Controlling Hazardous Releases While Protecting Passengers in Civil Infrastructure Systems

by

Sara P. Rimer

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Environmental Engineering)
in The University of Michigan
2016

Doctoral Committee:

Professor Nikolaos D. Katopodes, Chair
Assistant Professor Elizabeth Eve Bruch
Associate Professor Aline J. Cotel
Assistant Professor Branko Kerkez

© Sara P. Rimer 2016

All Rights Reserved

To my parents, my grandma, and Emerson;
in loving memory of my grandpa and Rita.

ACKNOWLEDGEMENTS

Over the past six years, numerous organizations and individuals have selflessly contributed to my growth as an engineer, academic, and individual; and I am forever humbly indebted to each of them.

I must first and foremost thank my advisor, Nikolaos D. Katopodes, for his continued support and mentorship. From the first day I met him when I visited the department in 2010, I have always enjoyed his sense of humor and perspective on our chosen profession, and have admired his accomplishments and vision for his research. I am so grateful for the compelling research project he developed that I was able to pursue for my dissertation–a project that has challenged me and allowed me to develop skills in computational fluid dynamics *as well as* physical modeling of fluid dynamics, *and* the bridging of the physical sciences with social science. His guidance and mentorship has been integral to my success in graduate school, and he has served as the singular teaching inspiration in all of my educational pursuits. His work philosophy, standard for high-quality research, and pure delight exuding from him in all of his lectures will motivate me through all of my career.

The other professor who has had the most significant impact on my doctoral education, and for whom I will always be so grateful to is Professor Aline Cotel. Aline epitomizes what a "mentor" should be as she has selflessly guided and encouraged me in all of my educational and professional decisions. She has brought me under her wing for all of the professional organizations in our field, and also on a broader level has brought women together in these organizations. Her pure joy of of fundamental fluid mechanics, and her bold and assertive approach to experimental fluid mechanics have galvanized my intellectual pursuits. And above all, she has shown me how caring for students and advocating for a diverse professional population does not have to be mutually exclusive from pursuing interesting and impactful research. I am so fortunate to have had Aline not only as a mentor and advisor, but as a friend throughout this whole experience. I will always cherish the time I have been able to spend with her, and I will continue to be inspired by her overall adventurousness, curiosity, and kindness. I cannot thank her enough.

Professor Branko Kerkez was integral as I designed and developed the real-time control component of my physical prototype. Additionally, he has helped me envision the future direction of my research. I am so grateful he came to Michigan during my tenure as a graduate student as his expertise–as well as his care as a mentor–has helped me push the future direction of my career and work. And I will continue to be motivated by his work-ethic, and his approach to building a research group and vision.

I am also fortunate that I was able to bring Professor Elizabeth Bruch into my research, as her expertise in social science added nuance to my interdisciplinary pursuits. Additionally, she gave me confidence and encouragement when I most needed it. As an engineer who is also deeply interested in social issues, I am so grateful to have her as a part of my team, and subsequently develop an added layer of expertise under her guidance and care.

Some of the most important people during one's education are the staff members who provide unseen but unwavering support through everything. Enough thanks and acknowledgements cannot be given to the lab technicians who have supported me on the physical model that was built for my experiments. In particular, Bob Spence built a exquisite wind tunnel from scratch, of which I am still in awe. I have been fortunate to work with him through the design and construction of this, and am extremely grateful for his time, expertise, and support. Additionally, Jan Pantolin and Rick Burch were dependable sources of support for all of my other laboratory needs. Finally, I will always be indebted to Nancy Osugi, Jessica Randolph, and Rebi Varghese for all of their assistance.

I was very lucky for having a supportive hydraulics core of faculty in my department, who not only inspired me intellectually, but who were such generous mentors through all of my education, including Professors Steve Wright, Avery Demond, and Valeriy Ivanov. Additionally, outside of my research group, I must acknowledge many more professors who gave their time and care to my education, including Professors Bill Schultz, Jonathan Bulkley, Adda Stefanopoulos-Zekkos, Rick Riolo, Anna Stefanopoulou, Shelie Miller, Luis Bernal, and James Vincent.

Beyond faculty member support, I am still in awe of all of the institutional support provided by various entities throughout University of Michigan committed to my success. The Women In Science and Engineering (WISE) Program was a refuge on campus for me, and their encouragement and consideration was always so heartening through everything. In particular, I will be forever grateful for Cinda-Sue Davis and Debbie Taylor for their generosity. The Rackham Graduate School has shown benevolent and unconditional support for my education and my outreach, and the knowledge I have gained because of it will forever influence and guide me. The Office of Graduate Education in the College of Engineering has also served as a caring entity through this experience. I am forever grateful for Shira Washington, Tiffany Porties, Andria Rose, and Kim Elliott for the opportunities they have given me, especially regarding my annual attendance at the Society of Women Engineers conference.

One of the most instrumental experiences I had in my graduate education was taking the *Ethnographic Writing* course with Ruth Behar of the Anthropology Department. Through this experience, I was able to grow in my writing and storytelling in a way I never thought was possible. She continues to be the inspiration behind my writing aspirations.

When debugging my various codes for my research, I often felt a sense of deep awe for the community of selfless “strangers” who formed the open-source software communities that were integral to the development of the various models in this research. In particular, the communities of Repast Simphony, OpenFOAM, Paraview, Python, and ZeroMQ have been especially helpful for my research.

Beyond my research, the other major component of my graduate education has been my experiences in Liberia, of which I am forever indebted to Herbert Winful. Looking back, it is almost impossible to believe that the opportunity he gave me as a first-year graduate student to teach undergraduate students in Liberia for a month, has now become a significant intellectual, and personal layer of my life. I am thankful for his support and advocacy through all of it.

Along with Professor Winful, I must also thank many others for their mentorship regarding Liberia, particularly from a research and intellectual inquiry perspective. This includes both Shanna Daly and Lisa Lattuca, who have provided me with the formative inspiration and knowledge on how I would like to “question engineering education” and how to carry out rigorous research in the field.

Of course, my gratitude is also for all of the Liberian students whom I have had the honor of teaching, and who inspire me daily with their curiosity and perseverance. I am so proud to have been a part of the Liberia Society of Women Engineers SUCCESS camp, and look forward to continued involvement and growth over the years. I am also grateful for the families and students who have helped me with my research endeavors while in Liberia, especially Edith Tarplah and and Veronica Badio who served as my Liberian home and family.

I am forever indebted to my undergraduate mentors, Sivaram Narayan (Central Michigan University), Hayley Shen (Clarkson University), and Hungtao Shen (Clarkson University) for pushing the idea of pursuing a PhD, for their mentorship as I applied, and for providing me with research experiences as an undergraduate that served as the basis of my success in graduate school. Without their encouragement, I would have never considered this professional path, nor believed I would be successful in it.

I must acknowledge the assistance and support my fellow peers gave me in my research, namely April Warnock and Boyun Wang. I only wish we would have been able to collaborate more. In addition, I am thankful for the other students in our ecohydrology/hydraulics group who enhanced this experience, including Lingli He, Pengchuan Wang, Jongho Kim, Chase Dwelle, Elizabeth Agee, Frank Sedlar, Chris Warren, and Arie Reath.

A significant portion of my graduate studies was spent with the Graduate Society of Women Engineers (GradSWE) who gave me what I deem "the friends I never knew I never had"; including Katherine Avery, Liz Getto, Mai Le, Ines Stuckert, Jenna Campbell, Jennifer Dolan, Alicia Salazaar, Sahithya Reddivari, Elizabeth Dreyer, and again Lingli He. Also, I am grateful for the support and friendship of Lauren Stadler throughout all of these years. I feel so proud to know such intelligent, hardworking, and resilient women.

I feel so fortunate to have been able to spend such formative years of my young adulthood in close geographic proximity with family, and I am grateful for the quality time I have been afforded to be with them, as well as for their love and support.

My grandma, Sarah Turbett, has served as one of my closest friends throughout this whole process, and has allowed me to use her home and love as a place of refuge. I continue to be inspired by her intelligence, resilience, and grace, and am proud to be a part of the large and loving family she has cultivated.

I am so grateful for the significant time I was able to spend with my grandpa, Marvin Rimer, my first few years of graduate school before he passed. His genuine curiosity for and pride of everyone and everything he encountered continues to inspire me daily; he became his best version of himself as he aged, and I am thankful to have witnessed that growth. Additionally, his humble working-class background has given me an added-layer of pride and perspective for this educational and professional accomplishment.

The numerous members of my extended family who were present at my oral defense are representative of the unconditional support that is demonstrated by the wonderful family I am so humble to have. My uncle and aunt, Dan and Helene Rimer, have served as additional guardians and cheerleaders throughout this experience, and their warmth, inquisitiveness, and vicarious delight throughout my doctoral education gave me the motivation I needed to persist. My brother and sister-in-law, Kyle and Megan Rimer, have been a wonderful support through my schooling. Additionally, my brother's deep empathy as a teacher for members of the rural, working-class community where we were raised inspire the type of empathy and care I hope to cultivate in my own classroom when I teach. And of course, I am grateful for all of the care and support from the rest of my aunts, uncles, and cousins I am so fortunate to have.

Most importantly, I must thank my parents, Janice and Kenneth Rimer, for their acceptance and grooming of my curiosity, assertiveness, fearlessness, exuberance, and sense of community; and for supporting and loving me unconditionally through everything. I am always in admiration of the love and care they have for each other and their family.

And finally, thank you to Priyank, for believing in me and caring for me – I pray this is only the beginning.

Funding for this research came from National Science Foundation Division of Civil, Mechanical and Manufacturing Innovation (NSF-CMMI) Grant 0856438, a Horace H. Rackham Predoctoral Fellowship, a Michigan Institute for Computational Discovery and Engineering Fellowship, a Rackham Graduate Student Research Award, and a Rackham Merit Fellowship.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF ABBREVIATIONS

CBR Chemical, Biological, and Radiological

CFD Computational Fluid Dynamics

MEMS Microelectromechanical Systems

ABM Agent-based Modeling

CBR Chemical, Biological, Radiological

MEMS Microelectromechanical Systems

FD Finite Difference

FV Finite Volume

FE Finite Element

DNS Direct Numerical Simulation

MPC Model Predictive Control

UD Upwind Differencing

CD Central Differencing

OOP Object Oriented Programming

OpenFOAM Open Source Field Operation and Manipulation

VTK Visualization Toolkit

BFGS Broyden-Fletcher-Goldfard-Shanno

DAKOTA Design and Analysis toolKit for Optimization and Terascale Applications

NI National Instruments

DAQ Data Acquisition

PISO Pressure Implicit with Splitting of Operator

SIMPLE Semi-Implicit Method for Pressure Linked Equations

CFD Computational Fluid Dynamics

LIST OF NOMENCLATURE

∂V The enclosed surface bounding the volume V of hypothetical element in computational mesh.

Δx Constant mesh size in x-direction.

Δy Constant mesh size in y-direction.

$\dot{q}$ Mass loading rate of contaminant release

ϵ Turbulent energy dissipation rate per unit mass for Reynolds-averaged Navier-Stokes model

Fr Froude number

Γ_e Effective diffusivity of the contaminant

$\overline{\mathbf{U'U'}}$ Reynolds stress tensor for Reynolds-averaged Navier-Stokes model

$\overline{\mathbf{U}}$ Mean velocity for Reynolds-averaged Navier-Stokes model

μ Dynamic viscosity of fluid

ν Kinematic viscosity of fluid

ν_t Turbulent viscosity for Reynolds-averaged Navier-Stokes model

ϕ Property of hypothetical element in computational mesh.

Re Reynolds number

Re_{cr} Critical Reynolds number

ρ Density of fluid

$\mathbf{A}_f$ Normal vector on face f of element in computational mesh pointing toward point N.

$\mathbf{d}$ Vector connecting points P and N for hypothetical element in computational mesh.

$\mathbf{f}$ Vector of unknown port velocities for model predictive control.

$\mathbf{g}$ Gravitational constant

$\mathbf{H}(\mathbf{U})$	Correction operator for PISO algorithm.
$\mathbf{U}$	Velocity vector $[U_x, U_y, U_z]$
$\mathbf{U}'$	Fluctuating velocity for Reynolds-averaged Navier-Stokes model
$\mathbf{U}_{ref}$	A reference velocity for Reynolds-averaged Navier-Stokes model
$\mathbf{x_s}$	Coordinates of the contaminant release
$\mathbf{x}_p$	Vector of coordinates corresponding to the location of the p_{th} protected point for model predictive control.
a_N	Matrix coefficient corresponding to neighboring cell N for PISO algorithm.
a_P	Matrix of coefficients to cell P for PISO algorithm.
C	Transported scalar
$d\mathbf{A}$	Infinitesimal surface element with outward pointing normal on the surface ∂V for hypothetical element in computational mesh.
F	Flux through the cell face for PISO algorithm.
f	Source of the contaminant
i	Number of ports for model predictive control.
k	Turbulent kinetic energy for Reynolds-averaged Navier-Stokes model
l	Characteristic length related to geometry underlying flow field
p	Pressure of fluid
q	Mass load of contaminant release
t_s	Time of contaminant release
U_0	Latest available velocity field for PISO algorithm.
U_i	Velocity for port i for model predictive control.
U_{max}	Maximum velocity of a port for model predictive control.
V_P	Control volume for hypothetical element in computational mesh.

ABSTRACT

Controlling Hazardous Releases while Protecting Passengers in Civil Infrastructure Systems

by

Sara P. Rimer

Chair: Professor Nikolaos Katopodes

The threat of accidental or deliberate toxic chemicals released into public spaces is a significant concern to public safety, and the real-time detection and mitigation of such hazardous contaminants has the potential to minimize harm and save lives. Furthermore, safe evacuation of occupants during such a catastrophe is of utmost importance. An illustrative example of such a scenario is the 1995 sarin-gas terrorism attacks in the Tokyo subway system, where sarin-gas permeated multiple subway lines after being released by the perpetrators, causing twelve deaths and hundreds of injuries. Not only were emergency response teams ill-prepared to deal with the catastrophe, but the *civil infrastructure system itself* lacked ability to detect and mitigate this chemical attack, or to survey and disseminate evacuee behavior to accelerate evacuation. The specific objective of this research is to develop intelligent public infrastructure systems capable of automatically responding to and minimizing hazardous contaminant disasters through (i) **the real-time sensing and control of contaminants** and, (ii) **the modeling of and communication to occupants as they evacuate**.

This dissertation first works to address such public safety scenarios through the development of a flow control computer model that combines computational fluid dynamics and model predictive control optimization techniques. Public spaces defined by a long conduit (e.g. airport terminal) allow us to assume unidirectional, ambient flow. We set up our long conduit domain with a series of sensor arrays and actuators along the wall boundaries, which are used to detect and mitigate the contaminant. The limitations of the computational flow control model is tested when used in real-time scenarios by building a physical model with its own programmable sensor-actuator control system.

Additionally, this dissertation addresses the evacuation of occupants inside of public spaces when faced with a threatening, dynamic environment by developing an evacuation agent-based model in which the agents (i.e. the evacuees) make egress decisions within a simplified public space that has a spreading contaminant. This agent-based evacuation model is coupled with the computational flow control model, which subsequently provides the agents with a realistic contaminant they must interact with as they evacuate, and in which the contaminant is controlled based on agent location and potential contaminant exposure.

This research is novel in its ability to bridge a social science computational model with a physical systems computational model allowing both systems to interact with each other. Additionally, this research is able to demonstrate real-time aptitude of cyber-physical fluid flow control through the construction and deployment of a physical prototype able to detect and mitigate the contaminant through a sensor-actuator-controller system. This research will be used by civil infrastructure systems desiring to improve their resilience and response to such hazardous contaminant threats.

CHAPTER I

Introduction

1.1 *"Breathing in a vacuum"*

It was just after 8:00 AM and Kiyoka Izumi was on her way to work. This day began the same as any others. Her daily commute took just a little over 30 minutes consisting of three subway lines, thus she needed to change trains twice. As most standard jobs in Japan begin at 8:30 AM, the morning commute can become quite overwhelming with trains packed to the brim with passengers rushing to be on time. Kiyoka always left early enough such that it would allow her to just miss the peak of morning commute passengers, being one of the few who would arrive early to work. She was on the second leg of her trip in the first car of the Chiyoda line that was just beginning to fill up, when she took a breath and all of a sudden felt an intense, sudden pain, her chest seeming to freeze as though she was trying to breathe in a vacuum. This void of breath soon turned to acute coughing followed by terrible nausea. She realized she was not the only one – everyone else in her car was also choking and coughing. Without much thought, she left her car at the Kasumigaseki stop to catch the Hibiya line for one final stop before reaching her office. As she left the car, a station attendant was waved in by other passengers to remove a strange plastic bag, that attendant dying soon after. As she reached the Hibiya stop, the most bizarre sight emerged of a completely abandoned Hibiya train, which immediately coincided with with an emergency announcement for everyone to evacuate the station. It was then that Kiyoka *began to realize how sick she felt, and as she evacuated, she was overwhelmed and horrified by the sight of station attendants and civilians randomly lying on the ground, unconscious, many with spoons shoved in their throats to keep them from choking on their tongues. As she says of the scene she observed: "'hell' describes it perfectly."*

[story adapted from *Murakami* (2010)]

On March 20, 1995, the most serious attack on Japanese soil since World War II occurred at the hands of the *Aum Shirinkyo* doomsday cult and terrorist group [*Juergensmeyer* (2005)]. During the peak of morning rush hour, five members of the cult walked into five different trains of three different subway lines on the Tokyo subway system carrying approximately 1 litre each of liquid sarin in two plastic bags covered by newspaper. Each of them dropped their respective bags on the ground and punctured them with sharpened umbrellas. As sarin is one of the most volatile of all nerve agents, the liquid soon began to vaporize and spread through the trains, eventually killing 12 civilians, seriously injuring over 50, and affecting over 5000, including Kiyoka. Due to the movement of passengers and trains, the strategic release of the sarin gas in such a location as a subway system is ripe for causing maximum spread and corresponding exposure for occupants in the system.

Sarin was developed by the German Nazis during World War II to be used as a chemical agent in warfare. It was actually discovered somewhat by accident as German researchers in the 1930s were in the process of developing improved insecticides, and instead invented the first class of nerve agents [*Evison et al.* (2002)]. Sarin was a later iteration of and ten times more potent than the nerve agent first discovered. In fact, the tiniest of pin drops of the liquid form of sarin is enough to kill a human being. While the Germans never actually used sarin against the Allies during World War II, by some estimates, they did have a stockpile of up to 10 tons of it. Since then, the chemical has been used multiple times in conflict to a varying degree, with production in the U.S. officially ending in 1957, and the United Nations Chemical Weapons Convention banning the chemical agent in 1993 [*Henderson* (1999)]. However, even with widespread ban of chemical agents such as sarin, their allurement still attracts both governments involved in conflict and terrorist organizations with a desire to cause widespread terror and destruction on a civilian population. Actually, the 1995 attack was not the first time that the Aum Shirinkyo cult first used the chemical: a similar attack was carried out in 1994 when sarin was released creating a cloud of gas that permeated through various targeted neighborhoods of Matsumoto, Japan killing eight and harming over 200 [*Okumura et al.* (1998)]. Most recently, sarin gas was used by the government of Syria during its Civil War [*Dolgin* (2013)].

Obviously, the development and use of sarin in warfare and terrorism is not the first time what is known as Chemical, Biological, Radiological (CBR) weaponry has been used. In fact, stories of CBR weaponry use date back to the Ancient Greek times [*Szinicz* (2005)], with their most prolific use in modern times occurring during World War I. Nevertheless, sarin represents a clear example of the type of threat CBR weapons pose to civilian populations. Additionally, while present-day usage of CBR weapons is rare, intolerable, and formally banned by the majority of the world, its use is still present, and is often on the imaginations of the most creative and deviant of groups wanting to spawn harm.

What is most alarming is that though many of the most technologically advanced countries have been at the forefront of developing modern CBR weapons and technologies, the civil defense of these countries against such threats is laughably underdeveloped, with most focusing on personal protective equipment rather than more extensive and intelligent civil infrastructure protection [*Henderson* (1999); *Falkenrath et al.* (1998)]. Fortunately, since the attacks of September 11, 2001, and the more recent localized terrorist threats, the focus on civil defense, particularly regarding potential CBR attacks has been on the upswing.

In the U.S., there have been a few studies on the potential widespread threat of CBR attacks on the civilian population in highly populated environments. For example, in Manhattan there have been a few attempts to understand the basic urban consequences if a CBR agent were released into the city's subway system. In 1966, U.S. Army researchers carried out a field test in which *bacillus subtilis* was released into the 23rd St subway station via smashed bulbs, and its concentration was measured around the city after different locations to learn about its dispersion. Within only five minutes after its release, the bacteria was detected at every subway station between 14th and 59th street; and within only four days, it was estimated that over one million people were exposed to it, demonstrating its reach. Furthermore, even though the tracer agent released in this study was deemed safe enough to ingest by subway passengers for the low concentration used, it was still quite a controversial test as *bacillus subtilis* is actually known to cause food poisoning [*Carlton* (2001)].

This sort of field test was carried out once again in 2013, and yet again in 2016, both funded by the Department of Homeland Security with researchers from the Brookhaven Laboratory. The most recent study used aerosol-dispensed particles as what would be more characteristic of biological agents such as anthrax, and was once again released in the subway system to measure its dispersion.

Yet, what limits the success of these field tests is that the scope of what they are able to portray is often futile to the largeness of the problem at hand. Thus, a question exists as to how to better understand the large-scale dispersion of CBR agents in highly populated environments with minimal invasiveness. Computational methods modeling these scenarios serve as an attractive alternative to disruptive, intensive, and often controversial field tests.

The physical spread of a CBR agent can be viewed as a rich fluid dynamics problem. Many recent Computational Fluid Dynamics (CFD) studies have been carried out to model urban environments that have been exposed to a CBR agent [*Settles*

(2006), *Branscomb et al.* (2002)]. However, these CFD studies often only investigate the outdoor urban environment, with *only a few modeling the indoor environment* [*Camelli and Löhner* (2004)]. Moreover, all of these investigations are concerned with understanding the spread of the contaminant, *without consideration as to how to control it.* Finally, the concern of civilians in such scenario, particularly as a means to reduce exposure to such contaminants, can also be a means of enriching and accompanying this overall civil engineering scenario.

The ability to detect and mitigate a hazardous contaminant in a public space in real time, while also considering the safe and efficient evacuation of civilian occupants is the motivation behind this dissertation and the attempts to solve such a problem are explored throughout this thesis.

1.2 The Control of Fluid Flow

Flow control is the manipulation of the flow field of a fluid (e.g. water, air) to meet some specific desire. It is considered a subfield of the branch of science known as fluid mechanics, the physics of fluids. The concept of flow control can be demonstrated quite well by looking at evolution in the natural world of species that have exploited a fluid in their environment for their own benefit. For example, consider a tall, cylindrical cactus with complex surface geometry. Because cacti face high windspeed, it has been hypothesized they have evolved via natural selection to have longitudinal cavities and corresponding spines so that wind forces, such as drag and vortex shedding, are minimized [*Bearman and Harvey* (1993)].

The finesse and efficiency with which these aspects of nature are able to control their surrounding fluid is a nontrivial problem that researchers are only at a fragment of being able to understand and recreate. However, the examples demonstrate the beauty, complexity, and prospects of the field of flow control.

Much like the majority of the engineering sciences, the pragmatic utility of flow control proceeded the scientific development and understanding of the field. The existence of humans designing tools that are able to harness the flow field of a fluid for improved performance can be seen with the development of weapons in ancient civilizations, such as boomerangs [*Gad-el Hak* (1989)]. Additionally, the emergence of agricultural societies over 8,000 years ago is what brought the complicated water distribution systems of canals and aqueducts of different Ancient empires [*Mays et al.* (2007)].

While flow control has always been inherent in civil engineering hydrodynamics (i.e. liquid flow), the field of flow control as it can be seen today was majorly shaped by the modern development of aerodynamics (i.e. gas flow), which came about with the advent of aircraft in the late 19^{th} and early 20^{th} centuries. In particular, the development of "boundary layer theory" by Ludwig Prandtl in 1904 [*Prandtl* (1904)] was integral to describing the phenomena of separation of fluid around a stationary or moving object in a flow field; subsequently, boundary layer theory serves as the basis of modern aircraft design. Prandtl first introduced experiments that were able to manipulate the development of the boundary layer around an object, which can for example reduce pressure drag and increase lift, both improving the speed and efficiency of aircraft performance. Prandtl's initial work on boundary layer theory continues to be at the center of flow control applications[1].

