, 2017), the advection term is separated from the N-S equations and solved by the present method with the 3rd-order backward interpolation. The diffusion term and the Poison equation are solved by a V-cycle multigrid solver with Gauss-Seidel as the smoother. The details of the N–S equation solver can be found from chapter 3. The computing domain is with a uniform grid of 200×100 cells so $\Delta x = \Delta y = 0.08$ m. The inlet velocity is horizontal and uniform: $u = 1$ m/s and the outlet is set to be the free flow boundary.

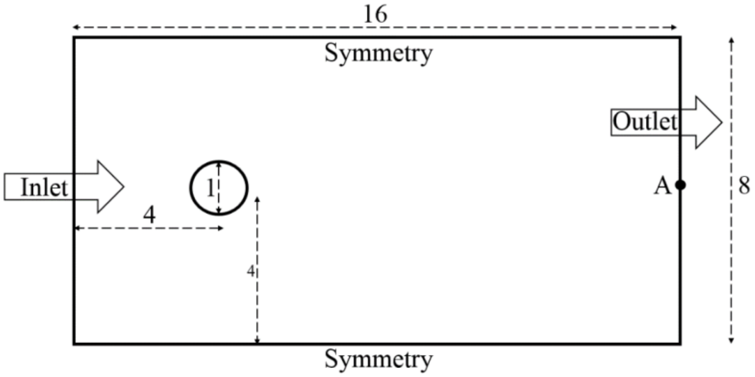
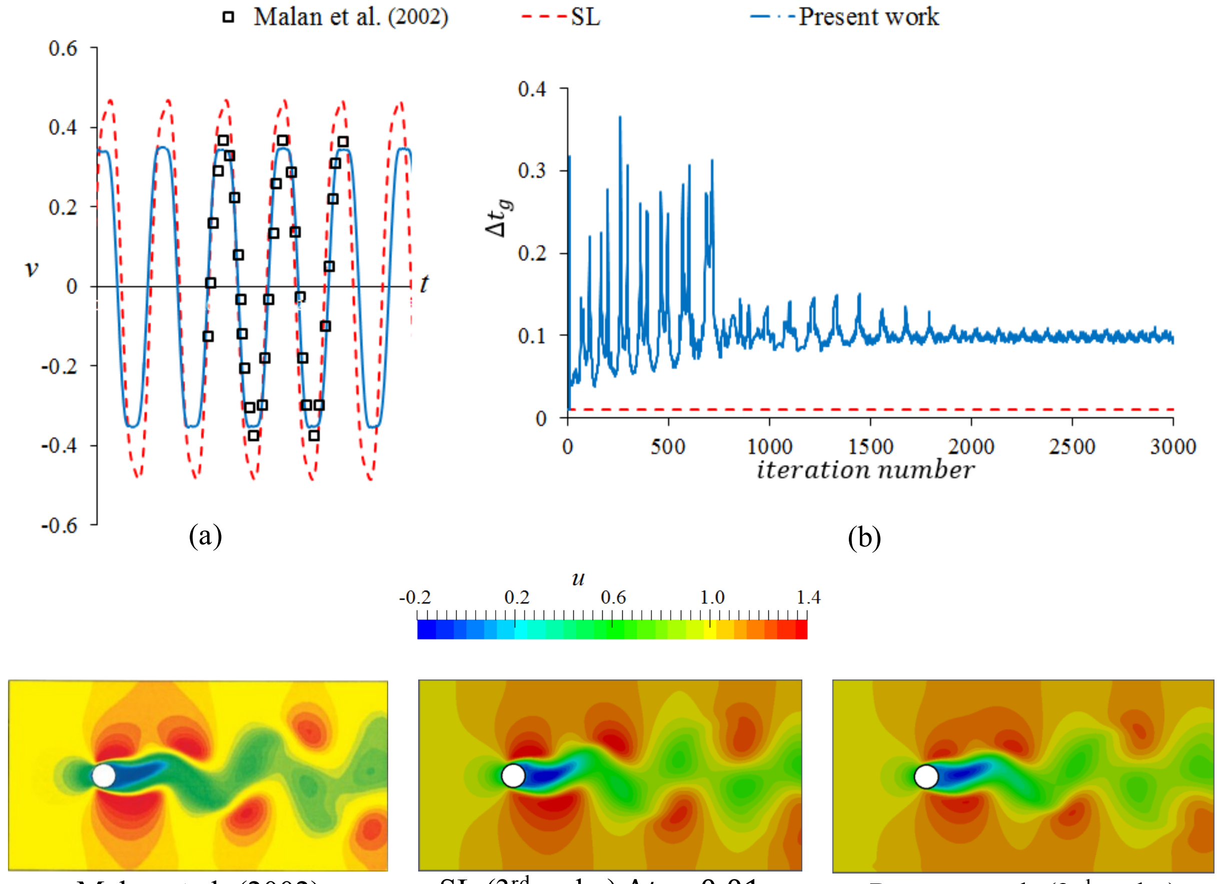


Figure 4-10 An unsteady airflow around a cylinder governed by the 2-D N-S equations.

Figure 4-11(a) compares the results of the proposed adaptive time stepping and the constant time step SL method with the initial time step of 0.01 s, and the previous study by Malan et al. (2002), who solved the problem by a robust artificial compressibility scheme which is equipped with a locally generalized preconditioner. It shows that by adjusting the time step dynamically, the proposed method improves the simulation accuracy, especially for peak velocities. Figure 4-11(c) compares visually the velocity contours by the three studies: although the 3rd-order conventional SL method in generally compares well to the result from Malan et al. (2002), apparent discrepancies can be observed near the wake regions with sharp gradients; the proposed method clearly shows an improvement of the results for these regions. Figure 4-11(b) illustrates that how the present work modifies the global time step dynamically from the initial time step of 0.01 [s] in the beginning to a relatively stable time step of about 0.1 [s] at a later stage.



(a)

(b)

Malan et al. (2002)

SL (3rd-order) $\Delta t = 0.01\ s$

Present work (3rd-order)

(c)

Figure 4-11 Comparison of the present work for solving the full 2-D Navier-Stokes equations with the conventional SL method (initial time step of 0.01 s) and the literature for (a) the velocity at the point A in Figure 4-10; (b) the history of the adapted global time steps; and (c) the velocity contours.

$$NMSE = \frac{1}{N}\sum_{i=1}^{i_{max}} \frac{(P_i - M_i)^2}{\bar{P}\bar{M}}, \bar{P} = \frac{1}{N}\sum_{i=1}^{i_{max}} P_i, \bar{M} = \frac{1}{N}\sum_{i=1}^{i_{max}} M_i \tag{4-24}$$

where $i_{max} = 31$ is the total data points for comparison avaiable from the study of Malan et al. (2002); P_i is the result of the present work and M_i is the reference result from Malan et al. (2002).

Table 4-1 The normalized mean square errors (NMSE) and CFL numbers for the 3rd-order SL method with different initial time steps and the present adaptive time stepping method when the flow becomes periodical.

	3rd-order Conventional SL			Present Work
Δt (s)	$0.005\ s$	$0.01\ s$	$0.04\ s$	0.087 s ~ 0.106 s
NMSE	33.5	34.9	20.1	8.6
CFL	0.06	0.13	0.5	1.09 ~ 1.33

Table 4-1 compares the NMSE for different initial time steps between the 3rd-order SL method with constant time steps and the present work, when the flow becomes periodical at the quasi-steady state. It shows that when solving the full N–S equations, the accuracy of the constant time step SL method heavily depends on the time steps; neither smaller nor greater time step ensures more accurate result; the present work is shown to be able to adjust the global time step dynamically around 0.1 s: from 0.087 s to 0.106 s so the NMSE is well reduced. A closer look at the corresponding CFL number shows that the proposed adaptive time stepping method tries to maintain $\mathrm{CFL} \approx 1$ (1.09–1.33) dynamically. It is understandable because the proposed method tries to adjust the time step, so the new departure points can stay as close as possible to the Eulerian grids. As a result, the arrival point and the new departure point is thus about "one CFL number" apart. In summary, with the proposed adaptive time stepping method, I am able to find a new global time step with reduced truncation errors. I would also like to point out that the performance of the proposed adaptive time stepping method, when solving the full N–S equations, depends on other factors such as the relative significance of advection associated numerical errors, and how other terms (e.g. diffusion terms) are affected by the new time step. However, this study provides a deeper understanding how the accuracy of a semi-Lagrangian method (either simple linear or high-order interpolations) is associated with time step and illustrates that it is possible to adjust the time step to reduce truncation errors.

4.4. Summary and Conclusion

This article investigates the impact of time steps on the simulation accuracy of the conventional linear and higher-order semi-Lagrangian method with the constant time steps by defining the error coefficient functions associated with the numerical truncation errors. By the analysis of these functions for both 1-D and 2-D advection problems, and the full N–S equation problem with linear and 3rd-order interpolation schemes, the three questions mentioned previously in the introduction are answered:

1. For a given interpolation method, what is the role that the time step plays in terms of simulation accuracy?

   The time step directly affects the spatial truncation errors (e.g. the dissipation error for the linear interpolation, and the dispersion error for the third-order interpolation), which could vary differently in x and y-directions. If the time step is adapted appropriately, the interpolating errors in x and y-directions can both be reduced. Therefore, by adapting the time step, I can control the error growths for a given interpolation scheme.
2. Is a smaller time step associated with a better simulation accuracy or vice versa?

   Neither smaller nor greater time step is monotonically associated with the accuracy of the SL method. A smaller time step can cause poorer simulation results or in other words, reducing the time step arbitrarily cannot guarantee better accuracy of the semi-Lagrangian method. The actual role of the time step is directly associated with the truncation error growth as explained previously.

3. How can the time step be adapted (i.e. adaptive time-stepping) to achieve more accurate results? An adaptive time-stepping semi-Lagrangian method is proposed to reduce the global numerical truncation errors in both x and y-directions using a local adaptive time stepping scheme to reduce the local error, and an averaging function for the global time step to march all grids at the same time. The performance of the new method is tested and confirmed for solving 1-D and 2-D advection problems and solving the 2-D N–S equations with different user-defined initial time steps and with uniform and non-uniform grids.

It is also noted that when solving the full N–S equations, the proposed adaptive time stepping method adjusts the new departure point of a semi-Lagrangian method to be as close as to the Eulerian grid: in the case study, its direct outcome is that $\mathrm{CFL} \approx 1$. Therefore, a time step is preferred for a global CFL number of one, which is true for both linear and high-order interpolation schemes. The further implication from this would be that a good choice of the initial time step for a semi-Lagrangian method can be based on $\mathrm{CFL} \approx 1$, e.g. using initial velocity boundary conditions to estimate a preferred time step. Future confirmation studies are necessary, so the initial time-step's use should be with caution. Future study is also needed to confirm and apply the proposed method by extending to the 3-D implementations and developing other adaptive time step schemes (e.g. for specific flow directions or regions).

Chapter 5 High-order temporal Fast Fluid Dynamics scheme by accurate estimation of the characteristic curve

The contents of this chapter are ready to submit "Mortezazadeh, M. and Wang, L., High-order temporal Fast Fluid Dynamics scheme by accurate estimation of the characteristic curve." . The contents are modified, and some parts of the original draft are removed to avoid repetition.

In the building and environmental problems, the existence of an accurate, fast, and reliable CFD solver is crucial for modeling indoor and outdoor environment problems. Building-related test cases, such as high-rise buildings and urban microclimate, are mostly large scales. So, numerical simulation of airflow for these typical problems are computationally too expensive. Using coarse grids and large time steps can significantly reduce the computational time but it also reduces accuracy. In the present work, a new method is proposed to improve the temporal accuracy of the Fast Fluid Dynamics (FFD) approach and provide accurate results even on the large time steps. The temporal accuracy is improved from the first order to the second order by considering the acceleration of the flow field and accurate estimation of the characteristic curves. Performance of the proposed method is investigated by simulating a pure advection problem. Then, an airflow around a cylinder is studied to compare the transient results. Next, two 3-D indoor and outdoor cases are simulated. Indoor airflow is related to forced convection in a single room. Outdoor problem is airflow simulation in a step-down canyon.

5.1. Introduction

As mentioned before, FFD method is an unconditional stable model which can simulate airflow on the coarse grids and large time steps. Using a large time step can significantly reduce the accuracy of conventional FFD method. In the present work, a new temporal high-order FFD method is developed to simulate the indoor and outdoor airflow problems with large time step. The temporal accuracy is improved by considering the acceleration of the fluid's particle inside the computational domain. So, the estimation of the characteristic curves will be improved. The proposed method is applied to a high-order FFD method, discussed in chapter 3, which is suitable for accurate simulation on the coarse grids.

Results show the proposed method can significantly improve the accuracy of the conventional FFD method, especially for large time steps and coarse grids. In the following sections, the methodology will be explained first. Then, the performance of the proposed method is shown for a pure advection problem. Later, unsteady airflow around a cylinder, forced convection inside a single room, and outdoor airflow around two buildings will be presented for different time steps and grid resolutions.

5.2. Methodology

In the present work, dimensionless Navier-Stokes equations for the turbulence airflow problems is divided into 3 sub-equations (1. diffusion and source terms, 2. advection terms, and 3. pressure gradient) by using the fractional step method (see chapter 2):

$$\frac{\partial U}{\partial t} = \left(\frac{1}{Re} + \nu_t\right) \nabla^2 U + \vec{f} \tag{5-1}$$

$$\frac{\partial U}{\partial t} = -(U \cdot \nabla) U \tag{5-2}$$

$$\frac{\partial U}{\partial t} = -\nabla p \tag{5-3}$$

In this method, an intermediate velocity domain is estimated by solving Eqs. (5-1) and (5-2). The pressure domain based on the intermediate velocity fields is calculated by solving the Poisson equation which is defined as follows:

$$\nabla^2 p = \frac{1}{\Delta t}\nabla \cdot U \tag{5-4}$$

By using the new pressure field, Eq. (5-3) is solved and new velocity field is calculated. Here, to capture the turbulence behavior of the flow, I use zero-equation turbulence model.

$$\nu_t = 0.03874LV \tag{5-5}$$

where L is the length of the shortest distance inside the domain from the wall, and V is the local mean velocity (Chen and Xu, 1998; Mortezazadeh and Wang, 2018). Note that to speed up the simulation, the Eqs. (5-1) and (5-4) are solved based on a V-cycle multigrid method (Mortezazadeh and Wang, 2016). Additionally, parallel computing (OpenMP) is used to speed up the computation (Mortezazadeh and Wang, 2018).

5.2.1. FFD method

In the conventional semi-Lagrangian method, the characteristic curve is approximated by assuming a constant slope for the characteristic line (line blue in Figure 5-1). Thus, the characteristic curve is a straight line. But this assumption can cause some errors in estimating the exact position of the departure points because the curvature of the characteristic curves is not considered.

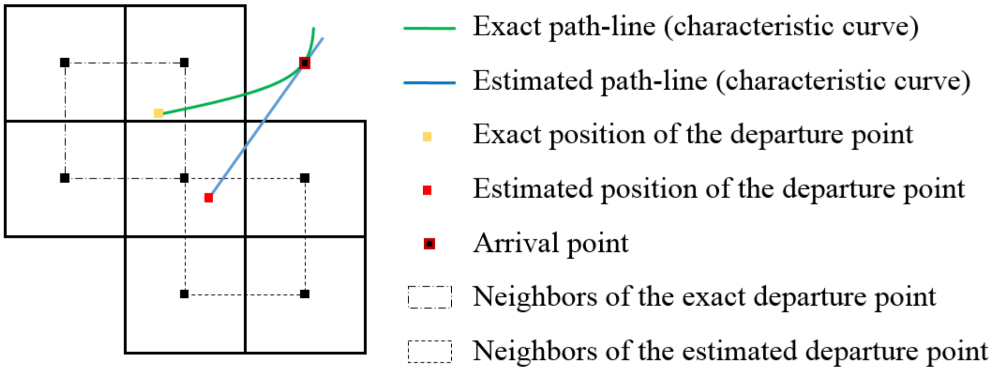

Exact path-line (characteristic curve)
Estimated path-line (characteristic curve)
Exact position of the departure point
Estimated position of the departure point
Arrival point
Neighbors of the exact departure point
Neighbors of the estimated departure point

Figure 5-1 Characteristic curve on an Eulerian computational domain.

In the present work, a new idea based on Kinematics (Beggs, 1983; Whittaker, 1988) is proposed to accurately construct the characteristic curves and estimate the position of the departure points. Using the initial conditions of a particle, including its position, velocity, and acceleration, the path-line can be determined and consequently the characteristic curve of the particles within the time step can be constructed. The velocity and acceleration of the domain play an important role in describing the motion of a particle. The velocity demonstrates the direction and the magnitude of the particle's motion and it equals the first derivative of the path-line, $\frac{d\vec{X}}{dt}$. The acceleration is the change rate of velocity magnitude and its direction, and represents the curvature of the path-line. It is calculated by the second order of derivative, $\frac{d^2\vec{X}}{dt^2}$. In other words, as shown in Figure 5-2, the velocity and acceleration represent the slope and curvature of the path-line, respectively. In the conventional semi-Lagrangian method, the first order of spatial derivative over the time is considered to construct the characteristic curve. However, in the present work, the second order of this derivative is also considered.

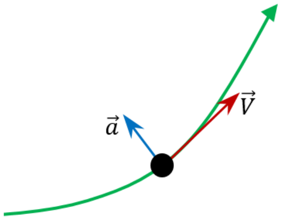


Figure 5-2 Slope and curvature of the path-line.

Advection term, Eq. (5-2), is the nonlinear part of Navier-Stokes equations. In the semi-Lagrangian method, advection equation is written in the Lagrangian perspective:

$$\frac{\partial \emptyset}{\partial t} = -(U \cdot \nabla)\emptyset \qquad \rightarrow \qquad \frac{d\emptyset}{dS} = 0 \tag{5-6}$$

where $\emptyset$ is equal to U in the Navier-Stokes equations, and S is the characteristic curve shown as Eq. (5-7):

$$\frac{dS}{dt} = U \tag{5-7}$$

Note that in the general form, velocity is a function of time and space, $U_{(t,x)}$. In the conventional semi-Lagrangian method, $U_{(t,x)} = U + \boldsymbol{O}(\Delta \boldsymbol{t})$ is considered as a constant value within one time step. So, by using the integral function along the characteristic line within one time step $[n, n+1]$, the characteristic curve can be written as follows:

$$\int_{n}^{n+1} dS = \int_{n}^{n+1} U dt \qquad \rightarrow \qquad S^n \approx S^{n+1} - U\Delta t \tag{5-8}$$

$S^n = (x_d, y_d)$ shows the position of the departure point and $S^{n+1} = (x_a, y_a)$ shows the arrival point. So:

$$x_d \approx x_a - u\Delta t \tag{5-9}$$

$$y_d \approx y_a - v\Delta t \tag{5-10}$$

By combining Eqs. (5-9) and (5-10), the path-line of the arrival particle is constructed:

$$y_d \approx \frac{v}{u}(x_d - x_a) + y_a \tag{5-11}$$

This curve is shown by the blue line in Figure 5-1.

In the present work, I consider the velocity as a function of time and it can be written as $U_{(t,x)} = at + U + \boldsymbol{O}(\Delta \boldsymbol{t}^{2})$ where a is the acceleration of the arrival cell. Then by using the integral function, Eq. (5-7) can be written as:

$$\int_{n}^{n+1} dS = \int_{n}^{n+1} (at + U_0)dt \qquad \rightarrow \qquad S^n \approx S^{n+1} - a\frac{\Delta t^2}{2} - U_0\Delta t \tag{5-12}$$

Now, the position of the departure point can be found by:

$$x_d \approx x_a - a_x\frac{\Delta t^2}{2} - u\Delta t$$

$$y_d \approx y_a - a_y\frac{\Delta t^2}{2} - v\Delta t \tag{5-13}$$

This curve is shown by the green line in Figure 5-1. The accelerations are calculated by:

$$a_x = \frac{u^n - u^{n-1}}{\Delta t} + O(\Delta t^2)$$

$$a_y = \frac{v^n - v^{n-1}}{\Delta t} + O(\Delta t^2) \tag{5-14}$$

So, according to the above procedure, I could improve the accuracy of the conventional semi-Lagrangian method from first order to the second order, and consequently improve the accuracy of FFD.

In the following, I will show the performance of the proposed method for large time step simulation. The proposed high-order temporal approach is also added to the high-order backward forward sweep interpolating scheme (chapter 3) which is a powerful model for coarse grid simulation.

First, a pure advection problem is solved and the performance of the proposed method for estimation of the characteristic curve will be investigated in comparison with the conventional FFD method. Then some real 2-D and 3-D problems will be investigated.

5.3. Results

5.3.1 Pure advection problem

Here, the motion of a 2-D projectile under the influence of gravity is simulated by using the semi-Lagrangian method (the advection solver of FFD). In this problem, the projectile is considered on the floor and it starts to move under an initial condition. The projectile is a 2-D square inside a 2-D domain as shown in Figure 5-3. Here, $\Delta x = \Delta y = 0.01$. The initial condition is:

$$\emptyset = \begin{cases} 1 & 0.2 < x < 0.4 \; and \; 0.2 < y < 0.4 \\ 0 & \text{otherwise} \end{cases} \tag{5-16}$$

and:

$$u_0 = 4, a_x = 0 \tag{5-17}$$

$$v_0 = 4, a_y = g = -10$$

So, based on the proposed method the characteristic curves are:

$$x_d = x_a - 4\Delta t \tag{5-18}$$

$$y_d = y_a + 5\Delta t^2 - 4\Delta t$$

Note that the characteristic curves for the conventional semi-Lagrangian method in this problem, is:

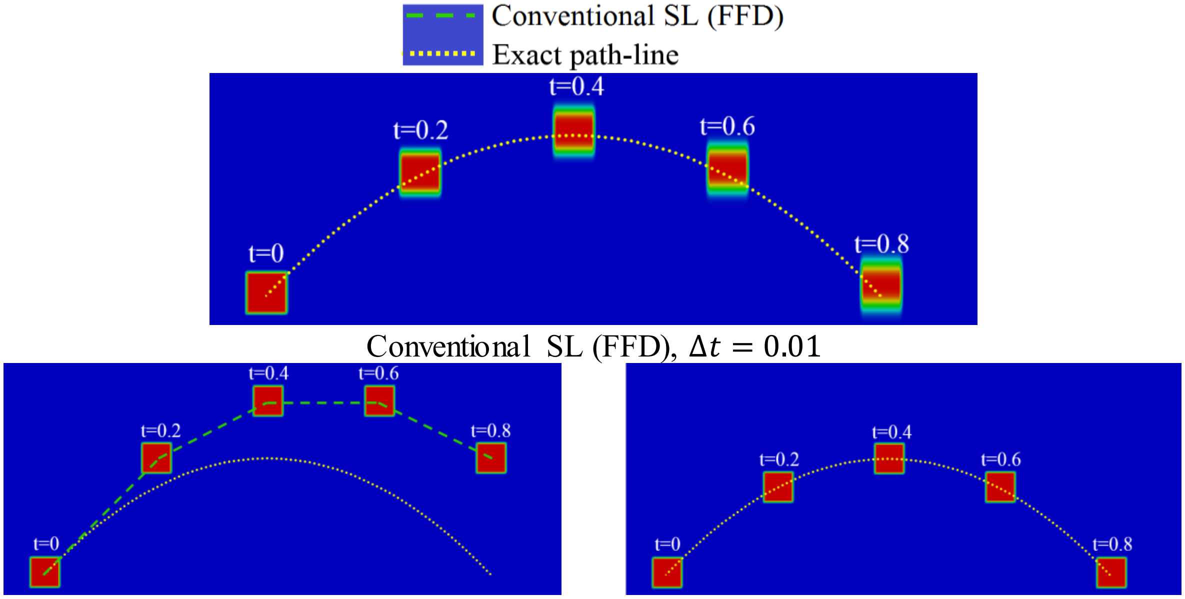
$$x_d = x_a - 4\Delta t \tag{5-19a}$$

$$y_d = y_a - 4\Delta t \tag{5-19b}$$

where $5\Delta t^2$ is missed in Eq. (5-19b). When the time step is small, the effects of the acceleration can be neglected, because the truncation error in $\vec{U}_{(t,x)} = \vec{U} + O(\Delta t)$ is small and also $5\Delta t^2$ is negligible. Figure 5-3(a) shows that for the conventional semi-Lagrangian method the projectile follows the path-line, because of the small time step. One can notice that the diffusion errors are growing during the simulation. So, using very small time step can increase the numerical errors such as diffusion errors. Figures 5-3(b) and 5-3(c) show the results of the conventional semi-

Lagrangian method and the proposed algorithm by using larger time step. Using large time step will reduce both the computational time and the numerical accumulation error. But the large time step can also reduce accuracy of the transient results of the conventional FFD method. When the large time step is used, the assumption of using constant velocity within one-time step will generate large deviation from the exact path-line. But the proposed method can prevent this problem by considering the acceleration and curvature of the particle's movement and consequently accurate estimation of the characteristic curve. Figure 5-3(d) shows the results of very large time step. As you can see, the conventional semi-Lagrangian method literally failed while the proposed algorithm can accurately estimate the new position of the projectile. In this problem, I can easily speed up the simulation, for instance, 8 times faster or even more without any effect on the transient accuracy.

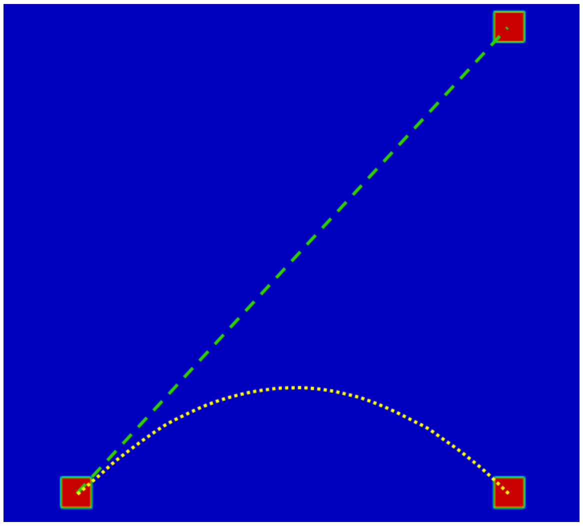
Note that conventional semi-Lagrangian (SL) method is only related to the advection term in the FFD model. In the current section (5.3.1), I only considered the advection term and discussed conventional SL; in the following sections, I will focus on the full Navier-Stokes and FFD model.



Conventional SL (FFD), $\Delta t = 0.01$

Conventional SL (FFD), $\Delta t = 0.2$

Present work, $\Delta t = 0.2$



Conventional SL (FFD) vs Present work, $\Delta t = 0.8$

Figure 5-3 2-D projectile motion, $t = 0.8\ [s]$.

5.3.2 Transient wind flow around a cylinder

In the building and environment problems, such as airflow around buildings, vortex and recirculation regions can occur. Thus, a reliable CFD solver should accurately capture this phenomenon. In this section, a well-known 2-D unsteady problem is simulated to study the performance of the proposed method for capturing the recirculation areas. Figure 5-4 shows the schematic of the 2-D cylinder problem.

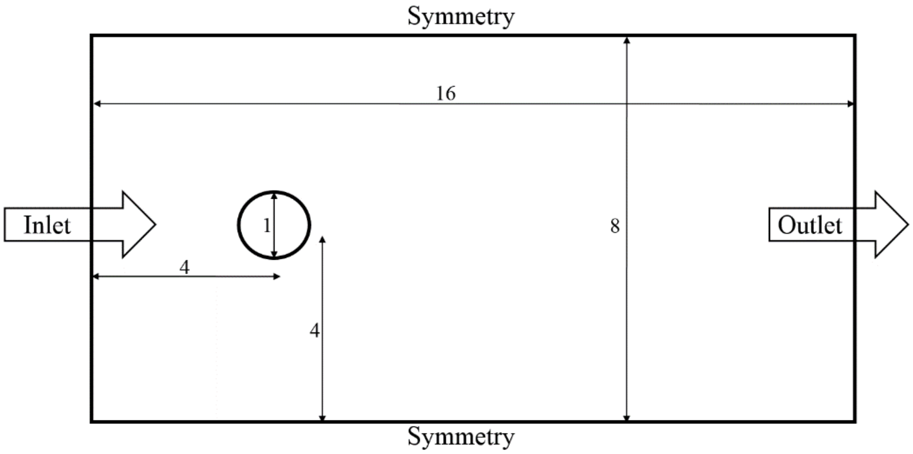
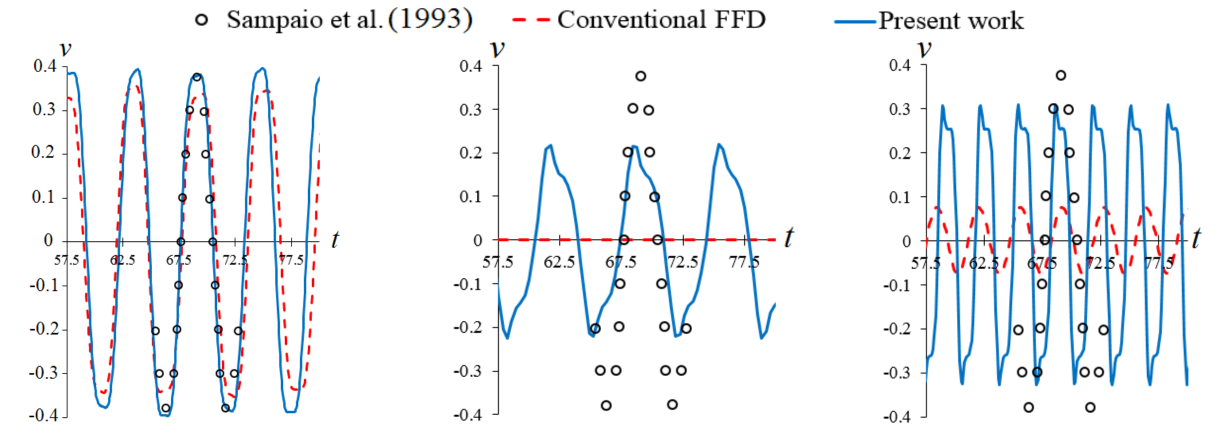


Figure 5-4 Schematic of 2-D cylinder problem

Here, the velocity at the inlet is $u = 1, v = 0$. At the outlet, the pressure is $p = 0$. The Reynolds number is $Re = 100$. Note that the Reynolds number is based on the diameter of the cylinder. In this problem, when the Reynolds number is higher than 40, an oscillating flow occurs behind the cylinder because of the generation of Von Karman vortex street (Panton, 1984). This test case is a complicated benchmark problem because of highly non-uniform airflows and large recirculation region behind the cylinder. Here, the results are compared with Sampaio et al. (1993). Figure 5-5 shows v-velocity at the middle of the outlet for different time and space steps. Here it can be seen that FFD method with conventional semi-Lagrangian approach is able to provide acceptable results by using small time step and fine grids (see Figure 5-5(a)). By using large time step and coarse grids, FFD method generates unreliable and low accurate results. Figure 5-5(b) shows the simulation for the present work and conventional FFD method for the very large time step. As you can see, the conventional FFD method cannot capture transient behavior of the flow and the recirculation area behind the cylinder. A comparison between the results of the present work with $\Delta t = 0.35$ and $\Delta x = \Delta y = 0.08$ and the conventional FFD with $\Delta t = 0.2$ and $\Delta x = \Delta y = 0.16$ shows that I can achieve even better and more accurate results on the coarser grids and larger time step by the proposed method. Figure 5-6 shows the velocity contours of the problem with $\Delta t = 0.35$ and $\Delta x = \Delta y = 0.08$.