Since the birth of modern fluid flow control by Prandtl, the continued development of flow control techniques and their applications have mirrored that of the political landscape of the industrialized world, and the corresponding major scientific breakthroughs. World War II and the Cold War propelled much of the scientific development in the middle of the 20th century. Therefore, much of fluid flow control during that time revolved around improved aircraft/watercraft and missile/torpedo

[1] For more elucidation on the fluid mechanics of these phenomena, the reader is directed to the excellent film series by the National Committee for Fluid Mechanics Films which can be viewed at the following link: `https://www.youtube.com/playlist?list=PLfF--3o8i4r82vJ0kjCVYgqKgyVM5QwN0`.

performance. Additionally, the corresponding development of computational methods during this same time period revolutionized the field of fluid dynamics by driving the development of CFD [*Gad-el Hak et al.* (2003)], subsequently contributing to breakthroughs in the design and refinement of watercraft and aircraft. During the 1970s and 1980s, the energy crisis and corresponding environmentalism focused flow control efforts on improving the fuel efficiency of civilian vehicles (e.g. reducing drag) on land, sea, and air. Additionally, the formation of wave theory in the 1950s instigated the development of wave and free-surface control in hydraulic applications. Since, flow control continues to develop in manners that reflect the expansion of computational capacity and performance, plus improved knowledge and understanding of turbulence. Furthermore, the scope of flow control applications has expanded beyond aerodynamics and hydraulics, influencing such fields as biofluids [*Quarteroni and Rozza* (2003)].

In the most recent decades, the development of Microelectromechanical Systems (MEMS) has completely transformed the underlying objectives of flow control. No longer is flow control primarily passive; instead, the ability to use MEMS devices by inputting energy into a flow field in order to exploit it has expanded the reach of the field. Therefore, the methods of flow control have progressed beyond a priori design optimization, to the ability to dynamically control a flow field utilizing MEMS. For example, consider the airfoil of an airplane: mini-jets have now been placed along said airfoil to pulsate air in strategic locations and times such that the boundary layer is changed for the dynamic needs of airplane flight [*Gad-el Hak et al.* (2003)].

What is most exciting about the development of MEMS and its application to flow control, is the corresponding recent developments of data and information science. With the ability to use sensors to read information about the flow field, and use MEMS to react and control the flow field based on the information read, the ability to carry out flow control in real-time is the newest endeavor to this frontier.

1.3 No Need to Panic

On the 15 of April in 1989, almost 100 people were killed, and over 700 were injured, at the Hillsborough football stadium in Sheffield during a professional soccer (football) game [*Richardson* (1993)]. These deaths and injuries occurred prior to the start of the match as fans were entering the stadium, moving through "turnstiles" to enter their assigned "pens." As fans tried to reach their respective pens before the match was to begin, they began pressing against crush barriers and fences due to overcrowding. To ease overcrowding at the entrance to the stadium, officials decided to open an exit. However, rather than reduce the load of the crowd, the opening of this exit instead lead to an influx of even more people. Eventually, one of the barriers gave way, and fans began trampling over one another in an effort to escape, subsequently leading to the large number of deaths and injuries, and what is considered the worst disaster in British sporting history (as well as one of the worst worldwide).

These types of human chaos in public spaces – sometimes known as "crushing" or "stampedes" – are nothing new, particular for sporting events and large festivals. In fact, as recently as 2015, over 2,000 people were killed in Mecca, Saudi Arabia in a human stampede during the Hajj Pilgrimage[2].

However, what is notable about the Hillsborough football stadium crushing event is the recent decision by a British jury that the victims were actually "unlawfully killed" [*Mansfield* (2016)]. For the past 27 years, police and safety officials pushed a narrative that those caught up in the incident were "drunk" and "rowdy," with their "reckless" behavior contributing to the disaster, and thus the death and injuries were their personal fault. Yet, after evidence was provided to the contrary – that actually those caught in the stampede were often behaving rationally, sometimes working to save one another – the jury instead ruled that the fault was at the hands of the police

[2] An interesting interactive article on the Hajj crushing event can be found here: `http://nyti.ms/2cjo51y`.

and safety officers due to the decisions they had made.

The decision by the jury falls in line with the individual and social psychological research that has occurred over the past few decades. Researchers have found that contrary to the narrative often pushed by everyday discourse – that under situations of distress, humans are likely to act with "hysterics" and "panic" and without regards to those around – that actually, individuals behave quite rationally, and make decisions to protect themselves and others, while also trying to mitigate or minimize the disaster at hand [*Quarantelli* (2001)].

This case study of the Hillsborough football stadium crushing demonstrates two points:

1. First, that in chaotic situations such as human stampedes, individuals actually behave and make decisions *rationally*; a heightened sense of peril or fear does not equate to a panic behavior.

2. Second, public officials and officers have a duty to the public to make decisions, design these spaces such that large-scale evacuation disasters such as those described are minimized.

Therefore, engineers have a duty to take into consideration the potential for disasters and the subsequent safety of human occupants when designing public spaces, structures, and buildings[3].

Now obviously, the evacuation out of large-scale structures such as stadiums is nothing new. Additionally, engineers have responded to such catastrophes accordingly. For example, many stadiums no longer include standing-only sections in their designs [*Bale* (2000)]. However, there still exists a need to continually improve upon

[3] Actually, the structural engineering firm that had worked on the area of the Hillsborough Stadium where the disaster occurred was found partly culpable in the recent jury's decision. It was found that the firm had performed some miscalculations regarding the safety aspects of their work. After the jury's decision, the firm even issued an apology. `http://www.liverpoolecho.co.uk/news/liverpool-news/hillsborough-structural-engineers-say-sorry-11243837`

these public spaces, their design, and the response to such disastrous events. Furthermore, this need stretches further than just focusing on stadiums or large-scale public activities: a variety of spaces exist (e.g. airports, subway tunnels) that are vulnerable to catastrophic events that could cause unnecessary human death and injury if design of these public spaces or immediate response to assist with evacuation is poor[4]. Subsequently, the need to better understand evacuation dynamics in order to improve the design of spaces, and response to catastrophic events is a rich research area intersecting both the fields of individual and social psychology, and civil engineering.

In the past few decades, many efforts have been made to understand evacuation dynamics from a building. These efforts are usually limited to studying human behavior in past real-world evacuations where comprehensive data is often lacking [*Aguirre et al.* (2011)]. Furthermore, it is usually impractical to recreate emergency situations in order to study evacuation dynamics due to its difficulty, expense, and potential danger. Thus, a research need exists to understand evacuation dynamics in complicated situations where there is limited data and an inability to recreate real-life evacuation scenarios. The development of computational models to simulate evacuation situations is therefore appealing. Most promising is the use of bottom-up computational methods, such as agent-based modeling, that allow for heterogenous human behavior. Much like flow control, an exciting yet unexplored opportunity for these sorts of models is (i) their coupling with physical, dynamic environments, and (ii) their potential to be deployed in real-time. The former is explored throughout this dissertation, while the latter is pondered upon in its conclusion.

[4] Ironically, false alarms and subsequent evacuations out of airports seemed to be occurring almost weekly during August 2016. `http://nyti.ms/2aMS57G`, `http://nyti.ms/2bvTYFU`

1.4 Overview of Intellectual Contribution

This dissertation seeks to develop a comprehensive mean of approaching the civil engineering scenario described at the beginning of this introduction from two fronts: (i) the detection and mitigation of a hazardous contaminant released into a public space and (ii) the safe and efficient evacuation of occupants inside such a space during this scenario. There are three components to this research:

1. This research first addresses such public safety scenarios through the development of a flow control computer model that combines computational fluid dynamics and model predictive control optimization techniques, which we will call the *CFD flow-control model*. The CFD flow-control model is for wall-bounded public spaces defined by a long conduit (e.g. airport terminal), which allows us to assume unidirectional, ambient flow. The CFD flow-control model uses the underlying Navier-Stokes equations with a scalar transport equation for the contaminant injected into the flow field. These equations are programmed using the open-source CFD software suite, OpenFOAM. We set up our long conduit domain with a series of sensor arrays and actuators along the wall boundaries, which are used to detect and mitigate the contaminant (see Figure 1.1).

   This CFD flow-control model is based on a model developed previously by a student in this research group who was able to demonstrate boundary control of a contaminant in a similar elongated conduit *Warnock* (2013b). This student also investigated contaminant control in free-surface flows. This research builds upon this model by exploring the influence of changes in "protected areas" in the domain. Additionally, this research applies the CFD flow-control model to a physical prototype that has been constructed, and an evacuation behavioral computational model that has been developed.

2. This research tests the limitations of the CFD flow-control when used in real-

time scenarios by building a physical prototype with its own programmable sensor-actuator control system. This physical prototype is able to ‘detect’ and ‘mitigate’ in real-time a ‘contaminant’ released into the ambient flow in a test section. The basis of the prototype design is a blower wind tunnel, where an upstream fan pushes air through a series of screens, honeycomb, and contraction to provide uniform flow at the beginning of a test section. The test section reaches over 6 m long (with a cross-sectional area of 0.6×0.6 m^2) and has a programmable sensor-actuator control system installed.

3. Additionally, this research investigates the evacuation of occupants inside of public spaces when faced with a threatening, dynamic environment by developing an Agent-based Modeling (ABM) in which the agents (i.e. the evacuees) make egress decisions within a simplified public space that has a spreading contaminant. A basic evacuation ABM has been developed to test the emergent behavior of agents individually assigned a specific trait related to their evacuation decision-making. For this hypothetical scenario, agents must evacuate and also try to minimize their exposure to the ‘contaminant.’ Each agent is assigned a hypothetical ‘risk tolerance’ trait, which is their likelihood to expose themselves to the contaminant if it allows them to move closer to the exit. The evacuation ABM was programmed using Repast agent modeling toolkit. The evacuation model is coupled with the CFD flow-control model, which subsequently provides the agents with a realistic contaminant they must interact with as they evacuate.

This research distinguishes itself in three particular ways. **First**, while active feedback flow control is already a research area in many different engineering disciplines [*Gad-el Hak* (1989), *Bewley et al.* (2001), *Jameson et al.* (1998)], it rarely involves the removal of a mass from an ambient fluid flow–an application specific to civil engineering context. **Second**, most active feedback flow control work is limited in its capability to be implemented in real-time [*Gad-el Hak* (1989), *Bewley et al.* (2001), *Duriez et al.* (2014)]. By applying the computational model to a customized physical prototype, this research has the the potential to develop significant techniques to progress the real-time capability of active feedback flow control. **Third**, while many different evacuation ABMs have been developed for different emergency scenarios [*Pan et al.* (2007), *Pelechano et al.* (2005)], most do not incorporate dynamic environments, such as a contaminant penetrating a public space.

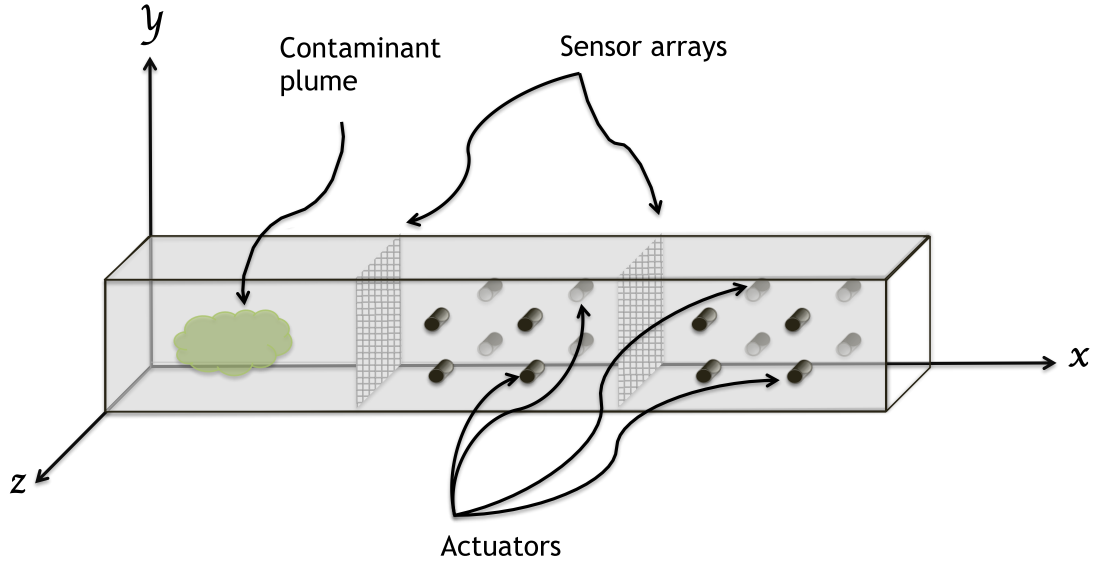


Figure 1.1: Sensor-array schematic for underlying flow control problem.

1.5 Structure of Thesis

This dissertation organizes itself around the three areas discussed in Section 1.4:

- The CFD flow control model is developed in Chapter II, including the underlying equations used, and how they are computationally estimated. This chapter presents basic simulations demonstrating the CFD flow-control model, particularly focusing on the influence of changing the protected area of the model's domain.

- A physical model designed and built to test the real-time feasibility of the CFD flow control model is presented in Chapter III. The physical model is able to detect and remove a contaminant in real-time, however, potentially undesirable mixing and fluid rotation also occur as a result of the control. The fluid behavior in the physical is compared to the CFD flow control model.
- Third, the evacuation ABM is developed, presented, and coupled with the CFD flow control model to simulate the control of a contaminant in consideration of civilians present in Chapter IV.

The dissertation concludes with reflection on this research, discussing its accomplishments, and also limitations in Chapter V. The possibility for future research is also discussed, particularly envisioning the long-term vision to move this research forward.

CHAPTER II

Flow Control

The flow control strategy used is essentially the combination of two problems: fluid dynamics and optimization. Actually, it is the area of "optimal control" that is what describes this bridging. While optimal control is just one area within control engineering, it is the method used most often for physical systems that wish to stay true to their corresponding set of governing equations.

This chapter builds the CFD flow-control model by (i) first, describing one classification of flow control methods that is widely used by the flow control community in Section 2.1; (ii) then developing the CFD model for the underlying fluid flow in Section 2.2; (iii) followed by bringing in optimization to control the fluid flow 2.3. Case runs are carried out and described in Section 2.4 for results based on where the contaminant should be minimized, and how the capacity of the boundary actuators influences the optimizing behavior. Finally, discussion on these results, and further implications for the CFD flow-control model takes place in Section 2.5.

2.1 Classification of Flow Control Methods

As discussed earlier, the main contributions to the field of flow control lie in the applications of aerodynamics, primarily in the applications to vehicle performance. Thus, naturally researchers at the frontier of modern flow control have classified the different methods behind flow control, resulting in a breakdown that can be seen in Figure 2.1. Broadly speaking flow control strategies can be separated into two main categories: passive and active. An active flow control system can be simplified to being a system with a set of sensors and actuators: sensors used to measure the state of the flow field, and actuators used to input energy into the system to change the flow field. Whereas passive flow control is majorly a priori shape optimization (e.g. airfoil shape design).

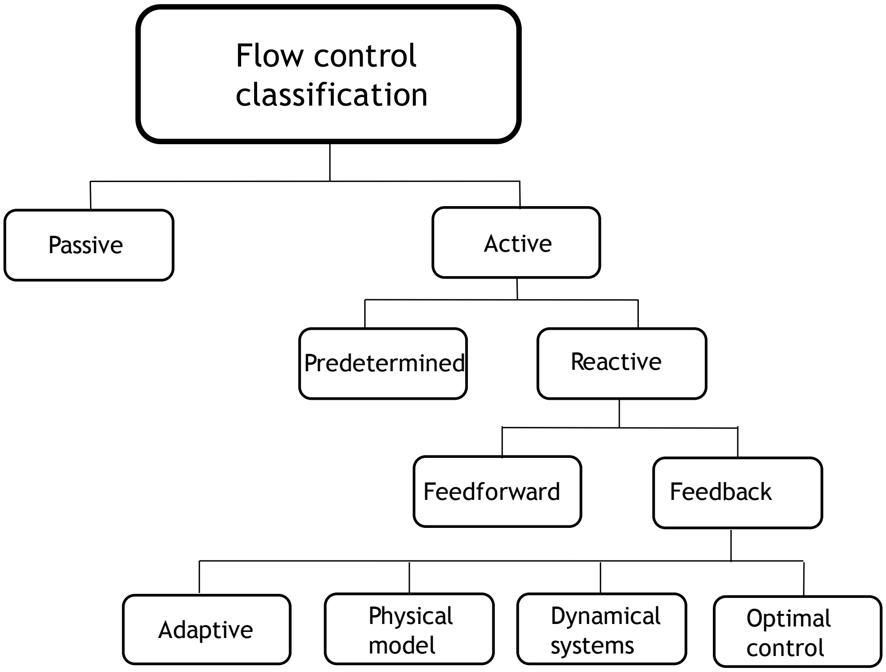


Figure 2.1: Classification of control strategies.

As can be seen in Figure 2.1, active flow control can be predetermined such that energy is input into the flow field without regards to its state. Simply speaking, predetermined active flow control is an active flow control system absent of sensors. Thus, nothing in the flow field is measured, but a predetermined action is still taken by the actuators (see Figure 2.2).

Reactive active flow control is categorized by two types: feedback and feedforward. In both of these types of flow control, sensors are used to read the state of the flow system, and a control action is carried out based on what is read. However, the major difference between feedback and feedforward is that active feedback flow control systems continue to read the flow field and adjust the control action accordingly until the desired state of the controlled variable is met. Figure 2.2 shows the different control loops for predetermined, feedforward, and active feedback flow control.

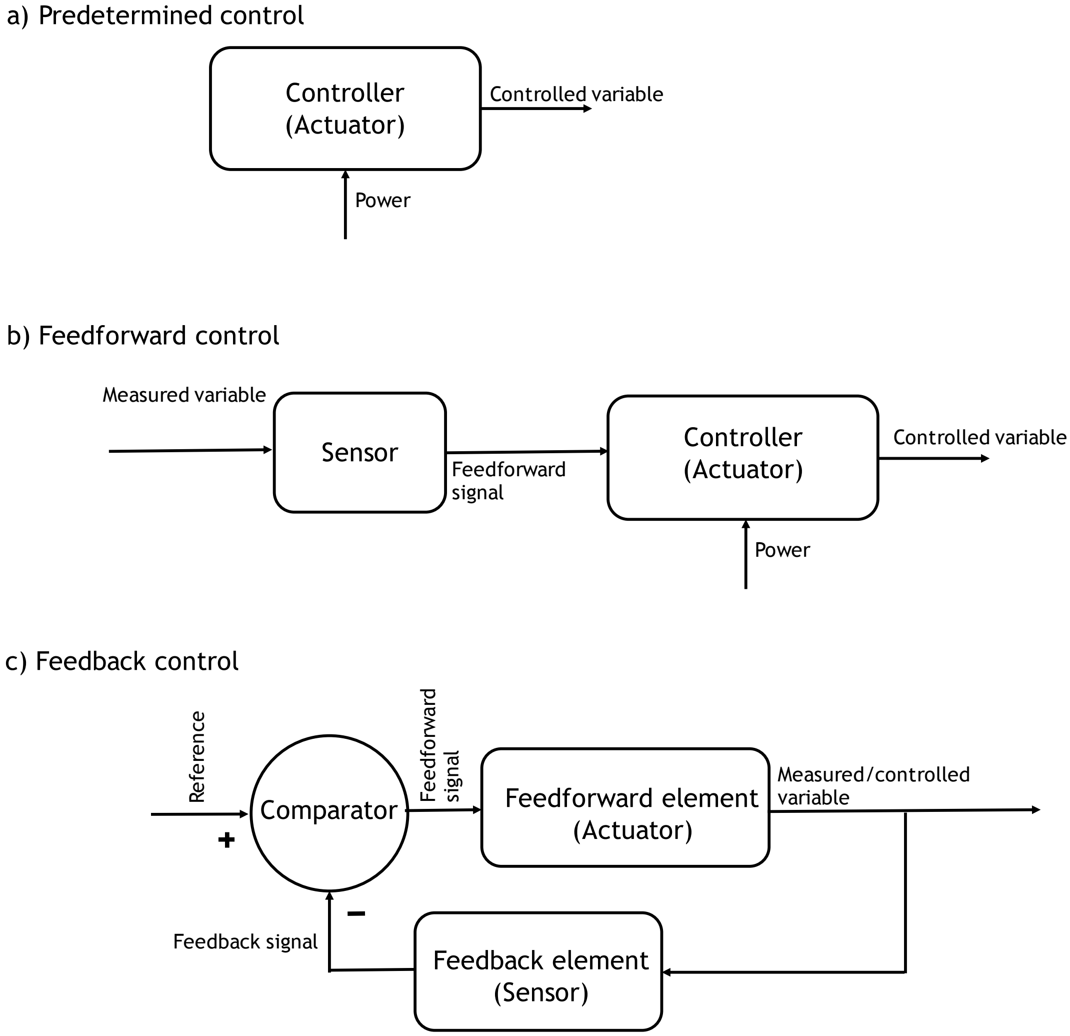


Figure 2.2: Different types of control loops for active flow control.

The type of control that has the most potential is active feedback flow control, in which sensors are used to read a flow field, and the system then has the potential to change its control actions via actuators based on what is read. These systems in particular have the potential to carry out control in real-time, thus control actions can react to real changes in the environment.

Bewley and Moin (1994) have specified four sub-categories of active feedback flow control. It must continue to be noted that these four subcategories are most likely somewhat limiting due to their specificity (i.e. aerodynamics of vehicle performance) of application, and the broader field of engineering control has a multitude of active feedback control strategies that might also prove fruitful to flow control. However, for this underlying research, the four subcategories still provide a basis for how to look at the flow control problem at hand. Additionally, in Chapter III, a distinction will be made between prior work done in this research group using one type of flow control technique (i.e. dynamic systems) versus the one used throughout this dissertation (i.e. optimal control). These four subcategories for active feedback flow control are discussed further.

1. *Adaptive schemes.* Adaptive schemes use an empirical approach to flow control. A flow control algorithm is developed by iteratively training itself to produce the desired output based on given measurements. System identification and controller determination are carried out without regard for the physical phenomena taking place. Instead, parameters are adjusted by the user and/or the optimization scheme itself to tune the feedback control law. Usually, adaptive systems are based on some sort of control law for linear dynamic systems and are best used for simple systems. As will be discussed in Chapter V, the application of adaptive schemes to well-defined physical systems has a lot of future possibilities, particularly as using adaptive methods with high fidelity to nonlinear partial differential equations is the focus of more recent developments [*Washabaugh et al.* (2016); *Duriez et al.* (2014)].

2. *Intuitive-based schemes.* Intuitive-based flow control methods are usually used when the physical phenomena taking place is well understood a priori. Intuitive-based control methods usually develop first from physical experiments. Many examples of intuitive-based methods can be seen in applications which involved the active cancellation of turbulence, or delaying the onset of turbulent transition in boundary layers. However, similar to adaptive schemes, intuitive-based methods are ill-suited for problems in which the physical processes are not completely understood, such as for turbulent flows.

3. *Dynamical-systems schemes.* Dynamical-systems schemes, often referred to as reduced-order models, extrapolate from linear systems theory and aim to decompose a complex physical phenomena, such as turbulence, into a finite number of representative modes. For example, the complexity of turbulent flow lends itself well to a dynamical systems approach as by having large spatial and temporal scales, it can be reduced down to systems of a much lower dimension.

4. *Optimal control.* The final approach, optimal control, provides a rigorous and systematic method to derive feedback control laws. Unlike the other three approaches, optimal control theory can be applied directly to the equations of motion that govern the flow, such as the Navier-Stokes equations. The goal of optimal control is to minimize a cost functional that is applied to the governing flow equations and specifically written for the physical problem at hand. The cost functional can be developed to represent a wide array of flow properties, such as turbulent kinetic energy, drag, etc. While on the first look, this type of control seems the most desirable as it is the truest to the governing equations at hand. However, the cost (e.g. computational expense) of carrying out optimal control makes it less desirable for certain applications.

Because it is the strategy that is most able to be true to the underlying physics of fluid dynamics, the control strategy used throughout this research is optimal control. One simple way of looking at optimal flow control is such that it can be reduced down to essentially the bridging of two different problems: fluid dynamics and optimization.

Exploiting a fluid field can be looked at as a means of searching for the best, or rather "good enough" solution to manipulating the flow field for some particular need. Relating back to the earlier discussion of flow control, the problems of fluid flow around an airfoil can be looked at as an optimization problem in which the objective is to design the airfoil such that drag is "minimized."

In particular, the optimal control method is used in conjunction with what is known as Model Predictive Control (MPC) and is presented in further detail in *Grune and Pannek* (2011), in which the "optimal flow control" strategy is implemented for only a short time period before reading in new information and adjusting this optimal control strategy. The details on the implementation of MPC are presented later in this chapter in Section 2.3.2. Before we are able to present on the optimization, we must first develop a model for what exactly we are wanting to optimize, which is discussed throughout this next section.