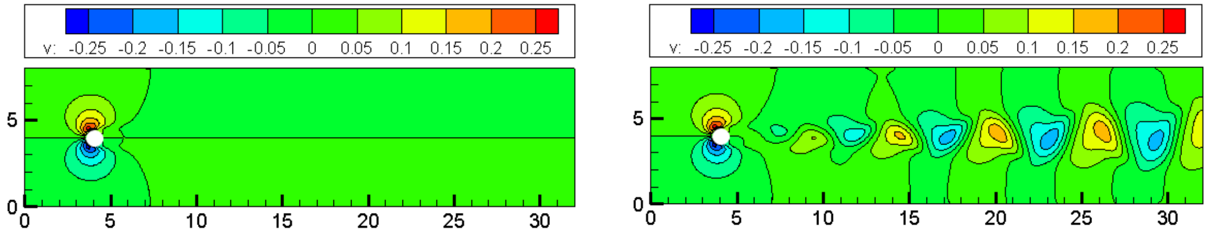


$\Delta t = 0.05, \Delta x = \Delta y = 0.04$
(a)

$\Delta t = 0.35, \Delta x = \Delta y = 0.08$
(b)

$\Delta t = 0.2, \Delta x = \Delta y = 0.16$
(c)

Figure 5-5 v-velocity history at the middle of the outlet, 2-D cylinder test case.



Conventional FFD

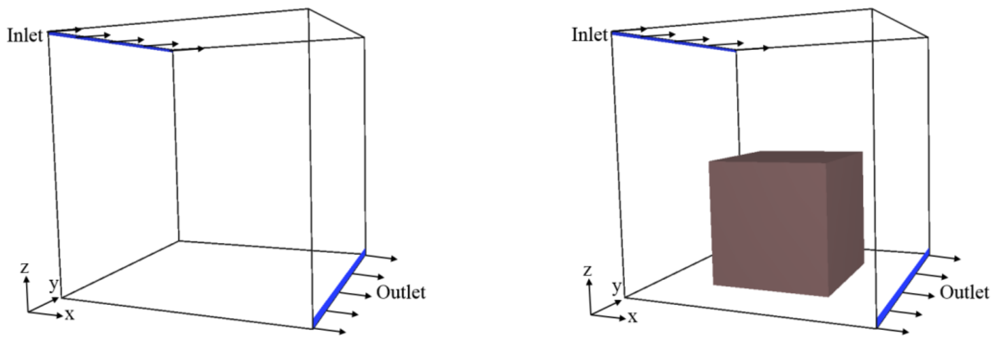
Present work

Figure 5-6 v-velocity contour around the 2-D cylinder, Δt=0.35 and Δx=Δy=0.08.

As it is shown in Figure 5-67, the reader can see the capability of the proposed method to capture the vortex shedding behind the cylinder for the large time step and coarse grids. So, this method is appropriate for the large scales, such as building and environment problems, where using the large time steps and coarse grids are inevitable.

5.3.3 Forced convection airflow in a single room

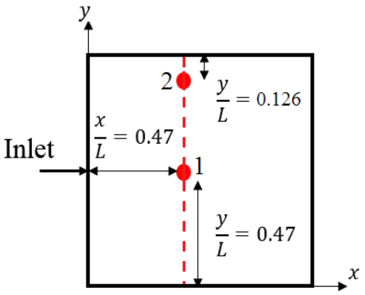
In this section, isothermal forced convection in a single room is simulated to demonstrate the ability of the solver for indoor airflow problems. Figure 5-7 shows the schematic of the test case.



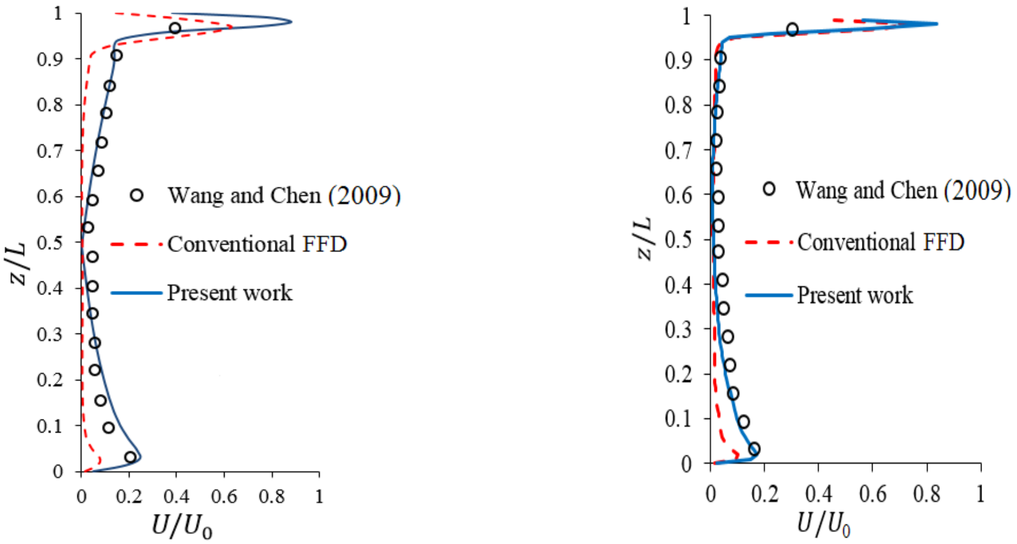
Empty single room Single room with furniture

Figure 5-7 Schematic of the forced convection problem in a room.

Dimensions of room in x, y, and z directions equal to $L = 2.44$ [m]. The height of the inlet and outlet equal to 0.03 [m] and 0.08 [m], respectively. The velocity of supply air is $U_0 = 0.455$ [m/s]. For the other faces of the room and also the block inside the room, I will consider the non-slip boundary condition. In these cases, a set of non-uniform grids generated by CFD0-Editor is used to provide finer grids near the inlet and the room walls. For the room without and with the furniture, the total numbers of grids are 64000 and 316875, respectively. The larger numbers of grids are used for the second case because of using finer grids near the furniture inside the room. In this problem for both cases, the time step is $\Delta t = 1.0$. Figures 5-8(a) and 5-8(b) show the conventional FFD is not able to correctly capture the recirculation area. But the proposed method is obviously able to simulate the complicated phenomenon like a big recirculation domain inside the single room.



Position of plotted result and velocity contour

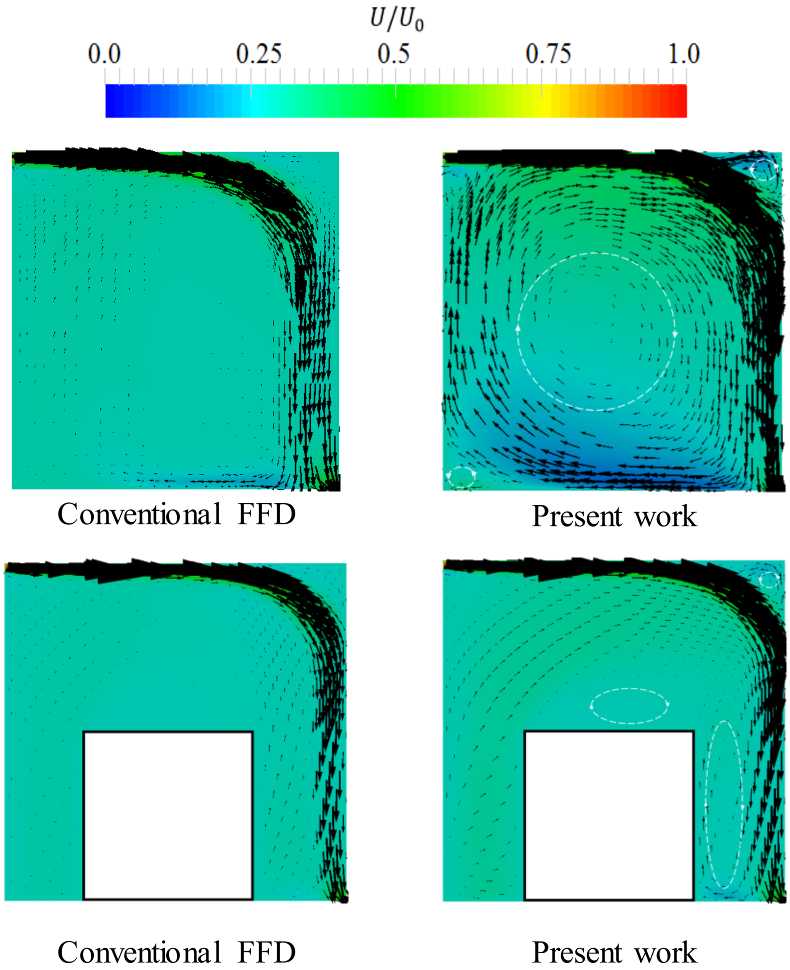


(a) Velocity profiles at Point 1 (Empty single room)

(b) Velocity profiles at Point 2 (Single room with furniture)

Figure 5-8 Velocity profiles along a vertical line at the marked points.

Figure 5-9 shows the velocity vector distribution inside the room. Here the reader can see the proposed method is able to accurately capture the recirculation area inside the single room with no furniture and also the small recirculation area behind the furniture in the room with the block.



Conventional FFD

Present work

Conventional FFD

Present work

Figure 5-9 Velocity vectors and contours at the vertical surface x-z at the center of the room.

5.3.4 *Airflow simulation in a step-down street canyon*

In this section, airflow around a step-down street canyon is simulated. Building engineers are interested in studying aerodynamics of airflow in urban areas because of many concerns, such as air quality, pedestrian comfort, energy consumption of the buildings, and contaminant dispersion (Blocken et al., 2012; Toparlar et al., 2015; Addepalli and Pardyjak, 2015; Jandaghian et al., 2018). In this test case, the upwind building height is greater than the downwind building height, so it can affect the airflow inside the street canyon. Here, H, $L = 0.27H$, and $W = 0.27H$ are the dimensions of the buildings and $S = 0.7H$ is the width of the street. $Re = 53680$, which is based on the height of the tallest building. Velocity at the inlet follows the power law and equals $u_{(z)} = u_{\infty}(\frac{z}{H})^{0.205}$, where u_{∞} is the reference velocity. Figure 5-10 shows the geometry of the problem.

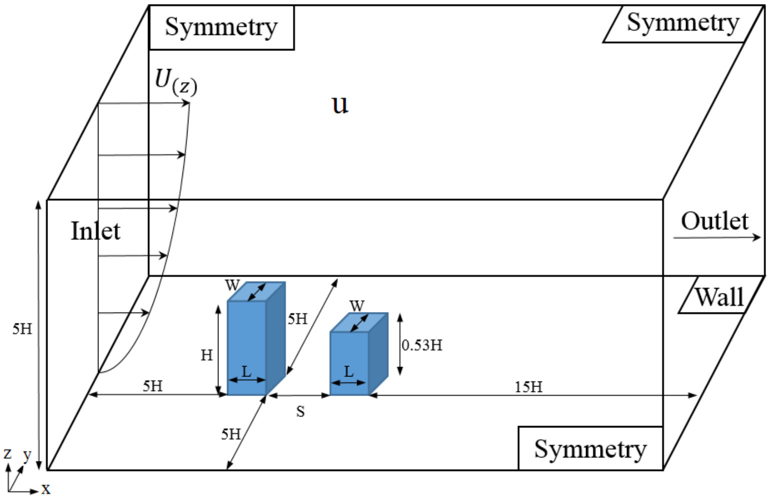


Figure 5-10 Schematic of step-down canyon problem.

The short building will be located inside the recirculation area behind the tall building. Figure 5-11(a) shows the streamline of the airflow around the buildings. As you can see, there are some vortex regions inside the street canyon. Figure 5-11(b) shows a comparison between the conventional FFD, present work, and Addepalli and Pardyjak (2015). Results indicate that the proposed method can significantly improve the accuracy of the conventional FFD method. In this test case, the number of girds on each face of the building is 8 which is less than the minimum grid resolution (1/10) based on the AIJ guidelines (Tominaga et al., 2008). So, the proposed method is recommended for modeling the outdoor airflow in urban area. The computational time is around 30 minutes.

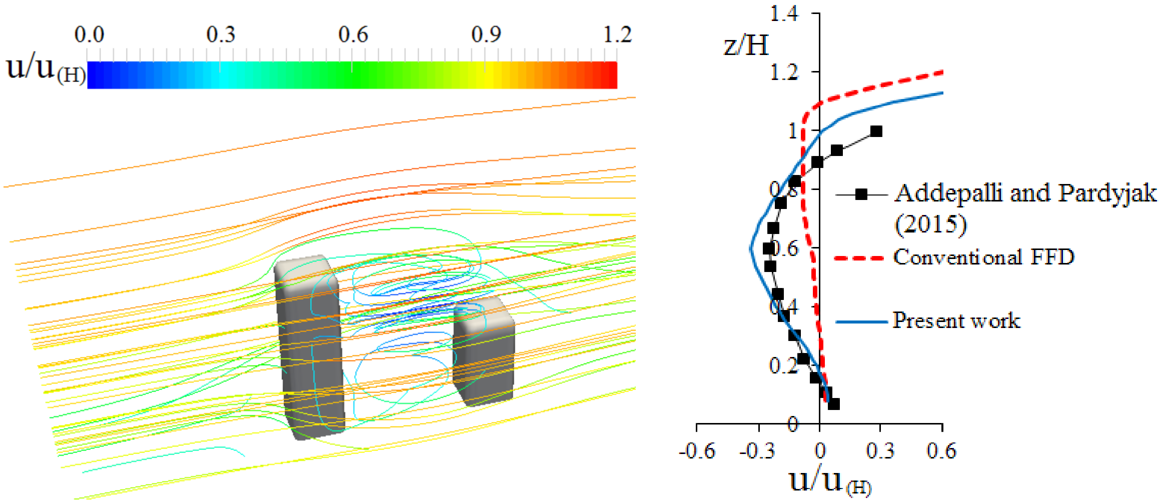


Figure 5-11 Step-down canyon problem: (a) Streamline around the buildings (Present work), (b) Comparison of mean stream-wise x-velocity along the vertical profile in the middle of street canyon.

5.4. Summary and Conclusion

In the present work, a new solver based on the semi-Lagrangian approach is proposed to accurately simulate airflow problems for the large time steps and coarse grids. In the conventional semi-Lagrangian method, the curvatures of the characteristic curves are not considered. This causes the approximation of the departure points not to be accurate, especially when the time step is large and the flow field is highly non-uniform. Based on the proposed method, the curvatures are estimated by calculating the acceleration of the arrival cells. Additionally, to reduce the numerical error, especially on the coarse grids, I used high-order backward forward interpolation scheme. The effectiveness of the proposed method is more significant when the large time steps and coarse grids are used.

Chapter 6 SLAC – a semi-Lagrangian artificial compressibility solver for steady-state incompressible flows

The contents of this chapter have been published in "Mortezazadeh, M. and Wang, L., 2019. SLAC–a semi-Lagrangian artificial compressibility solver for steady-state incompressible flows. International Journal of Numerical Methods for Heat & Fluid Flow." . The contents are slightly modified.

The semi-Lagrangian-based solvers are typically pressure-based, i.e. semi-Lagrangian pressure-based (SLPB) solvers, where a Poisson equation is solved for obtaining the pressure field and ensuring a divergence-free flow field. As an elliptic-type equation, the Poisson equation often relies on an iterative solution, so it can create a challenge of parallel computing and a bottleneck of computing speed. This study proposes a new density-based semi-Lagrangian method, i.e. the semi-Lagrangian artificial compressibility (SLAC), which replaces the Poisson equation by a hyperbolic continuity equation with an added artificial compressibility term, so a time-marching solution is possible. Without the Poisson equation, the proposed SLAC solver is faster, particularly for the cases with more computational cells, and better suited for parallel computing. The study compares the accuracy and the computing speeds of both SLPB and SLAC solvers for the lid-driven cavity flow and the step-flow problems. It shows that the proposed SLAC solver is able to achieve the same results as the SLPB whereas with a 3.03-times speedup before using the OpenMP parallelization, and a 3.35 times speedup for the large grid number case (512×512) after the parallelization. The speedup can be improved further for larger cases because of increasing the condition number of the coefficient matrixes of the Poisson equation.

6.1. Introduction

Conventional FFD method is a pressure-based type of solver (PB), originally developed by Harlow and Welch (1965) for unsteady flow problems and then later extended by Patankar and Spalding (1972) for steady-state calculations. In these methods, computations of the velocity and the pressure fields are decoupled (Blazek, 2005). Here, the pressure domain is calculated by solving the Poisson equation and then the velocity is obtained by the pressure domains. On irregular grids, solving the Poisson equation is not straightforward (Min and Gibou, 2006). In addition, solving the Poisson equation often needs iterative solvers, making it a real challenge for parallelizing semi-Lagrangian-based solvers on modern high-performance computing infrastructures, such as OpenMP and/or GPU, especially for large data cases (e.g. tens of millions of grids), which may only be solved in a parallel manner. In comparison, density-based (DB) solvers offer another way of solving fluid flows without the need of Poisson equation for pressure, e.g. the artificial compressibility (AC) based solver for steady-state incompressible flow problems (Kwak et al., 2005). The AC method was developed by Chorin (Chorin, 1997) and has been used by many researchers in different areas (Ramshaw and Mousseau, 1990; Tamamidis et al., 1996; Hejranfar et al., 2009; Degroote et al., 2010). By defining an artificial compressibility coefficient, the continuity equation is transformed to a hyperbolic equation instead of the elliptic one for PB methods, and can be relatively easily solved by marching in the pseudo-time until the divergence-free velocity field is satisfied at the steady state. Therefore, the Poisson equation of pressure is not required. Additionally, the continuity equation in AC based method is naturally parallelizable and perfect for GPU computing applications.

This study proposed the SLAC – a new Semi-Lagrangian Artificial Compressibility solver for steady-state incompressible flows to accelerate the convergence of the semi-Lagrangian schemes.

This acceleration is achieved by preventing the need for using iterative solvers, such as Gauss-Seidel, to calculate the pressure fields. Note that here, although both the semi-Lagrangian-based fluid solver and the artificial compressibility model are existing methods, the novelty of this study is the combination of these two methods for facilitating a better parallel computing process, which has not been done by previous studies to the knowledge of the authors. In this study, diffusion terms are solved by a three-level V-cycle multigrid method to speed up convergence. The interpolation scheme for the semi-Lagrangian method is the linear interpolation to reduce the computational cost. However, the proposed method is not limited to linear interpolation scheme but applicable for high-order schemes (Mortezazadeh and Wang, 2017). Additionally, the advection term and also the pressure domain in the proposed method are solved by using a parallel programming code. Here, the benefit of using artificial compressibility methods in comparison with pressure-based methods is shown by two well-known problems, i.e. cavity driven flow and step flow problems. This chapter starts by introducing the methodology and the numerical procedure in the second section, followed by the numerical results for the two cases in the third section. Then, a discussion section follows about the convergence speed, accuracy-related parameters and potential paralleling application of the proposed method. Finally, a conclusion of the present work is presented.

6.2. Methodology

As mentioned in the previous section, there are two main categories to solve the incompressible Navier-Stokes equations, named pressure-based (PB) method and density-based method, i.e. artificial compressibility (AC) method. Semi-Lagrangian pressure-based (SLPB) method has comprehensively explained in section 2.2.1. In the following, details of SLAC algorithm is presented.

6.2.1. Semi-Lagrangian artificial compressibility method

Artificial compressibility (AC) method is based on the adaption of the compressible time-marching algorithms to low Mach numbers (Reboud et al., 2003). Using AC method for unsteady problems needs to implicitly solve time derivative terms. For example, an implicit procedure of dual time-stepping algorithm was developed for transient problems (Merkle, 1987; Reboud et al., 2003). This chapter focuses on applying the AC method to semi-Lagrangian-based fluid solvers at steady state.

For the SLAC method, Eq. (6-1) is changed by adding the artificial compressibility term:

$$\frac{1}{\rho}\frac{\partial \rho}{\partial t} + \nabla \cdot U = 0 \tag{6-1}$$

At the steady state, the additional AC term will vanish, so the steady-state results are the same as those of Eq. (2-1). The artificial compressibility coefficient, or the artificial sound speed, $\beta > 0$, is defined by:

$$p = \beta^2 \rho \tag{6-2}$$

The artificial compressibility coefficient determines the stability and convergence rate of the AC scheme (Turkel, 1987; Malan et al., 2002; Esfahanian et al., 2012), and its choice is often case dependent. For example, for an inviscid and high Reynolds flow problem, Turkel (1987) showed that if β was close to the local convective velocity, the AC simulation achieved a better convergence rate. Malan et al. (2002) pointed out the following two considerations when choosing the value of β: 1). The Mach number associated with β should be kept below unity; and 2). The value of β should be considered with a distinction between convective dominated and diffusion dominated regions: leading to considering β differently for the convection and diffusion problems. Eq. (6-3) shows that the selection of β may also be related to the time step. The transient term, or the storage term will vanish for a steady-state solution, i.e. the steady-state solution should be unique for different combinations of β and time step, as long as the simulation converges. However, the proper selection of both terms may affect the simulation path to reach the convergence, e.g. from a numerical point of view, both terms act like certain levels of relaxation terms, although both terms have their own physical implications. For achieving a convergent simulation, choosing both terms can be problem dependent as also mentioned by the previous studies (Turkel, 1987; Malan et al., 2002), and similar to choosing relaxation factors, multiple trials sometimes may be needed. A general advice from this study is to choose β to be less than or close to one. More details are discussed in the later sections of the case studies.

Combing Eqs. (6-1) and (6-2), I obtain:

$$\sigma\frac{\partial p}{\partial t}+\nabla\cdot U=0 \tag{6-3}$$

where $\sigma=\frac{1}{\rho\beta^2}$. Here, Eq. (6-3) is turned into the pressure equation by the inclusions of the AC and pressure terms so there is no need for the Poisson equation as required by the SLPB method. Meanwhile, the removal of the Poisson equation will need the pressure gradient term kept in Eq. (2-3). The semi-Lagrangian method and the FSM solution procedure are applied as similarly as that in the SLPB method. In summary, the proposed SLAC method is as follows:

$$\frac{\partial U}{\partial t}=-\nabla p+\frac{1}{Re}\nabla^2 U+f \tag{6-4}$$

$$\frac{\partial U}{\partial t}+(U\cdot\nabla)U=0 \tag{6-5}$$

$$\frac{1}{\rho\beta^2}\frac{\partial p}{\partial t}+\nabla\cdot U=0 \tag{6-6}$$

As a result, for the pressure domain, the original elliptic-type of equations is transformed into a system of hyperbolic equations, which can be solved relatively easily by marching in time. Additionally, the solution of Eq. (6-6) does not need iterations, perfect for parallel computing.

For a better comparison, Table 6-1 compares the conventional SLPB and the proposed SLAC methods by different steps and whether numerical iterations are needed. It shows that the SLAC only needs three steps and one equation (Eq. 6-4) needs an iterative solution. Therefore, fewer arithmetic operations are needed by the SLAC method than the SLPB method.

Table 6-1 Comparison of the conventional semi-Lagrangian pressure-based (SLPB) method and the proposed semi-Lagrangian Artificial Compressibility (SLAC) methods.

Steps	Semi-Lagrangian Pressure-Based (SLPB)	Need Iteration	Semi-Lagrangian Artificial Compressiblity (SLAC)	Need Iteration
Step 1	$\frac{\partial U}{\partial t} = \frac{1}{Re}\nabla^2 U + f$ (2-3)	Yes	$\frac{\partial U}{\partial t} = -\nabla p + \frac{1}{Re}\nabla^2 U + f$ (6-4)	Yes
Step 2	$\frac{\partial U}{\partial t} + (U \cdot \nabla)U = 0$ (2-4)	No	$\frac{\partial U}{\partial t} + (U \cdot \nabla)U = 0$ (6-5)	No
Step 3	$\nabla^2 p = \frac{1}{\Delta t}\nabla . U$ (2-5)	Yes	$\frac{1}{\rho\beta^2}\frac{\partial p}{\partial t} + \nabla \cdot U = 0$ (6-6)	No
Step 4	$\frac{\partial U}{\partial t} = -\nabla p$ (2-6)	No		

In the present work, the conventional semi-Lagrangian method with linear interpolation scheme is used to solve the equation of advection term in the Navier-Stokes equations, i.e. Eq. (2-4) in the SLPB, or Eq. (6-5) in the SLAC. Note that the proposed SLAC is not limited to the linear scheme but applicable to higher-order schemes.

6.3. Results

In this section, two benchmark problems for incompressible flows, i.e. cavity driven and step-flow problems, are employed to investigate the performance of the proposed SLAC method in terms of accuracy and computing speed compared to the conventional SLPB method.

6.3.1. Lid-driven cavity problem

In a 2-D isothermal lid-driven cavity problem (Wang et al., 2017), a unit-length square is set up with the top lid moving at a constant speed of $U_x = 1$, and all other faces of the square are considered as no-slip walls as shown in Figure 6-2. The laminar airflow is modeled for two different Reynolds numbers, i.e. Re = 1,000 and 5,000. Different grid resolutions were tested from 100×100 as the coarsest grid case to 500×500 for the finest. All the simulations were performed on a PC system with the Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz and 12 GB RAM.

In this test case, different artificial compressibility numbers are tested: β = 0.2, 0.4, 0.6, 0.8, and 1.0, and the same flow results are obtained. Figure 6-1 shows the velocity streamlines for $Re = 5{,}000$: three recirculation areas are formed due to the movement of the top lid with one central large region, and two other smaller ones at the lower corners, showing that the SLAC method is able to capture the recirculation at different scales. Note that the convergence criterion is when the infinite norm is less than 10^{-5}, the simulation is considered to reach the steady state:

$$E_{L_\infty} = \frac{\max|u^{n+1} - u^n|}{\max|u^n|} \tag{6-7}$$

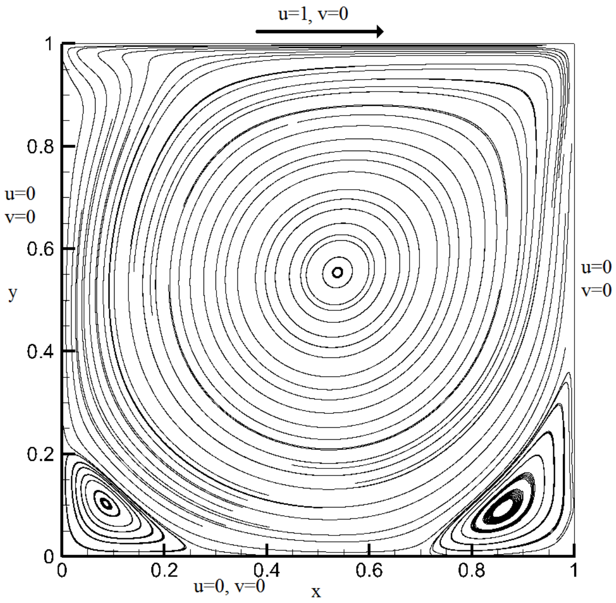
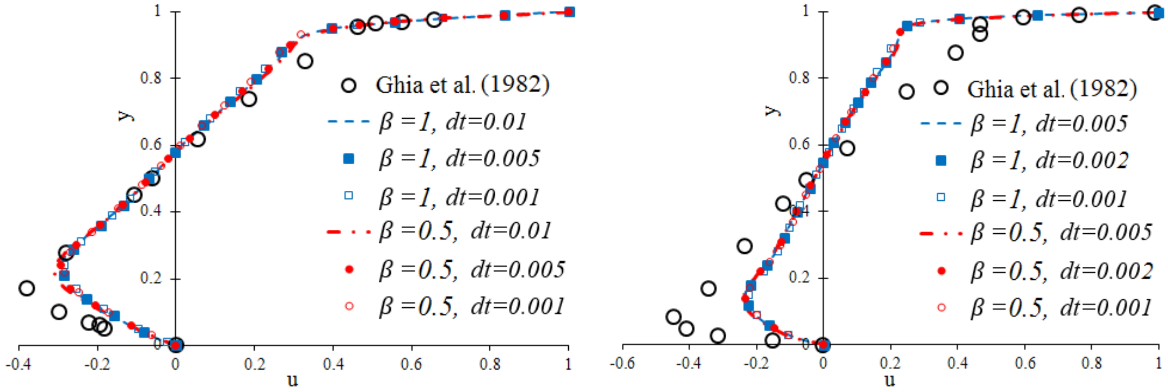


Figure 6-1 Calculated streamlines of the 2-D lid-driven cavity problem for $Re = 5{,}000$ using the proposed SLAC method.



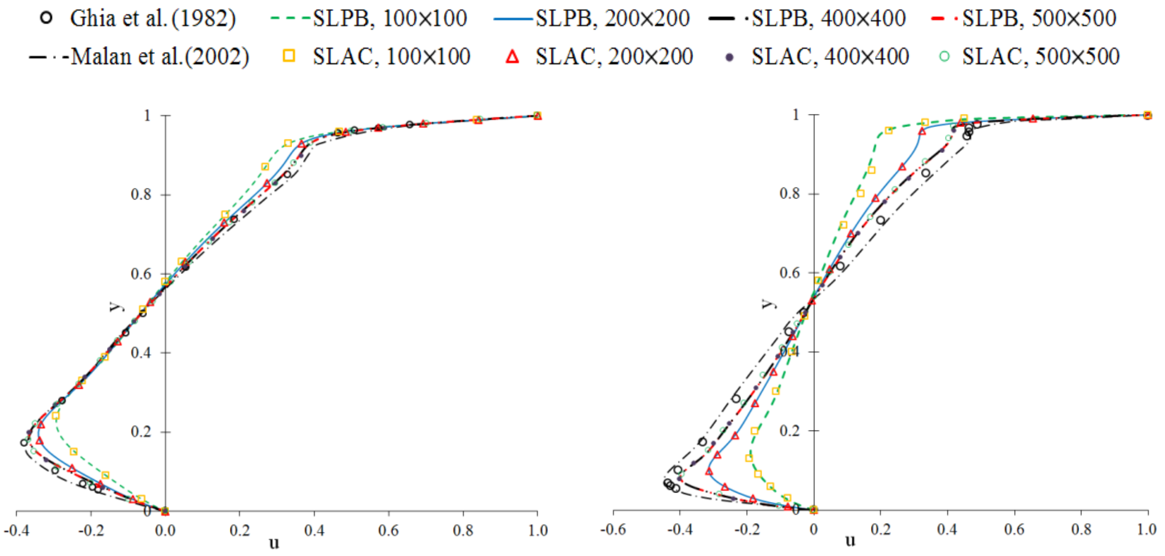
(a) $Re = 1{,}000$

(b) $Re = 5{,}000$

Figure 6-2 Comparing different time step dt and β for the lid-driven cavity problem with 100×100 grids when using the SLAC method.

Figure 6-2 shows that different values of β and dt of the SLAC method can reach the same steady-state results. For this problem, as suggested by the previous study (Malan et al., 2002) and also found in this study, the value of β was chosen to be equal to or less than one, and a value more than one could cause convergence issues. Comparing Figure 6-2(a) and Figure 6-2(b), a higher *Re* number seems to need a smaller time step for the same value of β to ensure convergence, which may be related to the fact that the increase of local advection velocity may need a smaller time step to capture the information transferred between the departure and arrival points. Therefore, the selection of β should be combined with the consideration for selecting the time step dt, both of which have been noted as affecting the convergence rate and being case dependent by the previous studies (Turkel, 1987; Malan et al., 2002). A good start would be to use $\beta \leqslant 1$, e.g. 0.5, and an appropriate value of dt, e.g. 0.01.

The steady-state results in Figure 6-2 are consistent among different β and dt, which however may still be improved when compared to the measurement results. Figure 6-3 illustrates that this can be achieved by increasing the grid resolutions for both cases: when doubling the grids in each coordinate directions, the accuracy improves significantly. For this problem, the grid-independent results can be achieved for the grid number beyond 400×400. Meanwhile, the accuracy may be further improved by adopting higher-order spatial schemes for the semi-Lagrangian methods as explained in my previous study (Mortezazadeh and Wang, 2017). Since the current study focuses on the comparison between SLPB and SLAC methods, I will not dwell on how to improve the simulations, for which readers with interests can refer to my previous study. Here, for the SLAC method, it is important to check if the simulations converge to the same results as the SLPB method at the steady state. In Figure 6-3, a closer observation shows both SLAC and SLPB achieved the same results for the x-velocity component, u, along the vertical center line in the y-direction of the domain. Both results are also close to those from the previous experimental and numerical studies by Ghia et al. (1982) and Malan et al. (2002).



(a) $Re = 1{,}000, dt = 0.01.$ (b) $Re = 5{,}000, dt = 0.005.$

Figure 6-3 Lid-driven cavity flow results for the coarse and fine grids by the SLPB and SLAC ($\beta = 1$) methods compared to the previous studies.

To investigate the computational cost, e.g., for $Re = 1{,}000$, the number of iterations and the computing time for the SLPB is 4,200 and 384 [s], respectively for the 200 × 200 grids, when compared to 4,500 iterations with the computing time of 127 [s] for the SLAC. So, the number of iterations for both methods are about the same but the SLAC is about 3.02 times faster than the SLPB. A more detailed discussion about the computing cost is provided in the later section.

6.3.2. Step-flow problem

The lid-driven cavity problem is a case with enclosed computational domain. Here, I chose to study the step-flow problem (Armaly et al., 1983) with an open domain for three different Reynolds numbers. Figure 6-4 shows the schematic of the step-flow problem in a channel. Here, H is equal to 2, and the length of the channel is 20, which is long enough to allow for a free outflow at the right side of the domain. S is the half of H. Here, the inlet flow, coming from a long channel, is fully-developed with the profile defined by (Zuo et al., 2012):

$$U = \begin{cases} u = -6(y-S)^2 + 6(y-S) \\ v = 0 \end{cases} \tag{6-8}$$

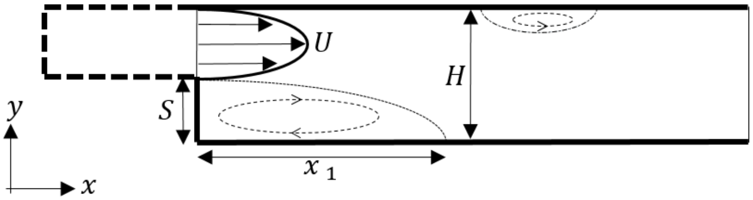


Figure 6-4 Schematic of the step-flow inside a 2-D channel.

Figure 6-5 shows the calculated streamline results for both SLPB (upper) and the SLAC (lower) methods at three different Reynolds numbers. A greater *Re* number results in a longer length of the recirculation zone. In the meantime, the streamline results of SLPB are pretty close to those of SLAC for different Re numbers.

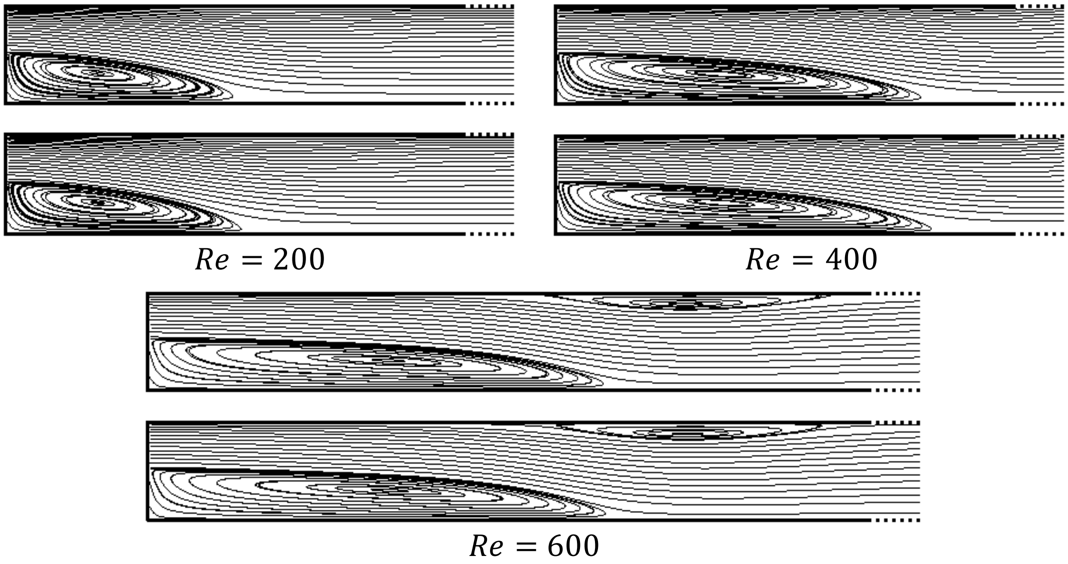


Figure 6-5 Calculated streamlines for the step-flow problem by the SLPB (upper) and SLAC (lower) ($\beta = 1.0$).

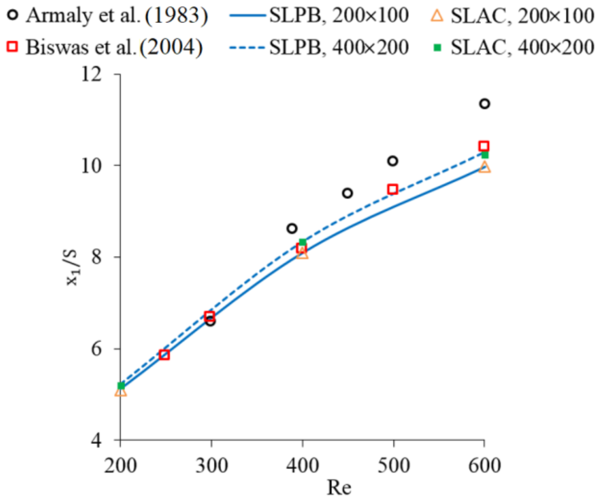


Figure 6-6 The dimensionless recirculation length x_1/s for different grid resolutions when compared to the previous studies.

Figure 6-6 shows the results in comparison with the previous studies (Armaly et al., 1983; Biswas, et al., 2004). Armaly et al. (1983) measured the velocity distribution and reattachment length by using Laser-Doppler measurements in Reynolds number range of 70<Re<8,000. They found that the flow characteristic is laminar when Re<1,200, and also is mostly 2-D when Re<400. So the increasing discrepancy with the Reynolds number in the figure could be attributed to the increasing 3-D behavior of the flow. In this problem, the time step is dt = 0.01, for which the artificial compressibility coefficient of β=1.0 is suitable. The convergence of a higher value of β (e.g. β = 2.0) may be possible but a generally consistent convergence cannot always be guaranteed for this problem: a suitable time step must be chosen accordingly. Once again, choosing the value of β is problem-dependent and sometimes, multiple trials should be conducted for better and faster convergence. An β value less than or equal to one is again advised here. The SLAC results are also compared with the numerical simulations by Biswas et al. (2004), and it shows a better agreement with their work. Table 6-2 shows the computational time for the SLPB and SLAC methods: at best, the latter is about 2.6 times (Re=400) faster than the former for the step-flow problem.

Table 6-2 Computational time comparison between SLPB and SLAC methods in the step-flow problem.

Grid Size	Reynolds	Computational time [s]	
		SLPB	SLAC
$dx = 0.1$ $dy = 0.02$	200	459	214
	400	429	163
	600	415	164

6.4. Discussions

6.4.1. Convergence evaluation

Here, I investigate the computing cost due to solving the Poisson equation for the SLPB method. The condition number is an important parameter for evaluating convergences for a linear iterative solver. A small condition number represents a well-posed problem and thus a better convergence (Pyzara et al., 2011), and in comparison, a large condition number is for an ill-posed and slow converging solution.

For solving a linear system:

$$A_{matrix} x_{unknown} = B, \ \det(A_{matrix}) \neq 0 \tag{6-9}$$

Here, for simplicity I consider $A_{matrix} = A$. The condition number of Eq. (6-9) is defined by:

$$cond(A) = ||A|| \cdot ||A^{-1}|| \tag{6-10}$$

where $||A||$ is the norm of A and A^{-1} is the inverse of A. There exists a connection between the condition number and the matrix size (Pyzara et al., 2011). If the matrix is well-posed, increasing the matrix size will not affect the condition number significantly so a well-posed CFD problem tends to maintain a convergence rate even with large grid numbers. In comparison, an ill-posed problem is often associated with slow convergence, especially on finer grids.

Here, taking the 2-D lid-driven cavity flow as an example, I investigate the condition numbers and their association with the convergences of the Poisson equation and the diffusion equation. The diffusion equation and the Poisson equation can be discretized as follows:

$$\frac{U_{i,j}^{*}-U_{i,j}^{n}}{\Delta t}=\frac{1}{Re}\left[\frac{U_{i+1,j}^{*}-2U_{i,j}^{*}+U_{i-1,j}^{*}}{\Delta x^{2}}+\frac{U_{i,j+1}^{*}-2U_{i,j}^{*}+U_{i,j-1}^{*}}{\Delta y^{2}}\right]+RHS \tag{6-11a}$$

$$\frac{p_{i+1,j}^{*}-2p_{i,j}^{*}+p_{i-1,j}^{*}}{\Delta x^{2}}+\frac{p_{i,j+1}^{*}-2p_{i,j}^{*}+p_{i,j-1}^{*}}{\Delta y^{2}}=\frac{1}{\Delta t}\frac{\partial U_{i}^{**}}{\partial x_{i}} \tag{6-11b}$$

Where the 2nd-order central differencing and implicit method are implicitly solved. Here * means the intermediate values. Note that for simplicity, in Eq. (6-11a), the RHS term includes f_i for the SLPB, and $-\nabla p+f_i$ for the SLAC. Here, the matrix coefficient A for the 2-D case can be written as:

$$A=\begin{bmatrix} a & b & 0 & 0 & \dots & 0 & c & 0 & \dots \\ b & a & b & 0 & \dots & \dots & 0 & c & 0 \\ 0 & b & a & b & 0 & \dots & \dots & \dots & c \\ 0 & 0 & b & a & b & 0 & 0 & 0 & 0 \\ \vdots & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 \\ c & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & c & 0 & 0 & 0 & 0 & \ddots & \ddots & \ddots \\ \vdots & 0 & c & 0 & 0 & 0 & 0 & b & a \end{bmatrix} \tag{6-12}$$

where the values of a, b, and c for the diffusion equation Eq. (6-11a) are:

$$a=\frac{1}{\Delta t}+\frac{2}{Re}\left[\frac{1}{Re\Delta x^{2}}+\frac{1}{Re\Delta y^{2}}\right],\quad b=-\frac{1}{Re\Delta x^{2}},\quad c=-\frac{1}{Re\Delta y^{2}} \tag{6-13}$$

and the coefficients for the Poisson equation Eq. (11b) are:

$$a=\frac{2}{Re}\left[\frac{1}{Re\Delta x^{2}}+\frac{1}{Re\Delta y^{2}}\right],\qquad b=-\frac{1}{Re\Delta x^{2}},\quad c=-\frac{1}{Re\Delta y^{2}} \tag{6-14}$$

To calculate the condition number for each equation, the following settings are used: a uniform grid of $\Delta x = 0.01$; a coefficient matrix size of $A\!: 20 \times 20$; the time step of $\Delta t = 0.01$, and $Re = 1000$. The corresponding condition number of the coefficient matrix is then

$$cond(A)_{\text{diffusin}} = 1.3947, \quad cond(A)_{\text{Poisson equation}} = 178.0643 \tag{6-15}$$

So, the condition number of the Poisson equation is much greater than the diffusion term. If the size of the matrix increases for $A\!: 100 \times 100$, the condition numbers become:

$$cond(A)_{\text{diffusion}} = 1.3998, \quad cond(A)_{\text{Poisson equation}} = 4.13 \times 10^3 \tag{6-16}$$

Therefore, the condition number of the Poisson equation increases drastically with the size of the problem as a result of the ill-posed problem. Consequently, the convergence rate of the iterative solver is reduced significantly.

As explained previously, the major computing bottleneck of the SLPB method is the Poisson equation and the strength of the SLAC method is the absence of the Poisson equation so a faster simulation may be possible. Figure 6-7 shows the effect of increasing the number of grids on the computing cost. As expected, with the increase of grid numbers and consequently the size of the coefficient matrix, the convergence rate of solving the Poisson equation worsens and the computational time significantly increases. When using the SLAC method, the speedup factor ($\frac{t_{SLPB}}{t_{SLAC}}$) increases with the grid numbers. The speedup can be further improved by applying parallel computing techniques to the SLAC method, which is discussed later.

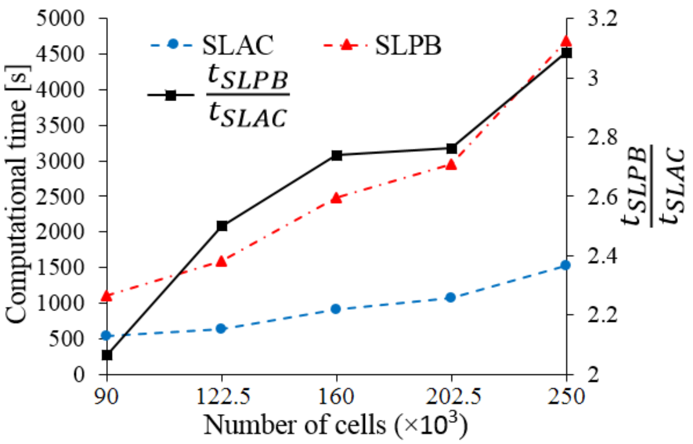
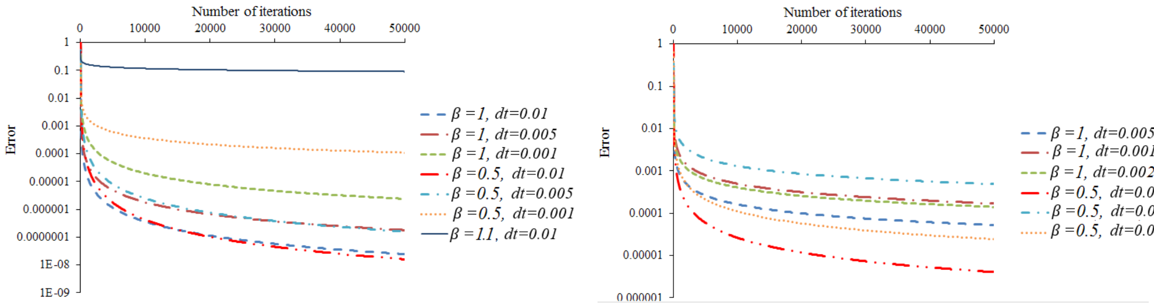


Figure 6-7 Effects of increasing the grid resolutions on computing costs and speedups for the lid-driven cavity problem with $Re = 1{,}000$.



$Re = 1{,}000$ $Re = 5{,}000$

Figure 6-8 Convergence for different β for the lid-driven cavity flow problem with 100×100 grid.

6.4.2. *Effect of artificial compressibility*

In this section, the effect of different values of β on the convergence rate in terms of mass conservation is investigated. As mentioned previously, the selection of β value is case dependent and sometimes several trials should be conducted to find the most suitable one in terms of convergence rate. For the lid-driven cavity flow problem, Figure 6-8 shows the effect of different values of β and dt on the convergence for the lid-driven cavity problem. Here, the convergence can be achieved when β is smaller than or equal to one. Additionally, larger time step seems to be associated with convergence speed. For $Re = 1{,}000$, if $\beta = 1.0$ and $dt = 0.01$, a faster convergence is achieved when compared to the other setups, and for $Re = 5000$, a combination of $\beta = 0.5$ and $dt = 0.005$ shows a better performance. Therefore, when considering the effects of both β and dt, a general advice is to choose a larger dt and $\beta \leq 1$ as a start trial for better and faster convergence. A large value of β may cause divergence, e.g. for $Re = 1{,}000$ and $dt = 0.01$, the setting of $\beta = 1.1$ causes the simulation to diverge. In the meantime, once reaching the final steady state, all the simulations with different β values in regard to the same convergence criterion will achieve the same and unique results as discussed previously.

6.4.3. Effect of parallel semi-Lagrangian artificial compressibility

To compare the parallel computing performance of the SLAC with the SLPB, I implemented the parallel versions of both solvers by using the OpenMP library (OpenMP, 2015). OpenMP is a well-known application programming interface (API) for parallelizing a computer program through the Hyper-Threading Technology, with which independent work or mathematic operations can be assigned to all accessible CPU threads. Here, I calculated the same problem of the lid-driven cavity flow with $Re = 1000$, $dx = dy = 1/512$, $dt = 0.004\ s$ using the OpenMP versions of both SLPB and SLAC methods. All the simulations were conducted on the PC system with 12 GB RAM and the Intel(R) Core (TM) i7-4790 CPU @ 3.60GHz, which has with 4 CPU cores and 8 threads. For evaluating the parallel computing speedup, the following equation from Amdahl's law (Scott et al., 2005) was used :

$$S_{PC} = \frac{1}{(1 - P_{solver}) + \frac{P_{solver}}{C_{PC}}} \tag{6-17}$$

where S_{PC} is the speedup or the ratio of real computational speed for the parallel simulation to the serial simulation. P_{solver} and C_{PC} are the portion of solver which is executed on multiple cores and the number of cores. Eq. (6-17) shows that if P_{solver}, I can reach the maximum speedup, which is equal to the number of cores (Scott et al., 2005) as shown by the linear line in Figure 6-9. But in general, sub-linear speedup is expected, due to practical limits such as load balancing and non-computational sections. Figure 6-9 shows the parallel efficiency and speedup on the present system with 4 cores: the SLAC speedup is about 3.6 and for SLPB is about 3.2 when comparing after and before implementing OpenMP for each method. The 3.6 times speedup of the parallel SLAC can be further improved when using other parallel computing techniques, for example, using graphics processing unit (GPU) computing, which will be the next step.

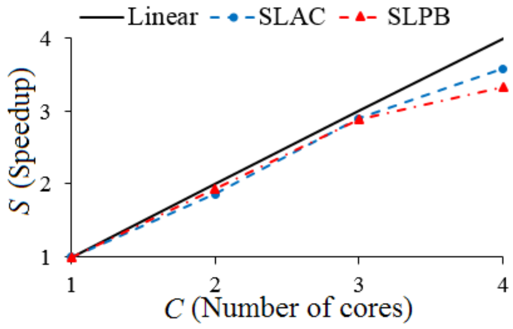


Figure 6-9 Parallel speedup relative to 4 cores on the PC system with 12 GB RAM and the Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz, which is with 4 CPU cores and 8 threads.

Table 6-3 Comparison of computing time [s] between the parallel SLPB and SLAC with non-parallel simulations for the lid-driven cavity problem with $512 \times 512 = 262$ k grids.

Non-Parallel Computing Time (s)		Parallel Computing Time (s)	
SLPB	SLAC	SLPB	SLAC
4711	1551	1411	433

Table 6-3 compares the speedups for the parallel SLPB and SLAC solvers. Before the OpenMP speedup, the proposed SLAC is 3.03 times faster than SLPB, while it increases to 3.35 times after the parallel implementation. Therefore, the OpenMP's SLAC speedup seems not quite impressive in this example, which may be due to the fact the OpenMP computing may lead to asynchronously iterative implementations, causing the side effect on the convergence rate of the iterative methods for solving other terms, e.g. diffusion terms (Üresin and Dubois, 1996).

6.5. Summary and Conclusion

In the present work, a new semi-Lagrangian method using the artificial compressibility algorithm is developed to speed up the conventional semi-Lagrangian method for solving the steady-state incompressible flows. The advantage of using the SLAC in comparison with the SLPB method is the elimination of the Poisson equation, which is the major bottleneck of the convergence speed and thus the computing time of the SLPB solver. For the SLAC method, an artificial compressibility coefficient, β, is defined, and the continuity equation is transformed to a hyperbolic equation of pressures, which can be solved by marching in time without the need of iterations. Both β and time step values can affect the convergence rate. Using larger value of dt and smaller value of β are recommended for better convergence, and large value of β could cause divergent simulations. This study also shows that the SLAC method can achieve the same results as the SLPB method while the speedup of about 3.03 times can be obtained. By using the parallel OpenMP, the SLAC speedup can be further increased to 3.35 times faster. Additionally, the parallel implementation of the proposed SLAC method is also straightforward and relatively easy, which makes it a convenient solver for steady-state incompressible flows and potentially for using the parallel computing techniques, for example, graphics processing unit (GPU) computing.

Chapter 7 LES model implementation with CityFFD, code validation, and city-scale simulation

The contents of this chapter are ready to submit "Mortezazadeh, M. and Wang, L., City Fast Fluid Dynamics – CityFFD equipped with LES for modeling urban microclimate airflows. " . The contents are modified, and some parts of the original paper are removed to avoid repetition.

Thermal and airflow simulations in the urban microclimates are important for the studies addressing public health and urban energy efficiency, and such studies are often conducted through conventional CFD solvers. The main limitation of these CFD tools is that they are computationally expensive for urban scale cases, and as a result, smaller computational domains or simpler and less accurate turbulence models such as RANS have to be used for saving computational time. This study developed a City Fast Fluid Dynamics (CityFFD) solver, a high-order FFD approach equipped with large eddy simulation (LES) turbulence model to simulate the urban microclimate in a faster and more stable manner than conventional CFD models. For the LES validation, two well-known test cases, including the airflow around a square cylinder and the natural ventilation airflow around a building, are modeled. The thermal solver is validated by simulating the natural convection inside a square cavity. The results show a good agreement with the experiment and other CFD models. For the application, CityFFD is applied to modeling the whole island of Montreal, Canada for the wind flows passing through building clusters and the urban temperature distribution during a heat wave. Compared to the horizontal area of a typical urban microclimate simulation of less than $2 \times 2\ km^2$, in the present study, the area of the test case is almost 400 times larger with a reasonable computing time: $45 \times 37\ km^2$.

7.1. Introduction

In Chapter 1, I comprehensively discussed the importance of urban microclimate in human health and comfort, and buildings energy performance. In the past decades, many engineers and researchers have applied CFD to studying the aerodynamics of airflow around buildings (Blocken et al., 2011; Tominaga and Stathopoulos 2011). Some researchers prepared a guideline for various parameters of CFD simulations, such as the size of the computational domain, grid resolutions near building objects, and numerical models of turbulence solvers (Tominaga et al., 2008). For example, Tominaga et al. (2008) mentioned that at least 10 grids are required for each side of a building for outdoor CFD simulation. The main concerns of the current CFD tools are the simulation speed and accuracy, especially when the turbulence models are used (Tominaga and Stathopoulos, 2010; Blocken, et al., 2012). As mentioned before, the turbulence models commonly used for the simulations of urban microclimates include three main categories: Reynolds-averaged Navier-Stokes (RANS) models, Unsteady Reynolds-averaged Navier-Stokes (URANS), and Large Eddy Simulation (LES) (Tominaga 2015). Tominaga and Stathopoulos (2010) demonstrated LES is more accurate and reliable in comparison with RANS. But the drawback of the LES model is a much higher demand for computational time. A typical CPU time consumption for a LES is almost 25 times more than the RANS (Tominaga and Stathopoulos, 2010). This drawback makes LES an unfavorable choice for large scale simulations (Gousseau et al., 2011; Tominaga and Stathopoulos, 2010; Tominaga, 2015). Generally, there are two main LES models: Smagorinsky and Dynamic LES (Hoffimann and Chiang, 2000(b)). The Smagorinsky LES model is a simple model which is able to provide relatively reasonable results, making it a commonly used model in turbulence modeling. Compared to the Smagorinsky, the Dynamic LES model is more complicated and time-consuming.

In the present work, I introduce a City Fast Fluid Dynamic (CityFFD) solver with the Smagorinsky LES model. CityFFD is a fast and accurate FFD model which is able to provide accurate results on the coarse grid and large time step. So, the model can provide the chance of using LES without much concern about the computational time. The proposed model is validated by two CFD benchmarks, the airflow around a square cylinder and the natural ventilation around a single room. Then, the thermal solver is validated by simulating natural convection in a square cavity. The results are shown to agree well with the experimental data. For the application, I simulate the whole island of Montreal, Canada with the area around $45 \times 37\ km^2$ that is much larger than the typical microclimates (area$< 2 \times 2\ km^2$) simulated by CFD (Toparlar et al., 2017). Montreal is selected for simulation because of its specific condition. Montreal is a city with low air conditioning rate, and consequently may be vulnerable to extreme events such as heat wave (Smoyer-Tomic et al., 2003). Therefore, in the current work, one of Montreal's hottest days in summer 2018 is modelled. As mentioned before, there were more than 50 deaths in Montreal because of a heat wave in summer 2018. The information of the current simulation study can be used by different users, namely, municipal offices, the governments, electric utilities, researchers and engineers, and etc. as comprehensively discussed in Chapters 1 and 2.