2.2 Computational Fluid Dynamics Model

It is naive to believe that any more than a brief overview on the fields of fluid mechanics can be presented throughout this thesis. Instead, the reader is directed to far more compelling and comprehensive texts on the subject such as *Anderson and Wendt* (1995); *Moukalled et al.* (2015); *Ferziger and Peric* (2012); *Wendt* (2008). However, for the sake of being loyal to the complex physics underlying the mechanics of fluids, a brief overview is provided here, particularly with regards to the development of computational methods for fluid dynamics.

Computational science has afforded scientists and engineers the ability to study and model physical systems to which a closed form solution is not possible or whose existence is unknown. It is interesting that the development of computation in the sciences and engineering began actually with brute force human computing[1]. Because many different computational applications involve thousands, and possibly millions, of basic individual computations, the use of humans to carry out these computations obviously has its limitations. Thus, the development of high-speed digital computational machines is what has truly led to the modern field of computational science as it is seen today. The advancement of computational capacity and efficiency over the past half decade, and the subsequent reduction in computational costs, has lead to the increased ability to model and simulate more diverse and complicated systems. Originally, computational science was seen solely as an aid to the theoretical and experimental arms of science. However, the ubiquity of computational science and its applications is now itself considered its own paradigm with unique theory and practice norms.

To understand the dramatic influence the field of computational science has had on the field of physics, one does not need to look any further than CFD. A complete solution to the equations used to describe the motion of fluid flow, known as the Navier-Stokes equations (described later in this chapter), are considered one of the greatest unsolved problems in physics of the century[2]. Thus, the ability to computationally estimate the solutions to these equations makes fluid dynamics a particularly relevant field for the application of computational science. In fact, CFD has completely revolutionized the inherent potential to the field of computational science, as it is often one of the driving forces behind faster, bigger, and more efficient computational machines and methods [*Graves Jr* (1982); *Wendt* (2008)].

The means to computationally estimate the solution to a set of partial differential equations involves reducing the equations to a set of solvable algebraic equations,

[1] The term "computer" actually referred to a person who performs calculations. Notably, many significant contributions in the sciences and engineering were carried out by labs of women "computers" *Des Jardins* (2010).

[2] A solution to the Navier-Stokes equation is incidentally one of the Clay Mathematics Institute "Millennium Prize Problems."

and iteratively calculate solutions throughout a domain while moving through space and time. Obviously, the ability to do this is nontrivial, and is itself the motivation behind the field of CFD.

For the most part, we can develop our CFD model using the following steps:

(i) Develop model of the physical system to be studied (i.e. geometric domain and governing equations),

(ii) Discretize the geometry of the physical domain to create computational domain,

(iii) Discretize governing equations into a set of algebraic equations,

(iv) Determine the initial and boundary conditions.

(v) Implement an iterative solver over space and time to find a solution.

Beyond these five steps, another decision must be made concerning the method of programming and subsequent deployment of software to assist in modeling our system. While in the past, this step was usually a commonplace decision that centered on choosing a procedural programming language to implement the above steps, the recent use of object-oriented programming instead of procedural programming for the ease of model development has favorably complicated this decision. Thus, as a final step for developing our CFD model, we have:

(vi) Employ CFD software to carry out the above steps.

The details of these steps for the CFD portion of the flow control model are presented throughout the following sections.

2.2.1 Geometric Domain

We choose our domain to represent a long conduit that is comparable to those in the public spaces being studied (e.g. airport terminal) where the length is significantly greater than the width. For computational simplification, we have the model be 2 −dimensional. Furthermore, we choose our geometric domain to be described using the *Cartesian coordinate system* (otherwise known as rectangular coordinates).

For this space, we assume that the bulk fluid enters at the upstream end of the space, and exits at the downstream end, thus signifying unidirectional ambient fluid flow. We then assume that a contaminant plume is "injected" at a single point in the upstream portion of the domain.

While not important until the control portion of this model in Section 2.3, we assume there are boundary ports along the wall of the domain serving as the "actuators" able to draw out the contaminant. These actuators can be compared to the ventilation ports seen inside of a public space (see Figure 2.3). Additionally, in our domain, we have "sensors" at the location of what are called the "protected points," which will determine how exactly we will control the flow field. Figure 2.4 shows our domain with example protected points. It should be noted that the image in Figure 2.3 shows a public space from a side view, while Figure 2.4 shows a top view of the system. In an actual public space, these protected points could be taken as a variety of objects in our space. As will be discussed in Chapter IV, we take these points to be "people" who are evacuating.

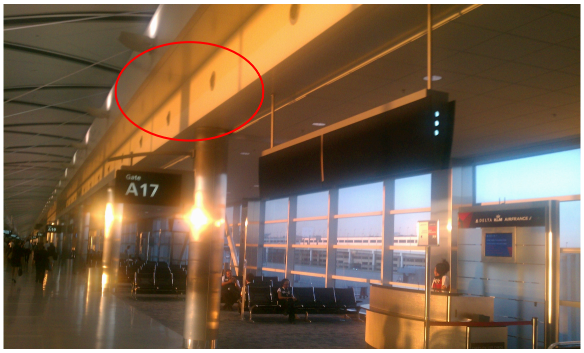


Figure 2.3: Example ports in a ventilation system.

This geometric domain obviously does not recreate that of a complex public space being studied. For example, it is 2 −dimensional. However, this simplified space instead allows for basic assumptions to be made about the underlying fluid flow which

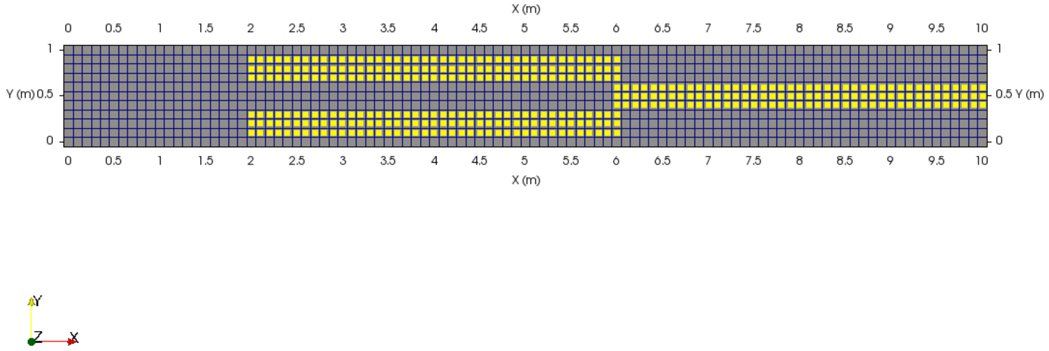


Figure 2.4: Example domain that is 10 m long and 1 m wide with protected points (denoted as yellow).

subsequently simplify the governing equations. Thus, the purpose is not to *simulate* the scenarios we are studying, but instead to provide a *model* that evinces the important fluid phenomena inherent to these scenarios of which we are trying to better understand.

2.2.2 Governing Equations

To start, we can simplify our flow model without major losses of understanding in our application by making the following basic assumptions:

- *Viscous.* The fluid is viscous, thus it has some resistance to shear stress.
- *Newtonian.* The fluid is Newtonian, which constitutes the rate of shearing stress of the fluid is linearly related to the angular deformation.
- *Incompressible.* The fluid is incompressible such that that the fluid density is

constant for its given velocity. This assumption can be made in our application because all of the fluid we will be modeling has low velocities such that the corresponding Mach number is less than 0.3 (where the Mach number is defined as the ratio of the speed of the fluid flow speed to the speed of sound).

- *Shock wave free.* Because the fluid is incompressible, this also allows us to assume the fluid is free of "shock waves." Thus, our computational grid and schemes to not need to take into consideration the severe discontinuities that would exist in the flow were there a shock wave.
- *Constant thermo-physical properties.* Our fluid is isothermal and has constant temperature conditions. This assumption allows us to exclude consideration of the energy equation in our model.

It should be noted that the assumptions above allow us to use many different types of fluids as our bulk fluid, depending on the application. For example, in hydraulic applications, it would be more appropriate to use water as the bulk fluid. While in public buildings, air is the more appropriate fluid. The assumptions apply to both of those applications. However, for this model, the bulk fluid used will be air.

We must also make assumptions regarding the contaminant that is released into our bulk fluid:

- *Non-reactive.* We assume our contaminant is non-reactive, which means it does not undergo any sort of chemical or biological transformation while in the flow.
- *Neutrally buoyant.* Our contaminant is neutrally buoyant, meaning the contaminant's density is equal to the density of the fluid in which it is immersed and gravitational effects can be dismissed.

Based on these assumptions, the contaminant in this problem behaves most like a tracer element (or often referred to as "passive scalar") released into the flow, which significantly simplifies the underlying equations used to describe our system. Furthermore, these assumptions simplify our system such that behavior of the contaminant is the same whether analyzing the system in $2-$dimensions from the side-view ($xy-$plane) versus top-view ($xz-$plane). The interchangeability of coordinate planes is important particularly in Chapter IV when human evacuation around a plume is analyzed from the top-view. When applying our model to an actual system at hand, this interchangeability in coordinate planes may not be appropriate. For example, control actions in a public space might be more appropriate for horizontal control (i.e. $xz-$plane), thus a top-view analysis would be more appropriate (or vice versa). Or as another example, many contaminants may be sensitive to gravitational effects, thus a side-view analysis would be more appropriate. For this chapter, we look at our system from a side-view (although our "neutral buoyancy" assumption means the fluid behavior is interchangeable for a top-view analysis).

It should also be noted that the assumptions above possibly disregard many of the critical physical, chemical, and biological processes that might occur in such a scenario. For actual scenarios in which specific contaminants are considered, these assumptions may not hold, and therefore, the underlying equations described below would need to reflect the specifics of the contaminant as the control scheme could significantly change. For example, much research has been carried out regarding fire in public spaces in which its heat flux leads to a phenomena of "horizontal buoyancy" in which stratification of smoke and the ambient fluid the smoke is impregnating is critical in control decisions for the smoke removal [*Wu and Bakar* (2000)]. Including more specific processes would be a rich research question in and of itself adding significant nuance to the problem at hand, and could serve as its own continued research project.

2.2.2.1 Conservation of Mass and Momentum

We can now define the "simplified" governing equations for our fluid flow as a result of the assumptions made. Viscous fluid flow is governed by a set of equations known as the Navier-Stokes equations, derived from the continuity equation and applying Newton's second law to fluid motion, and named after Claude-Louis Navier and George Gabriel Stokes. The Navier-Stokes equations are defined as:

$$\nabla \cdot \mathbf{U} = 0 \tag{2.1}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U}\mathbf{U}) = \mathbf{g} - \nabla p + \nabla \cdot (\nu \nabla \mathbf{U}) \tag{2.2}$$

where $\mathbf{U}$ is the velocity vector $[U_x, U_y, U_z]$, p is the pressure, $\mathbf{g}$ is the gravitational constant, and ν is the kinematic viscosity; and where (2.1) represents the conservation of mass, and (2.2) the conservation of momentum of a fluid. From left to right, the terms in (2.2) represent the rate of change of inertia, gravitational body force, pressure gradient, and dissipation terms.

2.2.2.2 Contaminant Transport

Additionally, we must also have another equation to represent the contaminant transport in the ambient fluid flow. As mentioned above, we have assumed that the contaminant in our model is neutrally buoyant and non-reactive. To model the contaminant, we can use the following scalar transport equation:

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{U}C) - \nabla^2(\Gamma_e C) - f = 0 \tag{2.3}$$

where C is the transported scalar, Γ_e is the effective diffusivity of the contaminant, and f is the source of the contaminant. The source, f, can be represented as an instantaneous release in both time and space using the following Dirac-Delta function

$$f = q\delta(\mathbf{x} - \mathbf{x_s})\delta(t - t_s) \tag{2.4}$$

where $\mathbf{x} = [x, y, z]^T$, $\mathbf{x_s}$ represents the coordinates of the release, t_s is the time of the release, $\dot{q}$ is the mass loading rate, and q is a mass load. For a continuous contaminant release (i.e. an instantaneous release only in space, but not time), the source term can just be represented as the following Dirac-Delta function

$$f = \dot{q}\delta(\mathbf{x} - \mathbf{x_s}). \tag{2.5}$$

2.2.2.3 Turbulence Model

The next decision that must be made regarding our CFD model is the turbulent model we will use. Turbulent flow is irregular fluid flow characterized by apparent chaotic and random changes in pressure and velocity of a fluid. Most fluid flow occurring in nature is turbulent. In order to distinguish when a fluid is turbulent, a dimensionless number known as the *Reynolds number*, which is defined as the ratio of inertial forces to viscous forces, or

$$Re = \frac{\rho U l}{\mu} \tag{2.6}$$

where ρ is the fluid density, U is the characteristic velocity, l is a characteristic length (e.g. diameter of pipe), and μ is the dynamic viscosity. For the scenario presented throughout this chapter with air at 15 °C, we have our characteristic length to be the domain width $l = 1$ m, the characteristic velocity as our maximum inlet velocity $U = 1$ m/s, our density to be $\rho = 1.225$ kg/m^3, and the dynamic viscosity to be

$\mu = 1.98 \times 10^{-5}\,\mathrm{N \cdot s/m^2}$. Thus, our Reynolds number is:

$$Re = \frac{(1.225\,\mathrm{kg/m^3})(1\,\mathrm{m/s})(1\,\mathrm{m})}{1.98 \times 10^{-5}\,\mathrm{N \cdot s/m^2}} = \mathbf{6187}. \tag{2.7}$$

In general, flows above what is known as the critical Reynolds number Re_{cr} number are considered turbulent, often with a laminar-to-turbulent transition range beforehand. Critical Reynolds numbers are dependent on the specific type of geometry of the flow domain. Some example geometries and their corresponding critical Reynolds number can be seen in Table 2.1: As our $Re = 6187$ is much greater than many of the Re_{cr} values–particularly Re_{cr} for open channel flow and pipe flow–we will treat the flow as turbulent.

Table 2.1: Critical Reynolds values Re_{cr} for various geometries [*Potter et al.* (2011)].

Geometry	Re_{cr}
Open channel	600
Pipe	2300
Golf ball	4×10^4
Smooth sphere	3×10^5
Free atmosphere	3.85×10^5
Flow over airfoil	5×10^5

While *Re* is a signifier for flows in the turbulent regime, it is not a strict physical distinguisher, and was developed empirically by Stokes and Reynolds in the 19th century. Turbulent flow is also often characterized by the existence of vortices, or the rotations of a fluid around an axis line. While some flows may have a low *Re* number (i.e. are laminar), they may still demonstrate turbulent flow (i.e. have turbulent structure such as rotation). Because a complete solution to the Navier-Stokes equations in 3-dimensions for turbulent flow has still yet to be found, developing an understanding of turbulence has largely been comprised of stochastic and qualitative efforts–the existence of vortices serving as one of the most promising guides to better understanding turbulence. One of the most important contributions to the understanding of turbulence came from *Kolmogorov* (1941) in which turbulent regime was described as having a spectrum of vortex sizes from the largest size (i.e. integral length) to the smallest length (i.e. Kolmogorov length). This spectrum of vortex sizes of a turbulent regime is described as "energy cascades" in which it is argued that energy is transferred from the large vortices to the smallest ones, which eventually dissipate this energy to the surrounding flow via viscosity. The idea of energy cascades propelled the understanding that turbulent flows also are characterized by diffusion and mixing.

Because turbulent flows are inherently included in the full Navier-Stokes equations, the most straightforward method of modeling turbulence is by solving the Navier-Stokes directly, known as Direct Numerical Simulation (DNS). However, in order to use the modeling method of DNS, the mesh used in discretization must be fine enough to entirely capture the full range of turbulent eddies, which would subsequently require a high computational cost. Therefore, the use of DNS modeling is prohibitive and is not usually computationally justified except for small-scale applications.

To overcome the prohibitive nature of DNS, different types of turbulence models have been developed for different needs of turbulent modeling accuracy. The three models most used to model turbulence, in order of increasing accuracy are:

1. *Reynolds-averaged Navier-Stokes (RANS) models.* RANS models use ensemble-averaged versions of the governing equations.

2. *Large Eddy Simulation (LES) models.* LES models use fully resolved large-scale eddies, and paramaterized small-scale eddies, and are more accurate than RANS models.

3. *Detached Eddy Simulation (DES) models.* DES models are a hybrid of RANS and LES models. DES models use RANS modeling at near-wall regions, and

LES models at the interior flow and are the most accurate of the three.

Currently for this project, fully-resolved or near fully-resolved turbulent modeling is not necessary since the focus of this research is on the real-time control of the fluid flow and not necessarily on the subsequent effects on the flow's turbulence. Thus, RANS modeling is sufficient for our turbulent modeling needs. However, depending on the nature of the application of this project, the need for a better turbulent model might become necessary. For example, if the means of mitigating the contaminant via injecting another chemical to react with the contaminant, turbulence might be more important as mixing via turbulent diffusion is desired. Or as will be discussed later in the the dissertation, increased mixing due to turbulence could be of concern for its effect on human occupants. In that sort of scenario, a different turbulence model might become necessary.

To develop the Reynolds-averaged Navier-Stokes models, we first decompose the velocity as

$$\mathbf{U} = \overline{\mathbf{U}} + \mathbf{U}' \tag{2.8}$$

where $\overline{\mathbf{U}}$ is the mean velocity and $\mathbf{U}'$ is the fluctuating velocity. The average velocity is formally defined as:

$$\overline{\mathbf{U}} = \lim_{T\to\infty} \frac{1}{T} \int\limits_0^T \mathbf{U}(x,t)dt. \tag{2.9}$$

We can substitute the decomposed velocity (2.8) into (2.1) and (2.2), and then average these new equations over time to obtain the Reynolds Averaged Navier-Stokes (RANS) equations:

$$\nabla \cdot \overline{\mathbf{U}} = 0 \tag{2.10}$$

$$\frac{\partial \overline{\mathbf{U}}}{\partial t} + \nabla \cdot (\overline{\mathbf{U}}\overline{\mathbf{U}}) = \mathbf{g} - \nabla \overline{p} + \nabla \cdot (\nu \nabla \overline{\mathbf{U}}) + \overline{\mathbf{U}'\mathbf{U}'} \tag{2.11}$$

where

$$\overline{\mathbf{U}'\mathbf{U}'} \tag{2.12}$$

of the last term in equation (2.11) is known as the Reynolds stress tensor. Because there are more unkowns than equations for RANS models, we have what is known as a *closure problem*, thus RANS turbulent models focus on creating a connection between the Reynolds stress tensor and the mean velocity field. There are various strategies to computationally solving the closure model, with the most common being the $k-\epsilon$ closure model. For this model, the turbulent kinetic energy k is used to characterize the turbulence, and the turbulent energy dissipation rate per unit mass ϵ is used to characterize the scale of the turbulence. The turbulent kinetic energy is defined as:

$$k = \frac{1}{2}\overline{\mathbf{U'} \cdot \mathbf{U'}} = \frac{1}{2}(U_x'^2 + U_y'^2 + U_z'^2) \tag{2.13}$$

with k estimated using

$$k = \frac{3}{2}(\mathbf{U}_{ref}\mathbf{I})^2 \tag{2.14}$$

where $\mathbf{U}_{ref}$ is a reference velocity typically taken as the average velocity of the flow at the boundary, and $\mathbf{I}$ is taken as 5% of the inlet velocity for fully developed turbulent flow. The turbulent energy dissipation rate per unit mass is defined as:

$$\epsilon = \frac{C_\mu^{3/4} k^{3/2}}{l} \tag{2.15}$$

where C_μ is a model constant, and l is the mixing length taken as $0.07 \cdot L$ for inlet boundaries where L is the characteristic length of the inlet. The transport equations for k and ϵ for fully defined incompressible flow are now defined as:

$$\frac{\partial k}{\partial t} + \overline{U}_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] + \left[\nu_t\left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i}\right) - \frac{2}{3}k\delta_{ij}\right]\frac{\partial \overline{U}_i}{\partial x_j} - \epsilon \tag{2.16}$$

$$\frac{\partial \epsilon}{\partial t} + \overline{U}_j \frac{\partial \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\nu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_j}\right) + C_{\epsilon 1}\frac{\epsilon}{k}\nu_t\left(\frac{\partial \overline{U}_i}{\partial x_j} + \frac{\partial \overline{U}_j}{\partial x_i}\right)\frac{\partial \overline{U}_i}{\partial x_j} - C_{\epsilon 2}\frac{\epsilon^2}{k} \tag{2.17}$$

where C_{ϵ_1}, C_{ϵ_2}, σ_k, and σ_ϵ are model constants, and ν_t is the turbulent viscosity defined as

$$\nu_t = C_\mu \frac{k^2}{\epsilon}. \tag{2.18}$$

For this model, the $k - \epsilon$ constants in this model are summarized in Table 2.2, and are standard values developed after model refinement.

Table 2.2: Empirical constants used in the $k - \epsilon$ closure model [*Versteeg and Malalasekera* (2007)].

C_μ	C_{ϵ_1}	C_{ϵ_2}	σ_k	σ_ϵ
0.09	1.44	1.92	1.00	1.30

2.2.3 Geometric Discretization for Computational Domain

The heart of CFD is converting a continuous system into a discrete space and set of equations that can be solved at every point in space and time. Discretization of a system's geometric domain (sometimes called *meshing*) is in and of itself a vital and intricate component of developing the CFD model that involves strategically partitioning said domain into smaller elements. The smaller elements then represent the components for which the algebraic equations derived from the PDEs are applied to and solved. The more "complex" the fluid flow in a certain section of the domain, the smaller and more refined the elements should be in order to capture the full flow phenomena. However, a more refined geometric domain leads to increased computational cost; thus, it is necessary to balance the need for more accurate solutions–with higher rates of convergence–against the computational expense it precipitates.

The most straight forward discretization is on a Cartesian (rectangular) grid, in which the domain is broken down into uniform, smaller rectangles, as can be seen in Figure 2.5. Fortunately, for the problem being studied throughout this dissertation, this sort of meshing is currently sufficient as the accuracy of the solution is not the most critical part of the model.

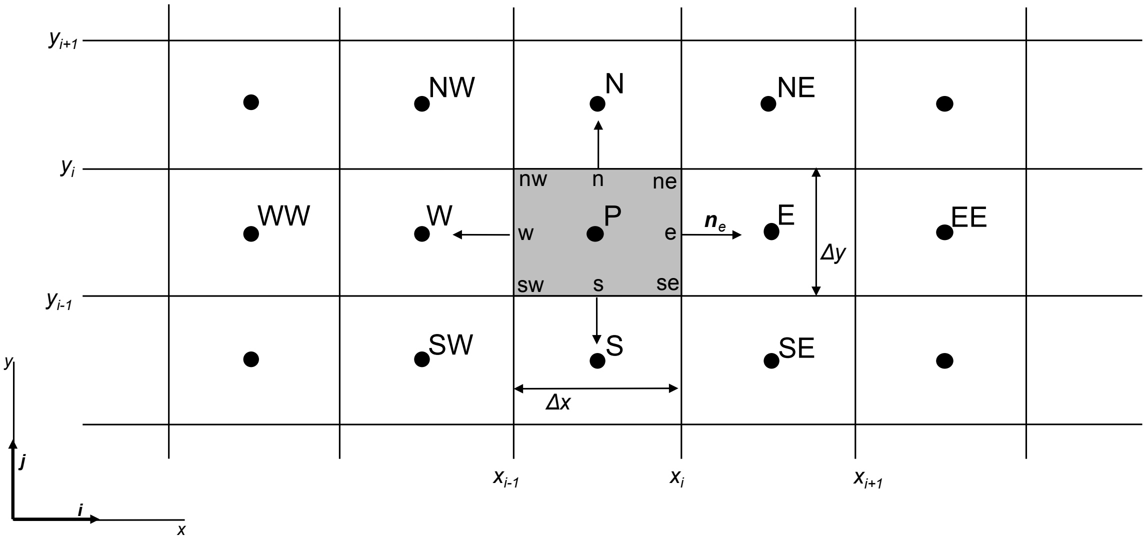


Figure 2.5: A meshed rectangular domain

.

However, for fluid flows with more complex geometries or in which there are critical areas in the fluid flow that must be elucidated, different meshing strategies become important. Decisions for more complex types of meshing usually are made concerning element shapes (e.g. 2-D triangle, 3-D pyramid); and whether or not the mesh should be structured (i.e. follows a uniform pattern) or unstructured (i.e. does not follow a uniform pattern).