7.2. Methodology

CityFFD solves the following non-dimensional Navier-Stokes equations:

$$\nabla \cdot U = 0 \qquad (7\text{-}1)$$

$$\frac{\partial U}{\partial t} + (U \cdot \nabla)U = -\nabla p + \left(\frac{1}{Re} + \nu_t\right)\nabla^2 U - \frac{Gr}{Re^2}T \qquad (7\text{-}2)$$

$$\frac{\partial T}{\partial t} + (U \cdot \nabla)T = \left(\frac{1}{Re \cdot Pr} + \alpha_t\right)\nabla^2 T + q \tag{7-3}$$

where

Dimensionless time: $t = \frac{U_\infty}{L_\infty} t^*$

Dimensionless length: $X = \frac{X^*}{L_\infty}$

Dimensionless velocity: $U = \frac{U^*}{U_\infty}$

Dimensionless pressure: $p = \frac{p^*}{\rho_\infty {U_\infty}^2}$

Dimensionless temperature: $T = \frac{T^* - {T_\mathrm{C}}^*}{{T_\mathrm{H}}^* - {T_\mathrm{C}}^*}$

Dimensionless heat source: $q = (\frac{L_\infty}{\rho_\infty c_{p_\infty} U_\infty ({T_\mathrm{H}}^* - {T_\mathrm{C}}^*)}) q^*$

Reynolds number: $Re = \frac{U_\infty L_\infty}{\nu_\infty}$

Grashof number: $Gr = \frac{g_i \beta_\infty {L_\infty}^3 ({T_\mathrm{H}}^* - {T_\mathrm{C}}^*)}{{\nu_\infty}^2}$

and Prandtl number: $Pr = \frac{\nu_\infty}{\alpha_\infty}$

where * denotes the dimensional value of the parameters. Here $g_i = (0, 0, -9.81)$ is gravity. Air Prandtl number for a wide range of temperature from -50 $^{\mathrm{o}}$C to 100 $^{\mathrm{o}}$C is constant $Pr = 0.71$. Details of CityFFD solver were investigated in the previous chapters.

One of the most well-known and accurate turbulence models is LES. LES was initially proposed by Smagorinsky (1963) to simulate atmospheric airflows. This model is based on a mathematical method to capture the turbulence characteristic of the flow. LES is developed to reduce computational cost by avoiding directly solving the smallest length scales which are the most time-consuming parts of the turbulence modeling. This is achieved by using low-pass filtering of the Navier-Stokes equations (Hoffmann and Chiang, 2000(b)). In the current work, the Smagorinsky model (SGS) is added to CityFFD to model turbulence behavior of the flow inside the urban microclimates. In this model, the turbulence viscosity in Eq. (7-2) is calculated by the following equation:

$$\nu_t = (c_s\Delta)^2\sqrt{2\bar{S}_{ij}\bar{S}_{ij}} \tag{7-4}$$

where c_s, Δ, and $\bar{S}_{ij}$ are the Smagorinsky constant, the filter width, and rate of the strain tensor, respectively. For most applications, the Smagorinsky constant is in the range of $0.1 < c_s < 0.24$ (Hoffmann and Chiang, 2000(b)). For the filter width, there are several options, and the following two options are mostly used by researchers:

$$\Delta = (\Delta x\Delta y\Delta z)^{1/3} \tag{7-5}$$

$$\Delta = (\Delta x^2 + \Delta y^2 + \Delta z^2)^{1/3}$$

In this work, I use the first one. The strain rate is given by:

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial X_j} + \frac{\partial U_j}{\partial X_i}\right) \tag{7-6}$$

where i-index shows the x, y, and z directions, e.g. $U_i = (u, v, w)$ and $X_i = (x, y, z)$. In the next section, I show the performance of the proposed LES method in comparison with other numerical results.

Turbulent thermal diffusivity is calculated based on the turbulent Prandtl number as follows:

$$Pr_t = \frac{\nu_t}{\alpha_t} \tag{7-7}$$

where $Pr_t = 0.3 \sim 1$. The turbulent Prandtl number, also named turbulent Schmidt number, varies according to the distance from the earth in the boundary layer (Koeltzsch, 2000). For simplicity, in the present work, I consider $Pr_t = 1$ and leave other value settings as future work.

7.3. Results

In this section, I first simulate 3 different CFD benchmarks to investigate the accuracy of CityFFD with LES. The first two validation cases are related to the wind and the last one is for the thermal solver.

7.3.1. Airflow around a square cylinder

In this section, the turbulent flow passing a square cylinder is simulated and the result is compared with the experiment and other numerical simulations. This problem is with $Re = 22000$ and the characteristic of this problem with the mentioned Reynolds number is quasi 2-D (Bouris and Bergeles, 1999). The schematic of the computational domain and the average streamline around the square cylinder are shown in Figure 7-1.

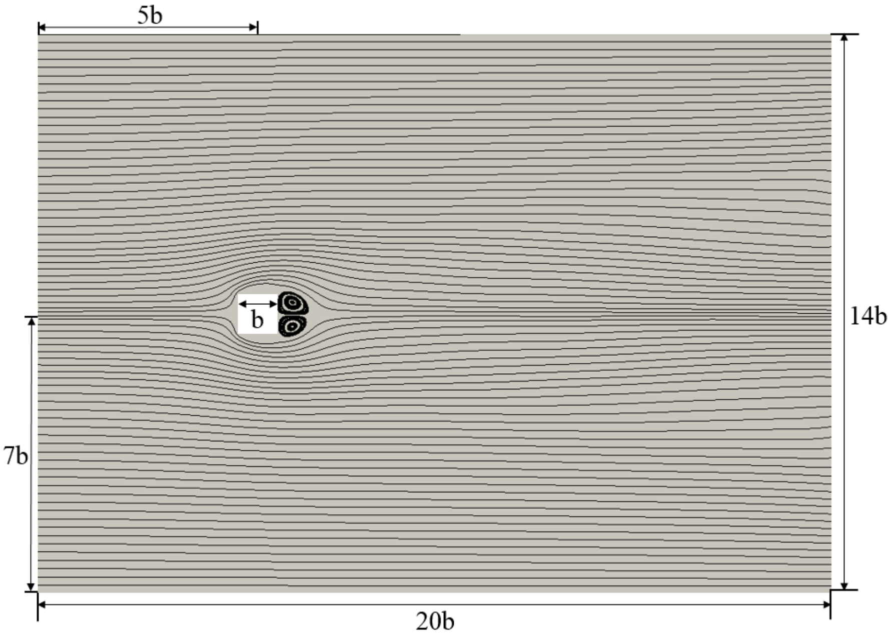


Figure 7-1 Average velocity field (Streamline) around 2D square.

In this problem, a uniform computational grid, $\Delta x = \Delta y = 0.05$, about 40 times coarser than the same case done by Bouris and Bergeles (1999), is considered and the time step is $\Delta t = 0.01$. Figure 7-2 shows the simulated average velocity along a horizontal line passing the center of square cylinder.

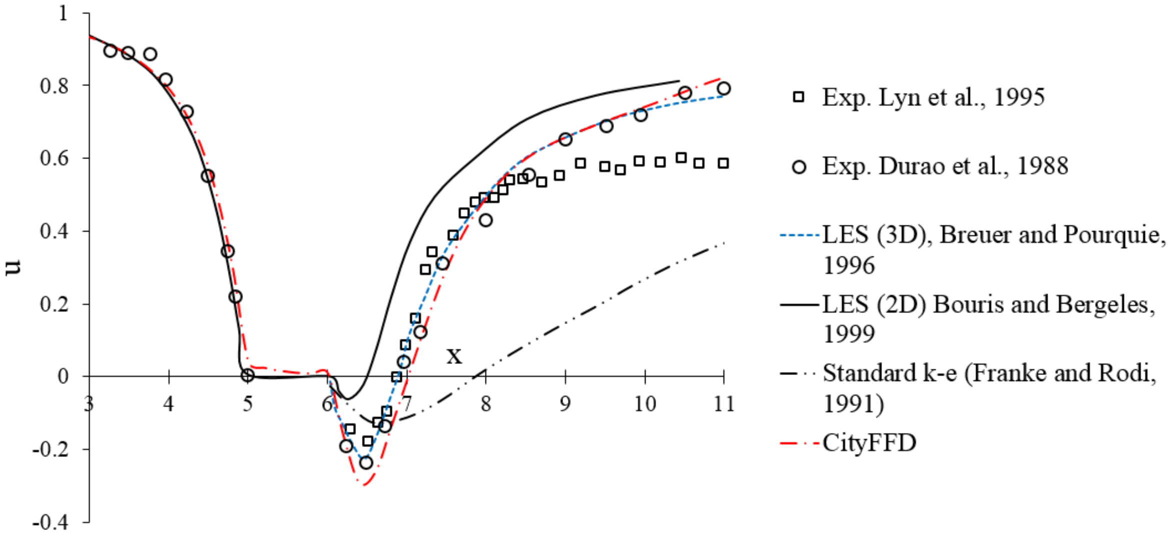


Figure 7-2 u along a horizontal line passed the center of the square cylinder.

Figure 7-2 shows that the proposed model has a good agreement with the experiment and another CFD based LES model proposed by Breuer and Pourquie (1996). It can be seen that $k - \varepsilon$ turbulence model overestimate the size of recirculation zone behind the cylinder and the results while LES models can generate more accurate results and capture the correct size of vortices behind the cylinder. Figure 7-2 also shows a good agreement between CityFFD simulation data and experiment. So, LES model used in CityFFD can accurately capture the recirculation regions.

7.3.2. Natural ventilation around a single building

The second validation study is based on the cubic building with an opening, which was first conducted experimentally by Jiang in 2003 (Jiang et al, 2003) in a boundary layer wind tunnel. The dimension of the cubic building model is shown in Figure 7-3. The cubic building is designed with a size of 250 mm (250×250×250 mm^3). There are two openings with the size of 84 mm (width) ×125 mm (height) located at the windward and leeward of the building, respectively. In this study, only the cross-ventilation case was considered, and both doors will be set as the open boundary condition. Here, the inlet wind profile is based on the power-law distribution with an exposure component of 0.17.

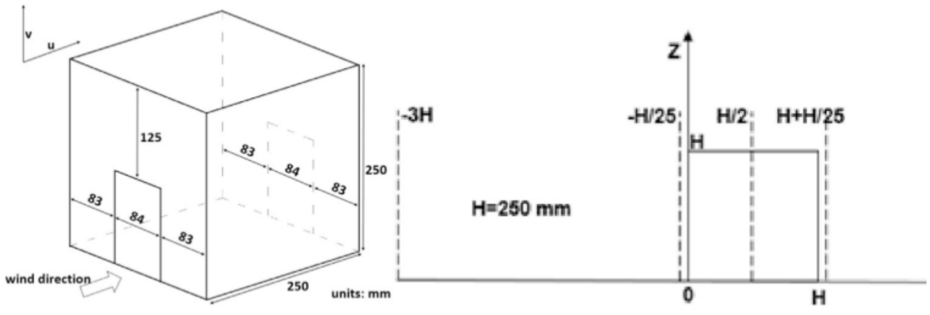


Figure 7-3 Dimension of the single cubic building model (Jiang et al, 2003).

In the previous study (Jiang et al, 2003), the velocities at different locations were collected for comparing the simulation results with the experimental results measured by Laser Doppler

Anemometer. In this case, the velocities at four locations were used for validation: X= -3H, X= -H/25, X=H/2, X= H+H/25) (see Figure 7-3). Generally, the results show a good agreement with the literature. For the windward part, X= -3H and X= -H/25, the simulation results are quite close to the experiment data at the different elevations. However, for the leeward part, X=H/2 and X=H+H/25, the results show a good agreement at the height from 0 to 0.25 m, while above 0.25 m (the height of the building) the simulation overestimates the velocity compared to the experimental data. In summary, in this typical single building for the cross-ventilation study, CityFFD shows an acceptable performance.

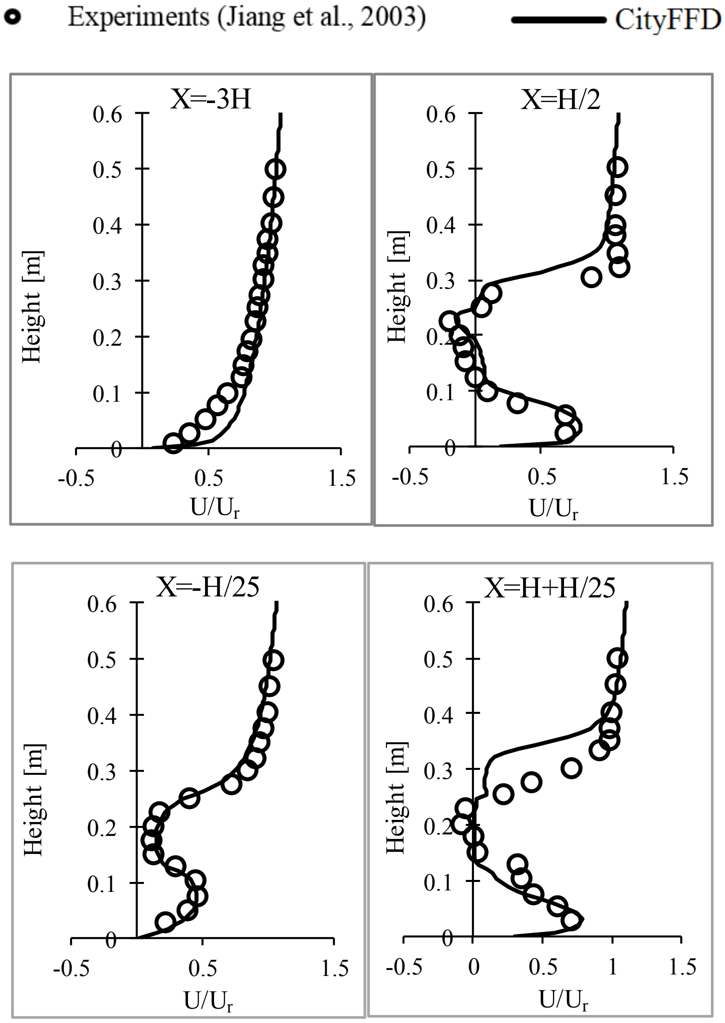


Figure 7-4 Comparison of mean velocity profile between simulation and experimental results.

7.3.3. Natural convection in a square cavity

Here, to validate the thermal solver, I simulate natural convection in a square block. In this problem, four faces of the block have the adiabatic thermal condition and two vertical faces have a constant temperature (Figure 7-5).

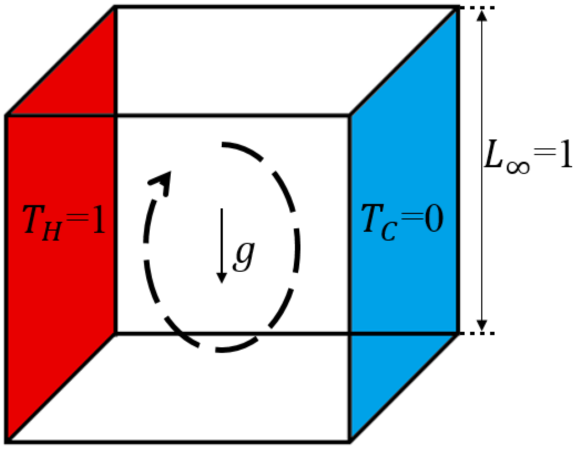
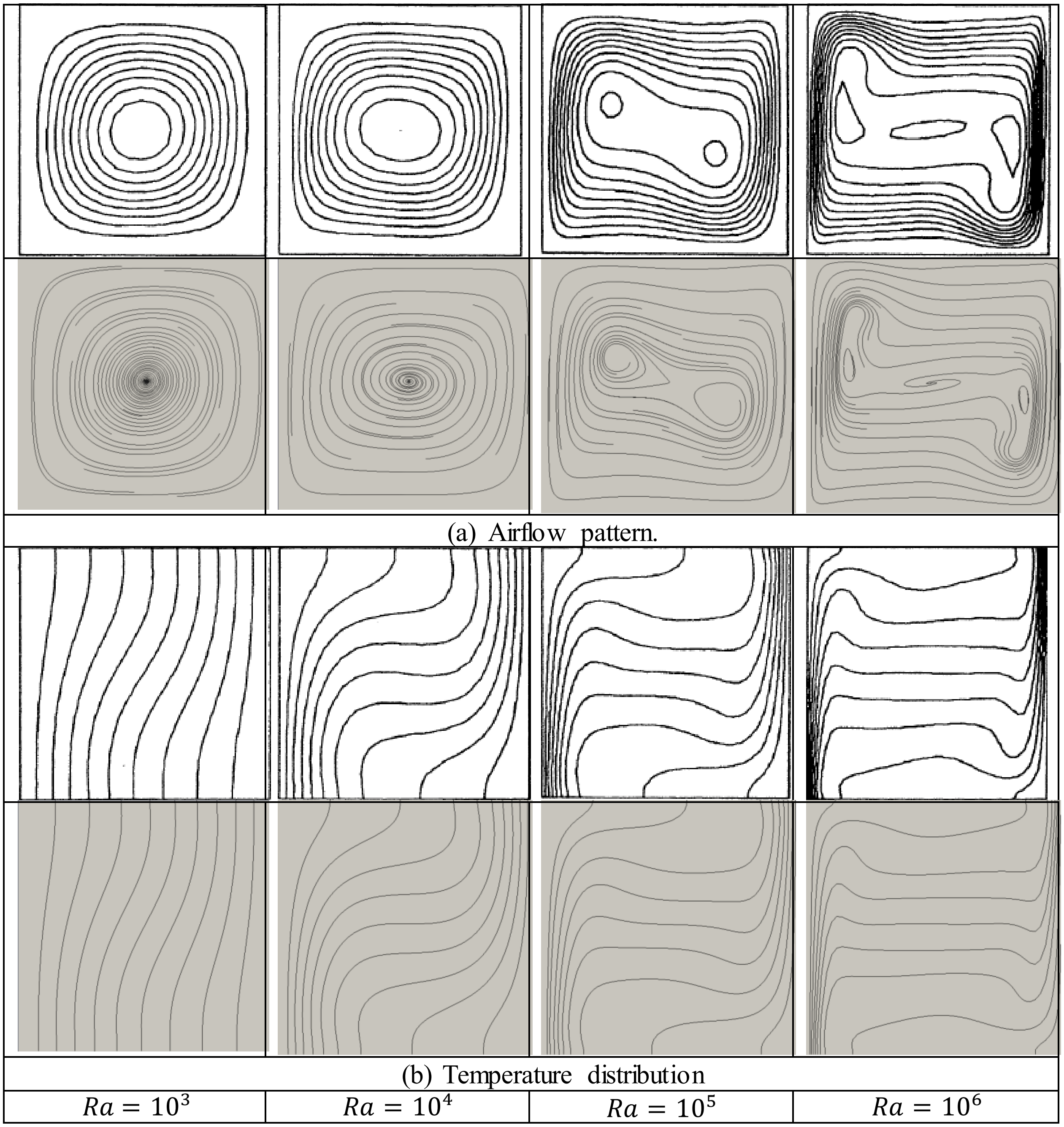


Figure 7-5 Schematic of natural ventilation problem in a square cavity.

Because of the buoyancy effect, the air inside the cavity and near the hot face goes up and a recirculation airflow is generated inside the cavity. Temperature distribution and airflow pattern inside the domain are directly related to the Grashof number. In these types of problems, another dimensionless number, Rayleigh number, is defined: $Ra =$In some problems such as natural convection, there is no explicit definition of reference velocity. So, dimensional velocity components are nondimensionalized with $\frac{\alpha_\infty}{L_\infty}$ which equals to thermal diffusivity divided by length scale. Detail of this procedure is provided in Appendix 2. For more information, I refer the reader to Appendix 2. Figure 7-6 shows the capability of CityFFD to model characteristics of the flow under the influence of temperature with high accuracy. In this problem, natural convection with a wide range of different Ra and consequently Gr is modeled to show the performance of CityFFD for even large temperature differences. Larger value of Ra means larger difference between T_H^*

and $T_C{}^*$, that mean higher buoyancy effect. For $Ra = 10^3$ and $Ra = 10^4$, a uniform vortex is generated inside the cavity, but for larger values of Ra two small vortices are created inside a big one. It also affects the temperature distribution. For larger value of Ra the temperature variation will change from vertical distribution to a almost horizontal one.



(a) Airflow pattern.

(b) Temperature distribution

$Ra = 10^3$	$Ra = 10^4$	$Ra = 10^5$	$Ra = 10^6$

Figure 7-6 Natural convection in a square cavity for different Rayleigh numbers, upper: de Vahl Davis (1983), lower: CityFFD.

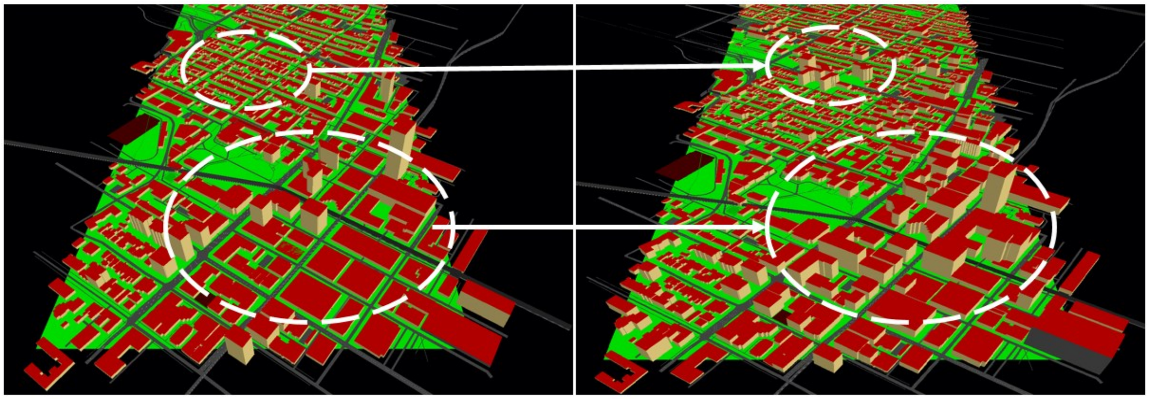
7.3.4. Simulation of Great Montreal, Canada

As mentioned before, Montreal is city with low air conditioning rate and vulnerable to heat stress problems (Smoyer-Tomic et al., 2003). More than 120 heatwave events happened in Montreal during 1942-1994, which is almost 2 times more than the average of events in Canada's cities. In the following, Montreal's microclimate is simulated for the hottest temperature recorded during the heatwave in summer 2018 (28 June-3July). Table 7-1 shows the weather information:

Table 7-1 Weather information July 2, 2018, 2:00:00 PM.

Air temperature	$\mathbf{35.3}$ ℃
Wind speed	$4.7\ m/s$
Wind direction (from north)	$230\ deg$
Relative humidity	42 %

To generate the geometry of the whole city, here I used OpenStreetMap data (Haklay and Weber, 2008). OpenStreetMap creates a free editable map of the world with the collaboration of more than 2 million users (Neis and Zipf, 2012). Buildings information in OpenStreetMap lacks attributes such as height. To access the correct shape of the building heights, I use a Google Earth API to find the correct elevations of the building roofs. Figure 7-7 shows the original OpenStreetMap and the modified data after using Google Earth API.



(a) OpenStreetMap (b) OpenStreetMap and Google Earth API

Figure 7-7 3-D building geometry and building's height modification.

The total simulation region is $45 \times 37 \times 1.5$ km^3 with more than 100,000 buildings. The total computational grids are 160 million. Uniform grids are used in this case and the grid's size is 25 m. Ground temperature is estimated around 37 °C. In this work, by considering a similar material for all the buildings and applying the effect of solar radiation, I estimated the surface temperature which equals to 40 °C. This case ran on the computer with a NVDIA GTX TITAN V graphics card. It took 50 hours to get the converged results.

Figure 7-9 presents the thermal distribution inside the city at the pedestrian level. In dense urban area, because of more buildings and slower wind velocity, higher temperature is expected.

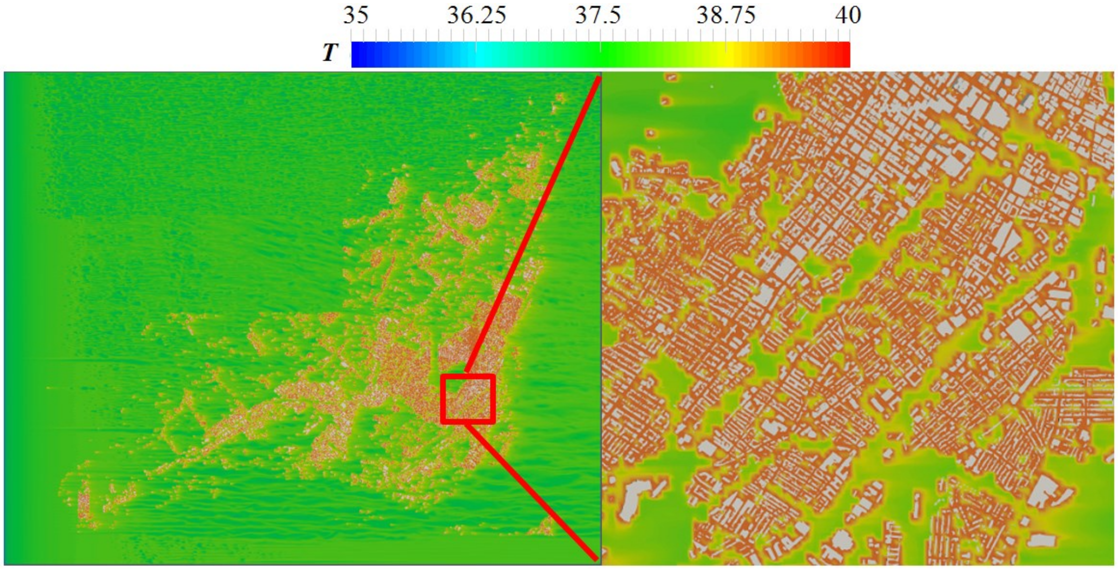


Figure 7-8 Temperature distribution in Montreal (July 2, 2018, 2:00:00 PM) (a) Whole Montreal island (b) Downtown of Montreal.

The results could show the most vulnerable overheating regions in the city. Higher temperature can be seen in downtown of Montreal. This region is a dense urban area so higher heat generation and lower wind velocity is expected. Thus, higher air temperature is trapped at the pedestrian level and in the street canyons. By this type of information, Montreal municipality or the government can identify high-risk areas and take actions to reduce the risks of heat stress.

The main purpose of this simulation is to demonstrate the performance of CityFFD for modeling real urban scale problems on personal computers. To improve the accuracy of the simulation for modeling extreme events, such as heatwaves, it is necessary to accurately calculate the building surface temperatures by considering the building envelope materials and indoor conditions. Additionally, transient simulation of airflow is crucial to capture the effect of diurnal temperature and wind. In this work, I didn't consider the terrain, such as mountain because of the lack of such information in OpenStreetMap. But these type of information can be obtained by using Google

Earth API. In the next chapter, for a further demonstration, I study the urban microclimate by considering the individual information of each building and its urban microclimate impact.

7.4. Summary and Conclusion

CityFFD, a high-order semi-Lagrangian approach equipped with large eddy simulation (LES) method, is developed to simulate urban microclimates in a fast manner. The following conclusions is obtained: a) The validation study shows that CityFFD can achieve accurate results and gather reliable velocity information at different locations, even for the case with indoor airflows involved. The results are in a good agreement with the experiments; b) The thermal solver is validated for various values of temperature differences. The results show the capability of CityFFD for modeling thermal airflow problems; c) For urban scale simulations, which typically requires high computing cost for conventional CFD tools, CityFFD is able to model large metropolitan cases with less computing power and better efficiency than conventional tools. Here a very large domain has been simulated by CityFFD which is considerably larger than typical microclimate domains that have been simulated by CFD. The simulation has been done on a personal computer without needs of supercomputers or clusters.

Chapter 8 Modeling building resilience against extreme weather by integrated CityFFD and CityBEM simulations

The contents of this chapter have been published in "Katal, A., Mortezazadeh, M. and Wang, L.L., 2019. Modeling building resilience against extreme weather by integrated CityFFD and CityBEM simulations. Applied Energy, 250, pp.1402-1417.". The contents are modified. The paper is based on the integration of two in-house models, CityFFD and CityBEM. CityBEM has been proposed by Ali Katal and is a Building Energy Model. Details of this model are briefly introduced in Appendix 3.

In the past decade, urban building energy models have been developed to address the increasing concerns over energy consumption and greenhouse gas emission due to rapid urbanization and building resilience as a result of climate change. These models can estimate energy consumption, GHG emission and resilience response of buildings in an urban area, and evaluate retrofit strategies for architects, engineers, researchers, and policymakers. It has been recognized that local microclimate and neighborhood effects play an important role in urban building energy modeling. Creating an urban building energy model also requires the collection of extensive building data, which is a time-consuming process. In this study, I developed an integrated platform by combining CityFFD (City Fast Fluid Dynamics), an urban-scale fast fluid dynamics model for microclimate modeling, and CityBEM (City Building Energy Model), a new urban building energy model with a library of 1,700 building archetypes for facilitating urban model creation. Local aerodynamics and heat transfer information are exchanged between both models at each time step. Graphics processing unit computing is also applied to CityFFD for simulation speedup. The simulation of the 1971 Montreal snowstorm of the century was conducted as a case study of more than 1,500 buildings of an island near Montreal, Canada for the investigation of their resilience against the three-day power outage due to the storm. Building retrofit analysis was also conducted to evaluate the added level of resilience. The results show that the proposed platform can produce high-resolution results of building thermal load, microclimate condition, and building behavior during weather extremes.