Again, for the model at its current state, a structured and uniform mesh is sufficient to expose the fluid phenomena we are most interested in understanding at this point. The meshed geometric domain for this model can be seen in Figure 2.6. As can be seen, the domain is uniformly meshed with $\Delta x = \Delta y = 0.1$ m.

Depending on the discretization method of the governing equations that is used, it is either the nodes or the elements of a discretized geometry for which the solutions are estimated.

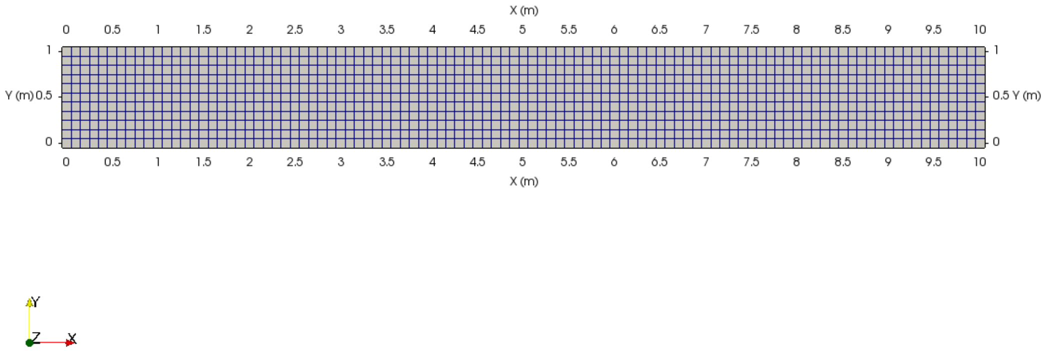


Figure 2.6: The meshed geometric domain for this model.

2.2.4 Discretization of Governing Equations

While a range of methods exist to solve PDE's, for the most part, there are three main methods to discretize PDE's in fluid dynamics:

Finite difference. The Finite Difference (FD) method is the oldest and most straightforward of these discretization approaches. It uses the Taylor Series expansion to represent the spatial and temporal derivatives within the Navier-Stokes equations to a given order and accuracy. While the FD method is the easiest to code, it is ill-suited for complex geometries and irregular boundaries, as a transformation must occur for domains that are not defined by a rectangular grid. Furthermore, as the approximations occur from point-to-point temporally and spatially, the state variables are not necessarily conserved which can lead to greater inaccuracy when applied to realistic simulations.

Finite element. The Finite Element (FE) method was originally developed for use in structural analysis before it was applied to fluid dynamics. In simple FE applications, the domain of the problem is broken into a set of unstructured 'finite elements'; the solution is then approximated by a linear shape function within each element which guarantees continuity of the solution across element boundaries. The FE method works well on coarse and complicated grids; however, is not well-suited when used for modeling turbulence. Furthermore, since the FE method integrates the equations over the entire domain, local conservation of state variables are not guaranteed, which is another disadvantage to this technique.

Finite volume. The Finite Volume (FV) method uses a solution domain that is discretized into polyhedral volumes (also called 'control volumes') with a 'cell center' and a set of faces. It then utilizes the integral form of the conservation equations applied to each control volume. The control volumes make up the entire domain without overlapping. Thus, the FV method works well for complex geometries. Furthermore, unlike the other two methods, the FV method is inherently conservative for each of the individual volumes. The inherent conservation for each individual is especially important when dealing with flows that have discontinuities, such as shock waves.

The FV method is most-widely used in CFD. Details on the exact means of discretizing using the FV method will be discussed below, and is largely drawn from the same derivation techniques and explanations in *de Villiers* (2006); *Jasak* (1996).

2.2.4.1 Gauss's Theorem

Unlike the FD method which uses the derivative to calculate the desired variable at each "node" in a meshed grid, instead the FV method uses volume integrals to calculate the variable value for each mesh element. Take the control volume of a hypothetical element from our mesh seen in Figure 2.7. Here we have our control volume V_P with a normal vector $\mathbf{A}_f$ on the face f pointing outward from the control volume V_P toward point N. Additionally, we have a vector $\mathbf{d}$ connecting our two points P and N, and is not necessarily orthogonal to our face f. For each element in our mesh, we set our field to be cell-centered such that

$$\int_{V_P} (\mathbf{x} - \mathbf{x}_P) dV = \mathbf{0}. \tag{2.19}$$

Furthermore, we assume each element's control volume is bounded by a set of flat faces, and each face is shared with only one neighboring control volume, no elements are overlapping, and all variables for an element are defined by the same control volumes. Faces can be either internal faces, between two elements; or boundary faces, at a boundary on the geometric domain. In order to ensure second-order accuracy regarding the truncation error, we must assume that all dependent variables in an element's control volume vary linearly around a point P and time t such that:

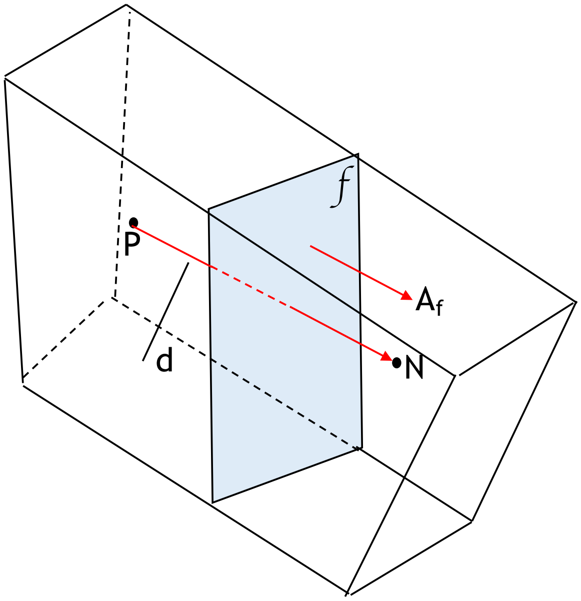


Figure 2.7: Hypothetical control volume for points P and N.

$$\phi(x) = \phi_P + (\mathbf{x} - \mathbf{x}_P) \cdot (\nabla\phi)_P, \tag{2.20}$$

$$\phi(t + \Delta t) = \phi^t + \Delta t \left(\frac{\partial \phi}{\partial t} \right)_t , \tag{2.21}$$

where

$$\phi_P = \phi(\mathbf{x}_P), \tag{2.22}$$

and

$$\phi^t = \phi(t). \tag{2.23}$$

The crux of the FV method is what is known as Gauss's Theorem (sometimes called the Divergence Theorem). Gauss's Theorem states that the rate of change of a property in a control volume is equivalent to the net flux of that property across its boundaries, or

$$\int\limits_V \nabla \cdot \phi dV = \int\limits_{\partial V} d\mathbf{A} \cdot \phi, \tag{2.24}$$

where ∂V is the enclosed surface bounding the volume V, and $d\mathbf{A}$ is an infinitesimal surface element with outward pointing normal on the surface ∂V. Essentially, Gauss' Theorem reduces the volume integral of an element to a surface equivalent. Thus, conserving the net flux of a variable across all of the components of an element's surface allows for the solution of that variable to be calculated for each element in the mesh.

Using Gauss's Theorem, we can convert our governing equations into a set of algebraic equations that can be solved at each element in our domain. For example, if we take our hypothetical control volume for our element P in Figure 2.7, along with equation 2.20, it follows that

$$\int\limits_{V_P} \phi(x) dV = \phi_P V_P \tag{2.25}$$

which is the midpoint interpolation. Because the faces of our elements are flat, we can also take the integration of the divergence operator for our hypothetical element such that

$$\int\limits_{V_P} \nabla \cdot \phi dV = \int\limits_{\partial V_P} d\mathbf{A} \cdot \phi = \sum_f \left(\int\limits_f d\mathbf{A} \cdot \phi \right) = \sum_f \mathbf{A} \cdot \phi_f. \tag{2.26}$$

With these equivalencies, we can now convert our governing equations into a set of solvable algebraic equations across the elements in our domain.

For brevity and ease, we will breakdown a general transport equation to demonstrate the discretization of the temporal, advection, diffusion, and source terms, which is the same discretization that occurs with the governing equations for our model (equations 2.1, 2.2, and 2.3). We take our general transport equation for an independent variable ϕ to be

$$\underbrace{\frac{\partial \phi}{\partial t}}_{\text{temporal derivative}} + \underbrace{\nabla \cdot (\mathbf{U}\phi)}_{\text{advection term}} - \underbrace{\nabla \cdot (\Gamma_\phi \nabla \phi)}_{\text{diffusion term}} = \underbrace{S_\phi(\phi)}_{\text{source term}} \tag{2.27}$$

where Γ_ϕ is the diffusivity, $\phi = \phi(\mathbf{x}, t)$ is a function of space and time around the point P, the temporal term $\frac{\partial \phi}{\partial t}$ represents the "accumulation" of ϕ in the control volume, the convection term $\nabla \cdot (\mathbf{U}\phi)$ represents the transport of ϕ due to the velocity field, the diffusion term $\nabla \cdot (\Gamma_\phi \nabla \phi)$ represents the transport of ϕ due to its gradients, and the source term accounts for sources or sinks that create or remove ϕ from the control volume.

2.2.4.2 Advection Discretization

To discretize our advection term, we use equation (2.26) such that

$$\int\limits_{V_P} \nabla \cdot (\mathbf{U}\phi) dV = \sum_f \mathbf{A} \cdot (\mathbf{U}\phi)_f = \sum_f (\mathbf{A} \cdot \mathbf{U}_f)\phi_f = \sum_f F\phi_f \tag{2.28}$$

where F is the volume flux through the face,

$$F = \mathbf{A} \cdot \mathbf{U}_f. \tag{2.29}$$

Since the net mass flux on an element must be zero (i.e. the FV continuity equation is obeyed), we have that

$$\int_V \nabla \cdot \mathbf{U} dV = \int_{\partial V} d\mathbf{A} \cdot \mathbf{U} = \sum_f \left(\int_f d\mathbf{A} \cdot \mathbf{U} \right) = \sum F = 0. \tag{2.30}$$

Now the next trick is to interpolate for ϕ at the face in order to find ϕ_f. There are a multitude of approaches to do so, some with higher accuracy than 2^{nd} order. However, higher order accuracy does not necessarily bring stable and non-oscillatory behavior. For this model, a combination of two interpolation approaches are used.

Central Differencing. The first approach known as Central Differencing (CD) is just a simple linear interpolation between the two element cell-centers,

$$\phi_f = f_x \phi_P + (1 - f_x)\phi_N, \tag{2.31}$$

where f_x is defined as the ratio of distances $\overline{fN}$ and $\overline{PN}$, such that

$$f_x = \frac{\overline{fN}}{\overline{PN}}. \tag{2.32}$$

A schematic of this interpolation scheme can be seen in Figure 2.8. This method has been shown to be 2^{nd} order accurate for even unstructured meshes. However, it has drawbacks in that for advective-dominant flow, the numerical solutions will become increasingly non-physical and will possibly diverge. As a method of alleviation, a higher order filter in the form of a dissipation term may be applied, particularly in structured mesh in which it is easier to implement.

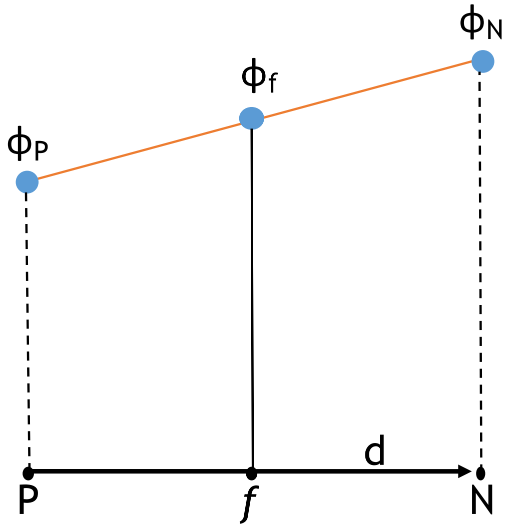


Figure 2.8: Schematic of interpolation across face values in the FV method.

Upwind Differencing. For a first-order accurate variation of Upwind Differencing (UD), the face value of ϕ_f is determined based on the direction of the face's flux such that

$$\phi_f = \begin{cases} \phi_P, & \text{if } F \geq 0 \\ \phi_N, & \text{if } F < 0. \end{cases} \tag{2.33}$$

While UD improves upon stability and boundedness on the numerical solutions, it is unfortunately prone to introducing numerical diffusion into the system, and additional approaches must be taken for element *Re* numbers above a critical value.

2.2.4.3 Diffusion Discretization

The diffusion term from equation (2.27) can be discretized such that,

$$\int\limits_{V_P} \nabla \cdot (\Gamma_\phi \nabla_\phi) dV = \sum_f \mathbf{A} \cdot (\Gamma_\phi \nabla \phi)_f = \sum_f (\Gamma_\phi)_f \mathbf{A} \cdot (\nabla \phi)_f \tag{2.34}$$

where the subscript f denotes a face interpolated quantity. An interpolated solution to $(\Gamma_\phi)_f$ can be found in the same manner as the CD scheme described above using an equivalent form of equation (2.31). For mesh and geometric domain of this model, because all of the elements are orthogonal to each other, determining the right hand side of equation (2.34) is straightforward. The vector $\mathbf{A}$ is parallel to $\mathbf{d}$, and the face gradient of ϕ can be expressed as the following:

$$\mathbf{A} \cdot (\nabla \phi)_f = |\mathbf{A}| \frac{\phi_N - \phi_P}{|\mathbf{d}|}. \tag{2.35}$$

It should be noted that this approach is different than a simple interpolation of $(\nabla \phi)_f$ across cell-centers, and is actually more accurate with less truncation error (although both approaches are still 2^{nd} order accurate). For meshes with elements that are non-orthogonal to their neighbors, another step must be taken in addition to equation (2.35) to estimate the interpolated value and correct for non-orthogonality.

2.2.4.4 Source Terms

The source terms encompass all terms in the generic transport equation that are not written as advection, diffusion, or temporal terms, and are usually a function of ϕ plus other variables. To discretize it, we linearize the source term such that:

$$S_\phi(\phi) = S_c + S_p \phi \tag{2.36}$$

where S_c and S_p are constant and linear components of the source terms, respectively, and may be functions of ϕ. Using the same midpoint interpolation as equation (2.25), we calculate the volume integral as

$$\int_{V_P} S_\phi(\phi) dV = S_c V_P + S_p V_P \phi_P. \tag{2.37}$$

2.2.4.5 Temporal Discretization

In order to discern exactly how to discretize the temporal component of our PDE, let us first return to equation (2.27) and integrate the equation over the control volume and in time to produce the integral form of the generic transport equation:

$$\begin{aligned}\int_t^{t+\Delta t} \left[\frac{\partial}{\partial t} \int_V \phi t dV + \int_V \nabla \cdot (\mathbf{U}\phi) dV - \int_V \nabla \cdot (\Gamma_\phi \nabla \phi) dV \right] dt \\ = \int_t^{t+\Delta t} \left[\int_V S_\phi dV \right] dt.\end{aligned} \tag{2.38}$$

After substituting in our equivalencies from equations (2.28), (2.34), and (2.37), we now have:

$$\begin{aligned}\int_t^{t+\Delta t} \left[\left(\frac{\partial \phi}{\partial t} \right)_P V_P + \sum_f F \phi_f - \sum_f (\Gamma_\phi)_f \mathbf{A} \cdot (\nabla \phi)_f \right] dt \\ = \int_t^{t+\Delta t} [S_e V_P + S_p V_P \phi_P] dt.\end{aligned} \tag{2.39}$$

Temporal discretization must occur for the temporal derivative, as well as the spatial terms (i.e. advection, diffusion, and source term). However, the method for temporal discretization does not need to be the same for all terms, as long as the accuracy is 2^{nd} order.

For ease of implementation and low computational cost while still preserving 2^{nd} order accuracy, the preferred method for time discretization is the **Second-Order Backwards Differencing**. To derive it, the Taylor series expansion is taken using three time levels:

$$\phi^{n-2} = \phi(t - \Delta t) \tag{2.40}$$

$$\phi^{n-1} = \phi(t) \tag{2.41}$$

and

$$\phi^n = \phi(t + \Delta t). \tag{2.42}$$

The Taylor series expansion for ϕ^{n-2} is:

$$\phi^{n-2} = \phi^n - 2\left(\frac{\partial \phi}{\partial t}\right)^n \Delta t + 2\left(\frac{\partial^2 \phi}{\partial t^2}\right)^n \Delta t^2 + O(\Delta t^3) \tag{2.43}$$

and for ϕ^{n-1} is

$$\phi^{n-1} = \phi^n - \left(\frac{\partial \phi}{\partial t}\right)^n \Delta t + \frac{1}{2}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^n \Delta t^2 + O(\Delta t^3). \tag{2.44}$$

Combining equations (2.43) and (2.44) yields a second order approximation of the temporal derivative at n:

$$\left(\frac{\partial \phi}{\partial t}\right)^n = \frac{\frac{3}{2}\phi^n - 2\phi^{n-1} + \frac{1}{2}\phi^{n-2}}{\Delta t} \tag{2.45}$$

If the temporal variation in the face fluxes and derivatives are neglected, then equation (2.45) gives:

$$\frac{\frac{3}{2}\phi^n - 2\phi^{n-1} + \frac{1}{2}\phi^{n-2}}{\Delta t} V_P + \sum_f F\phi_f^n - \sum_f (\Gamma_\phi)_f \mathbf{S} \cdot (\nabla\phi)_f^n = S_c V_P + S_p V_P \phi_P^n \tag{2.46}$$

which is a fully implicit second order accurate discretization for the hypothetical transport equation. The major drawback to the Backward Differencing method is the truncation error due to the lack of temporal variation in face fluxes and derivatives, which subsequently results in artificial diffusion. To handle this error, a cell-face Courant number is defined as

$$CFL = \frac{\mathbf{U}_f \cdot \mathbf{d}}{\Delta t} \tag{2.47}$$

where Δt is chosen such that CFL is less than 1 for stability.

2.2.5 Boundary Conditions

Most CFD models become well-posed when sufficient boundary and initial condition assumptions are made that allow for the forward in time and space iteration of the computational solution. In fact, to ensure that a numerical solution can be developed for a CFD model, the underlying set up of the model is said to be well-posed when the solution to the PDE governing equation exists, is unique, and continuously depends upon its initial and boundary conditions. There are a multitude of boundary and initial conditions that can be used to help define the computational model. Fortunately, for the problem investigated throughout this research, the most basic of these assumptions are made, and thus discussed throughout this section.

There exist two types of boundary conditions that must be discussed: (i) numerical boundary conditions, and (ii) physical boundary conditions. The most basic description of the boundary conditions is that they are either (i) fixed value, also known as a Dirichlet condition, or (ii) fixed gradient, also known as a Neumann condition. Additionally, there is a boundary condition that is implemented for the boundary layer flow of fully-developed flow, in this case called wall-treatment. Together, these three boundary conditions are used in this model.

2.2.5.1 Numerical Boundary Conditions

For the fixed value and fixed gradient boundary conditions must be developed for the diffusion and advection terms previously derived. Consider an arbitrary element from our mesh along the boundary of a domain with a boundary face b. It is assumed that the boundary conditions a b is the same for the whole of the element's boundary face.

Fixed Value. For the fixed value conditions, let ϕ_b be the value of ϕ at the boundary face b.

- *Advection.* The fixed value boundary condition for the discretized advection term given in equation (2.28), is as follows:

$$\int_{V_P} \nabla \cdot (\mathbf{U}\phi) dV = \sum_f F\phi_f \tag{2.48}$$

  where the boundary face is

$$(F\phi_f)_{f=b} = F_b\phi_b \tag{2.49}$$

  and F_b is the specified flux across the boundary face.

- *Diffusion.* The fixed value boundary condition for the discretized diffusion term given in equation (2.34), is as follows:

$$\int_{V_P} \nabla \cdot (\Gamma_\phi \nabla\phi) dV = \sum_f (\nabla_\phi)_f \mathbf{A} \cdot (\nabla\phi)_f \tag{2.50}$$

  where the boundary face is

$$\left((\nabla_\phi)_f \mathbf{A} \cdot (\nabla\phi)_f\right)_{f=b} = (\nabla_\phi)_b \mathbf{A} \cdot (\nabla\phi)_b. \tag{2.51}$$

  For the diffusion case, if the boundary was non-orthogonal (which is not the case for this model), an additional interpolation step would need to be made for **A**.

Fixed Gradient. A fixed gradient boundary condition prescribes a normal gradient g_b on a boundary face such that

$$\left(\frac{\mathbf{A}}{|\mathbf{A}|} \cdot \nabla\phi\right)_b = g_b. \tag{2.52}$$

- *Advection.* For the advection term, ϕ is calculated using the gradient and cell centered value as the following:

$$\phi_b = \phi_P + \mathbf{d}_n \cdot (\nabla\phi)_b = \phi_P|\mathbf{d}_n|g_b. \tag{2.53}$$

- *Diffusion.* Because the face gradient is prescribed, the diffusion term for the fixed gradient is simply

$$(\Gamma_\phi)_b|\mathbf{A}|g_b. \tag{2.54}$$

2.2.5.2 Physical Boundary Treatments

For the physical aspects of the model we have built, we make the following boundary condition assumptions:

- *Inlet boundary.* The velocity at the inlet is prescribed as a constant parabolic value (i.e. the flow is fully developed), with its vertex reaching a constant velocity of 1 m/s. The boundary condition on the inlet for pressure is zero gradient.
- *Outlet boundary.* For the outlet boundary, it must be calculated such that the overall mass balance for the domain is satisfied. For this application, we can do this by specifying the pressure at the outlet boundary, but not the velocity. The pressure at the outlet is given atmospheric pressure (or relative zero pressure), with zero gradient boundary condition on velocity. It should be noted that this interdependence between the velocity and pressure will be discussed further in Section 2.2.6.
- *Port boundaries.* Port boundaries are given an orthogonal uniform velocity at their face, depending on what the optimized velocity is found to be. The boundary condition on the ports for pressure is zero gradient.

- *Symmetry plane boundary.* The component of the gradient normal to the boundary is fixed to zero for a symmetry plane boundary.

- *Impermeable no-slip walls.* At the walls, the no-slip condition applies, which means that the velocity of the fluid on the wall is equal to that of the wall itself. Thus, it is a fixed boundary layer condition. Additionally, the pressure gradient condition is zero gradient.

We must also take into special consideration the influence of the boundary layer at near-wall flow regions. In particular, additional constraints must be applied to account for the existence of high shear and large gradients brought on by turbulent flow. Before describing these constraints, however, the specific considerations of the near-wall flow must be described. For near-wall flow, there are three different velocity regions, majorly affected by the viscosity of the fluid [*Pope* (2001)]. For the region nearest the flow, the "no-slip" condition is present such that the velocity of the fluid at the wall is equal to the wall velocity itself. However, additionally, this implies that the velocity varies linearly as the distance from the wall increases, or rather

$$u^+ = y^+ \qquad\qquad 0 < y^+ < 5 \tag{2.55}$$

where u^+ is the non-dimensional velocity, and y^+ is the non-dimensional distance from the wall.

Moving away from the wall is the region known as the logarithmic inertial layer where Reynolds stresses are more dominant in the flow. For that region, we have a log-law relationship that describes the velocity profile:

$$u^+ = \frac{1}{\kappa}\ln y^+ + C \qquad\qquad y^+ > 30 \tag{2.56}$$

where κ and C are constants. The region $5 < y^+ < 30$ is known as the buffer region where both viscous and Reynolds stresses affect the flow, so neither equation (2.55) or equation (2.56) hold. The log-law relationship for velocity profile in a boundary layer can be seen in Figure 2.9.