8.1. Introduction

Rapid urbanization with the increased energy consumption, especially in the building sector, and increased greenhouse gas (GHG) emissions draw a lot of attention to the understanding of building energy usages at an urban scale, i.e. so-called urban building energy modeling (UBEM) (Chen et al., 2017). Through UBEM, municipal governments, urban planners, and building and environment researchers can investigate the effect of future potential energy savings through new technologies, standards/codes, and energy management policies on existing or new constructions in terms of urban energy usage and their associations with GHG emissions and regional environmental qualities. There exist two main UBEM approaches: the top-down and the bottom-up models. In top-down models, a group of buildings are analyzed as a single group unit, and they do not provide the energy consumption of each individual building (Howard et al., 2012; Swan and Ugursal, 2009). These models are incapable of modeling different energy demand-supply scenarios and retrofitting strategies and cannot provide a detailed analysis of a specific neighborhood (Fabri and Tarabusim, 2014). In contrast, the bottom-up models simulate each building individually by statistical and/or physics-based methods in aggregations to the urban, state or country scale (Robinson et al., 2009). The bottom-up models can provide detailed analyses of every single building, evaluate the impact of new technologies, predict the future energy consumptions of a specific existing neighborhood, and even of future urban developments. The bottom-up models can be categorized into two different types: statistical and physics-based models.

The statistical models use historical energy usage of end-use buildings or some sample buildings to calculate the total energy uses. The historical data of energy consumption and economic indicators are provided by governments' sources, which may not be available and accessible for all urban areas (Swan and Ugursal, 2009). Another major limitation of such statistics-based models is poor characterizations of energy services and coarse spatial/temporal resolution of the analysis (Howard et al., 2012). In contrast, the physics-based models apply heat and mass balance equations to each individual building with the capability of achieving any spatial and temporal resolutions. Physics-based models require buildings' geometrical and non-geometrical parameters including buildings' shape, glazing, envelope thermal properties, occupancy rate and schedule to create the model and calculate buildings energy consumption (Cerezo et al., 2014). Physics-based models do not require the historical energy consumption data as required by the top-down and statistical bottom-up models. Physics-based models only require buildings parameters and weather data to analyze buildings energy consumption. Therefore, they can be used for the analysis of future city infrastructures, which makes them a promising method for future urban-scale energy consumption and GHG emission analysis studies.

In recent years, several physics-based UBEM tools have been developed, such as CitySim, Urban Modeling Interface (UMI), CityBES. CitySim has been established by Ecole Polytechnique Fédérale de Lausanne University (Robinson et al., 2009), and uses a simplified resistor-capacitor network model to estimate the energy usage by buildings at the scale of an urban district. Urban Modeling Interface (UMI) (Davila et al, 2016) is based on the 3-D modelling software platform Rhino (McNeel R. Rhino 6 for Windows 2018) and uses EnergyPlus. UMI has been firstly used to estimate hourly energy demand of the Boston city. CityBES has been developed by the

Lawrence Berkeley National Lab (Hong et al., 2016), and is an open web platform for simulating city building energy efficiency.

Most of these existing UBEM platforms use weather data from one or several nearby weather stations for the energy analysis of all buildings. Therefore, they have not considered the impacts from localized microclimate environment. The airflow velocity and temperature around buildings are affected by building configurations, heights and neighboring building locations. Different wind velocity and temperature around buildings have also a direct impact on building thermal load in terms of local convective heat transfer coefficients and rates, and air infiltration through envelopes (Gracik et al., 2015). To better predict individual building energy use, therefore, it is important to consider local microclimate and aerodynamics conditions, and neighborhood effects on building energy consumptions (Quan et al., 2015; Hong et al., 2016).

Computational fluid dynamics (CFD) can be used to generate local microclimate conditions of airflow around buildings. However, the main challenge of using conventional CFD tools, such as ANSYS Fluent (2011) or OpenFOAM (2014), is the high computational time in solving urban-scale problems typically with several millions of grids. For example, previous studies show that solving a problem with 1-10 million of grids using commercial software on a personal computer takes several days or even weeks (Gousseau et al., 2011). Urban microclimate simulations need a fast solver, which cannot be done using existing CFD tools. Therefore, a fast CFD solver is needed for the simulation of such large-scale problems as urban microclimate modeling.

In the present work, I introduce a new urban scale model, which integrates two in-house models: CityFFD (Mortezazadeh and Wang, 2018) and the CityBEM. The CityFFD model is based on a 3-D Fast Fluid Dynamics (FFD) solver (Mortezazadeh and Wang, 2017) for the prediction of local microclimate and neighborhoods. Compared to the conventional CFD model, the CityFFD solver is a high-order semi-Lagrangian-based model, which is unconditionally stable, fast and accurate to simulate urban aerodynamics. The model is written in the NVIDIA CUDA (2011) for Graphics processing unit (GPU) computing to achieve superior performance on a personal computer.

The CityBEM is a new energy model for buildings and other infrastructures (e.g. tunnels) covering all key heat and mass transfer mechanisms for the calculations of building heating/cooling loads including solar radiation, wall conductive heat transfer, internal heat gain, and infiltration. The model is considerably faster than other simulation engines with acceptable accuracy for urban scale energy analysis. The two urban scale models are integrated as follows: the average air velocity and temperature at each building surface are calculated by CityFFD for the calculations of the heat transfer and infiltration through building envelope as the input boundary conditions for CityBEM. The building surface temperatures are calculated by CityBEM to be applied as the boundary conditions for the CityFFD solver at the next time step simulation. Therefore, the integrated CityFFD and CityBEM can simulate both outdoor airflow and temperature, building thermal load and indoor air temperature within an urban area in a fast manner without the need of using a supercomputer. Meanwhile, a new comprehensive archetype library is defined, which includes 1,700 archetypes of different buildings types and ages to cover all the buildings located in the urban area under investigation. The chapter is organized as follows: the methodology is introduced in Section 2 followed by a case study of an urban area, the Ile-des Soeurs located in Montreal, Canada, for the evaluation of building and city resilience against the extreme events such as ice storms, snowstorms and power outages in Section 3. Section 4 is the conclusion and discussion about current and future studies. As mentioned before, details of CityBEM are available in Appendix 3.

8.2. Methodology

8.2.1. Overview

The process of modeling an urban-scale problem can be divided into two main steps: model preparation, and model simulation. The model preparation includes all the tasks to prepare the 3-D model of the city for the simulation: preparing GIS geometrical input data, weather data, and creating archetype building dataset. When the city model is ready, the simulation step applies the integrated model by running the CityFFD and CityBEM (Figure 8-1) in a sequential and interactive manner, i.e. Ping-pong (Figure 8-2) (Hensen, 1995). The CityFFD solver takes building geometry and weather data as inputs to simulate approaching wind conditions, and temperature distributions around the buildings. Then the average wind velocities, directions, and temperatures around each surface of buildings calculated by CityFFD, along with the geometry and archetype data are used as the inputs for CityBEM for the calculations of heating/cooling demands, energy uses, and indoor air temperatures of each building. The building surface temperatures calculated from CityBEM are then used as the boundary conditions of CityFFD for the next time step. The framework of the integrated model is shown in Figure 8-1. The method of "Ping-pong" data transfer means that the data are only transferred once between the two solvers at each time step (Figure 8-2). A more detailed flow diagram of solving a problem with an integrated framework is shown in Figure 8-6.

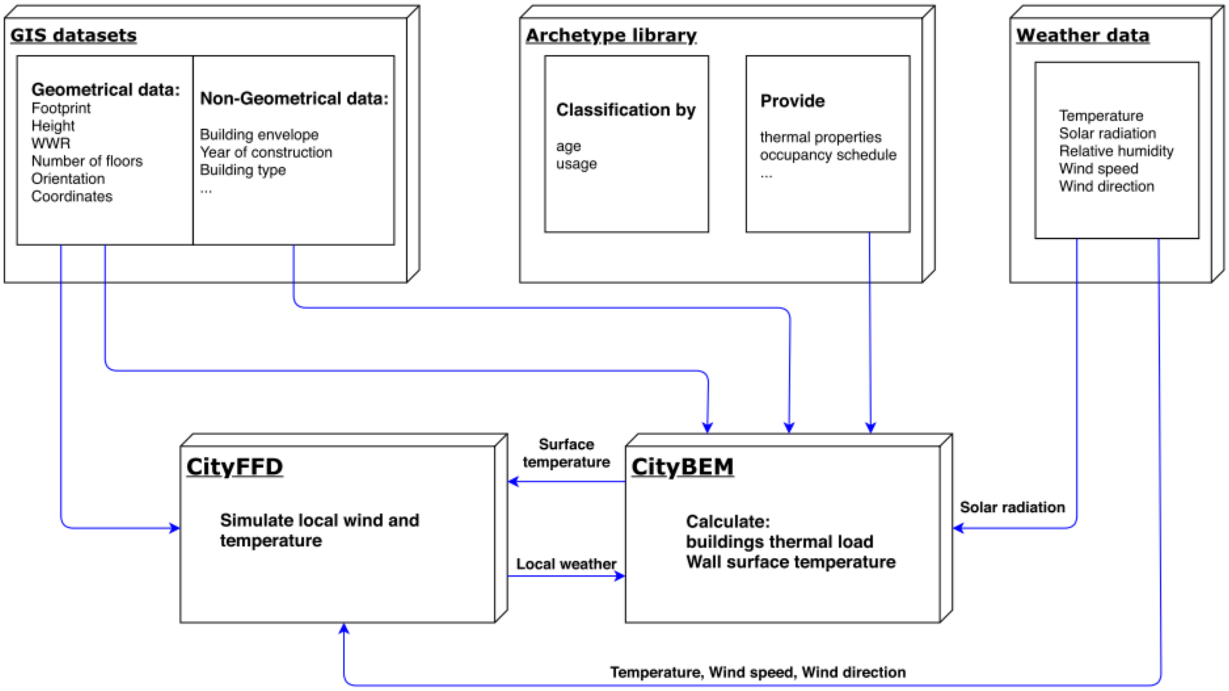


Figure 8-1 The integration framework of CityFFD and CityBEM.

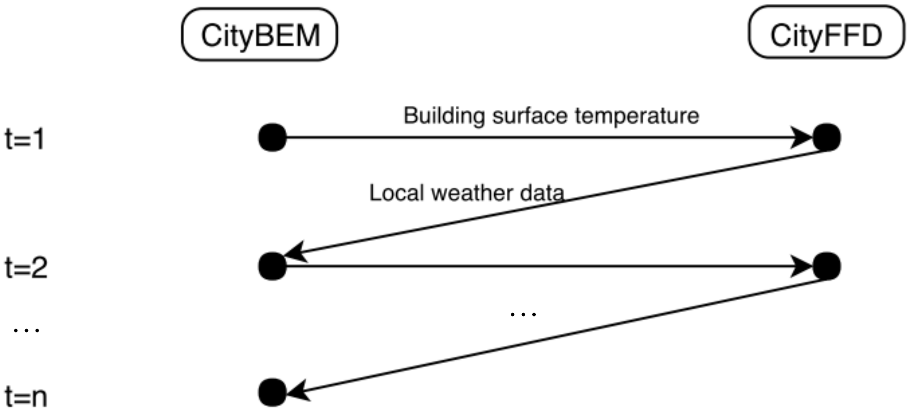


Figure 8-2 The integration procedure of CityFFD and CityBEM.

8.2.2. Model creation

Based on Figure 8-1, three input data sets are required for the simulation: geometry and physical data, archetype library, and weather data.

8.2.2.1. Building geometry and physical data

The geometrical data include building footprints, heights, window-wall-ratios (WWR), number of floors, orientations, and other physical data including building envelopes, occupancy information, year of constructions, and building types. For various urban regions, different geographic datasets can be integrated to provide the required data.

8.2.2.2. Archetype library

The geometry data sources do not provide all required data in section 8.2.2.1, such as building envelope and construction data. Although the information can be gathered for small groups of buildings, it remains a challenge to measure or collect the information for a large urban area with thousands of buildings. In the context of urban building modeling, for the characterization of such building properties, the building stock is divided into "archetypes". The creation of an archetype library requires two steps: segmentation and characterization. In the segmentation step, the building stock is divided into a couple of groups by use, age, shape, etc. In the characterization step, a set of properties such as building envelope, thermal properties, and occupancy schedule are assigned to each group (Reinhart and Davila, 2016).

8.2.2.3. Weather data

The input weather dataset contains hourly measured or existing environmental variables such as dry bulb temperatures, solar radiations, relative humidity, wind speeds and directions. Here, the dry bulb temperatures, wind speeds, and directions are the boundary conditions for the CityFFD model whereas the solar radiation data are the input data for CityBEM.

8.2.3. Simulation models

8.2.3.1. CityFFD - high-order city-scale FFD model

As mentioned in the previous section, modeling local microclimate and capturing the effects of neighborhoods on the aerodynamics in a city are remaining challenges of UBEM, mostly related to huge computational costs (Gousseau et al., 2011). Some mathematical constraints of the conventional CFD programs, e.g. a small Courant–Friedrichs–Lewy (CFL) number required for the stability of a simulation, lead to the long computational time, especially for a typical desktop personal computer (PC) (Mortezazadeh and Wang, 2017). Details of CityFFD have been investigated in the previous chapters.

The turbulence closure here is achieved by using a zero-equation turbulence model, which is a fast and relatively accurate (Chen and Xu, 1998):

$$\nu_t = 0.03874LV \tag{8-1}$$

Similar to the previous chapter, here Pr_t is equal to 1. Details of the proposed method have been presented in chapters 3-7.

The CityFFD model uses hourly typical meteorological year (TMY) weather data as the boundary conditions. The code is developed by the NVIDIA CUDA computing language with a typical speedup of more than 15 times faster than the central processing unit (CPU) counterpart.

8.2.3.2. CityBEM – Fast energy modeling of buildings and infrastructures

CityBEM is a physics-based simulation model for urban thermal loads and energy uses. The model takes as the inputs including building information: geometry, construction materials, lighting, HVAC, etc.; and building uses and operations: occupancy schedules, lighting, plug-loads, and thermostat settings, so it is possible to run the annual calculations on an hourly or shorter-time-step basis. General information about CityBEM is presented in Appendix 3.

8.2.3.3. Integration of CityFFD and CityBEM

The CityFFD model provides local aerodynamics conditions around each building to CityBEM including average temperatures, average wind velocities and local exterior surface convective heat transfer coefficient around external surfaces of each building, calculated using all mesh cells on each surface of the building (See Figure 8-3). The local outdoor temperature and wind velocity are used for the calculation of exterior convective heat transfer coefficient. The building surface temperatures calculated by CityBEM are used as boundary condition by CityFFD for the next time step simulation. Because of using the ping-pong method in transferring data between CityFFD and CityBEM, it is important to start the simulation with an accurate initial condition. For this purpose, at the first time-step, the data is transferred between CityFFD and CityBEM in an inner loop until the convergence with a criterion of 1×10^{-3}. The converged building surface temperatures are then used as the initial data for CityFFD simulation. Figure 8-4 shows the flow diagram of parameters initialization and transient simulation. For the transient simulation stage in Figure 8-4, the integrated model allows for running sub-loops at each time step as the first time-step does. However, because the city-scale climatic data will not change much with the CityBEM data, especially for the case study in this chapter (Section 8.3), the data are therefore exchanged only once for the rest of the time steps ("Ping-pong"). This also helps to save computing time. It might be possible that the Ping-pong method becomes inapplicable for other cases, for which I am conducting more studies and will report as future work.

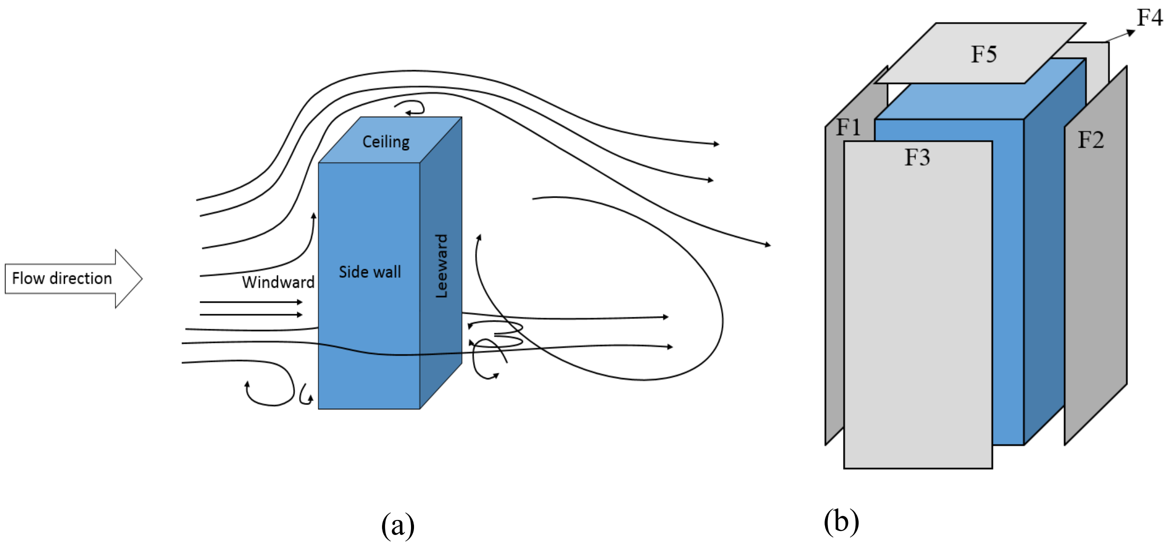


Figure 8-3 CityFFD data provided for the CityBEM model.

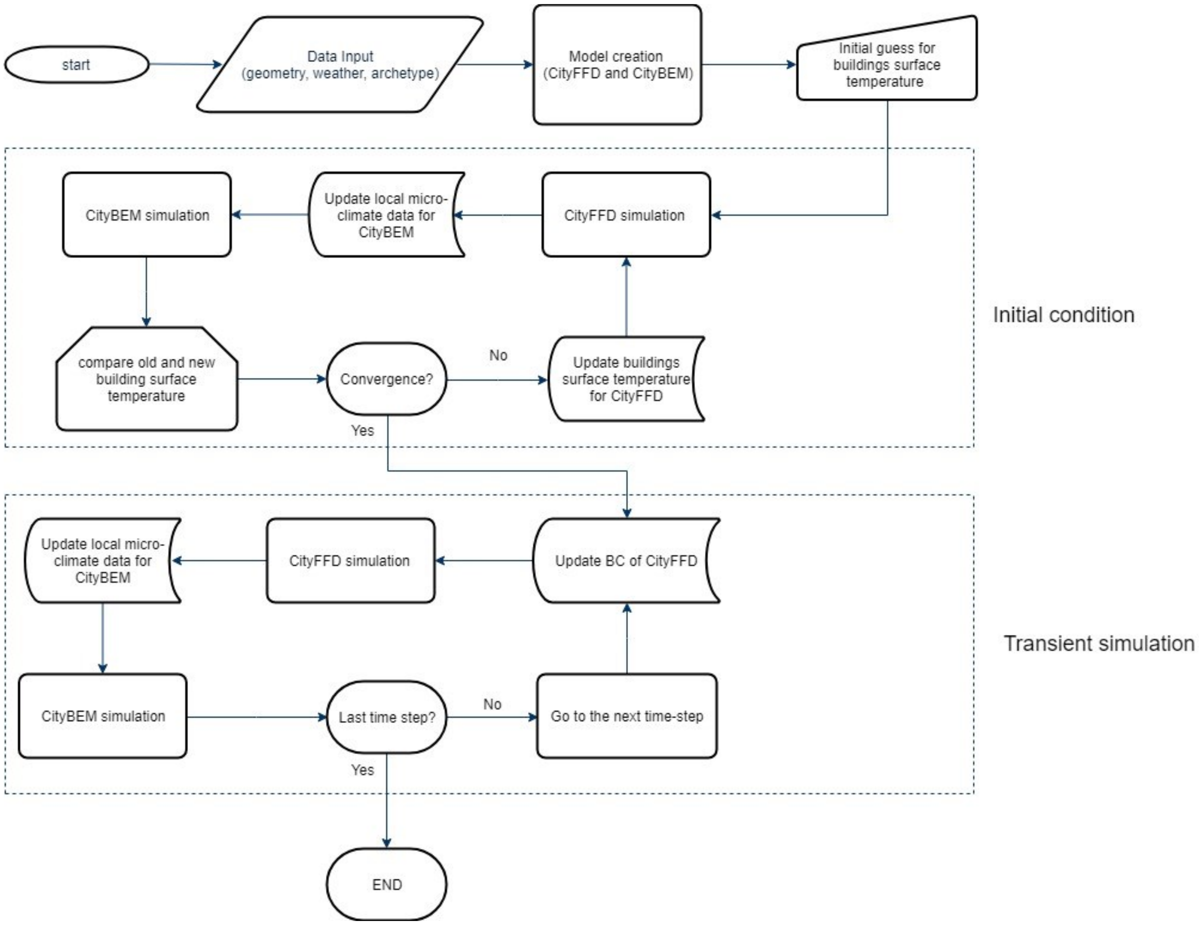


Figure 8-4 Flow diagram of solving a problem by integration of CityFFD and CityBEM.

8.3. Case study - urban building energy simulation of Ile-des Soeurs

As a case study, the Ile-des Soeurs community in Montreal, Canada was selected. In this section, the preparation of the input data including building geometrical data, physical property data, and weather data for the Ile-des Soeurs is explained first. Then, a hypothetical power outage in the region based on the real event of the Montreal "snowstorm of the century" (SOC) in March 1971 was simulated to study the thermal resilience of the region. The effect of building envelope retrofitting on the thermal performance of the buildings is then studied to demonstrate using the proposed urban models for a real scenario.

8.3.1. Urban district

As a part of the city of Montreal, Quebec, Ile-des Soeurs is an island with a total area of $3.74\ km^2$, around 1,500 buildings, and 19,000 populations (SC. 2011-data products). The island is primarily composed of multi-unit residential apartments, condos and townhouses. The various types of buildings with different years of constructions and the available database of buildings geometry make the island a good test case for the current study. Figure 8-5 shows the aerial map view of the Ile-des Soeurs with years of constructions (Rocha, 2018).

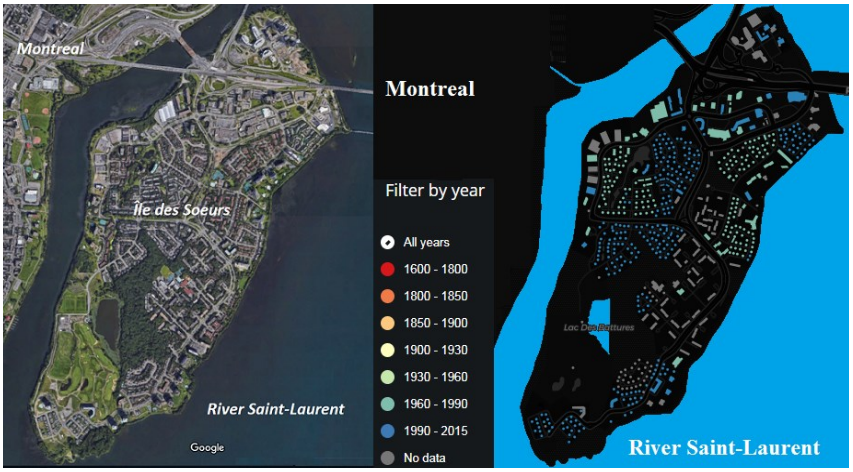


Figure 8-5 The Ile-des Soeurs aerial view map (left) (Google) and Ile-des Soeurs building year of construction map (right) (Rocha, 2018).

8.3.2. Data preparation

Multiple global and local datasets including Google Maps (Google), OpenStreetMap (Haklay and Weber, 2008), City's property assessment office (VdM. Consultation) were used to acquire buildings' geometrical and physical data. The CityFFD model of the island with the enclosing computational domain is shown in Figure 8-6. The inlet and outlet boundary conditions were varied according to the hourly wind speeds and directions provided by the historical weather data at the Montréal-Trudeau International Airport (YUL) (ECCC. Historical Data, 2018). The inlet wind profile is defined using the power-law function. The upper surface of the domain was modeled as a symmetric condition by setting the normal gradients of the velocity components to be zero (Tominaga et al., 2008).

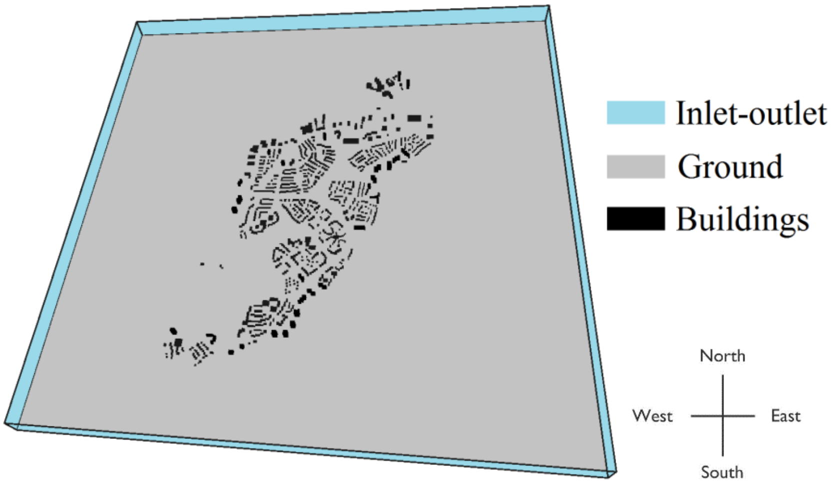


Figure 8-6 The CityFFD simulation model of Ile-des Soeurs.

Table 8-1 Archetype segmentations for the estimation of operation hours and average loads.

Building type	Operation hours		Occupancy		Appliances		Lighting	
	Time	Day	Usage rate	Load W/m^2	Usage rate	Load W/m^2	Usage rate	Load W/m^2
Detached house	00:00-0:00	7	0.60	2.8	0.60	2.4	0.10	8.0
Apartment building	00:00-0:00	7	0.60	4.2	0.60	3.0	0.10	8.0
Office building	07:00-18:00	5	0.55	7.0	0.55	12.0	0.55	12.0
Department store	08:00-21:00	7	0.60	9.3	1.00	1.0	1.00	20.0
Hotel	00:00-00:00	7	0.58	5.6	0.37	1.0	0.41	8.0
Restaurant	06:00-00:00	7	0.46	19.7	0.20	4.0	0.64	20.0
Sport, terminal, theatre	08:00-22:00	7	0.60	9.3	0.00	0.0	1.00	14.0
School	08:00-17:00	5	0.50	21.3	0.50	8.0	0.50	15.0
Daycare center	07:00-19:00	5	0.40	15.5	0.40	4.0	0.40	15.0
Hospital	00:00-00:00	7	0.54	10.8	0.62	4.0	0.62	9.0

Other building information also needs to be defined, such as WWR, building envelopes, and occupancy schedules. WWR and envelope properties are required for the calculation of solar radiation through glazing and conductive heat transfer through building exterior walls. Occupancy information is needed for the calculation of internal heat gain from occupants. For this information, an archetype library of 1,700 archetypes classified according to the ages and types is defined in this work. Here, buildings were classified by three segmentation parameters: 1) building type for occupancy and internal load estimation, 2) building type for WWR estimation, and 3) year of constructions to estimate building envelope properties. For the internal load estimation, following the ASHRAE 90.1 building code (Thornton et al., 2011), the building stocks were divided into 10 groups for the estimation of operation hours and average loads (Ahmed et al., 2017) as summarized in Table 8-1.

The building usage rates are calculated from the average usage rates of the building during operational hours. The geometrical datasets here do not provide building WWR information. Therefore, the WWR is considered as an archetype parameter. Following the recommendations from the U.S. Department of Energy (Liu, 2018), 17 reference building types were considered with the WWR values ranging from 11% to 90% (Winiarski et al., 2007). For the year of construction, 10 construction periods were considered by using the dataset averaged over the data provided by the Quebec construction code of different years (RdbQ, 2018) as shown in Table 8-2.

Table 8-2 Archetype segmentation by year of construction for the estimation of building envelope characteristics.

Period	**U-value** $(\boldsymbol{W/m^2K})$			
	Roof	Wall	Floor	Window
1958	1.42	1.7	0.45	5.7
1964	1.42	1.7	0.45	5.7
1974	0.6	1.0	0.45	5.7
1981	0.35	0.6	0.45	5.7
1990	0.25	0.45	0.45	5.7
1995	0.25	0.45	0.45	3.3
2002	0.25	0.35	0.25	2.0
2006	0.25	0.35	0.25	2.2
2010	0.25	0.35	0.25	2.2
2013	0.18	0.26	0.25	1.6

As mentioned previously, the historical weather data during the SOC from the Montreal-Pierre Elliott Trudeau International Airport station were used including dry bulb temperature, wind velocity, wind direction, and total solar radiation. As for the computing costs and resources, the total number of grids in the CityFFD simulation is around 10 million and the computational time of each time step is almost 12 minutes on a PC with 12 GB RAM and the Intel(R) Core(TM) i7-4790 CPU@3.60GHz and the NVIDIA GeForce GTX 745 graphic card. The computational time of the CityBEM solver for all buildings was about less than 3 seconds for onetime-step, which is considerably faster than a typical city scale analysis.