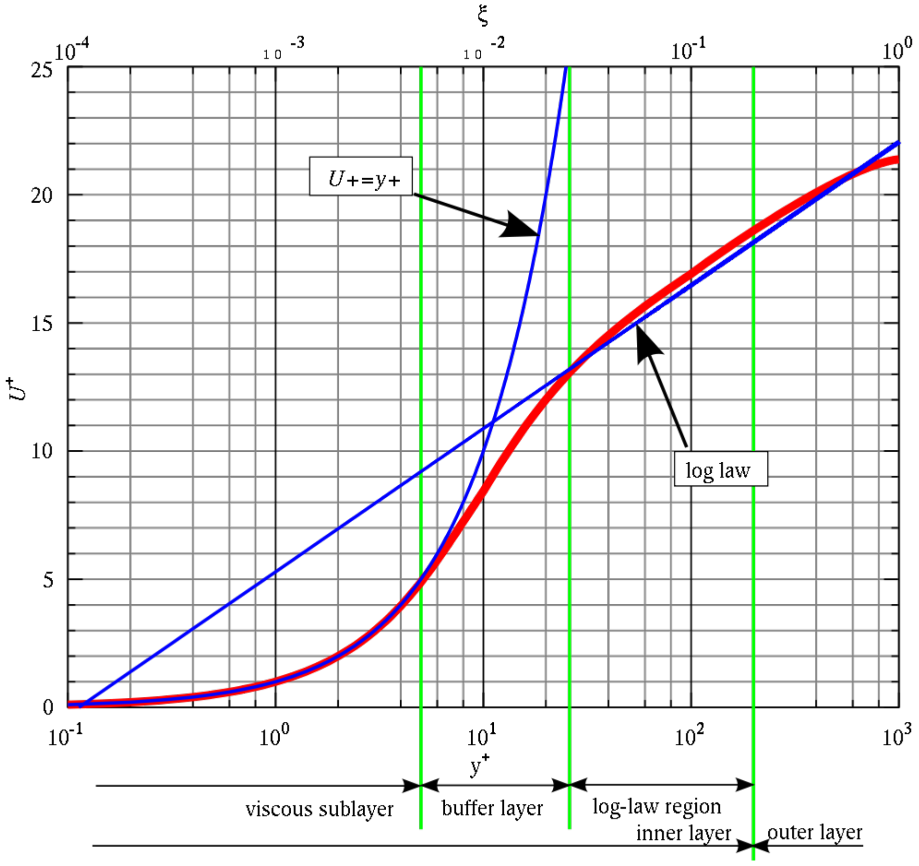


Figure 2.9: Graph demonstrating the law of the wall.

There are many ways to take into account the boundary layer. For example, a refined mesh closer to a boundary is one method. However, to fully resolve the boundary layer can be computationally expensive. Thus, when full resolved flow is not critical to the solution of the model, semi-empirical *near wall treatments* are used instead. These wall treatments estimate u^+ for the location y^+ based on equation (2.56) and are sufficient estimations for this model that reduce computational cost and increase computational speed.

A table summarizing the boundary conditions used for this model can be seen in Table 2.3.

Table 2.3: Boundary conditions for the model at hand.

	$p(\mathrm{m}^2/\mathrm{s}^2)$	$U(\mathrm{m/s})$	$C(\text{units})$	$k(\mathrm{m}^2/\mathrm{s}^2)$	$\epsilon(\mathrm{m}^2/\mathrm{s}^3)$
Inlet	zero gradient	parabolic $U_{max} = 1$	0	7.4×10^{-4}	4.7×10^{-5}
Outlet	0	zero gradient	zero gradient	zero gradient	zero gradient
Walls	zero gradient	0	zero gradient	wall function	wall function
Ports	zero gradient	normal fixed value	zero gradient	zero gradient	zero gradient

2.2.6 Solving Navier-Stokes

One of the major issues when solving the incompressible Navier-Stokes equations is the weak coupling between the pressure and velocity terms. In particular, the conservation of mass of equation (2.1) does not include the pressure term when flow is incompressible, and with the conservation of momentum of equation (2.2) the pressure is not a primary variable. Thus, in order to solve the set of Navier-Stokes equations, we must develop a coupling between pressure and velocity.

For this research, the method used to fully solve the Navier-Stokes equations known as the Pressure Implicit with Splitting of Operator (PISO) procedure for transient systems (a variation of the method known as Semi-Implicit Method for Pressure Linked Equations (SIMPLE) for steady-state flows), and was developed by *Issa* (1986). A simplified version of the PISO algorithm can be seeing in Algorithm 1.

To solve for pressure, we must first write the momentum equation in a semi-discretized form:

$$a_P \mathbf{U}_P + \sum_N a_N \mathbf{U}_N = \frac{\mathbf{U}_0}{\Delta t} - \nabla p \tag{2.57}$$

where $\mathbf{U}_0$ is the latest available velocity field, a_P is the matrix of coefficients to cell P, and a_N is the matrix coefficient corresponding to neighboring cell N. We now introduce a correction operator $\mathbf{H}(\mathbf{U})$ such that

$$\mathbf{H}(\mathbf{U}) = \frac{\mathbf{U}_0}{\Delta t} - \sum_N a_N \mathbf{U}_N. \tag{2.58}$$

Substituting $\mathbf{H}(\mathbf{U})$ into equation (2.57) yields the following velocity field:

$$\mathbf{U}_P = \frac{\mathbf{H}(\mathbf{U}) - \nabla p}{a_P} \tag{2.59}$$

which interpolating at the face then yields

$$\mathbf{U}_f = \frac{(\mathbf{H}(\mathbf{U}))_f - (\nabla p)_f}{(a_P)_f}. \tag{2.60}$$

Next, we discretize the continuity equation such that:

$$\nabla \cdot \mathbf{U} = \sum_f \mathbf{A}_f \cdot \mathbf{U}_f = \sum_f F = 0 \tag{2.61}$$

where F is the flux through the cell face. Using equations (2.60) and (2.61), we can derive F as

$$F = \mathbf{A}_f \cdot \mathbf{U}_f = \mathbf{A}_f \cdot \left(\frac{(\mathbf{H}(\mathbf{U}))_f - (\nabla p)_f}{(a_P)_f} \right). \tag{2.62}$$

Additionally, substituting equation (2.60) into equation (2.61) yields the pressure equation:

$$\nabla \cdot \left(\frac{\nabla p}{a_P} \right) = \nabla \cdot \left(\frac{\mathbf{H}(\mathbf{U})}{a_P} \right) = \sum_f \mathbf{A}_f \cdot \left(\frac{\mathbf{H}(\mathbf{U})}{a_P} \right)_f. \tag{2.63}$$

Finally, we have the Navier-Stokes equations for the pressure-velocity coupling:

$$a_p \mathbf{U}_P = \mathbf{H}(\mathbf{U}) - \sum_f \mathbf{A}_f \cdot p_f \tag{2.64}$$

$$\sum_f \mathbf{A}_f \left(\frac{\nabla p}{a_P} \right)_f = \sum_f \mathbf{A}_f \cdot \left(\frac{\mathbf{H}(\mathbf{U})}{a_P} \right)_f. \tag{2.65}$$

These equations are used in the PISO loop in Algorithm 1.

Algorithm 1 PISO algorithm

```
Decide on previous values of $F$ and $p$.
Predict $\mathbf{U}$ using momentum equation
PISO loop:
for Number of PISO corrector steps do
    Update $\mathbf{H}(U)$                    ▷ Mass fluxes at cell faces
    Solve for $p$
    Correct for $\mathbf{H}(U)$
    Correct for $\mathbf{U}$ and $F$          ▷ Using new $p$
$t = t + \Delta t$
```

2.2.7 Software Employment

An often overlooked but nontrivial aspect of implementing CFD models is the programming paradigm used. When computational methods in engineering were first deployed, the most straightforward approach was to utilize what is known as *procedural programming*, often called *imperative programming*, in which a list of instructions is given to a computer regarding what to do step-by-step. These procedures usually involve a combination of routines and subroutines for the different computations that must be carried out for the computational model. Some of the earliest procedural programming languages include C, Fortran, and COBOL which have been around since the 1950s. In fact, the Fortran language was originally developed by IBM specifically for computationally intensive areas within engineering and science. For procedural programming, no matter how complicated the physical model is, it is usually reduced to low-level mathematics, in which the basis of the program is defining and employing functions for manipulation of individual floating-point values.

However, procedural programming has been limited in its ability to encompass or reach more complex problems. Almost concurrently with the development of procedural programming, another programming paradigm was also being developed, known as Object Oriented Programming (OOP), which is an approach to problem solving in which *objects* are the basic unit, and what are known as *classes* are the “blueprints” to the different objects. To develop an OOP, a class is first defined, and then objects are built from the defined classes. Put another way, a class is a concept and object is an embodiment of that concept. For procedural programming, data and the related functions are treated separately; for OOP, data and the related functions are grouped together into classes, and then employed individually as objects.

OOP has streamlined the ability to develop and employ programs for complex applications. The ability to implement a range and quality of solvers settings and solution algorithms for various physical models is what sets OOP apart from procedural programming. Frankly, using OOP, more complicated behavior can be modeled with less code. Additionally, it is easier to modularize and develop monolithic general-purpose computational tools using OOP.

2.2.7.1 OpenFOAM

In the late 1980s–motivated by similar applications for FE methods used in structural analysis, and with the desire to capitalize on the capabilities of OOP for CFD applications–researchers headed by Henry Weller from Imperial College in London developed what is currently known as Open Source Field Operation and Manipulation (OpenFOAM). The OpenFOAM CFD Toolbox is an open-source CFD software code of a set of C++ libraries that is fully customizable and is now overseen and maintained by OpenCFD Ltd. at ESI Group, and distributed by the OpenFOAM Foundation.[3] It is becoming a favored and popular CFD software system amongst engineers and scientists due to its extensive libraries and models ready for use, its accessible syntax, and its unstructured grid capabilities. It uses the FV method when solving the partial differential equations governing fluid dynamics problems.

At the core of OpenFOAM is the ability to create executables, known as applications. These applications fall into two categories: (i) solvers, designed to solve a specific problem in continuum mechanics; and (ii) utilities, designed to perform tasks

[3] It is available free for download at `http://www.openfoam.com`.

that involve data manipulation. The reach of OpenFOAM is extensive because of these applications. For example, custom objects can be created by a user (e.g. a boundary condition) that can be used with already existing solvers without the need for redeveloping the source code.

Furthermore, the most distinguishing and appealing aspect of OpenFOAM is its syntax for tensor operations and PDEs, which closely resembles the equations to be solved. Take for example, the transport equation (2.3) that governs the contaminant transport for this model:

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{U}C) - \nabla^2(\Gamma_e C) - f = 0.$$

In OpenFOAM, it is coded as the following:

```
fvScalarMatrix ContTransEqn
    (
        fvm::ddt(C)
      + fvm::div(U, C)
      - fvm::laplacian(gammaE, C)
      - f ==
      0
    );
ContTransEqn.solve();
```

For this research, the OpenFOAM application `pisoFoam` was modified to include solving for the above transport equation (2.3) for the contaminant. As can be inferred from the name, the `pisoFoam` application is a transient solver for incompressible flow, which can utilize a generic turbulent model, and uses the PISO iterative solver from Section 2.2.6 for the pressure variable[4].

[4] The PISO algorithm was actually developed within the same research group at Imperial College in London as the founding members of the OpenFOAM software suite.

2.2.7.2 ParaView for Visualization

Another software decision made for the development of this CFD model is the visualization software used. While different software exists for flow visualization and analysis (e.g. Tecplot, VisIT), the desire also existed to utilize another open-source software for post-processing. Thus, used in this research is ParaView, an open-source, freely available software system for 3D computer graphics, image processing, and visualization that was developed starting in 2000 in a coordinated effort between Kitware Inc. and Los Alamos National Laboratory. ParaView is an application built on top of the Visualization Toolkit (VTK) software system, an open-source set of C++ libraries for data processing.

OpenFOAM already has a robust utilization with ParaView, with a built-in reader module that can be easily accessed with the command `paraFoam`. One of the appealing components of ParaView is the ability to use Python scripting for post-processing.

2.3 Optimal Flow Control

The concept of "optimal control" is actually quite basic: use mathematical optimization to find the best control action. Flow control in aerospace engineering has focused on the reduction of turbulence, drag, and boundary layer manipulation to improve the aerodynamic performance of aircraft in fluid flow. A first approach has been to optimize the design of an aircraft's airfoil, and/or the aircraft's wing-body design by using passive optimal flow control. Jameson et al. were able to optimize the wing and wing-body combinations for a long range transport aircraft in viscous, compressible fluid flow [*Jameson* (1988); *Jameson et al.* (1998); *Jameson* (1999)]. In particular, a cost function was selected to represent some sort of design parameter, such as the drag coefficient or the lift to drag ratio, and optimization was carried out using a quasi-Newton optimization technique.

Yet, the early uses of optimal control were concerned more with shape optimization, and thus were largely *passive* forms of control without feedback. Later techniques in aerospace engineering focused on active flow control of turbulence via wall transpiration (i.e. suction and blowing along a boundary). The initial idea for drag reduction and turbulence control via transpiration was developed by *Lions* (1966) and *Abergel and Temam* (1990). *Bewley et al.* (2001) then extended the idea of wall transpiration for turbulence control into a feedback control problem. They utilized optimal control by developing a cost function based on the drag distributed along a channel wall in order to then reduce that drag. The cost function was minimized using an adjoint sensitivity approach, and updating the control with a gradient algorithm. Altogether, with this technique, they were able to reduce drag along boundary walls by up to 17%. *Bewley et al.* (1993) continued the use of active feedback control for drag reduction by developing a cost function based on the turbulent kinetic energy of the channel, while *Berggren* (1998) developed a cost function based on the vorticity of the flow (i.e. the curl of the velocity).

For civil and environmental engineering, the use of optimal control has been extensively applied to flood wave control in civil hydraulic systems, and contaminant detection and release and/or mitigation in environmental bodies of water. *Sanders and Katopodes* (1999) developed an optimal approach to gate motion of a reservoir so as to prevent overtopping of a dam and downstream flooding when faced with an incoming flood wave. Furthermore, *Sanders and Katopodes* (2000) developed a method for identifying an optimal location and flow diversion to actively intervene with an oncoming flood wave. *Piasecki and Katopodes* (1997a) developed a numerical model for the optimal release of hazardous contaminants into shallow rivers and estuaries in order to minimize the environmental impact of the contaminants. *Alvarez-Vzquez et al.* (2009) used optimal control and adjoint equations in order to optimally dilute a contaminant to a certain concentration in a section of a river via the injection of pure water. *Katopodes* (2009) demonstrated the feasibility of using sets of sensors and actuators to detect and mitigate sudden contaminant releases in channel flow.

One area of potential application for optimal flow control systems is by using existing ventilation systems currently in public infrastructure settings. *Sreedharan et al.* (2011) have developed a Bayesian-based model to be able to characterize a biological or chemical contaminant in a building's ventilation system in real-time. *Gao et al.* (2012) show the results of four different hybrid ventilation designs (i.e. natural and mechanical ventilation components) with the potential to minimize fire-induced carbon monoxide concentration. *Shih et al.* (2011) demonstrate the possibility of utilizing an air curtain to curtail the spread of gaseous ethanol released from a malfunctioning machine inside a clean room.

2.3.1 Model Predictive Control

For optimal control in a dynamic system, we follow the method of MPC model predictive control (MPC) discussed in *Grune and Pannek* (2011). The goal of model predictive control (MPC) is to either *track* or *stabilize* a system. Suppose we are given a controlled process with a state $x(n)$ that is measured at discrete time instants t_n for $n = 0, 1, 2, \ldots$. We *control* our system at each time instant by selecting a control input $u(n)$. This control input will influence the behavior of the future state of the system. For our specific application, we define our cost functional based on the concentration of the contaminant for a given time period at chosen "protected points" within our domain, and constrained by the velocities of actuators along the boundary. When the cost function is minimized, a vector of velocities are determined that are prescribed to the corresponding actuators on the domain's boundary and results in minimized contaminant concentration at the protected points. For each minimization iteration, we use the DAKOTA optimization toolkit with a quasi-Newton solver. Together, the OpenFOAM CFD and DAKOTA optimization make up the *CFD flow-control model.*

For *tracking control*, we select our control input $u(n)$ such that our system $x(n)$ follows a reference x^{ref} as best as possible. For this problem, let $x(n) \in X = \mathbb{R}^d$ and $u(n) \in U = \mathbb{R}^m$. To reduce our tracking control problem to *stabilization control* problem, we select our reference to be a constant such that $x^{ref}(n) = x_* = 0$ for all $n \geq 0$.

For a linear or nonlinear system of the form

$$x^+ = f(x, u) \tag{2.66}$$

where $f : X \times U \to X$ is a known and general nonlinear map that assigns a state x and a control u to the successor state x^+ at the next time instant. For any given control sequence $u(0), \ldots, u(N-1)$ with a horizon length $N \geq 2$, and starting at the current state $x(n)$, we can iterate equation (2.66) to construct a prediction trajectory x_u such that

$$x_u(0) = x(n), \qquad x_u(k+1) = f(x_u(k), u(k)), \qquad k = 0, \ldots, N-1. \tag{2.67}$$

We now can obtain predictions $x_u(k)$ for the state of a system $x(n+k)$ at time t_{n+k} in the future, which subsequently allows us to predict the behavior of our system on the discrete time interval $t_n, \ldots, t_{n+N}$ based on the chosen control sequence $u(0), \ldots, u(N-1)$.

Our optimal control comes in when we must choose our control sequence $u(0), \ldots,$ $u(N-1)$ such that the distance between x_u and $x_* = 0$ is minimized. We can define a cost function $\ell(x_u(k), u(k))$ to represent the distance between $x_u(k)$ and $x_* = 0$ for $k = 0, \ldots, N-1$ (i.e. the greater the distance between $x_u(k)$ and $x_* = 0$ the higher the cost to our control problem). Our optimal control problem now reads

$$\text{minimize} \qquad J(x(n), u(\cdot)) = \sum_{k=0}^{N-1} \ell(x_u(k), u(k)). \tag{2.68}$$

We can choose any function we would like to represent our cost function. For example, we could choose the standard distance formula between two points on a Euclidean space to represent our cost function. Furthermore, we can add in penalization for the different possible control values $u(k)$ to favor certain control actions over others (e.g. if we would like to choose a control action that uses less energy than another).

Because we would like to eventually distribute control actions over more than one time interval, we must also develop a means of *feedback* to our system. Thus, we can put our control action $u(n)$ in feedback form by define μ to map the state $x \in X$ into the set U of control values such that

$$u(n) = \mu(x(n)). \tag{2.69}$$

An illustration of this complete feedback control system at time step t_n can be seen in Figure 2.10.

2.3.2 Model Predictive Control Applied to CFD Model

For the CFD model developed in Section 2.2, our cost functional for MPC is

$$J(\mathbf{f}) = \sum_{p=1}^{N_p} \int_0^T C(\mathbf{x}_p)dt, \tag{2.70}$$

subject to

$$|U_i| \leq U_{max} \tag{2.71}$$

where $\mathbf{f}$ is the vector of unknown port velocities for i number of ports with $\mathbf{f} = (U_1, U_2, \ldots, U_i)$, $\mathbf{x}_p$ is a vector of coordinates corresponding to the location of the p^{th} protected point, N_p denotes the total number of protected points in the domain, and U_{max} is the maximum velocity of a port.

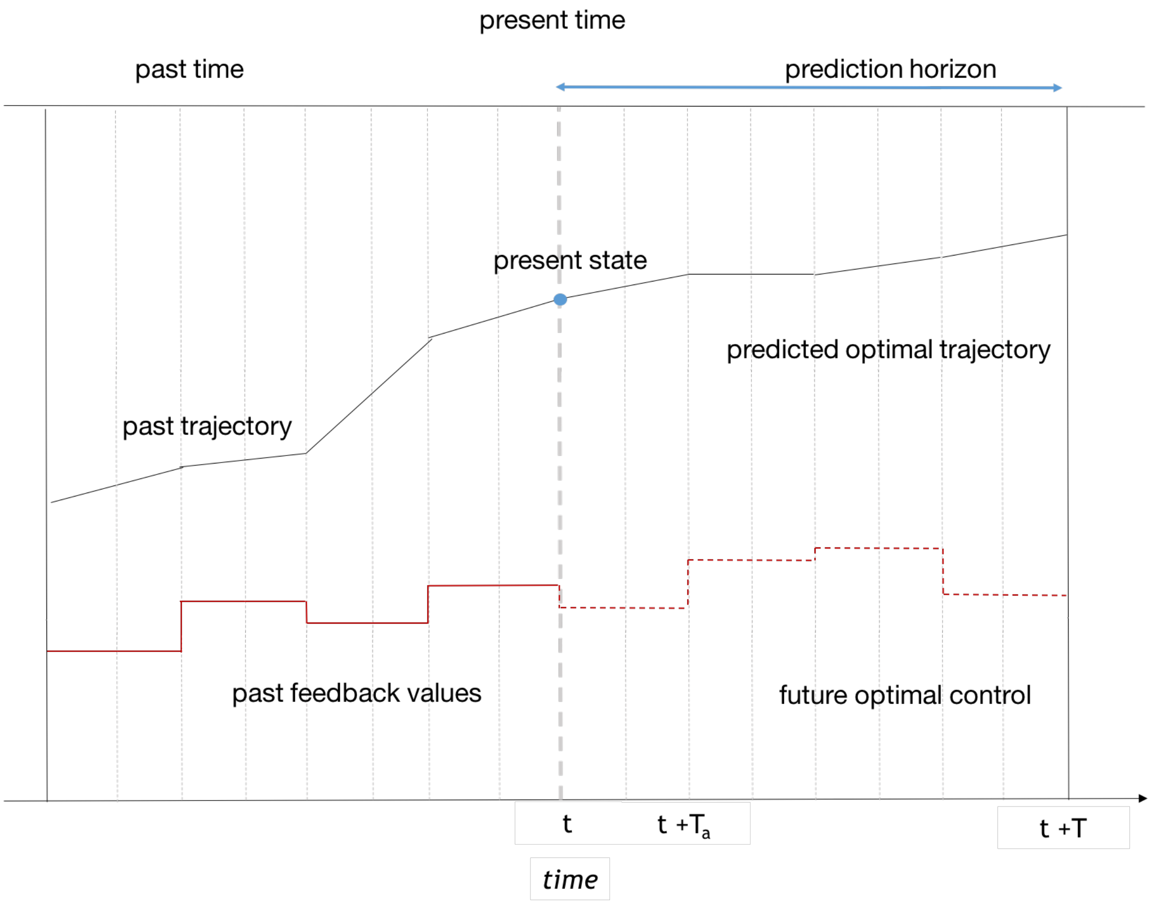


Figure 2.10: Conceptual framework of Model Predictive Control.

We minimize equation (2.70) (i.e. minimize the concentration of the contaminant at the given protected points) subject to the underlying equations of the CFD model. After minimizing equation (2.70), we end with a vector of velocities $\mathbf{f}$ that are to be prescribed to the corresponding control ports on our domain boundary.

It should be noted that for our problem of minimizing the concentration at the protected points with more than one port, there is not necessarily a single solution to the optimization problem. Furthermore, small errors in the optimization trajectory are unimportant as long as the end results achieves the goal of sufficiently mitigating the plume. This fortunately allows for some forgiveness in the optimization routine which is especially important in real-time control scenarios.

This optimization occurs for a given prediction horizon T, but is only actually implemented for a shorter time period T_a. The optimization then occurs for another prediction horizon T and applied again for only a time T_a. An image demonstrating the advancing prediction horizon can be seen in Figure 2.11. This iterative process occurs until a given termination criterion is met. The algorithm for the MPC optimization can be seen in Algorithm 2.

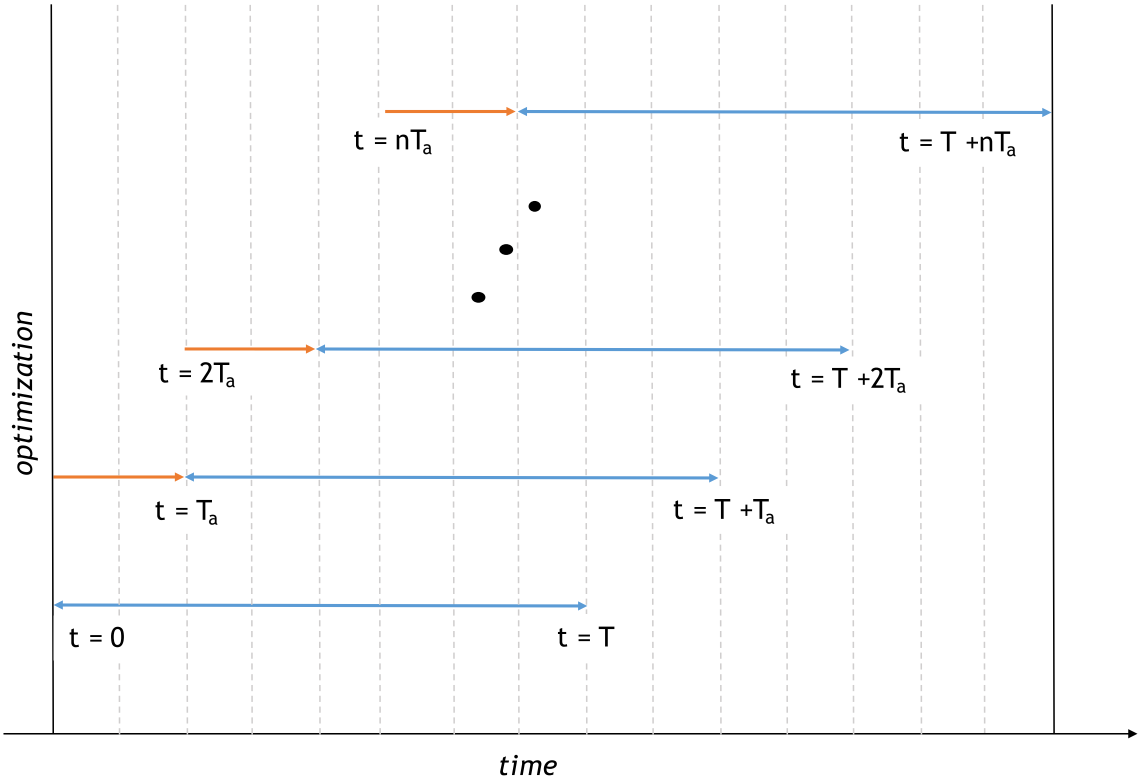


Figure 2.11: Prediction horizon for MPC routine.

2.3.3 Optimization Method

To implement MPC, we must iteratively optimize over small time horizons. Thus, we optimize our system for every prediction horizon T from Figure 2.11. Simply speaking, optimization involves finding the “best” result under given circumstances. In general, an optimization problem is defined mathematically as the following: Find

Algorithm 2 Model Predictive Control Algorithm

```
Initialize $T$, $T_a$, $T_f$, $n_{max}$, $\delta$, $i = 0$
while $iT_a < T_f$ do
    if $J > \delta$ then
        Find $\mathbf{f}_i$ that optimizes $J(\mathbf{f})$ over $iT_a$ to $(T + iT_a)$;
        Apply $\mathbf{f}_i$ for period $iT_a$ through $(T_a + iT_a)$;
    else
        Set $\mathbf{f}_i = \vec{0}$;
        Apply $\mathbf{f}_i$ for period $iT_a$ through $iT_a + T_a$;
    i=i++
```

$\mathbf{x}$, which minimizes or maximizes $f(\mathbf{x})$ subject to

$$d_i(\mathbf{x}) \leq a_i \qquad i = 1, 2, \ldots, m \tag{2.72}$$

and

$$e_i(\mathbf{x}) = b_i \qquad i = 1, 2, \ldots, p \tag{2.73}$$

where $\mathbf{x}$ is an n-dimensional vector called the "design vector"; $f(x)$ is the objective function; and $d_i(\mathbf{x})$ and $e_i(\mathbf{x})$ are inequality and equality constraints, respectively, that can be either linear or nonlinear. For such an optimization problem, there are many different ways this optimization may occur. For an objective function, there can exist many different local minima or maxima (i.e. a solution holds for a given parameter space), and sometimes there exists a global maximum or minimum (i.e. a solution that holds for the entire domain). Particularly when there exist constraints on an optimization problem, the likelihood of finding a local maximum/minimum exists.

A variety of individual optimization methods can be used for any given optimization problem. The decision of the optimization routine used is usually dependent on the type of system trying to be optimized. Some example questions that must be asked about an optimization problem are: Is the the realm of the cost functional where optimization is to take place noisy (e.g. high fluctuations of the system at hand), or does it demonstrate smooth behavior? Is there just one, or more than one objective that needs to be met? Is the problem constrained or unconstrained? Does the solution need to be global, or is local sufficient?

Usually, optimization methods fall into one of two categories: (a) they are gradient-based or (b) they are non gradient-based. A gradient-based optimization method uses the derivative in the optimization routine as it searches for the maximum or minimum. If the system being optimization is smooth, continuous, and single-objective, then a gradient-based method can be used for the optimization as it is usually the quickest mode of optimization. Additionally, gradient-based methods can be used for optimization problems with linear or nonlinear constraints.

For this research, the popular Broyden-Fletcher-Goldfard-Shanno (BFGS) quasi-Newton optimization method is used. The reason for specifically using a quasi-Newton scheme over a conjugate-gradient based scheme is due to previous work by a student that found the scheme to be quicker when employed with the CFD model, and often was able to find a better local minimum than the conjugate-gradient based scheme used [*Warnock* (2013a)].

The difficulty when employing an optimization scheme with a computational model is that the optimization scheme often does not have access to the underlying governing equations; instead, it is fed constraints and the cost-functional value. Thus, for a derivative-based optimization where either the gradient or Hessian needs to be calculated, it is difficult to do so without the underlying equations available. Instead, numerical estimates for calculating these derivatives are employed. Numerical estimation is usually an iterative process where guesses as to what the minimum or maximum might be, the cost function recalculated based on those guesses, and new guesses are made based on the analysis of the new cost functional. For gradient-based optimization, this analysis is with regards to determining a new direction to take for the next maximum/minimum "guess." This process occurs until the changes in the cost function are small enough for a given criterion. A numerical method for the BFGS quasi-Newton method is now described.

For a gradient-based algorithm the goal is to use gradient information, in conjunction with the function, and with respect to the unknown parameters $\mathbf{f}$. For gradient-based method, each iteration is dependent on a search direction p:

$$p_k = -B_k^{-1}\nabla J_k \tag{2.74}$$

where B_k is dependent on the algorithm used and J is the cost functional. For a Newton-type method, we define our search direction from a second-order Taylor series expansion about $J(\mathbf{f}_k + p)$ such that

$$J(\mathbf{f}_k + p) \approx J_k + p^T\nabla J_k + \frac{1}{2}p^T\nabla^2 J_k p. \tag{2.75}$$

This approach is such that it constructs a quadratic function around each $\mathbf{f}_k$ and minimizes this function. As a result, the optimal search direction is the p_k that minimizes the approximate function $m_k(p)$. If we take the derivative of $m_k(p)$ and set it to zero, our direction is

$$p_k = -(\nabla^2 J_k)^{-1}\nabla J_k \tag{2.76}$$

where $B_k = \nabla^2 J(\mathbf{f}_k)$ is the Hessian. While this method is second order accurate and converges quite quickly, calculating the Hessian at ever iteration is actually quite computationally costly. Thus, a method known as quasi-Newton has been developed that approximates the Hessian for B_k. For this method, we begin again with the Taylor series expansion such that:

$$\nabla J(\mathbf{f} + p) = \nabla J(\mathbf{f}) + \nabla^2 J(\mathbf{f})p + O(||p||) \tag{2.77}$$

where the error O is the same as the magnitude of p. Substituting $\mathbf{f}_k$ for $\mathbf{f}$ and $(\mathbf{f}_{k+1} - \mathbf{f}_k)$ for p yields:

$$\nabla J_{k+1} = \nabla J_k + \nabla^2 J_k(\mathbf{f}_{k+1} - \mathbf{f}_k) + O(||\mathbf{f}_{k+1} - \mathbf{f}||). \tag{2.78}$$

From this derivation, we can see that if $\mathbf{f}_{k+1}$ and $\mathbf{f}_k$ are in the region near the actual solution $\mathbf{f}$, the error term will be far less significant than the second order term of the expansion, and therefore we can deduce:

$$\nabla^2 J_k(\mathbf{f}_{k+1} - \mathbf{f}_k) \approx \nabla J_{k+1} - \nabla J_k. \tag{2.79}$$

We can now substitute our B back into the above equation such that

$$B_{k+1}(\mathbf{f}_{k+1} - \mathbf{f}_k) = \nabla J_{k+1} - \nabla J_k. \tag{2.80}$$

This method is known as the secant method and provides us with an method to iteratively construct the approximate Hessian. There exist many different ways to approximate the Hessian, which for the BFGS method is the following:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{2.81}$$

where $s_k = \mathbf{f}_{k+1} - \mathbf{k}$ and $y_k = \nabla J_{k+1} - \nabla J$. Equation (2.81) can be substituted into equation (2.74). Finally, a quadratic model of the objective function at the current value of p is

$$m_k(p) = J_k + \nabla J_k^T p + \frac{1}{2} p^T B_k p. \tag{2.82}$$

This equation is minimized by equation (2.74). The algorithm for the BFGS quasi-Newton method is seen in Algorithm 3.

Algorithm 3 Quasi-Newton BFGS algorithm

```
Define starting point $\mathbf{f_0}$, convergence tolerance $\epsilon$, inverse Hessian approximation, $H_0$;
while $||\nabla J_k|| > \epsilon$ do
    Compute search direction using equation (2.74);
    Carry out a line search where $\mathbf{f}_{k+1} = \mathbf{f}_k + a_k p_k$ for a direction step size $a_k$;
    Set $s_k = \mathbf{f}_{k_1} - \mathbf{f}_k$ and $y_k = \nabla J_{k+1} - \nabla J_k$;
    Determine the new approximate Hessian using equation (2.81);
    $k \leftarrow k + 1$;
```

2.3.4 Software Deployed for Optimal Control

Just like the employment of software for the CFD model described in Section 2.2.7, the decision on how to implement the optimization scheme computationally is nontrivial. The most straightforward means of implementing an optimization scheme is by coding it directly into the CFD model. However, this strategy can be tedious if a variety of optimization algorithms are to be tried. Similar to OpenFOAM, in the 1990s researchers from the Lawrence Berkeley National Laboratory and Sandia National Laboratory developed a C++ library of classes for nonlinear optimization named OPT++ [*Meza et al.* (2007)]. These libraries were particularly developed for simulation-based nonlinear optimization.

2.3.4.1 DAKOTA

Almost simultaneously as OPT++ was developed, the Design and Analysis toolKit for Optimization and Terascale Applications (DAKOTA) Design Analysis Kit for Optimization and Terascale Applications software toolkit was also developed by Sandia National Laboratories [*Adams et al.* (2009)]. DAKOTA contains algorithms for both gradient and non-gradient based optimization methods. Furthermore, DAKOTA has the capacity for parameter estimation with nonlinear least squares methods, uncertainty quantification with sampling, and sensitivity analysis. These diverse capabilities of DAKOTA makes it especially appealing for use in a wide array of applications.

2.3.4.2 OPT++ Package

While there are many different optimization packages and algorithms available within DAKOTA, for this research, we use the quasi-Newton solver library, OPT++, specifically developed for nonlinear optimization. The OPT++ library contains both conjugate-gradient based optimization routines as well as quasi-Newton based optimization routines. As stated in Section 2.3.3, the optimization routine specifically used in this work is based on the popular quasi-Newton optimization method.

The decision to chose one optimization routine over another depends on the type of problem being solved. In work carried out by a previous student, the CONMIN (CONstrained MINimization) library, which is a conjugate-gradient based optimization algorithm developed by NASA [*Vanderplaats* (1973)], was used for optimization. However, in the most recent work by this research group ,the CONMIN routine was compared to the OPT++ optimization algorithm; for the boundary control problem, OPT++ was shown to perform more accurately and reliably. Therefore, the OPT++ optimization library is the optimization algorithm that will be employed throughout this work.

2.3.4.3 Interfacing with OpenFOAM

One of the most appealing components of the DAKOTA software toolkit is its ability to be interfaced with external models quite easily. For this work, optimization of the numerical fluid model developed with OpenFOAM is achieved by 'loosely coupling' the model with the DAKOTA software toolkit. That is, the information needed for the optimization problem is passed to DAKOTA's optimizer from OpenFOAM, which then receives new parameter guesses from DAKOTA at each iteration. The communication and preparation of necessary files between OpenFOAM and DAKOTA is achieved through a combination of Python and Bash scripts. These scripts were adapted from research carried out by a previous student [*Warnock* (2013a)] and the interfacing routine can be seen in Figure 2.12. Additionally, the OpenFOAM-DAKOTA workflow can be seen in Figure 2.13.

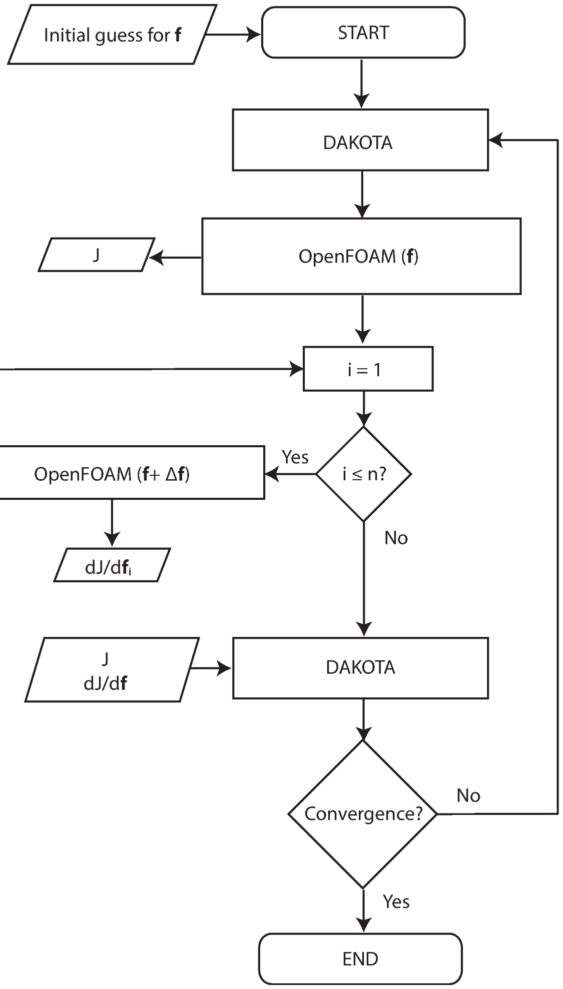


Figure 2.12: OpenFOAM-DAKOTA interfacing routine.

2.4 Results

The overall objective for this research when investigating the CFD flow-control model is what sort of influence coupling it with the evacuation model might have on the overall behavior of the control actions. Thus, the main results for this section were concerning the placement of protected points, and how that might change the control action.

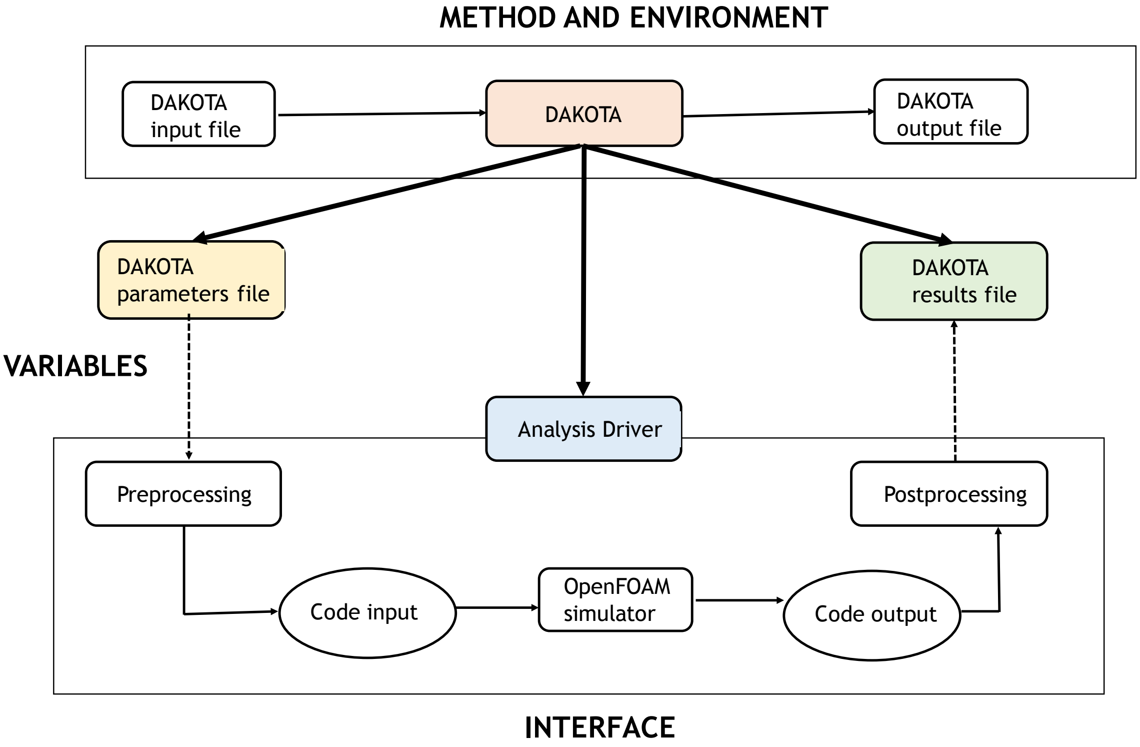


Figure 2.13: Workflow for DAKOTA coupled with OpenFOAM.

It should be noted in the figures displayed below that the yellow points are the protected points; the ports are distinguished by the grey sets of points along the boundary, the arrows at the port represent the direction and magnitude of the control action being applied, and the value C is the concentration in units. The label of port locations for the six port scenarios can be seen in Figure 2.14; the label of port locations for the two port scenarios can be seen in Figure 2.15

2.4.1 Case 1. General Flow Control

The first case is the very basic case in which there are two types of protected point arrangements: (i) the protected points are in the middle of the domain and (ii) the protected points are along the boundary of the domain. Simulation snapshots showing the results for these two scenarios can be seen in Figures 2.16 and 2.18, respectively.

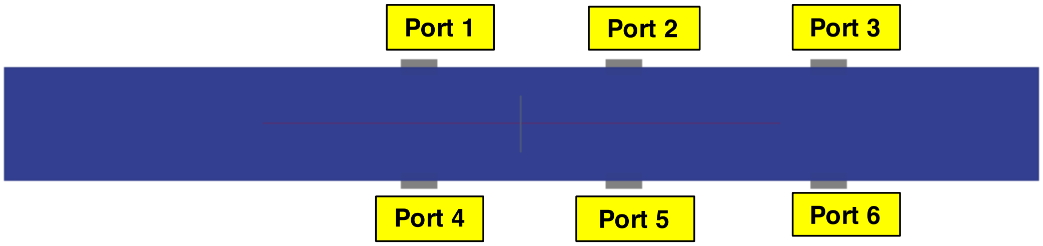


Figure 2.14: Location and labels of ports for six port domain.

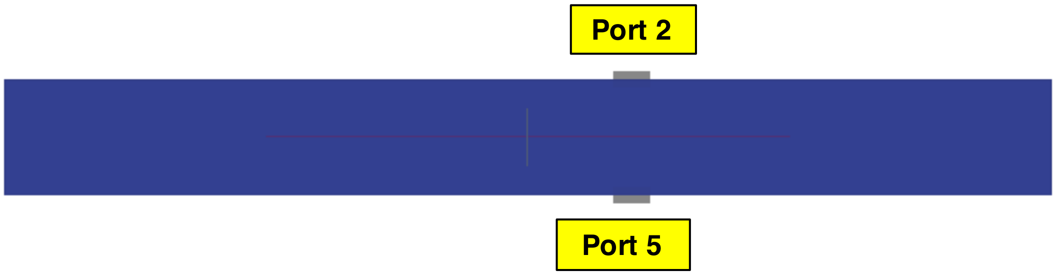


Figure 2.15: Location and labels of ports for two port domain.

The control action is actually quite obvious for both of these scenarios. It follows almost effortlessly that for the protected points in the middle, the control scheme will be to draw the contaminant to the side, and out of the domain if possible. Likewise, for the protected points along the boundary, the control action would certainly be to push the contaminant to the middle of the domain.

These two examples demonstrate a very basic control action for our domain. As will be seen in Chapter IV, these two scenarios actually match up quite well with the evacuation behavior.

2.4.2 Case 2. Location of Boundary Points

Because the first case showed almost obvious control actions, it was desired to change the control problem such that the control action might not be as obvious. Therefore, simulations were carried out with a domain where the protected points

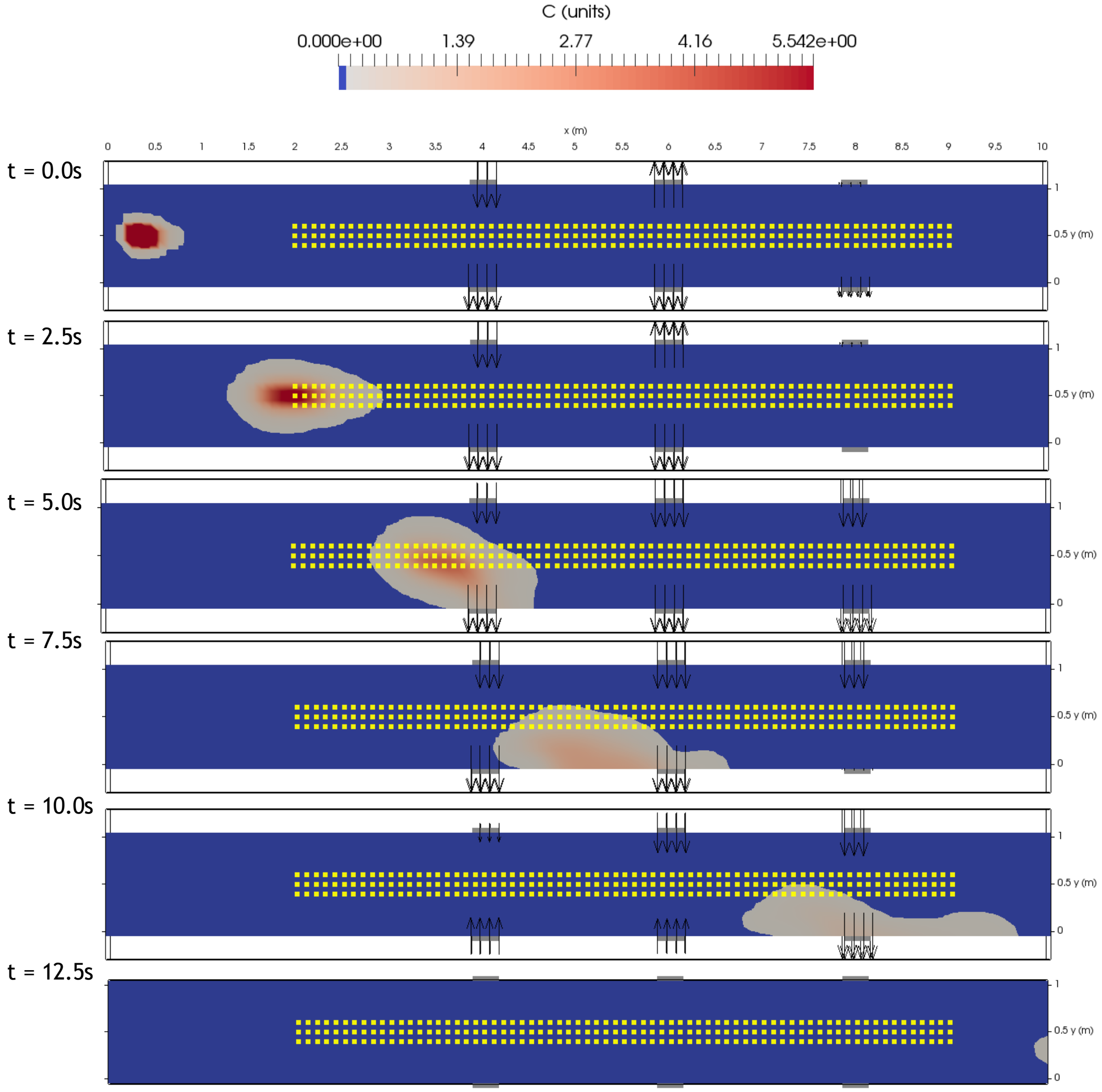


Figure 2.16: CFD flow-control for protected points in the middle of the domain.

were randomly distributed. The result from this simulation can be seen in Figure 2.20. As can be seen, for this example, the control action is quite similar to when all of the protected points were in the middle of the domain. The contaminant is pushed to one side of the domain, and drawn out as quickly as possible. We continue changing the protected points in various patterns in the domain as can be seen in Figures 2.22, 2.24, and 2.26.

Figure 2.26 is especially interesting as there is a clear opening for two of the ports

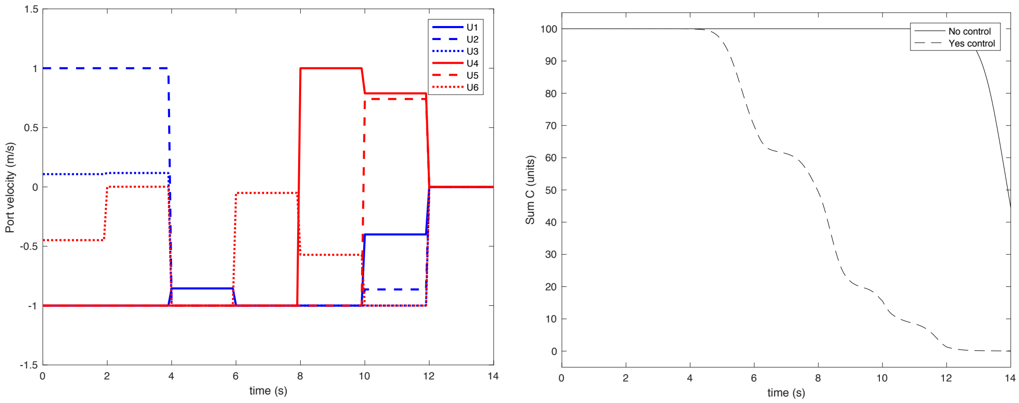


Figure 2.17: Corresponding port velocities, and comparison of total concentration at protected points with and without control for protected points in the middle of the domain. Positive velocity is taken as velocity normal to port.

to draw out the contaminant without affecting any of the protected points; however, the control action does not behave in this manner. The most likely explanation is that the control actions are sensitive to other parameters, such as the prediction horizon, or potentially the number of boundary actuators.