8.3.3. Model verification

Uncertainties in input data are one of the main limitations of UBEM models (Mosteiro-Romero et al., 2017). Occupant behavior is an important factor in the uncertainty of the physics-based models (Sun and Hong, 2017). The archetype library used in UBEM tools is another source of uncertainty which can affect the accuracy of the simulations (Cerezo et al., 2017). Therefore, validation and calibration processes are important in UBEM tools for better simulation. Most of UBEMs have been calibrated using monthly, or yearly measured data and only a few of them have been validated against hourly measured data.

In this case study, the annual space heating and cooling energy consumption in the buildings were calculated and compared with the corresponding metered data provided by the Hydro-Quebec (Hydro- Québec, 2018) (see Figure 8-8). Hydro-Quebec provides the annual detailed electricity consumption of buildings. In Canada, about 63 percent of residential building energy use is for space heating and cooling (Natural Resources Canada, 2013). To compare to Hydro-Quebec electricity consumption data, it is necessary to find the buildings using mostly electricity for heating and cooling. In Montreal, most of the residential buildings built before 1990 use electricity for space heating. Therefore, for the island modeled in this study, about 170 single-family residential building, most of which were built before 1990, was selected for the comparison. To calculate the annual heating and cooling energy consumption of the building, monthly average weather data were used in the CityFFD model as shown in Figure 8-7. The outdoor airflow and temperature were simulated for 12 months of the year by CityFFD at quasi-state simulations to obtain average local temperature around buildings' surfaces for CityBEM to calculate the annual thermal load of the buildings. The hourly thermal load was then multiplied by the number of hours per month to calculate the total monthly and then annual load of a building.

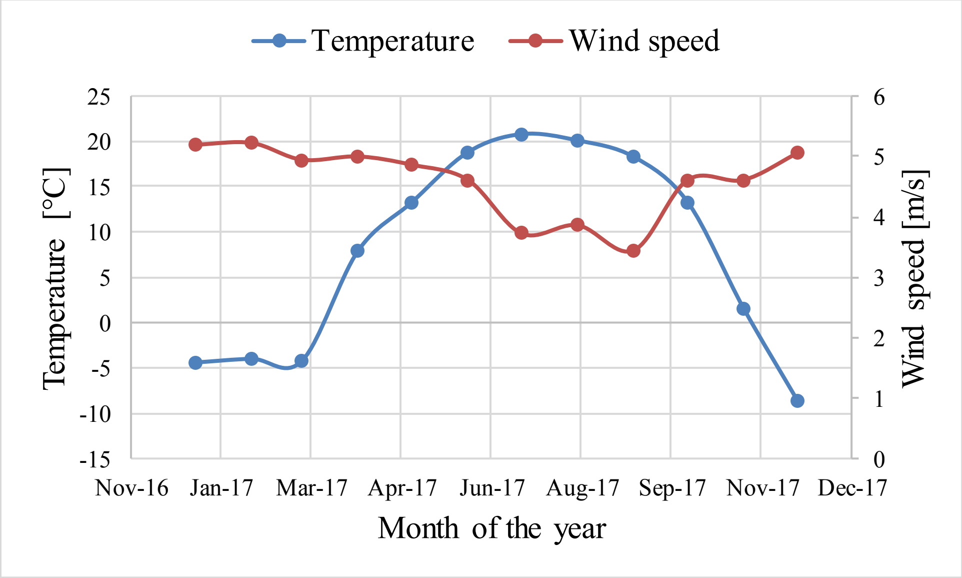


Figure 8-7 Monthly average weather data used for the calculation of the annual thermal load of the buildings.

Figure 8-8 shows the histogram of error percentage distribution in predicting the annual space heating and cooling energy consumption. The average error of the predicted energy consumption is 47%. According to the ASHRAE standard the average error for monthly energy consumption of a single building is around 15% (Quan et al., 2015) but for the urban level building energy simulation average error reported by previous works is around 69% (Quan et al., 2015) which has been considered the acceptable level of accuracy for UBEM, because of more uncertainties in the assumed data and modeling parameters. In this study, some of the buildings show the error greater than 100%, probably because some buildings may use natural gas heating instead of electrical heaters. The lack of data regarding space heating type details could be therefore one of the major sources of uncertainty, which may be reduced with more data from other official sources.

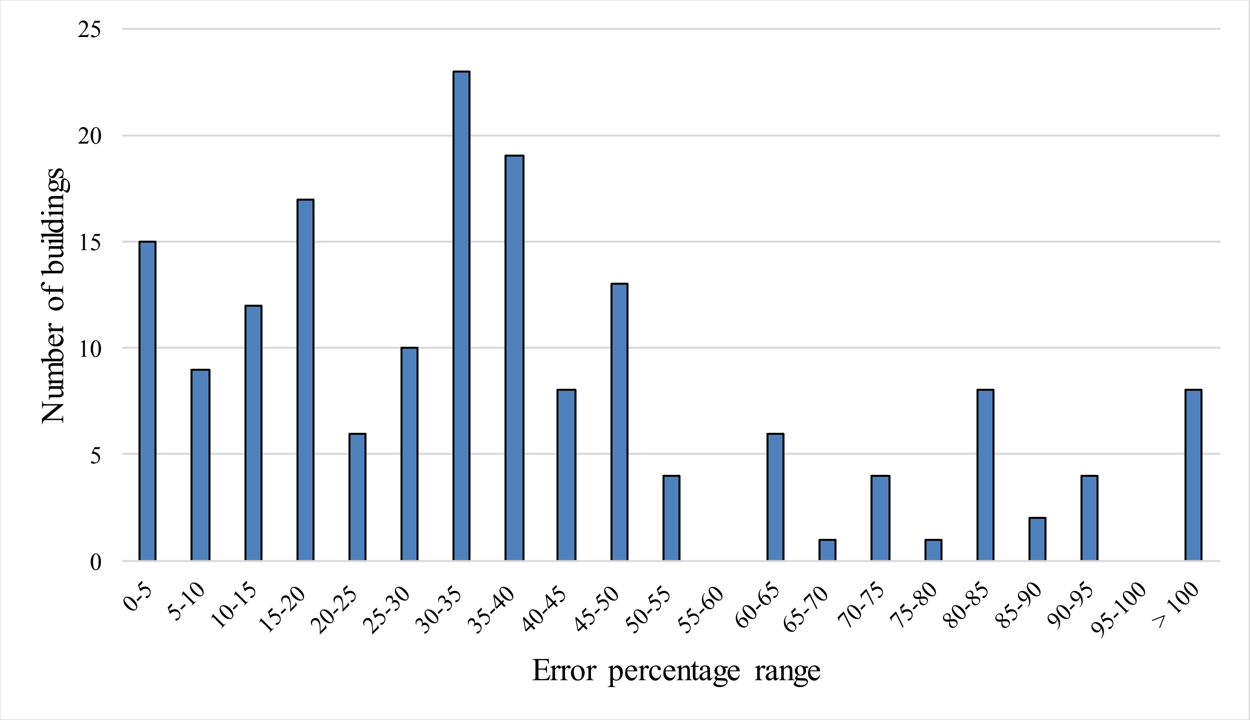


Figure 8-8 Histogram of error in calculating the annual energy consumption of buildings.

8.3.4. Modeling building resilience during the snowstorm of the century

One of the major applications of urban energy modeling is the investigation of the city/community thermal (or energy) resilience against extreme weather events. In this study, I modeled one historical snowstorm on March 4th, 1971, the so-called Snowstorm of the Century (SOC) of Montreal. 47 cm snow was dumped on Montreal with a maximum 110 km/h wind, resulting in broken power lines and cables and a major power outage lasting for full ten days on the island. Because most of the households in Montreal relied on electric heating, the power outage caused a severe drop in indoor air temperature in these households. In this section, three consecutive days after the storm were simulated by the proposed integrated urban model to study the snowstorm impact on indoor and building temperatures of the Ile-des Soeurs community with a focus on the thermal resilience. Figure 8-9 shows the input weather data of these three days which is provided by Environment Canada (ECCC. Historical Data).

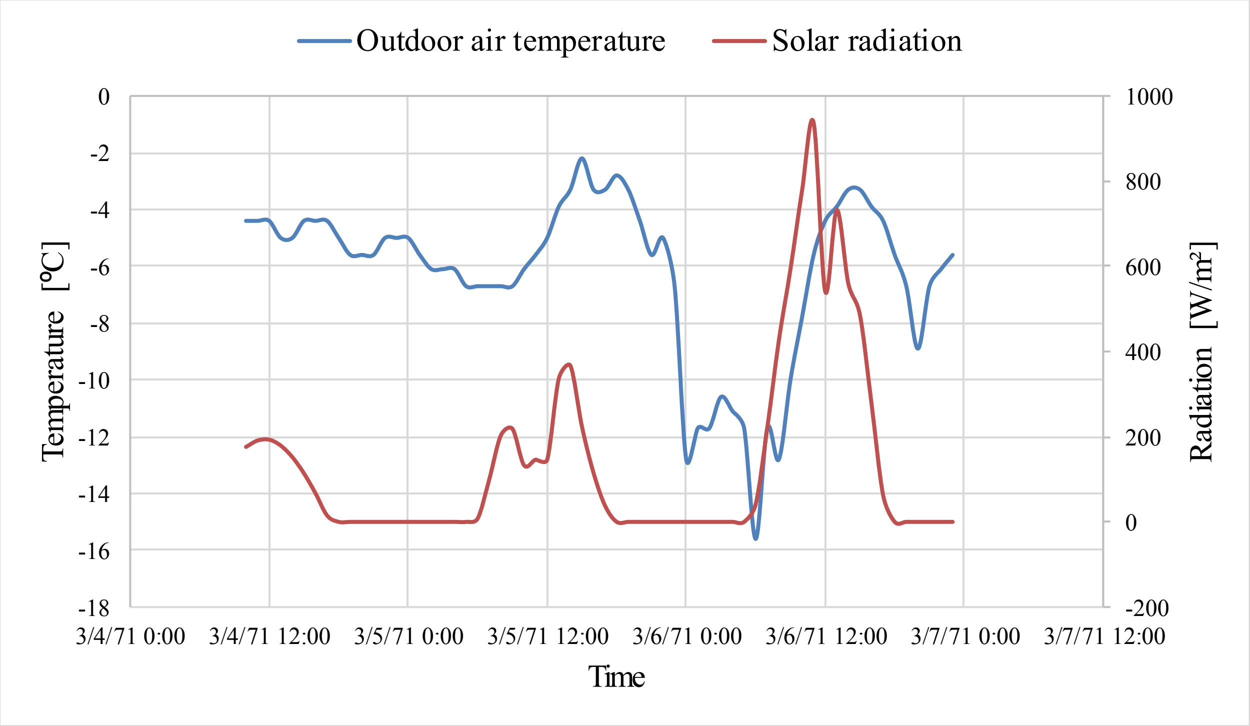


Figure 8-9 Input weather data during the March 4-6 snowstorm in 1971.

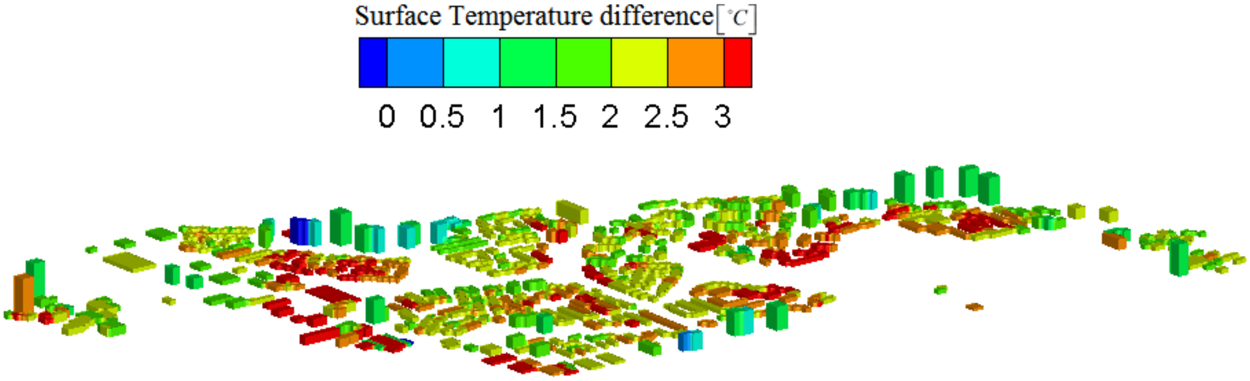


Figure 8-10 Buildings surface temperature difference with and without using the local microclimate data calculated by CityFFD.

In order to show the necessity of using local microclimate data provided from CityFFD for CityBEM simulation, first, I did the CityBEM simulation with constant weather data for all the buildings. Then, the simulation has been repeated using integrated CityFFD and CityBEM model considering the effect of local microclimate on CityBEM simulation. The difference of calculated buildings surface temperature at a selected time step is shown in Figure 8-10. The average building surface temperatures calculated by the integrated model (with local detailed microclimate conditions considered) is about 2.5 °C higher than that by using the historical weather data from one Environment Canada weather station for the March 4 – 7, 1971 for all buildings. The increased surface roughness parameters in urban areas than non-urban regions, where weather stations are often located, often result in lower wind velocities and thus lower building exterior heat transfer effects (Bornstein and Johnson, 1977). CityFFD is able to capture these local variations of wind and heat transfer effects and their impacts on the reduced heat loss from building exterior surfaces. So, the building exterior surface temperatures become generally higher than those by using the weather data (without local variations and applied to all buildings) from the weather station. With and without the local weather conditions considered, the calculated building surface temperatures vary from 0 to 3 °C, depending on different building' materials and convective heat transfer coefficients as a function of local aerodynamics around buildings. The temperature difference of 3 °C could become critical in terms of building survivability and resilience in the power outage condition during storms as discussed in the later section (see Section 8-3.5).

To show the effect of local aerodynamics on building's thermal performance, the effects of local wind conditions (i.e. different wind directions and speeds) around a group of buildings are shown in Figure 8-11. The left figure shows the results for the south wind of $1.39\ m/s$ at 2:00 pm and the right is for the east wind of $2.22\ m/s$ at 11:00 pm of March 6th. In comparison, the local airflow patterns and velocities around the buildings varied significantly. A closer look near a high-rise building highlighted by the arrow reveals that the average wind velocity around the specified surface of the building is $0.1\ m/s$ and $6\ m/s$, respectively. In the left figure, the airflow is perpendicular to the particular surface as indicated by the arrow, and the wind velocity near the wall is significantly low, creating a stagnation point. While in the right figure the airflow is parallel to the surface. Therefore, the air moves over the surface without considerable resistance and the local wind velocity is higher than the left. Higher wind velocity leads to a higher rate of surface convection (see Appendix 3). The resultant convective heat transfer coefficient is 3.95 $(W/m^2 \cdot ℃)$ for the left case, and 41.18 $(W/m^2 \cdot ℃)$ for the right.

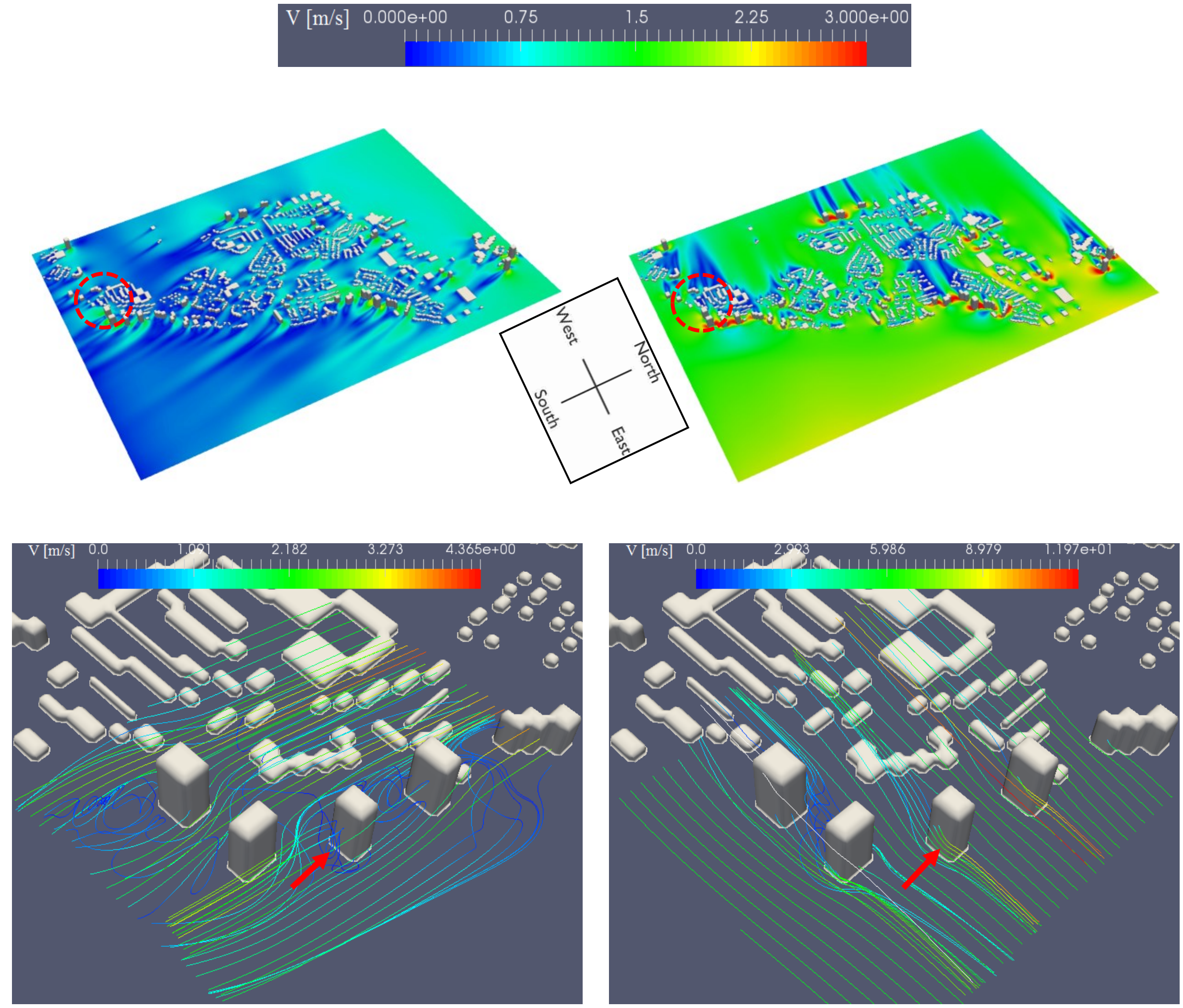
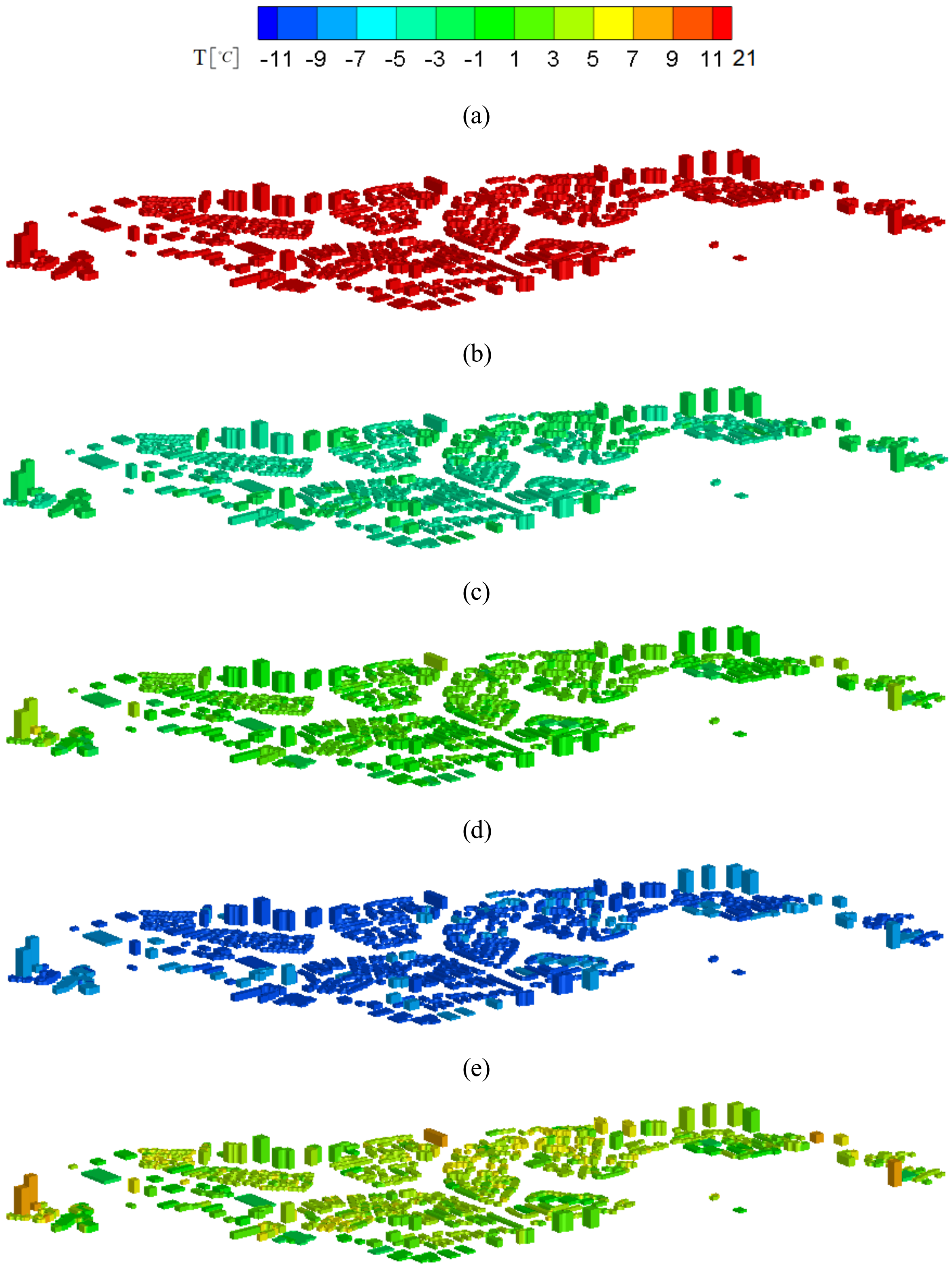


Figure 8-11 Local microclimate variations with different wind directions and speeds modeled by CityFFD: left (1.39 m/s south wind) and right (2.22 m/s east wind).



(a)

(b)

(c)

(d)

(e)

(f)

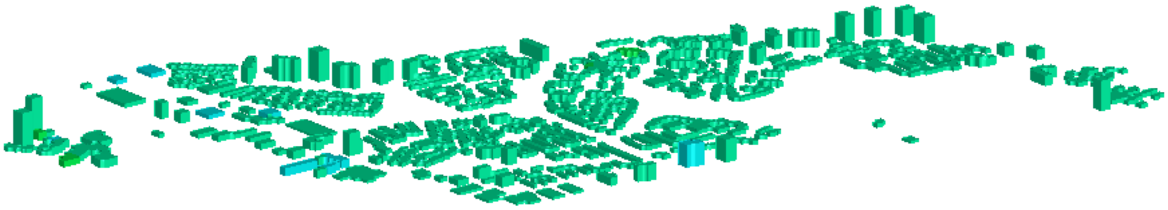


Figure 8-12 Buildings temperature map during power outage caused by snowstorm, from top to bottom: (a). March 04, 13:00, (b). March 05, 01:00, (c). March 05, 13:00, (d). March 06, 01:00, (e). March 06, 13:00, and (f). March 07, 01:00.

Figure 8-12 shows the building indoor air temperatures at different times after the power outage. The initial temperatures for all buildings were set to be around 21 °C at about 13:00, March 4th, 1971. After about 12 hours to 1:00, March 5th (Figure 8-12(b)), most of the building temperatures dropped below zero, although the ambient temperature did not decrease significantly during the same period, i.e. < 1 °C, as shown in Figure 8-12. Later around 13:00, March 5th (Figure 8-12(c)), with the increases of both the ambient temperature to around -4 °C and the solar radiation, the temperatures returned to above zero degree for most of the buildings. Then due to the sudden decrease of the ambient temperature to around -12 °C, all buildings reached the lowest temperatures at about 1:00 of March 6th (Figure 8-12(d)): the lowest temperature could be close to the ambient level, less than -11 °C, for many buildings. In the following day of March 7th (Figure 8-12(e)), the solar radiation peaked around the noon, so all buildings were heated returning to the above-zero-degree level: some buildings could be as high as around 8 °C under the "sunshine". All the temperatures then later again dropped back to below zero at the night after 12 hours (Figure 8-12(f)). Therefore, the time history of the calculated indoor temperatures illustrates the major impacts of the ambient conditions on indoors, especially the solar radiation. It also shows that without heating, the temperatures dropped so quickly to below zero just within a few hours for all buildings, although there existed a certain level of delays for some buildings.

One way to identify the group of buildings most vulnerable to the SOC is to compare the resultant temperatures at one of the worst scenarios during the storm. Figure 8-13 shows the indoor temperatures when the outdoor temperature was the lowest during the storm. The highest temperatures were observed for quite a few high-rise multi-unit residential buildings (MURBs) at the left (the North), lower (the West), and the right sides (the South) of the island. These are the newer constructions and the left-side and the right-side communities are the newest ones. In comparison, some MURBs were with the lowest temperature, e.g. a few buildings in the lower part of the island, because these buildings were older buildings. Therefore, the year of construction is one of the most significant parameters in terms of storm vulnerability. On the other hand, for buildings with similar years of constructions, e.g. for the right-side (the southern) community, some low-rise residences showed lower temperatures than the high-rises nearby. These high-rises were insulated as well as the low-rises and have more surface areas for benefiting from passive solar heating. This indicates that the building type also plays an important role in terms of vulnerability.

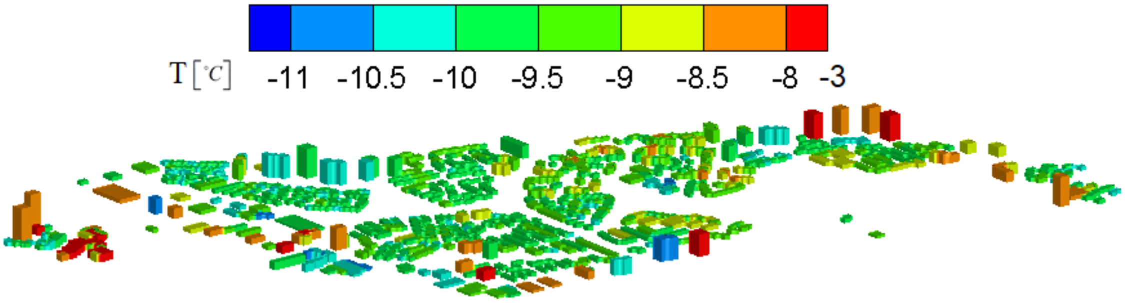


Figure 8-13 Buildings indoor air temperature at the lowest outdoor temperature during the storm.

A more vulnerable building is less resilient against an extreme weather event. Here, one way to define "building resilience" is that a building remains safe to occupy during a power outage. For example, if a building is well-insulated or designed to remain functional probably from passive solar heating in the winter, it is considered being resilient enough to sustain the power outage condition. A parameter, the Passive Survivability-Winter (PSW) (O'Brien and Bennet, 2016), is thus defined and applied here to evaluate the resilience of buildings, which is the time (in hours) from when heating is shut off to when the indoor operative temperature reaches 15 °C (59 °F) from an original heating set-point of 21°C (70°F). Figure 8-14 shows the calculated PSW values for all buildings. The PSW varies from 1 to more than 3 hours depending on building types and year of constructions. Specifically, the rate of temperature decrease varied among buildings due to various building envelope materials, occupancy schedule, WWR, and local microclimate data. It should be noted that within the immediate few hours of the power outage, the sky was cloudy, and the calculated solar heat gain was found to be negligible, so the building temperatures dropped relatively fast. Higher buildings seem to have lower PSW values when compared to low-rise residences on the island. Without the added benefits from solar heating, these buildings have greater footage and surface areas and thus are subject to higher heat losses. Therefore, the building resilience against the extreme cold events is closely related to building type, year of construction and ambient condition, especially solar radiation, for the current study. Although the ambient conditions may be predicted, they are not controllable. In comparison, the first two factors are directly related to the thermal insulation levels, which can be managed to improve the survivability and energy performance. This study, therefore, demonstrates the possibility of retrofitting techniques to enhance building resilience.