2.4.3 Case 3. Influence of Number of Boundary Ports

To begin testing other parameters that might be influencing the control action, some similar simulations were carried out with only two ports rather than six. The patterns for having the protected points in the middle first then on the sides (Figure 2.28), and vice versa (Figure 2.30), were also used to compare with Figures 2.22 and 2.24 for six ports. When comparing the simulations of two ports versus six ports, the control behavior is actually quite different for the two protected point scenarios. What is most interesting is that in the two port scenario, nothing is really done by the ports to control the upstream behavior of the fluid to move the contaminant away from the protected points.

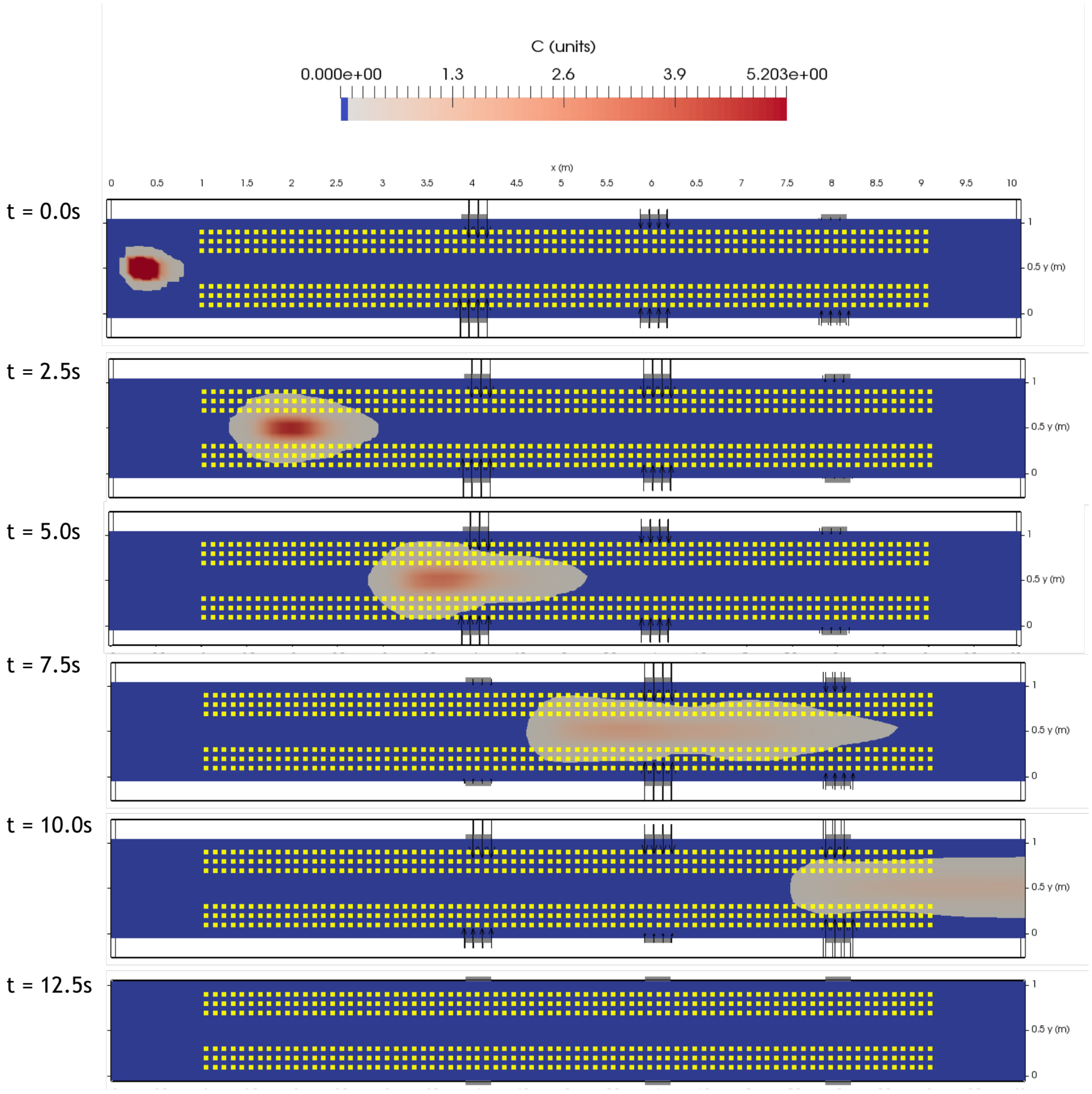


Figure 2.18: CFD flow-control for protected points in the sides of the domain.

2.5 Discussion

Many different runs were carried out with the CFD flow-control model to see what sort of unpredictable behavior might occur with the fluid for a variety of patterns of protected points. For most of these scenarios, the control action was quite predictable. Additionally, it is unclear what parameters may be be influencing the control action over others. For this reason, a sweep of parameters can possibly be carried out to

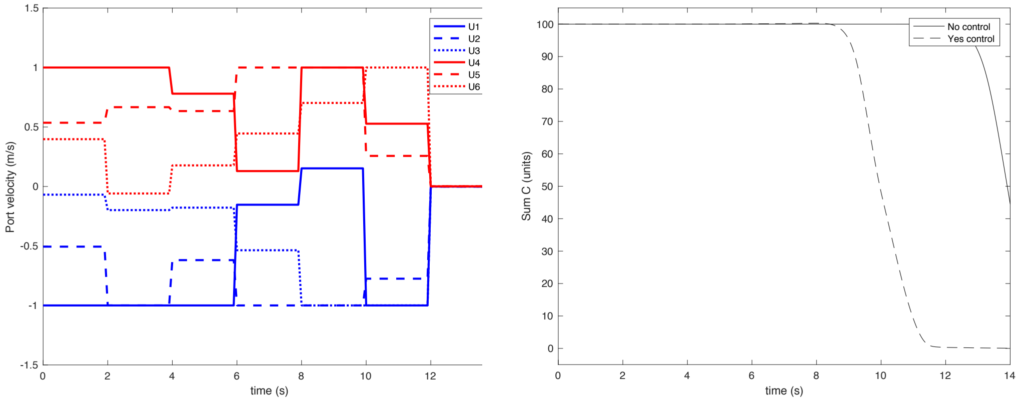


Figure 2.19: Corresponding port velocities, and comparison of total concentration at protected points with and without control for protected points on the sides of the domain. Positive velocity is taken as velocity normal to port.

better discern, included the prediction horizon, the convergence criteria used, and the velocities of the ports.

It must also be noted that as of now, there is only one parameter used within the cost functional: the cell concentration. It would be interesting to house more characteristics within the cost function. In particular, a cost to creating more turbulence could be added to the cost functional.

As will be shown in the following chapters, when applied to the physical prototype, and coupled with the evacuation ABM, some interesting fluid dynamics behavior arises.

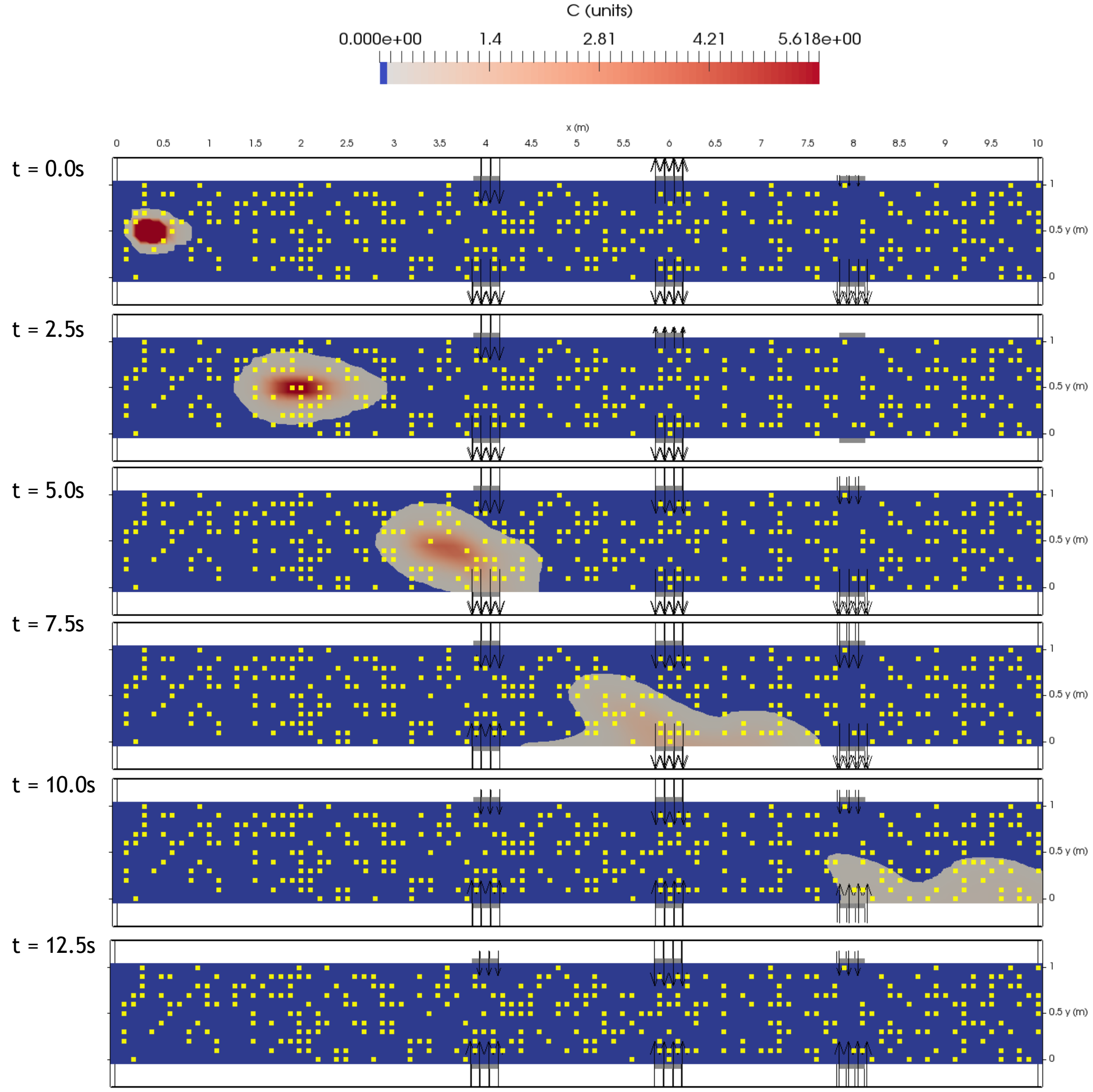


Figure 2.20: CFD flow-control for protected points randomly distributed in domain.

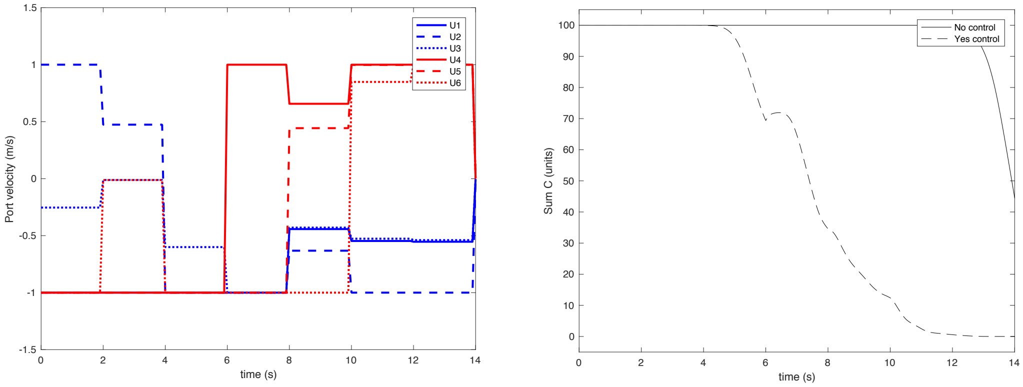


Figure 2.21: Corresponding port velocities, and comparison of total concentration at protected points with and without control for protected points in the sides of the domain. Positive velocity is taken as velocity normal to port.

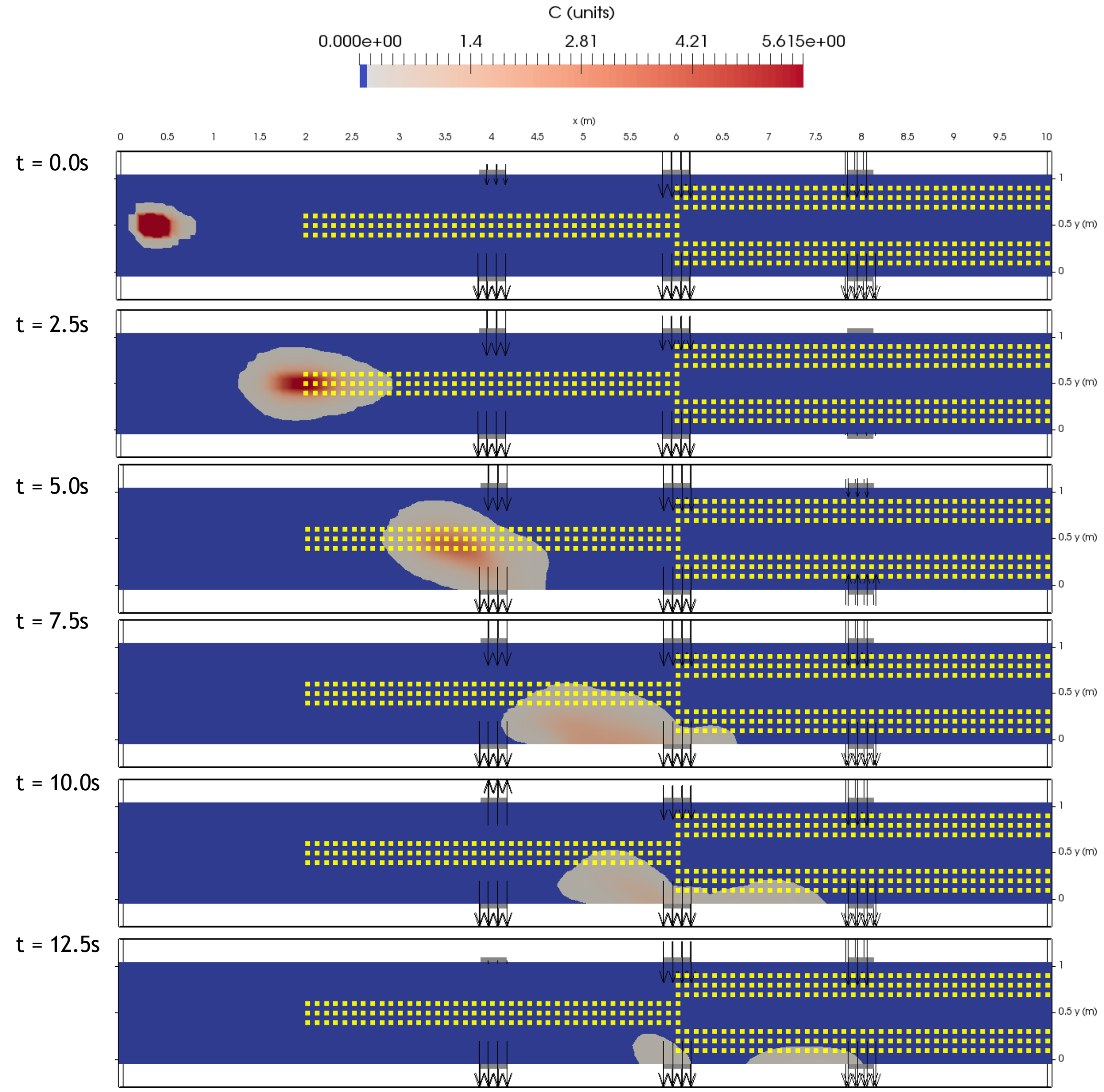


Figure 2.22: CFD flow-control for protected points first in the middle of the domain, followed by being on the side of the domain.

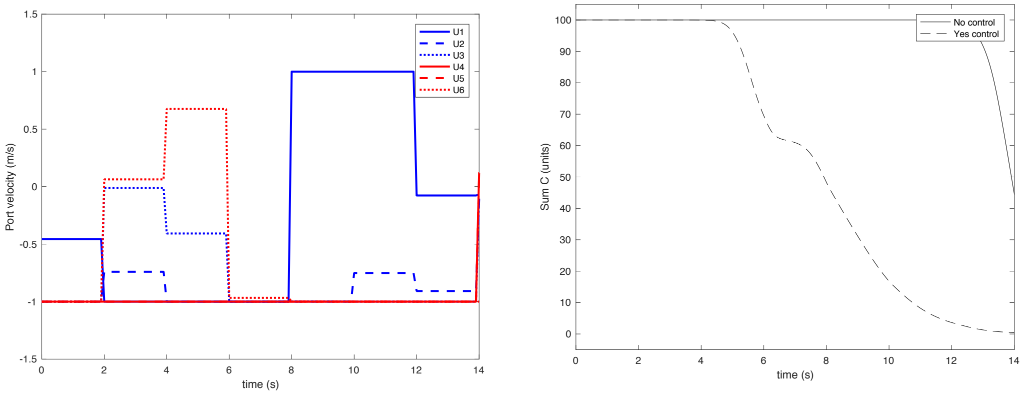


Figure 2.23: Corresponding port velocities, and comparison of total concentration at protected points with and without control for protected points first in the middle of the domain, followed by being on the side of the domain. Positive velocity is taken as velocity normal to port.

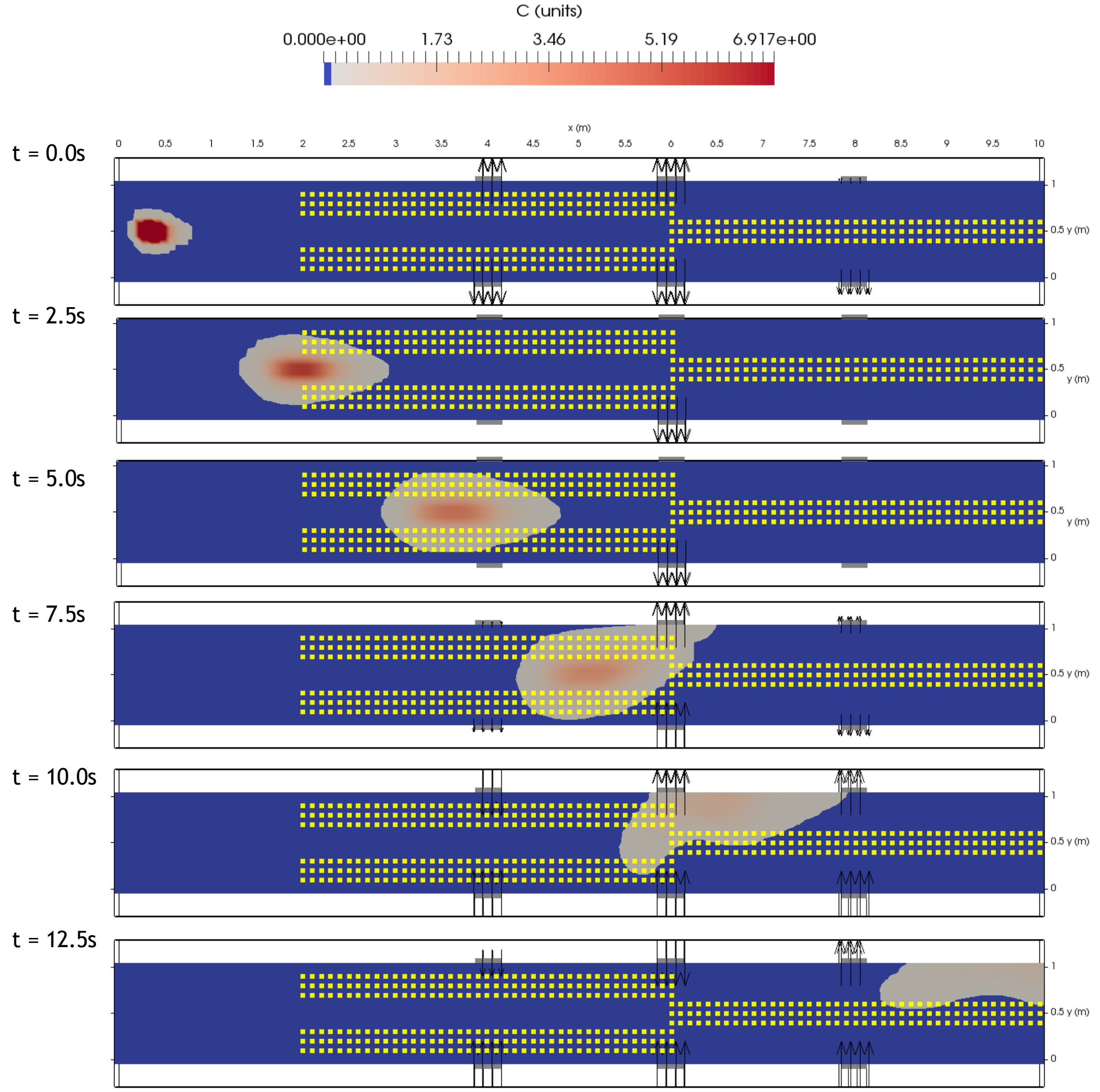


Figure 2.24: CFD flow-control for protected points first on the side of the domain, followed by being in the middle of the domain.

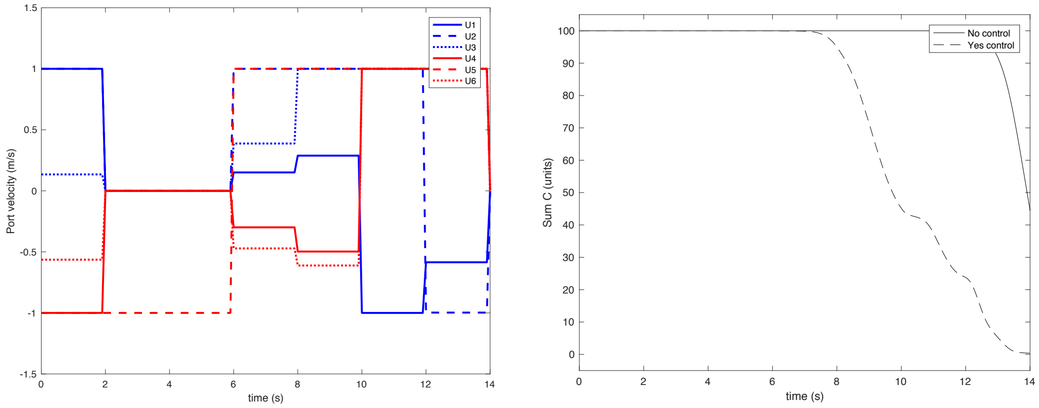


Figure 2.25: Corresponding port velocities, and comparison of total concentration at protected points with and without control for protected points first on the side of the domain, followed by being in the middle of the domain. Positive velocity is taken as velocity normal to port.