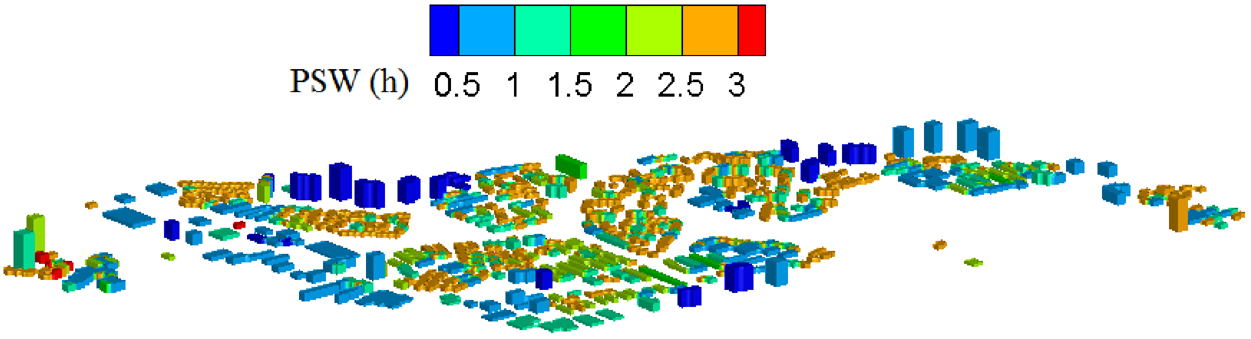


Figure 8- 14 Calculated building resilience in terms of "Passive Survivability-Winter" (PSW) after the power outage.

8.3.5. Building retrofit to improve resilience

Another major application of the proposed model is to evaluate building retrofitting strategies to improve their resilience against the winter power outage, for example, adding more thermal insulations is one of the common choices. Here, an 80-mm thick expanded polystyrene (EPS) layer was added to the outer surface of all the external walls of all buildings of the island. The R-value of the EPS layer is around 2.1 $m^2 K/W$, which doubles the insulations for old high-rise residential buildings, which are among the vulnerable buildings as shown in Figures 8-13 and 8-14. Figure 8-15 compares the temperature profile of one selected old high-rise building (one of the four buildings in the top right part in Figure 8-13) before and after adding the extra insulation layer. The PSW value increases about two more hours with the added insulation. Interestingly, it was also found that the extra insulation also prevents the building from reaching to 0 ℃ and thus avoid the freezing of water pipes and other problems for the whole duration of the three-day power outage. Therefore, in terms of the resilience against property damages, a single layer of extra insulation seems to be quite effective against the power outage for the building under consideration. For other buildings, a single retrofit measure may not be enough for keeping them from freezing, so it is also possible to evaluate other retrofitting options, preferably based on building specifics, e.g. using thermal storage and emergency heating devices etc. This case study is mostly for the demonstration purpose and more detailed analysis can be explored further in the future studies.

On the other hand, as noted previously from Figure 8-10, the calculated temperatures with the local microclimate information from CityFFD are on the average 2.5 °C higher than without it. For the selected building in Figure 8-15, this may be translated into that the building could remain above the freezing temperature for most of the time even without adding the insulation layer since the lowest temperature is around -3 °C. Accordingly, the current level of insulation for retrofitting may be overestimated and could be reconsidered for economic concerns. Moreover, for the same building, when the extra insulation layer was added, this temperature difference also means a few more hours of survivability time (i.e. PSW), which are critical for occupants and buildings themselves during extreme weathers. This analysis again shows the importance of including urban local microclimate into urban building energy model when evaluating building thermal response and resilience against weather extremes.

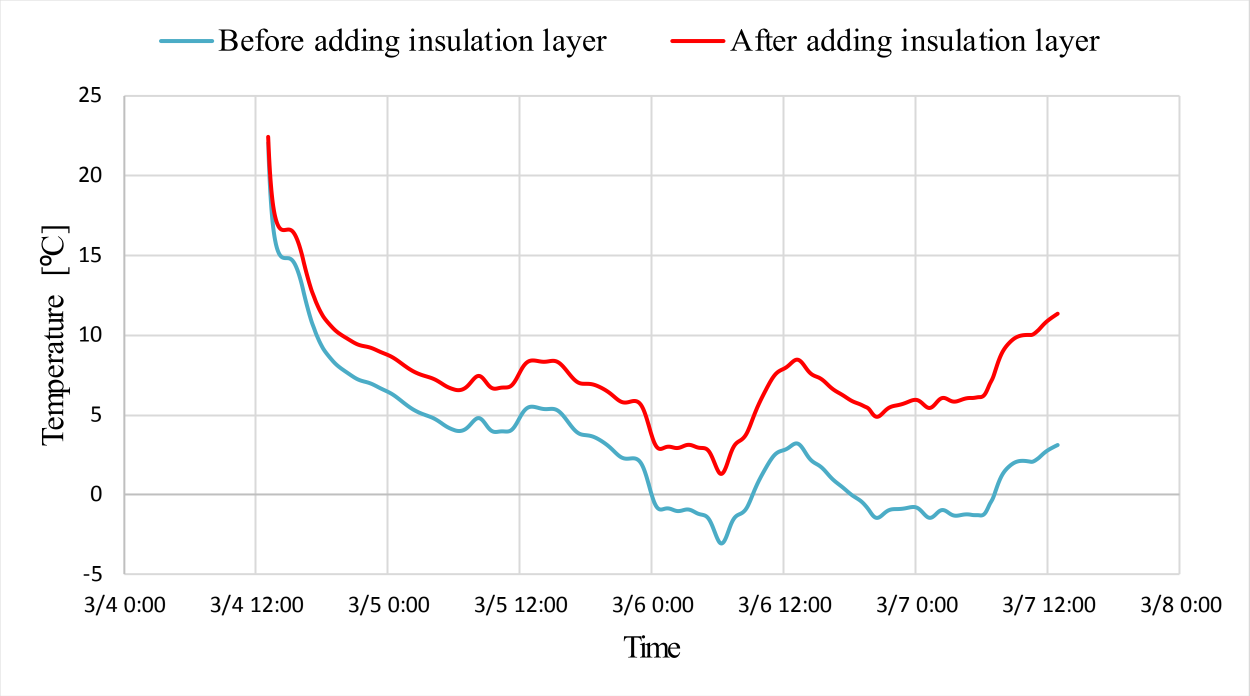


Figure 8-15 Effect of installing an insulation layer to the exterior surface of an old high-rise residential building on enhancing building resilience.

8.4. Summary and Conclusion

This study introduces a new urban-scale simulation framework of integrating a fast fluid dynamics model, CityFFD, and an urban building energy model, CityBEM, with the focus on exchanging local microclimate data between these two models at an urban scale. The microclimate from CityFFD can affect local infiltration and heat transfer through building facades. To simulate such a microclimate, CityFFD applies the high-order backward and forward sweep interpolation scheme and the semi-Lagrangian approach to providing the local aerodynamic information around buildings. With this information, CityBEM predicts heating/cooling load and building indoor and surface temperatures, which are then applied as the input boundary conditions for CityFFD at the following time-step simulation. A case study of about 1,500 buildings in the *Ile-des Soeurs*, Montreal, Canada was investigated for the effectiveness of the proposed integration model. A comprehensive archetype library of 1,700 archetypes was developed from this work to cover a wide range of building types. The results show that the proposed model can provide high-resolution results of local microclimate and airflow data around each individual building for a better prediction of building thermal responses and loads. The building resilience against the power outage during the 1971 snowstorm of the century in Montreal was studied and a retrofitting analysis was also conducted by adding external insulation layer to improve the resilience of the buildings.

My future work will focus on the further development of both CityFFD and CityBEM models and their applications for other resilience analysis for buildings and infrastructures, e.g. under extreme weather conditions, summer overheating and heat waves, and future climate changes. The plan includes also adding pollutant dispersion models for the study of outdoor and indoor characteristics such as air quality at urban scales, where because of many sources of pollutants, the airflow around the buildings and neighborhood effects can play an important role on the pollutant distribution, and thus human health. A fast and accurate simulation framework of the integration of the models like the proposed CityFFD and CityBEM is therefore needed for these problems.

Chapter 9 Conclusions and Future Work

9.1. Conclusions

This research established a fast and accurate CFD model to simulate urban microclimates even on personal computers. The proposed microclimate model (CityFFD) is a spatial and temporal high-order algorithm and capable of high accurate simulation by using large time step and coarse grid, which are two inevitable features of modeling urban scale problems. Here, it has concluded that:

- Currently, conventional CFD solvers are not suitable for modeling large scale problems, such as city/urban scales, because of some constraints such as CFL condition. Additionally, there are some CFD software developed for urban microclimates. But none of them are GPU based and cannot run big cases on personal computers.
- FFD method is a fast and unconditionally stable CFD model. But the drawback of typical FFD methods is low accuracy, especially on the coarse grids and large time steps. They may generate dissipation errors on coarse grids and near recirculation regions. To overcome these problems, a group of novel numerical algorithms has been developed.
- A new 4th-order interpolation scheme, backward forward sweep interpolating, was proposed in order to highly reduce dissipation errors even on coarse grids. The proposed model can also control dispersion errors near the sharp gradients. The other feature of this scheme is the speedup of the calculation in comparison with the conventional 4th-order schemes, because of using fewer arithmetic operations.
- Then, it is noted that the time step may affect the accuracy of a semi-Lagrangian method. In fact, a time step directly affects the position of departure points and consequently varies the magnitude of truncation error. Here, I demonstrated the smaller time step cannot always

guarantee better accuracy. So, by developing an adaptive time step based on minimizing the truncation errors, I could control the error growth during the simulation.

- Using large time step can affect the estimation of characteristic curves and generate errors in transient simulation and complex airflow problems, such as airflows with unsteady vortexes. In the conventional semi-Lagrangian methods, characteristic curves are estimated by the assumption of constant velocity which means the particles move along a straight line. But the flow's particles can have acceleration and then curvature in their path line. This assumption can generate the deviation between the estimated position of the particles and the real position, particularly for very large time steps. Here, I proposed an idea to estimating the characteristic curves by considering the velocity and acceleration of the particles. The proposed algorithm improves the temporal accuracy of the semi-Lagrangian method to the 2nd-order.
- My study demonstrated that the most time-consuming part of FFD method is the solution of the Poisson equation to calculate pressure domain. The condition number of the coefficient matrix in the Poisson equation is large leading to slow convergences and fluctuations. Here, by using the concept of artificial compressibility method, I replaced Poisson equation with the mass conservation equation, which is a hyperbolic equation and can be easily solved by a marching process in time. The proposed model, the Artificial Compressibility Semi-Lagrangian method (SLAC), is almost three times faster than conventional semi-Lagrangian methods in the case investigated, and the speedup rate can be further increased for larger problems, which may suffer from greater condition numbers of the Poisson equation and thus poorer convergences. The proposed method is also more compatible for parallel computing such as GPU programming.

- LES has been known for modeling urban microclimates and the LES-SGS model was added into CityFFD, which is then validated by the data from the literature. The CityFFD with LES was then demonstrated for the simulation of the whole Montreal island on a personal computer. The domain size is significantly larger than typical microscale problems simulated in the literature with an acceptable computing cost.
- At the end, CityFFD was applied to an extreme event in an urban area to demonstrating the impact of the urban microclimate on the building indoor environment. This work was based on a two-way interaction between CityFFD and CityBEM, which is a building energy model. I concluded that the local environment has a significant effect on the indoor conditions and building energy and thermal performance.

9.2. Future work

- For urban microclimate modeling, there are other important parameters which can affect human health and comfort, and building performance, such as humidity, solar radiation, pollutant dispersion, vegetation effect, and wind-driven rain. For example, humidity and solar radiation are two important health and comfort parameters during heatwaves. Adding these new components to CityFFD for a better understanding of the urban microclimate is necessary and important.
- The current turbulence model in CityFFD does not have wall functions so I cannot apply coarse grids to model the walls. Using wall function methods for turbulence models can help reduce the number of computational grids and may further speed up the simulation.
- Modeling very large-scale problems, such as 1 billion computational cells, needs some extra consideration for the GPU calculation. At this moment, CityFFD can run the cases around 200,000,000 cells but may have some technical problems for 1 billion grid case. To overcome

this issue, I may need to decompose the coefficient matrixes to the sub-matrixes. The boundary conditions in CityFFD are calculated by OpenMP. Meanwhile, the performance of CityFFD can be further improved by porting as many as calculations to GPU.

- Another issue which is particularly important for an accurate simulation of urban microclimate is to define accurate boundary conditions. Currently, weather station data are used for modeling the transient problems. Accurate boundary conditions for a microscale problem can be provided from mesoscale models (MMM), such as WRF and GEM-SURF. A two-way or one-way interaction between CityFFD and MMM models can provide a powerful tool for modeling urban microclimates.
- More validation of CityFFD for large scale simulations is also important. I have validated the model by using the data from the literature, such as wind tunnel test data. At urban scales, it is challenging to find the measured and detailed urban microclimate data for validation. Therefore, it is necessary to find a new way to validate CityFFD for modeling urban/city scale problems, e.g. comparing CityFFD to MMM, GEM-SURF and/or WRF etc.

References

Addepalli, B. and Pardyjak, E.R., 2015. “A study of flow fields in step-down street canyons.” Environmental Fluid Mechanics, 15(2): 439-481.

Ahmed, K., Akhondzada, A., Kurnitski, J. and Olesen, B., 2017. “Occupancy schedules for energy simulation in new prEN16798-1 and ISO/FDIS 17772-1 standards.” Sustainable cities and society, 35: 134-144.

Akbari, H., 2005. “Energy saving potentials and air quality benefits of urban heat island mitigation.” (No. LBNL-58285). Ernest Orlando Lawrence Berkeley NationalLaboratory, Berkeley, CA (US).

Amado, M., Poggi, F., Ribeiro Amado, A. and Breu, S., 2018. “E-City Web Platform: A Tool for Energy Efficiency at Urban Level.” Energies, 11(7): 1857.

ANSYS, 2011. “ANSYS FLUENT user’s guide - release 14.0”. Canonsburg, PA: ANSYS.

ASHRAE., 2013. “ASHRAE Handbook–Fundamentals.” Atlanta: ASHRAE.

Armaly, B.F., Durst, F., Pereira, J.C.F. and Schönung, B., 1983. “Experimental and theoretical investigation of backward-facing step flow.” Journal of fluid Mechanics, 127: 473-496.

Arnone, A., Liou, M.S. and POVINELLI, L., January 1993. “Multigrid time-accurate integration of Navier-Stokes equations.” In 11th Computational Fluid Dynamics Conference, 3361.

Athanailidi, P., Tenu, V. and Chronis, A., 2014, April. “Tensegrity systems acting as windbreaks: form finding and fast fluid dynamics analysis to address wind funnel effect.” In Proceedings of the Symposium on Simulation for Architecture & Urban Design. Society for Computer Simulation International, 2.

Ayhan, D. and Sağlam, Ş., 2012. “A technical review of building-mounted wind power systems and a sample simulation model.” Renewable and sustainable energy reviews, 16(1): 1040-

1049.

Bahadori, M.N., 1985. “An improved design of wind towers for natural ventilation and passive cooling.” Solar Energy, 35(2): 119-129.

Beggs, J.S., 1983. “Kinematics.” Taylor & Francis. ISBN 0-89116-355-7.

Bell, M.L., Davis, D.L. and Fletcher, T., 2004. “A retrospective assessment of mortality from the London smog episode of 1952: the role of influenza and pollution.” Environmental health perspectives, 112(1): 6-8.

Berger, M.J. and Oliger, J., 1984. “Adaptive mesh refinement for hyperbolic partial differential equations.” Journal of computational Physics, 53(3): 484-512.

Berger, M.J., 1985. “Stability of interfaces with mesh refinement.” Mathematics of computation, 45(172): 301-318.

Berge, E., Gravdahl, A.R., Schelling, J., Tallhaug, L. and Undheim, O., 2006, February. “Wind in complex terrain. A comparison of WAsP and two CFD-models.” In Proceedings from EWEC, 27.

Biswas, G., Breuer, M. and Durst, F., 2004. “Backward-facing step flows for various expansion ratios at low and moderate Reynolds numbers.” Journal of fluids engineering, 126(3): 362-374.

Blazek, J., 2005. “Computational fluid dynamics: principles and applications.” CompChemBook.

Blocken, B. and Carmeliet, J., 2004. “Pedestrian wind environment around buildings: Literature review and practical examples.” Journal of Thermal Envelope and Building Science, 28(2): 107-159.

Blocken, B., Stathopoulos, T., Carmeliet, J. and Hensen, J.L., 2011. “Application of computational fluid dynamics in building performance simulation for the outdoor environment: an overview.”

Journal of Building Performance Simulation, 4(2): 157-184.

Blocken, B., Janssen, W.D. and van Hooff, T., 2012. "CFD simulation for pedestrian wind comfort and wind safety in urban areas: General decision framework and case study for the Eindhoven University campus." Environmental Modelling & Software, 30: 15-34.

Blocken, B., 2014. "50 years of computational wind engineering: past, present and future." Journal of Wind Engineering and Industrial Aerodynamics, 129: 69-102.

Blocken, B., 2015. "Computational Fluid Dynamics for urban physics: Importance, scales, possibilities, limitations and ten tips and tricks towards accurate and reliable simulations." Building and Environment, 91: 219-245.

Blocken, B., Stathopoulos, T. and Van Beeck, J.P.A.J., 2016. "Pedestrian-level wind conditions around buildings: Review of wind-tunnel and CFD techniques and their accuracy for wind comfort assessment." Building and Environment, 100: 50-81.

Bonsal, B.R., Zhang, X., Vincent, L.A. and Hogg, W.D., 2001. "Characteristics of daily and extreme temperatures over Canada." Journal of Climate, 14(9): 1959-1976.

Bornstein, R.D. and Johnson, D.S., 1977. "Urban-rural wind velocity differences." Atmospheric Environment (1967), 11(7): 597-604.

Boscheri, W., Dumbser, M. and Zanotti, O., 2015. "High order cell-centered Lagrangian-type finite volume schemes with time-accurate local time stepping on unstructured triangular meshes." Journal of Computational Physics, 291: 120-150.

Boyer, H., Chabriat, J.P., Grondin-Perez, B., Tourrand, C. and Brau, J., 1996. "Thermal building simulation and computer generation of nodal models." Building and environment, 31(3): 207-214.

Breuer, M. and Pourquie, M., 1996. "First experiences with LES of flows past bluff bodies." In

Engineering Turbulence Modelling and Experiments. Elsevier, 177-186.

Brown, G., 2001. "Sun, Wind & Light." New York: Wiley. p. 18. ISBN 0-471-34877-5.

Bouris, D. and Bergeles, G., 1999. "2D LES of vortex shedding from a square cylinder." Journal of Wind Engineering and Industrial Aerodynamics, 80(1): 31-46.

Brunekreef, B. and Holgate, S.T., 2002. "Air pollution and health." The lancet, 360(9341): 1233-1242.

Bruse, M., 2004. "ENVI-met 3.0: updated model overview." University of Bochum. Retrieved from: www. envi-met. com.

Carlini, E., Falcone, M. and Ferretti, R., 2006. "A Time-Adaptive Semi-Lagrangian Approximation to Mean Curvature Motion." In Numerical mathematics and advanced applications. Springer, Berlin, Heidelberg, 732-739.

CBC. 45 years ago, "storm of the century" hits Quebec | CBC News 2016. https://www.cbc.ca/news/canada/montreal/storm-of-the-century-1.3477182 (accessed December 24, 2018).

Cerezo, C., Dogan, T. and Reinhart, C., 2014, September. "Towards standardized building properties template files for early design energy model generation." In Proceedings of ASHRAE/IBPSA conference.

Cerezo, C., Sokol, J., AlKhaled, S., Reinhart, C., Al-Mumin, A. and Hajiah, A., 2017. "Comparison of four building archetype characterization methods in urban building energy modeling (UBEM): A residential case study in Kuwait City." Energy and Buildings, 154: 321-334.

Cermak, J.E., 1984. "Physical modelling of flow and dispersion over complex terrain." In Boundary Layer Structure, Springer, Dordrecht, 261-292.

Chan, W.R., Nazaroff, W.W., Price, P.N., Sohn, M.D. and Gadgil, A.J., 2005. “Analyzing a database of residential air leakage in the United States.” Atmospheric Environment, 39(19): 3445-3455.

Chen, Q. and Xu, W., 1998. “A zero-equation turbulence model for indoor airflow simulation.” Energy and buildings, 28(2): 137-144.

Chen, Y., Hong, T., and Piette, MA. Automatic generation and simulation of urban building energy models based on city datasets for city-scale building retrofit analysis. Appl Energy 2017;205:323–35.

Choi, H.W., Kim, D.Y., Kim, J.J., Kim, K.Y. and Woo, J.H., 2012. “Study on dispersion characteristics for fire scenarios in an urban area using a CFD-WRF coupled model.” Atmosphere, 22(1): 47-55.

Coen, J.L., Cameron, M., Michalakes, J., Patton, E.G., Riggan, P.J. and Yedinak, K.M., 2013. “WRF-Fire: coupled weather–wildland fire modeling with the weather research and forecasting model.” Journal of Applied Meteorology and Climatology, 52(1): 16-38.

Courant, R., Isaacson, E. and Rees, M., 1952. “On the solution of nonlinear hyperbolic differential equations by finite differences.” Communications on Pure and Applied Mathematics, 5(3): 243-255.

Davila, C.C., Reinhart, C.F. and Bemis, J.L., 2016. “Modeling Boston: A workflow for the efficient generation and maintenance of urban building energy models from existing geospatial datasets.” Energy, 117: 237-250.

Degroote, J., Swillens, A., Bruggeman, P., Haelterman, R., Segers, P. and Vierendeels, J., 2010. “Simulation of fluid–structure interaction with the interface artificial compressibility method.” International Journal for Numerical Methods in Biomedical Engineering, 26(3-4): 276-289.

de Vahl Davis, G., 1983. "Natural convection of air in a square cavity: a bench mark numerical solution." International Journal for numerical methods in fluids, 3(3): 249-264.

Dino C. Room heat gain 2017. http://house-indesign.blogspot.com/2017/08/rumus-heat-gain-sumber-thermal-design.html (accessed December 24, 2018).

Dolean, V., Fahs, H., Fezoui, L. and Lanteri, S., 2010. "Locally implicit discontinuous Galerkin method for time domain electromagnetics." Journal of Computational Physics, 229(2): 512-526.

Dudhia, J., 1989. "Numerical study of convection observed during the winter monsoon experiment using a mesoscale two-dimensional model." Journal of the atmospheric sciences, 46(20): 3077-3107.

ECCC. Historical Data - Climate - Environment and Climate Change Canada 2018. http://climate.weather.gc.ca/historical_data/search_historic_data_e.html (accessed December 24, 2018).

Eriksson, K., Johnson, C. and Logg, A., 2004. "Adaptive computational methods for parabolic problems." Encyclopedia of computational mechanics, Chichester, UK: John Wiley & Sons, Ltd.

Esfahanian, V., Akbarzadeh, P. and Hejranfar, K., 2012. "An improved progressive preconditioning method for steady non-cavitating and sheet-cavitating flows." International Journal for Numerical Methods in Fluids. John Wiley & Sons, Ltd., 68(2): 210-232.

Fabri, K. and Tarabusim, V., 2014. "Top-down and bottom-up methodologies for energy building performance evaluation at meso-scale level—A literature review." J. Civ. Eng. Archit. Res, 1: 283-299.

Fedkiw, R., Stam, J. and Jensen, H.W., August 2001. "Visual simulation of smoke." In

Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 15-22. ACM.

Fei, X. and Xiaohong, T., 2006. "Stability and numerical dispersion analysis of a fourth-order accurate FDTD method." IEEE transactions on antennas and propagation, 54(9): 2525-2530.

Franke, J., Hellsten, A., Schlunzen, K.H. and Carissimo, B., 2011. "The COST 732 Best Practice Guideline for CFD simulation of flows in the urban environment: a summary." International Journal of Environment and Pollution, 44(1-4): 419-427.

Franke, R. and Rodi, W., 1993. "Calculation of vortex shedding past a square cylinder with various turbulence models." In Turbulent shear flows 8, Springer, Berlin, Heidelberg, 189-204.

Gander, M.J. and Halpern, L., 2013. Techniques for locally adaptive time stepping developed over the last two decades. In Domain decomposition methods in science and engineering XX, Springer, Berlin, Heidelberg, 377-385.

Garcia-Navarro, P. and Priestley, A., 1994. " A conservative and shape - preserving semi - Lagrangian method for the solution of the shallow water equations. " International Journal for Numerical Methods in Fluids, 18(3): 273-294.

Ghia, U.K.N.G., Ghia, K.N. and Shin, C.T., 1982. "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method." Journal of computational physics, 48(3): 387-411.

Ghiaus, C., Allard, F., Santamouris, M., Georgakis, C. and Nicol, F., 2006. "Urban environment influence on natural ventilation potential." Building and environment, 41(4): 395-406.

Google. Google Maps 2018. https://www.google.ca/maps (accessed December 24, 2018).

Gousseau, P., Blocken, B., Stathopoulos, T. and Van Heijst, G.J.F., 2011. "CFD simulation of near-field pollutant dispersion on a high-resolution grid: a case study by LES and RANS for a building group in downtown Montreal." Atmospheric Environment, 45(2): 428-438.

Gracik, S., Heidarinejad, M., Liu, J. and Srebric, J., 2015. "Effect of urban neighborhoods on the performance of building cooling systems." Building and Environment, 90: 15-29.

Grote, M. and Mitkova, T., 2012. "Explicit local time-stepping methods for time-dependent wave propagation." arXiv preprint arXiv: 1205.0654.

Gutiérrez, L.M.G. and Bermejo, R., 2005. "A semi-Lagrangian level set method for incompressible Navier-Stokes equations with free surface." International journal for numerical methods in fluids, 49(10): 1111-1146.

Haklay, M. and Weber, P., 2008. "Openstreetmap: User-generated street maps." IEEE Pervasive Computing, 7(4): 12-18.

Happle, G., Fonseca, J.A. and Schlueter, A., 2017. "Effects of air infiltration modeling approaches in urban building energy demand forecasts." Energy Procedia, 122: 283-288.

Harlan, S.L., Brazel, A.J., Prashad, L., Stefanov, W.L. and Larsen, L., 2006. "Neighborhood microclimates and vulnerability to heat stress." Social science & medicine, 63(11): 2847-2863.

Harlow, F. H. and Welch, J. E. 1965. "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface." Physics of Fluids, 8(12): 2182.

Hejranfar, K., Ezzatneshan, E. and Hesary, K. 2009. "A dual-time implicit preconditioned Navier-Stokes method for solving 2D steady/unsteady laminar cavitating/noncavitating flows using a Barotropic model." Available at: https://deepblue.lib.umich.edu/handle/2027.42/84312 (Accessed: 18 September 2017).

Hensen, J.L.M., 1995, September. "Modelling coupled heat and airflow: ping pong vs. onions." In DOCUMENT-AIR INFILTRATION CENTRE AIC PROC, OSCAR FABER PLC, 253-253.

Hirsch, JJ., 2010. "eQuest introductory tutorial, version 3.64." Camarillo, CA: James J. Hirsch & Associates.

Hoffmann, K.A. and Chiang, S.T., 2000(a). "Computational fluid dynamics volume I." Engineering Education System.

Hoffmann, K.A. and Chiang, S.T., 2000(b). "Computational fluid dynamics volume I." Engineering Education System.

Holland, D.M., Lockerby, D.A., Borg, M.K., Nicholls, W.D. and Reese, J.M., 2015. "Molecular dynamics pre-simulations for nanoscale computational fluid dynamics." Microfluidics and Nanofluidics, 18(3): 461-474.

Hong, T., Chen, Y., Lee, S.H. and Piette, M.A., 2016. "CityBES: A web-based platform to support city-scale building energy efficiency." Urban Comput.

Howard, B., Parshall, L., Thompson, J., Hammer, S., Dickinson, J. and Modi, V., 2012. "Spatial distribution of urban building energy consumption by end use." Energy and Buildings, 45: 141-151.

HQ. Getting an estimate |Hydro-Québec 2018. https://www.hydroquebec.com/portail/en/web/clientele/estimation-consommation/ (accessed December 24, 2018).

http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node19.html (accessed February 18, 2019).

https://globalnews.ca/news/4338998/heat-wave-blamed-53-deaths-montreal/

Huttner, S., 2012. "Further development and application of the 3D microclimate simulation ENVI-met." Johannes Gutenberg-Universität, Mainz.

Jacobi, P., Kjellen, M., McGranahan, G., Songsore, J. and Surjadi, C., 2010. "The citizens at risk:

from urban sanitation to sustainable cities." Routledge.

Jandaghian, Z., Dorostkar, M.M. and Wang, L.L., 2018. "Flow and pollutant dispersion in urban street canyons: Semi-Lagrangian approach with zero equation turbulence model." eSim 2018, Montreal.

Jiang, B.N., Lin, T.L. and Povinelli, L.A., 1994. "Large-scale computation of incompressible viscous flow by least-squares finite element method." Computer Methods in Applied Mechanics and Engineering, 114(3-4): 213-231.

Jiang, Y., Alexander, D., Jenkins, H., Arthur, R. and Chen, Q., 2003. "Natural ventilation in buildings: measurement in a wind tunnel and numerical simulation with large-eddy simulation." Journal of Wind Engineering and Industrial Aerodynamics, 91(3): 331-353.

Jin, M., Zuo, W. and Chen, Q., 2013. "Simulating natural ventilation in and around buildings by fast fluid dynamics." Numerical Heat Transfer, Part A: Applications, 64(4): 273-289.

Jin, M. and Chen, Q., 2015. "Improvement of fast fluid dynamics with a conservative semi-Lagrangian scheme." International Journal of Numerical Methods for Heat & Fluid Flow, 25(1): 2-18.

Kalmikov, A., Dupont, G., Dykes, K. and Chan, C.P., 2010. "Wind power resource assessment in complex urban environments: MIT campus case-study using CFD Analysis."

Karagkouni, C., Fatah gen Schieck, A., Tsigkari, M. and Chronis, A., 2014. "Performance-driven facades: Analysis of natural cross-ventilation in an indoor environment with Fast Fluid Dynamics and apertures optimization based on a genetic algorithm." Simulation, 90(8): 978-990.

Katal, A., Mortezazadeh, M. and Wang, L.L., 2019. "Modeling building resilience against extreme weather by integrated CityFFD and CityBEM simulations." Applied Energy, 250: 1402-1417.

Koeltzsch, K., 2000. “The height dependence of the turbulent Schmidt number within the boundary layer.” Atmospheric Environment, 34(7): 1147-1151.

Kolokotroni, M., Giannitsaris, I. and Watkins, R., 2006. “The effect of the London urban heat island on building summer cooling demand and night ventilation strategies.” Solar Energy, 80(4): 383-392.

Kwak, D., Kiris, C. and Kim, C. S. 2005. “Computational challenges of viscous incompressible flows.” Computers & Fluids. Pergamon, 34(3): 283-299.

Kwak, K.H., Baik, J.J., Ryu, Y.H. and Lee, S.H., 2015. “Urban air quality simulation in a high-rise building area using a CFD model coupled with mesoscale meteorological and chemistry-transport models.” Atmospheric Environment, 100: o167-177.

Lauritzen, P.H., Nair, R.D. and Ullrich, P.A., 2010. “A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid.” Journal of Computational Physics, 229(5): 1401-1424.

Leroyer, S., Bélair, S., Mailhot, J. and Strachan, I.B., 2011. “Microscale numerical prediction over Montreal with the Canadian external urban modeling system.” Journal of Applied Meteorology and Climatology, 50(12): 2410-2428.

Lind, S.J. and Phillips, T.N., 2012. “Bubble collapse in compressible fluids using a spectral element marker particle method. Part 1. Newtonian fluids.” International Journal for Numerical Methods in Fluids, 70(9): 1167-1187.

Liu, Y. and Harris, D.J., 2007. “Full-scale measurements of convective coefficient on external surface of a low-rise building in sheltered conditions.” Building and Environment, 42(7): 2718-2736.

Liu B. 90.1 Prototype Building Models -- Complete Package | Building Energy Codes Program

2018. https://www.energycodes.gov/901-prototype-building-models-complete-package (accessed December 24, 2018).

Loth, E., 2000. “Numerical approaches for motion of dispersed particles, droplets and bubbles.” Progress in energy and combustion science, 26(3): 161-223.

Lyn, D.A., Einav, S., Rodi, W. and Park, J.H., 1995. “A laser-Doppler velocimetry study of ensemble-averaged characteristics of the turbulent near wake of a square cylinder.” Journal of Fluid Mechanics, 304: 285-319.

Lynch, P., 2008. “The origins of computer weather prediction and climate modeling.” Journal of Computational Physics, 227(7): 3431-3444.

Mage, D., Ozolins, G., Peterson, P., Webster, A., Orthofer, R., Vandeweerd, V. and Gwynne, M., 1996. “Urban air pollution in megacities of the world.” Atmospheric Environment, 30(5): 681-686.

Malan, A.G., Lewis, R.W. and Nithiarasu, P., 2002. “An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part II. Application.” International Journal for Numerical Methods in Engineering, 54(5): 715-729.

Martin, S.L., Cakmak, S., Hebbern, C.A., Avramescu, M.L. and Tremblay, N., 2012. “Climate change and future temperature-related mortality in 15 Canadian cities.” International journal of biometeorology, 56(4): 605-619.

Maurice, M. J., Zucrow, J., and Hoffman, J. D. 1976. “Gas Dynamics.” New York: Wiley.

Merkle, C., 1987. “Time-accurate unsteady incompressible flow algorithms based on artificial compressibility.” in 8th Computational Fluid Dynamics Conference. Reston, Virigina: American Institute of Aeronautics and Astronautics.

McNeel R. Rhino 6 for Windows 2018. https://www.rhino3d.com/ (accessed December 24, 2018).

Miao, Y., Liu, S., Chen, B., Zhang, B., Wang, S. and Li, S., 2013. “Simulating urban flow and dispersion in Beijing by coupling a CFD model with the WRF model.” Advances in atmospheric sciences, 30(6): 1663-1678.

Min, C. and Gibou, F. 2006. “A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids.” Journal of Computational Physics. Academic Press, 219(2): 912-929.

Minkoff, S.E. and Kridler, N.M., 2006. “A comparison of adaptive time stepping methods for coupled flow and deformation modeling.” Applied mathematical modelling, 30(9): 993-1009.

Mirzaei, P.A. and Haghighat, F., 2010. “Approaches to study urban heat island–abilities and limitations.” Building and environment, 45(10): 2192-2201.

Moonen, P., Defraeye, T., Dorer, V., Blocken, B. and Carmeliet, J., 2012. “Urban Physics: Effect of the micro-climate on comfort, health and energy demand.” Frontiers of Architectural Research, 1(3): 197-228.

Mortezazadeh, M. and Wang, L.L., 2016. “A fast and accurate CFD solver for indoor airflows.” in: eSIM 2016, Hamilton, Canada.

Mortezazadeh, M. and Wang, L.L., 2017. “A high-order backward forward sweep interpolating algorithm for semi-Lagrangian method.” International Journal for Numerical Methods in Fluids, 84(10): 584-597.

Mortezazadeh M, Wang L (Leon)., 2018. “Modeling urban airflows by a new parallel high-order semi-Lagrangian 3D fluid flow solver.” COBEE 2018 4th International Conference on Building Energy, Environonment, Melbourne, Australia.

Mortezazadeh, M. and Wang, L., 2019. “SLAC–a semi-Lagrangian artificial compressibility solver for steady-state incompressible flows.” International Journal of Numerical Methods for Heat & Fluid Flow.

Mosteiro-Romero, M., Fonseca, J.A. and Schlueter, A., 2017. "Seasonal effects of input parameters in urban-scale building energy simulation." Energy Procedia, 122: 433-438.

Natural Resources Canada. Energy Efficiency Trends in Canada 1990 to 2010. 2013. doi:http://oee.nrcan.gc.ca/publications/statistics/trends11/pdf/trends.pdf.

Neis, P. and Zipf, A., 2012. "Analyzing the contributor activity of a volunteered geographic information project—The case of OpenStreetMap." ISPRS International Journal of Geo-Information, 1(2): 146-165.

Nirupama, N. and Simonovic, S.P., 2007. "Increase of flood risk due to urbanisation: a Canadian example." Natural Hazards, 40(1): 25.

NVIDIA, 2011. "NVIDIA CUDA C Programming Guide Version 4.1." Santa Clara: NVIDIA Corporation.

O'Brien, W. and Bennet, I., 2016. "Simulation-Based Evaluation of High-Rise Residential Building Thermal Resilience." ASHRAE Transactions, 122(1).

OpenCFD, 2014. "OpenFOAM v2.2.2 users manual 2014." http://www.openfoam.com/

OpenMP, 2015. "OpenMP Application Programming Interface." Available at: http://www.openmp.org/wp-content/uploads/openmp-4.5.pdf (Accessed: 26 September 2017).

Orlanski, I., 1975. "A rational subdivision of scales for atmospheric processes." Bulletin of the American Meteorological Society, 527-530.

Orme, M., 2001. "Estimates of the energy impact of ventilation and associated financial expenditures." Energy and Buildings, 33(3): 199-205.

Ozkeresteci, I., Crewe, K., Brazel, A.J. and Bruse, M., 2003, August. "Use and evaluation of the

ENVI-met model for environmental design and planning: an experiment on linear parks." In Proceedings of the 21st International Cartographic Conference (ICC), Durban, South Africa, 10-16.

Pantavou, K., Theoharatos, G., Mavrakis, A. and Santamouris, M., 2011. "Evaluating thermal comfort conditions and health responses during an extremely hot summer in Athens." Building and Environment, 46(2): 339-344.

Panton, RC., 1984. "Incompressible Flow." Wiley: New York.

Patankar, S. V. and Spalding, D. B. 1972 "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows." International Journal of Heat and Mass Transfer. Pergamon, 15(10): 1787-1806.

Patankar, S., 1980. "Numerical heat transfer and fluid flow." CRC press.

Penwarden, A.D. and Wise, A.F.E., 1975. "Wind Environment around Buildings," BRE Report, Department of the Environment, BRE, HMSO, London, U.K.

Pielke, R.A., Cotton, W.R., Walko, R.E.A., Tremback, C.J., Lyons, W.A., Grasso, L.D., Nicholls, M.E., Moran, M.D., Wesley, D.A., Lee, T.J. and Copeland, J.H., 1992. "A comprehensive meteorological modeling system—RAMS." Meteorology and atmospheric Physics, 49(1-4): 69-91.

Pope, S. B. 2000. "Turbulent Flows." Cambridge University Press.

Priestley, A., 1993. "A quasi-conservative version of the semi-Lagrangian advection scheme." Monthly Weather Review, 121(2): 621-629.

Pyzara, A., Bylina, B. and Bylina, J., 2011, September. "The influence of a matrix condition number on iterative methods' convergence." In 2011 Federated Conference on Computer Science and Information Systems (FedCSIS), 459-464. IEEE.

Quan, S.J., Li, Q., Augenbroe, G., Brown, J. and Yang, P.P.J., 2015. "Urban data and building energy modeling: A GIS-based urban building energy modeling system using the urban-EPC engine." In Planning Support Systems and Smart Cities, Springer, Cham, 447-469.

Ramshaw, J. D. and Mousseau, V. A. 1990. "Accelerated artificial compressibility method for steady-state incompressible flow calculations." Computers & Fluids. Pergamon, 18(4): 361-367.

RdbQ. Construction Code and Safety Code - Régie du bâtiment du Québec 2018. https://www.rbq.gouv.qc.ca/en/laws-regulations-and-codes/construction-code-and-safety-code.html (accessed December 24, 2018).

Reboud, J.L., Coutier-Delgosha, O., Pouffary, B. and Fortes-Patella, R., 2003, November. "Numerical simulation of unsteady cavitation flows: Some applications and open problems." In Fifth International Symposium on Cavitation, 1-10.

Reinhart, C.F. and Davila, C.C., 2016. "Urban building energy modeling–A review of a nascent field." Building and Environment, 97: 196-202.

Rice, J.R., 1960. "Split Runge-Kutta methods for simultaneous equations." Journal of Research of the National Institute of Standards and Technology, 60.

Robinson, D., Haldi, F., Leroux, P., Perez, D., Rasheed, A. and Wilke, U., 2009. "CitySim: Comprehensive micro-simulation of resource flows for sustainable urban planning." In Proceedings of the Eleventh International IBPSA Conference, No. CONF, 1083-1090.

Rocha, R., 2018. "Montreal is 375 years old, but how old are her buildings?."

Rosatti, G., Chemotti, R. and Bonaventura, L., 2005. "High order interpolation methods for semi-Lagrangian models of mobile-bed hydrodynamics on Cartesian grids with cut cells." International journal for numerical methods in fluids, 47(10-11): 1269-1275.

Saad, Y., 2003. "Iterative methods for sparse linear systems." siam, 82.

SC. 2011 Census of Population Program – Data products 2011. https://www12.statcan.gc.ca/census-recensement/2011/dp-pd/index-eng.cfm (accessed December 24, 2018).

Scott, L. R., Clark, T. and Bagheri, B., 2005. "Scientific parallel computing." Princeton University Press.

Shashua-Bar, L. and Hoffman, M., 2003. "Geometry and orientation aspects in passive cooling of canyon streets with trees." Energy and Buildings, 35: 61-8.

Shirolkar, J.S., Coimbra, C.F.M. and McQuay, M.Q., 1996. "Fundamental aspects of modeling turbulent particle dispersion in dilute flows." Progress in Energy and Combustion Science, 22(4): 363-399.

Skamarock, W.C. and Klemp, J.B., 2008. "A time-split nonhydrostatic atmospheric model for weather research and forecasting applications." Journal of computational physics, 227(7): 3465-3485.

Smagorinsky, J., 1963. "General circulation experiments with the primitive equations: I. The basic experiment." Monthly weather review, 91(3): 99-164.

Smoyer-Tomic, K.E., Kuhn, R. and Hudson, A., 2003. "Heat wave hazards: an overview of heat wave impacts in Canada." Natural hazards, 28(2-3): 465-486.

Spiegelman, M. and Katz, R.F., 2006. "A semi‐Lagrangian Crank‐Nicolson algorithm for the numerical solution of advection-diffusion problems." Geochemistry, Geophysics, Geosystems, 7(4).

Stam, J., July 1999,. "Stable Fluids." In Siggraph, 99: 121-128.

Staniforth, A. and Côté, J., 1991. "Semi-Lagrangian integration schemes for atmospheric models—A review." Monthly weather review, 119(9): 2206-2223.

Stathopoulos, T., Wu, H. and Bédard, C., 1992. "Wind environment around buildings: a knowledge-based approach." Journal of Wind Engineering and Industrial Aerodynamics, 44(1-3): 2377-2388.

Stewart, G.W., 1996. "Afternotes on numerical analysis." Siam, 49.

Su, P., Tian, X., Wang, Y., Deng, S., Zhao, J., An, Q. and Wang, Y., 2017. "Recent trends in load forecasting technology for the operation optimization of distributed energy system." Energies, 10(9): 1303.

Sullivan, P.P., McWilliams, J.C. and Moeng, C.H., 1994. "A subgrid-scale model for large-eddy simulation of planetary boundary-layer flows." Boundary-Layer Meteorology, 71(3): 247-276.

Sun, K. and Hong, T., 2017. "A framework for quantifying the impact of occupant behavior on energy savings of energy conservation measures." Energy and Buildings, 146: 383-396.

Sun, Z. and Xiao, F., 2017. "A semi-Lagrangian multi-moment finite volume method with fourth - order WENO projection." International Journal for Numerical Methods in Fluids, 83(4): 351-375.

Swan, L.G. and Ugursal, V.I., 2009. "Modeling of end-use energy consumption in the residential sector: A review of modeling techniques." Renewable and sustainable energy reviews, 13(8): 1819-1835.

Takacs, L.L., 1985. "A two-step scheme for the advection equation with minimized dissipation and dispersion errors." Monthly Weather Review, 113(6): 1050-1065.

Takahashi, K., Yoshida, H., Tanaka, Y., Aotake, N. and Wang, F., 2004. "Measurement of thermal environment in Kyoto city and its prediction by CFD simulation." Energy and Buildings, 36(8): 771-779.

Taleghani, M., Sailor, D.J., Tenpierik, M. and van den Dobbelsteen, A., 2014. "Thermal assessment of heat mitigation strategies: The case of Portland State University, Oregon, USA." Building and Environment, 73: 138-150.

Tamamidis, P., Zhang, G. and Assanis, D. N. 1996. "Comparison of Pressure-Based and Artificial Compressibility Methods for Solving 3D Steady Incompressible Viscous Flows." Journal of Computational Physics. Academic Press, 124(1): 1-13.

Tian, W., Sevilla, T.A., Zuo, W. and Sohn, M.D., 2017. "Coupling fast fluid dynamics and multizone airflow models in Modelica Buildings library to simulate the dynamics of HVAC systems." Building and Environment, 122: 269-286.

Thomas, L.H., 1949. "Elliptic problems in linear difference equations over a network." Watson Sci. Comput. Lab. Rept., Columbia University, New York, 1.

Thornton, B.A., Rosenberg, M.I., Richman, E.E., Wang, W., Xie, Y., Zhang, J., Cho, H., Mendon, V.V., Athalye, R.A. and Liu, B., 2011. "Achieving the 30% goal: Energy and cost savings analysis of ASHRAE Standard 90.1-2010 (No. PNNL-20405)." Pacific Northwest National Lab.(PNNL), Richland, WA (United States).

Tominaga, Y., Mochida, A., Yoshie, R., Kataoka, H., Nozu, T., Yoshikawa, M. and Shirasawa, T., 2008. "AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings." Journal of wind engineering and industrial aerodynamics, 96(10-11): 1749-1761.

Tominaga, Y. and Stathopoulos, T., 2010. "Numerical simulation of dispersion around an isolated cubic building: model evaluation of RANS and LES." Building and Environment, 45(10): 2231-2239.

Tominaga, Y. and Stathopoulos, T., 2011. "CFD modeling of pollution dispersion in a street canyon: Comparison between LES and RANS." Journal of Wind Engineering and Industrial Aerodynamics, 99(4): 340-348.

Tominaga, Y., Sato, Y. and Sadohara, S., 2015. "CFD simulations of the effect of evaporative cooling from water bodies in a micro-scale urban environment: Validation and application studies." Sustainable Cities and Society, 19: 259-270.

Tong, Z., Chen, Y., Malkawi, A., Liu, Z. and Freeman, R.B., 2016. "Energy saving potential of natural ventilation in China: The impact of ambient air pollution." Applied energy, 179: 660-668.

Toparlar, Y., Blocken, B., Vos, P.V., Van Heijst, G.J.F., Janssen, W.D., van Hooff, T., Montazeri, H. and Timmermans, H.J.P., 2015. "CFD simulation and validation of urban microclimate: A case study for Bergpolder Zuid, Rotterdam." Building and Environment, 83: 79-90.

Toparlar, Y., Blocken, B., Maiheu, B. and Van Heijst, G.J.F., 2017. "A review on the CFD analysis of urban microclimate." Renewable and Sustainable Energy Reviews, 80: 1613-1640.

TRNSYS. 2012. "A Transient System Simulation Program, version 17.1." Madison, Wisconsin: University of Wisconsin at Madison.

Turkel, E., 1987. "Preconditioned methods for solving the incompressible and low speed compressible equations." Journal of Computational Physics. Academic Press, 72(2): 277-298.

Ueberhuber, C.W., 2012. "Numerical computation 1: methods, software, and analysis." Springer Science & Business Media.

United Nations, "World Urbanization Prospects," 2014.

Üresin, A. and Dubois, M., 1996. "Effects of asynchronism on the convergence rate of iterative algorithms." Journal of Parallel and Distributed Computing, 34(1): 66–81.

US DOE. EnergyPlus Engineering Reference. Ref to EnergyPlus Calc 2010.

VdM. Consultation du rôle d'évaluation foncière 2018. https://servicesenligne2.ville.montreal.qc.ca/sel/evalweb/index (accessed December 24, 2018).

United States Fire Administration. 2008. Fire statistics, from http://www.usfa.dhs.gov/statistics/national/index.shtm.

Waibel, C., Bystricky, L., Kubilay, A., Evins, R. and Carmeliet, J., 2018. "Validation of Grasshopper-based Fast Fluid Dynamics for Air Flow around Buildings in Early Design Stage." Accessed February, 19.

Wang, M. and Moin, P., 2002. "Dynamic wall modeling for large-eddy simulation of complex turbulent flows." Physics of Fluids, 14(7): 2043-2051.

Wang, B., Cot, L.D., Adolphe, L., Geoffroy, S. and Morchain, J., 2015. "Estimation of wind energy over roof of two perpendicular buildings." Energy and Buildings, 88: 57-67.

Wang, C., Sun, J. and Ba, Y., 2017. "A semi-Lagrangian Vortex-In-Cell method and its application to high-Re lid-driven cavity flow." International Journal of Numerical Methods for Heat and Fluid Flow, 27(6): 1186–1214.

Whittaker, E.T., 1988. "A treatise on the analytical dynamics of particles and rigid bodies." Cambridge University Press.

Winiarski, D.W., Halverson, M.A. and Jiang, W., 2007. "Analysis of building envelope construction in 2003 CBECS (No. PNNL-20380)." Pacific Northwest National Lab.(PNNL), Richland, WA (United States).

Xiao, F. and Ikebata, A., 2003. "An efficient method for capturing free boundaries in multi‐fluid simulations." International journal for numerical methods in fluids, 42(2): 187-210.

Xiu, D. and Karniadakis, G.E., 2001. “A semi-Lagrangian high-order method for Navier–Stokes equations.” Journal of computational physics, 172(2): 658-684.

Xue, Y., Liu, W. and Zhai, Z.J., 2016. “New semi-Lagrangian-based PISO method for fast and accurate indoor environment modeling.” Building and Environment, 105: 236-244.

Yang, J.Y., Yang, S.C., Chen, Y.N. and Hsu, C.A., 1998. “Implicit weighted ENO schemes for the three-dimensional incompressible Navier–Stokes equations.” Journal of Computational Physics, 146(1): 464-487.

Zerroukat, M., 2010. “A simple mass conserving semi-Lagrangian scheme for transport problems.” Journal of Computational Physics, 229(24): 9011-9019.

Zonglin, J., 2004. “On dispersion-controlled principles for non-oscillatory shock-capturing schemes.” Acta Mechanica Sinica, 20(1): 1-15.

Zuo, W. and Chen, Q., 2010(a). “Fast and informative flow simulations in a building by using fast fluid dynamics model on graphics processing unit.” Building and environment, 45(3): 747-757.

Zuo, W. and Chen, Q., 2010(b). “Fast simulation of smoke transport in buildings.” In Proceedings of the the 41st International HVAC&R congress, 1-3.

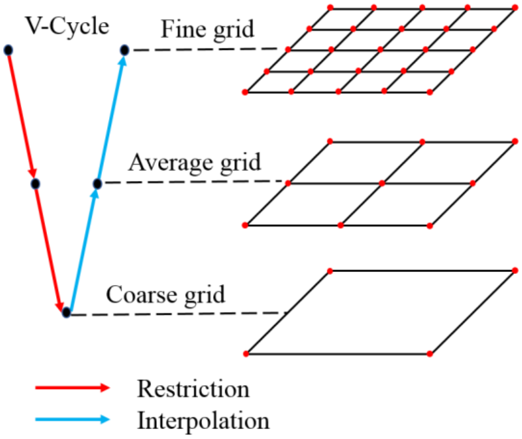
Zuo, W. and Chen, Q., 2011. “Validation of a fast-fluid-dynamics model for predicting distribution of particles with low stokes number.”

Zuo, W., Jin, M. and Chen, Q., 2012. “Reduction of numerical diffusion in FFD model.” Engineering Applications of Computational Fluid Mechanics, 6(2): 234-247.

Appendix 1 V-Cycle Multigrid

V-cycle geometric multigrid with three levels: fine grid, average, and coarse girds is shown in the following Figure. Multigrid method has three main steps:

1. Smoothing: solving the main equation on the fine grid by using a few iterations of a conventional iterative method, such as Jacobi and Gauss-Seidel methods.
2. Restriction: solving the error equation on the coarse grid by transferring the data from fine to coarse grids.
3. Interpolation or prolongation: transferring the calculated errors to fine girds and modifying the data calculated from smoothing step.



Interpolation and restriction in V-Cycle multigrid solver

Elliptic equations, Eqs. (2-3) and (2-5), can be written in a general matrix form after linearization:

$$AU_i = b_i$$

where A is the coefficients' matrix. This equation is solved by two iterations of the Gauss-Seidel method on the fine grid to find V_i, the approximate of U_i. Then I calculate the residual R_i:

$$R_i = b_i - AV_i$$

I then write the error equation $e_i = U_i - V_i$:

$$Ae_i = R_i$$

The next step is to transfer the solution of the error equation from the fine grid to the average grid, so-called the restriction step, by ten Gauss-Seidel iterations on the fine grid. For transferring, I used a first-order linear interpolation scheme in all three mesh directions. Then the whole procedure, recalculating residual, reconstructing error equation, and retransferring the error equation from average grid to coarse grid, will be repeated on the coarse grid. Then I go to the final step, the interpolation step when I transfer the results of the error equation from the coarse grid to the average grid by the linear interpolation. Here, I solve the error equation by two Gauss-Seidel iterations for smoothing followed by the linear interpolation to transfer the results of the error equation from the average grid to the fine grid. Now the new velocity values can be found from the correction:

$$U_i = e_i + V_i$$

If the convergence criterion is satisfied, I go to the next time step. Otherwise the whole procedure is repeated.

Appendix 2 Dimensionless form of governing equations for natural convection problems

$$x = \frac{x^*}{L_\infty}, u = \frac{u^*}{\alpha_\infty / L_\infty}, t = \frac{t^*}{{L_\infty}^2 / \alpha_\infty}, p = \frac{p^*}{\rho_\infty (\alpha_\infty / L_\infty)^2}, T = \frac{T^* - {T_\mathrm{C}}^*}{{T_\mathrm{H}}^* - {T_\mathrm{C}}^*}$$

$$\rho = \frac{\rho^*}{\rho_\infty} = 1, \mu = \frac{\mu^*}{\mu_\infty} = 1, \upsilon = \frac{\upsilon^*}{\mu_\infty / \rho_\infty} = 1, \alpha = \frac{\alpha^*}{\alpha_\infty} = 1, c_p = \frac{{c_p}^*}{{c_p}_\infty} = 1$$

So, Prandtl and Rayleigh Numbers are as follows:

$$Pr = \frac{\mu_\infty / \rho_\infty}{\alpha_\infty} = \frac{\alpha_\infty}{\upsilon_\infty}$$

$$Ra = \frac{g \beta_\infty {L_\infty}^3 ({T_\mathrm{H}}^* - {T_\mathrm{C}}^*)}{\alpha_\infty \nu_\infty}$$

The dimensionless governing equations are:

$$\nabla \cdot U = 0$$

$$\frac{\partial U}{\partial t} + (U \cdot \nabla) U = -\nabla p + Pr \nabla^2 U - Ra.Pr.T$$

$$\frac{\partial T}{\partial t} + (U \cdot \nabla) T = \nabla^2 T + Q$$

where:

$$Q = \frac{Q^*}{\frac{\rho_\infty {c_p}_\infty \alpha_\infty ({T_\mathrm{H}}^* - {T_\mathrm{C}}^*)}{{L_\infty}^2}}$$

Appendix 3 CityBEM

CityBEM is a building energy model developed by Ali Katal. The model treats a building as a single-block air capsule with an indoor air cavity enclosed by walls and windows. The total thermal load of the building (Q_t) consists of external and internal parts, including wall/window/roof/floor heat transfer (Q_{wall}), fenestration radiation heat $\left(Q_{fes}\right)$, and infiltration-related heat transfer $\left(Q_{inf}\right)$; and various internal loads (Q_{int}) ASHRAE handbook (2013).

$$Q_t = Q_{wall} + Q_{fes} + Q_{inf} + Q_{int}$$

Fenestration heat transfer includes the direct beam solar heat gain Q_b, diffuse solar heat gain Q_d, and conductive heat gain Q_c, which are calculated as.

$$Q_{fes} = Q_b + Q_d + Q_c$$

Building infiltration rate depends on tightness, prevailing wind speed and direction, which can be quantified by using either air change rates or leakage details.

$$Q_{inf} = F \cdot \rho \cdot C_{pa}(T_{out} - T_{in})$$

where F represents the infiltration rate (m^3/s), ρ and C_{pa} represent the density and specific heat of the infiltrated air, respectively.

The internal loads account for those from occupants, lighting, equipment, and appliances.

$$Q_{int} = Q_{occupants} + Q_{lighting} + Q_{equipment}$$

Here, all internal loads are calculated using the data provided by the archetype library in chapter 8.

Indoor air temperature calculation

The CityBEM model allows setting the indoor air temperature to be constant for the calculation of the mechanical heating/cooling energy demands or allowing the variation of indoor temperatures with the outdoor conditions during power outages.

$$mC_{pa}\frac{dT_{in}}{dt} = Q_t \rightarrow T_{in}^{t+1} = T_{in}^{t} + \frac{\Delta t}{mC_{pa}}Q_t$$

Where

m = the indoor air mass (kg)

C_{pa} = the indoor air heat capacity $(J/kg.K)$

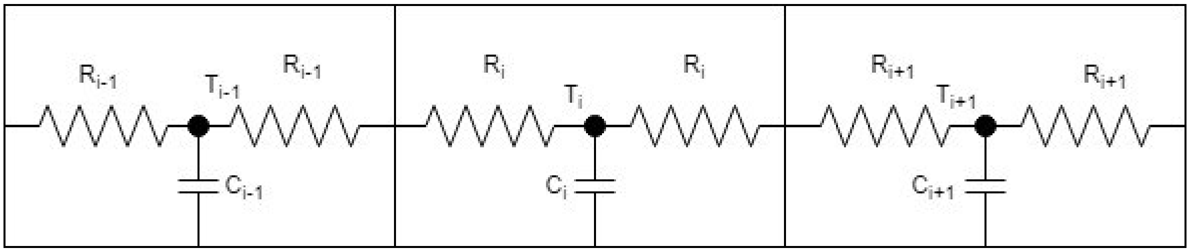
T^{t+1} = the indoor air temperature at time $t+1$ (K)

T^{t} = the indoor air temperature at time t (K)

Δt = the time step of the simulation (s)

Calculation of in-wall temperature

The heat balance equation is also solved for wall internal temperatures. Following Figure shows the thermal resistance network model used for modeling a wall. The right-hand side of the transient heat balance equation is discretized implicitly, and the tridiagonal matrix equation is solved using the Thomas algorithm. In this study, five thermal nodes were used for all the walls to calculate the temperature distribution inside the wall.



Thermal network model of a wall.

$$C_i \frac{\Delta T_i}{\Delta t} = \frac{T_{i-1}^{t+1} - T_i^{t+1}}{R_{i-1} + R_i} + \frac{T_{i+1}^{t+1} - T_i^{t+1}}{R_i + R_{i+1}} + S_i$$