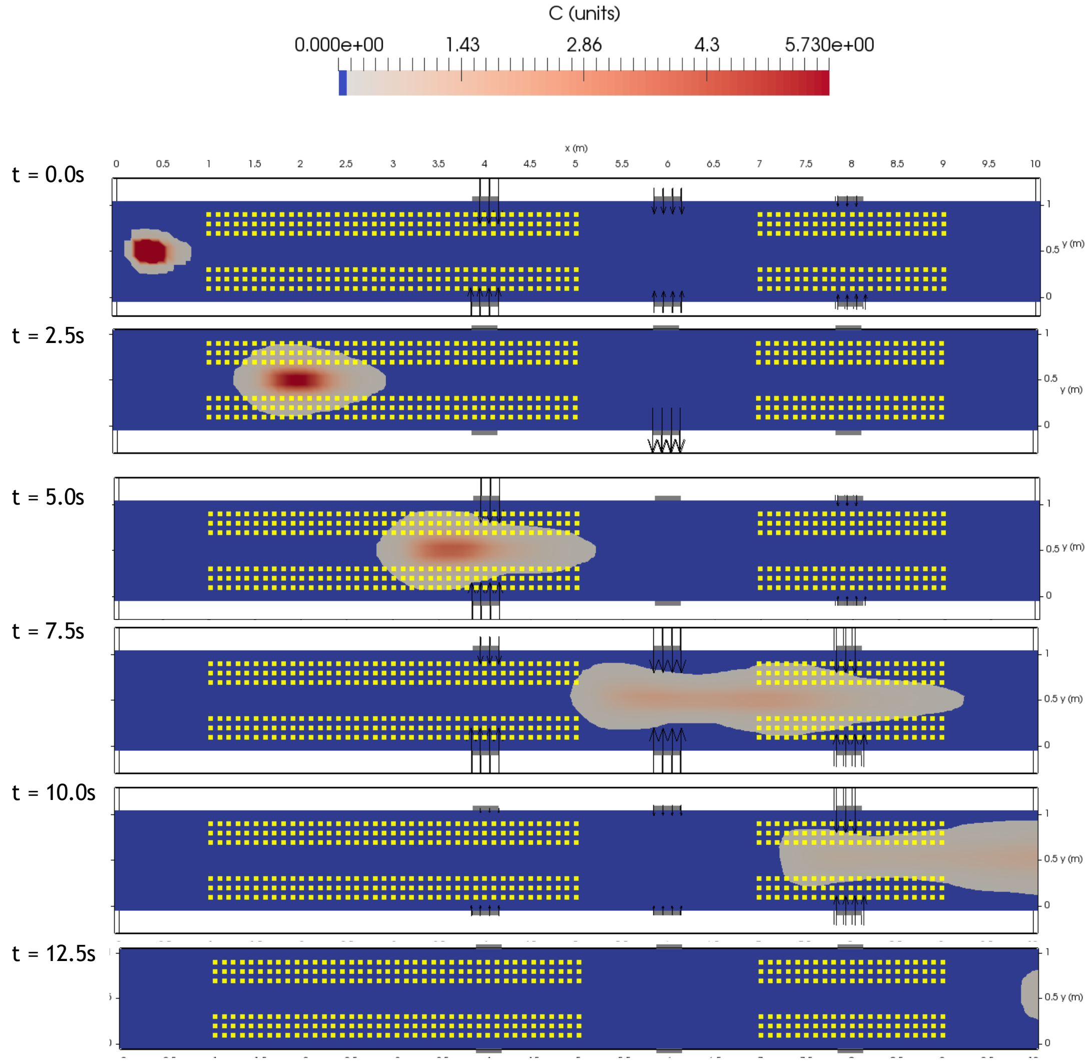


Figure 2.26: CFD flow-control for protected points distributed in groups throughout the domain.

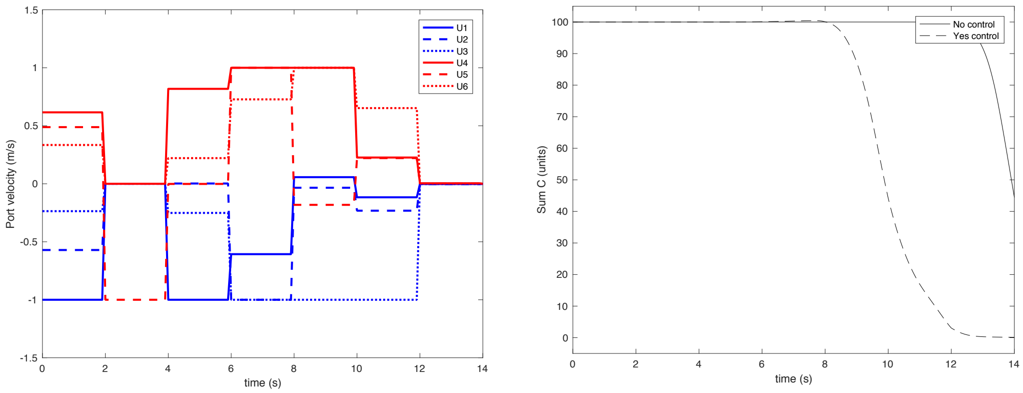


Figure 2.27: Corresponding port velocities, and comparison of total concentration at protected points with and without control for protected points distributed in groups throughout the domain. Positive velocity is taken as velocity normal to port.

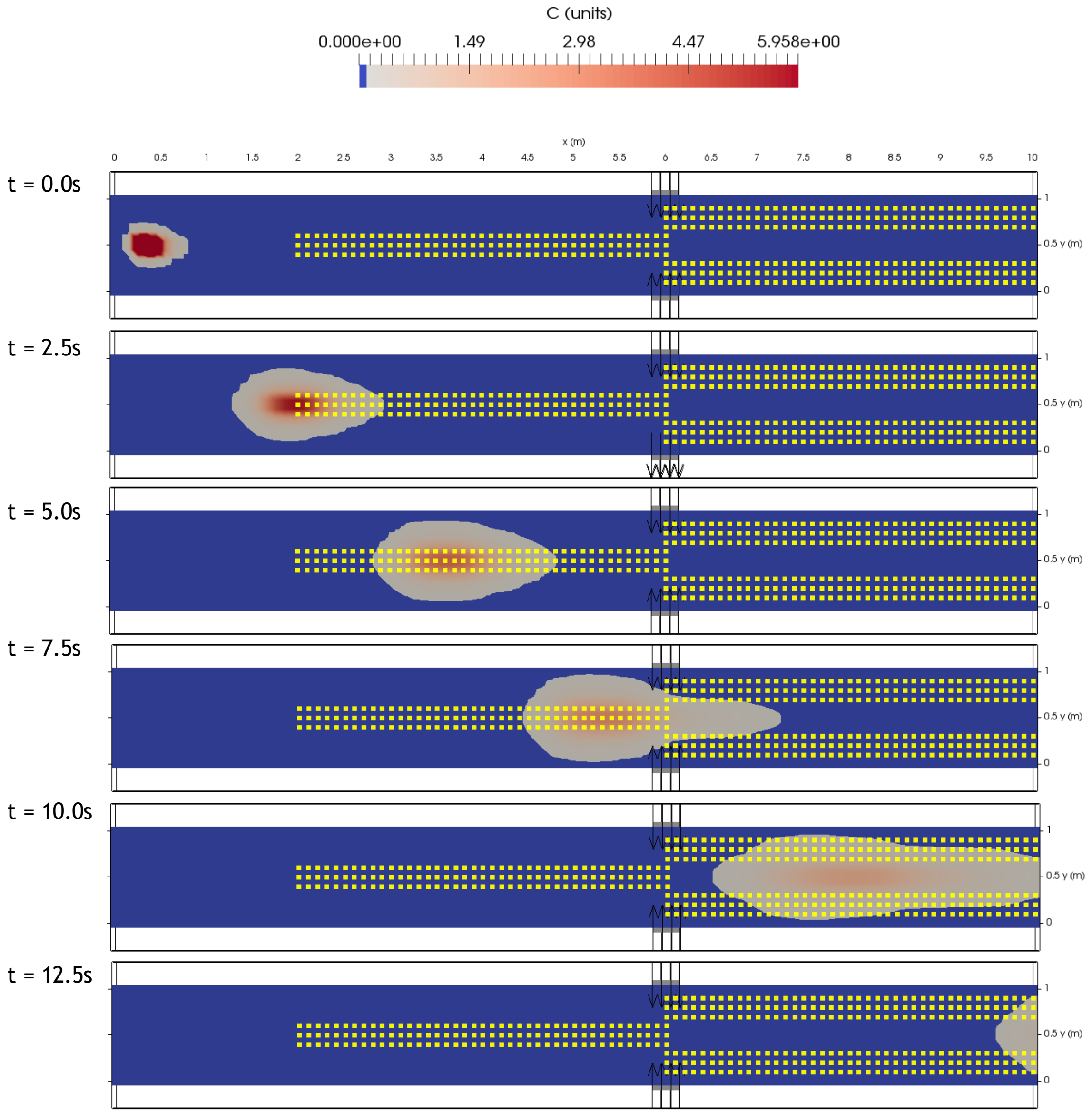


Figure 2.28: CFD flow-control for two ports with protected points first in the middle of the domain, followed by being on the side of the domain.

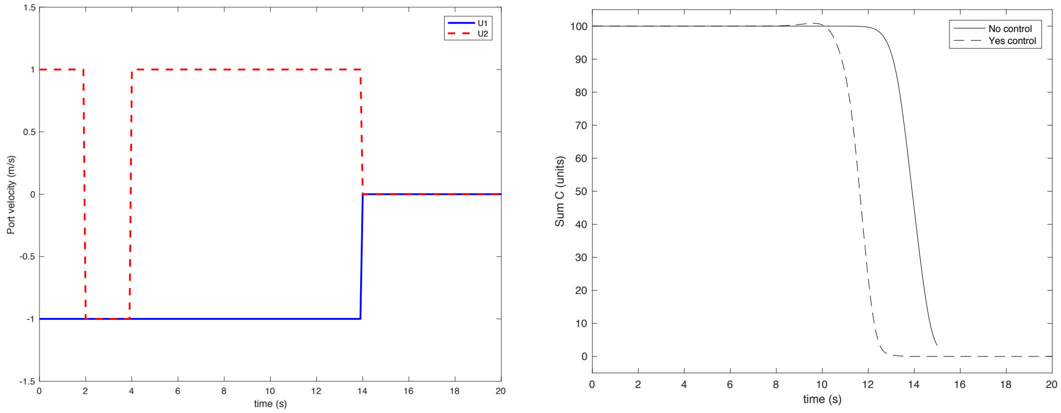


Figure 2.29: Corresponding port velocities, and comparison of total concentration at protected points with and without control for two ports with protected points first in the middle of the domain, followed by being on the side of the domain. Positive velocity is taken as velocity normal to port.

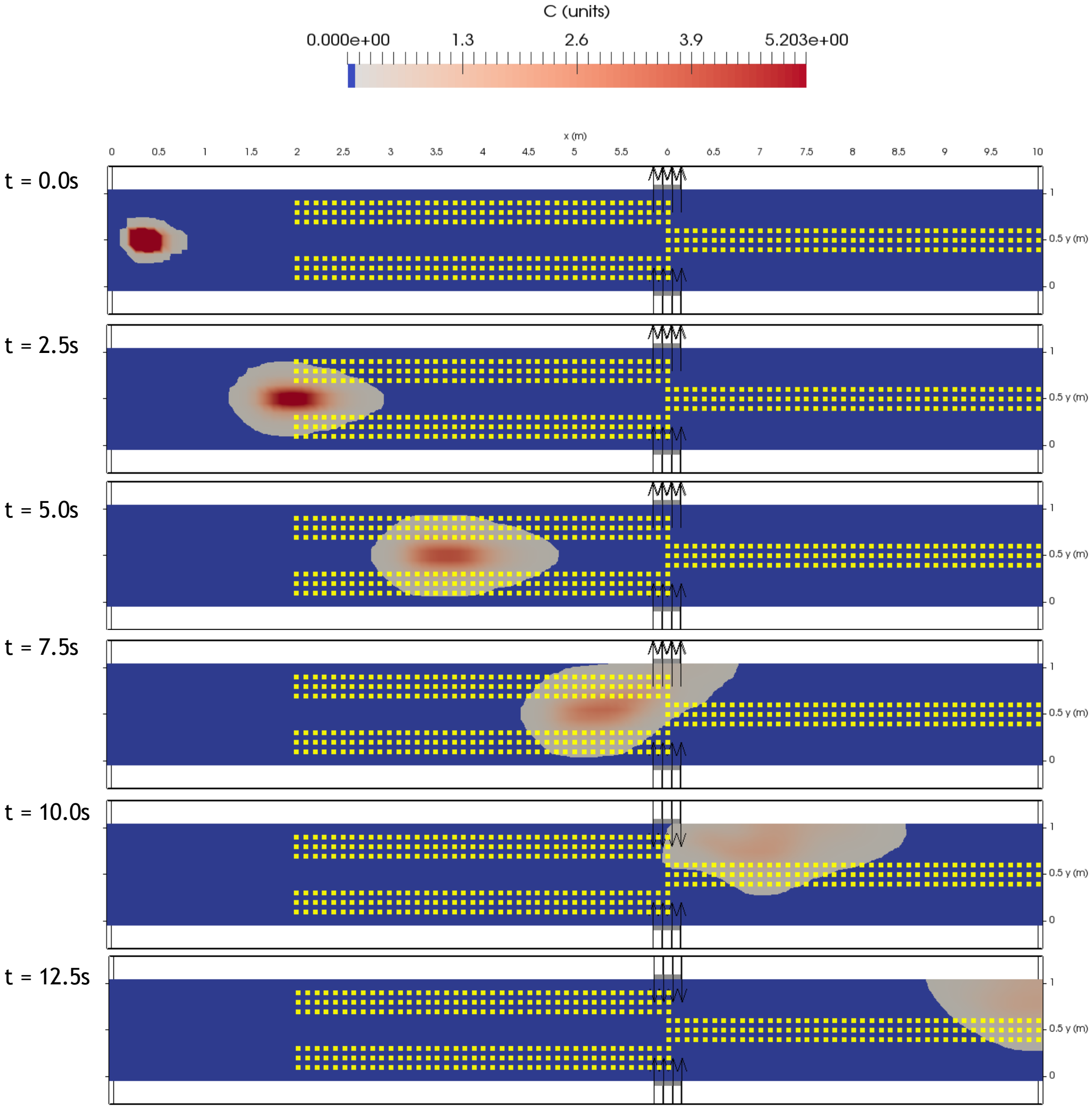


Figure 2.30: CFD flow-control for two ports with protected points first on the side of the domain, followed by being in the middle of the domain.

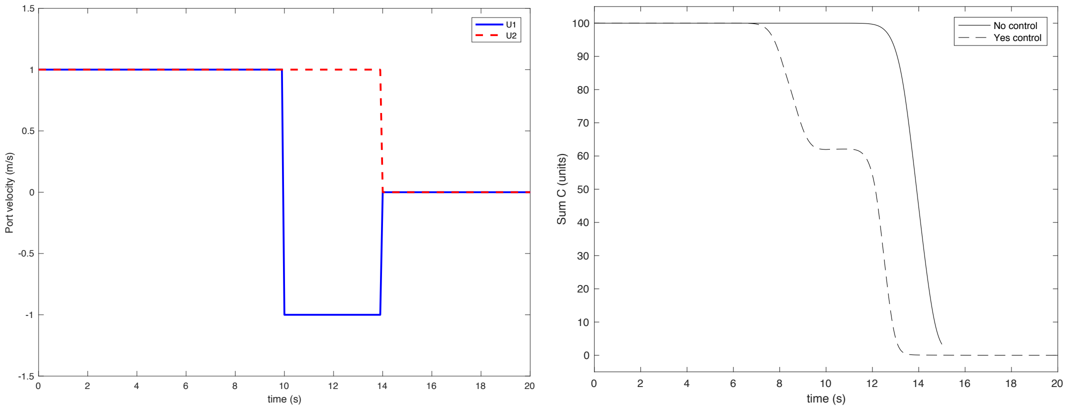


Figure 2.31: Corresponding port velocities, and comparison of total concentration at protected points with and without control for two ports with protected points first on the side of the domain, followed by being in the middle of the domain. Positive velocity is taken as velocity normal to port.

CHAPTER III

Physical Model

In order to test the feasibility of implementing the computational flow control model from Chapter II, we have built a physical prototype. There are two parts to the design of this physical prototype: (i) a blower wind tunnel (that provides uniform air flow into a "test section") and (ii) the control system (i.e. sensors, actuators, and controller) that detect and mitigate an injected "contaminant." The prototype has been built and is currently housed in the Department of Civil and Environmental Engineering's Structures Laboratory.

This chapter details the specifics behind the design of the wind tunnel (in Section 3.3) and the control system (in Section 3.4). Results demonstrating real-time control using this prototype are presented in Section 3.5 with the corresponding experimental limitations.

3.1 Motivation

For this research, our experimental design is really just a basic proof-of-concept: can we detect and mitigate a contaminant in real-time. A physical prototype has been designed, built, and tested by another student in this research group [*Wang et al.* (2013b)]. This prototype can be seen in Figure 3.1. For this prototype, water is used as the bulk fluid. The bulk fluid is supplied to the prototype via a hydraulic pump at the inlet to a 4-inch diameter acrylic pipe that is approximately 3-feet in length. Food dye is used as the 'contaminant' and is injected into the ambient flow just after the bulk fluid inlet. Together, the contaminant and the bulk fluid pass through a high density sponge used to improve the uniformity of the flow. A 'webcam' video camera is used to capture the colored contaminant in real-time and send its images to computer. The images from the camera read in the concentration of the contaminant and estimate its velocity. The controller then sends information to pressure-controlled actuators located on the prototype's boundary. The prototype uses a dSPACE real-time controller coupled with a Simulink C-code autogenerator when communicating with the actuators.

While this prototype has been successful in implementing real-time control with the hazardous contaminant, it is limited in that it currently only has one actuator port and is unable to be extended to include another port which would allow for *feedback* control. Furthermore, this prototype utilizes a simplified fluid dynamics model that is based on dynamical systems theory, and is thus not applicable for turbulent flows. Therefore, the physical prototype discussed throughout this dissertation builds upon this prototype by implementing a more comprehensive fluid dynamics model.

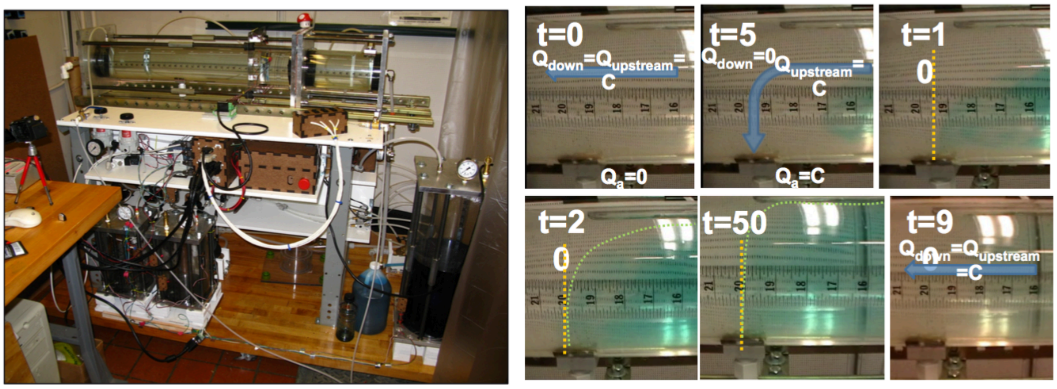


Figure 3.1: Water-based dynamical system based prototype developed by *Wang et al.* (2013b).

3.2 Background

Experimental methods for fluid mechanics usually fall in to one of two categories: (a) measurement or (b) visualization. As with most experimental methods, the desire to measure specific parameters for a given fluid flow drive the underlying experimental development. However, it must also be noted there is a great need for understanding the qualitative behavior behind some fluid dynamics phenomena, which is why fluid visualization is quite popular within the fluid dynamics community (although, it must also be stated that some visualization techniques are also used for measurement applications).

Studying a fluid phenomena experimentally allows researchers to develop knowledge that is unable to be gained numerically or theoretically. *Similitude* is one of the main concepts behind analyzing a fluid flow experimentally. In essence, similitude is the ability to gain insight into one type of flow by studying another type of flow that is *similar* to it. Thus, if a certain fluid phenomenon is difficult to study in its natural environment, a physical model can be developed and experimentally analyzed such that it has fluid behavior *similar* to the other. A “scaling” of a fluid system is the principle means of similitude in experimental fluid mechanics. For example, building a full-scale car in order to study air flow around it could be costly, whereas an alternative is to develop a scaled-down less expensive version of the vehicle to analyze; similitude practices are used to accomplish this “scaling-down” such that analysis can be carried out on the scaled-down model and subsequent inference can be made about the full-scale system. While there exists a multitude of methods developed over the decades to carry out experiments based on the idea of similitude, one of the most widespread physical methods to develop and carry out fluid experiments is through the use of a wind tunnel.

Because an object moving through a flow field has the same experience as an object with a flow field being pushed over it at the same velocity, the ability to make a scaled-down physical model to simulate an object moving through air is the basis behind wind tunnel designs. For this research, a wind tunnel is the starting point of the experimental model that is developed. Standard wind tunnel designs are built such that uniform flow is provided at the entrance to a test section with air as the bulk fluid.

Much like the discussion of flow control, the development of the standard design of wind tunnels coincides with that of aircraft design and the development of modern aerodynamics. As 19^{th} engineers and scientists were developing the ability to harness air for flight, the first wind tunnel was built more than 30 years before the Wright Brothers made their first controlled powered airplane flight in 1903. The underlying objective of the earliest wind tunnels are they same as they are today: to recreate the often unpredictable winds of nature with a steady, controllable flow of air using a machine[1].

The University of Michigan houses actually quite a few wind tunnels, the most famous being a closed-loop low-speed wind tunnel built in collaboration with the U.S. Air Force in 1956, which has a 5 ft × 7 ft ($\approx$ 1.5 m × 2 m|) cross-section and is large enough for a person to stand upright in. The School of Public Health built a draw-down wind tunnel with a 1 ft × 1 ft ($\approx$ 0.3 m × 0.3 m|) cross-section and was used to study human breathing of smoke.

The University of Michigan also has many water flumes, which are built and used with the same rationale as wind tunnels. One of the most exciting experimental facilities at the university with water as the fluid medium is the Marine Hydrodynamics Laboratory's physical model basin, which was built in 1905, reaches 360 ft (109.7 m) in length, is 22 ft (6.7 m) wide, and 10 ft ($\approx$ 3 m) deep[2]. Instead of pushing a fluid

[1] Actually, the first experiments carried out using a "wind tunnel" were often in open environments such as the openings of blowing caves.

[2] More about this tow tank can be read here: `http://mhl.engin.umich.edu/facilities/basin/`.

over a body, a tow tank pulls a body through a stationary fluid[3].

However, as has been continually reiterated throughout this writing, most of these experimental prototypes are used to study fluid flow over some sort of scaled-down body (primarily aircraft or vehicles). Instead, there existed a need to build a prototype with more flexible applications. Thus, as a motivation for the wind tunnel that was built for this research, many different wind tunnels that have been used for environmental applications, particularly to recreate atmospheric boundary layers, were explored as they were more likely to have a longer test section as the expanse of the systems studied is far larger than vehicle applications, and oftentimes understanding boundary layer development is a critical component of the corresponding research questions. Many atmospheric boundary layer wind tunnels were used as a motivation for the design of this prototype [*Farell and Iyengar* (1999); *Uehara et al.* (2000); *Chen et al.* (2009)].

3.3 Wind Tunnel Design

Over the years, a standard design has been developed for wind tunnels. The objective of wind tunnel design is such that at the entrance to a test-section, the flow of the air is as uniform (i.e. velocity is the same at every point in the cross-section) and laminar as possible (i.e. there should be no separation of flow). Most of the present-day design guidelines for wind tunnels are the result of decades of iterations and refinement of various wind tunnels used throughout research and industry. Different design criteria exist for different types of fluid flow (e.g. supersonic flow versus subsonic flow). However, for the most part, there are certain components that are a part of almost all wind tunnels.

[3] ...which is interestingly an equivalence observation first made by Leonardo Da Vinci around the turn of the 16^{th} century, but not pursued until multiple centuries later.

3.3.1 Experimental Fluid Flow Objective

Before the details on the actual design of the wind tunnel are presented, the conversation must first be brought back to the concept of *similitude* introduced earlier. The overall reason for developing this physical model is such that inferences can be made on the fluid flow present in our emergency scenarios described in Chapter I based on a *similar* version of fluid flow that is physically modeled. Most similitude practices involve what is known as *dimensional analysis*, in which the physical quantities of two different fluid flows that are *similar* are characterized in terms of the same fundamental parameters (e.g. length, velocity) for inferences to be made about the relations between them. To do this, the most important parameters of the flow being studied are identified, the remaining parameters are amalgamated into dimensionless numbers, and these dimensionless numbers are held to be the same for both the fluid flow being studied and its scaled version. Many famous dimensionless numbers are used across a variety of flows. For example, the Froude number:

$$Fr = \frac{U}{\sqrt{gD}} \tag{3.1}$$

is the ratio of a fluid's inertial forces to gravitational forces, where U is the fluid's velocity, g is gravity, and D is the characteristic length (or hydraulic depth in hydraulic applications), and is used in the field of hydraulics. The Froude number characterizes open-channel flow as either critical ($Fr = 1$), supercritical ($Fr > 1$), or subcritical ($Fr < 1$). Thus as an example of dimensional analysis, to study a scaled-down version of an open-channel, the physical model must have the same Fr number as the actual system being studied.

The ability to determine "similar" flow that is sufficient for this coupled analysis is nontrivial. For types of fluid flow that have been systematically studied over the years (e.g. flow over a vehicle), standard procedures have been developed and refined

Table 3.1: Geometries of two different example public spaces.

	Detroit-Windsor Tunnel	McNamara Airport Terminal
Cross-section	7 m × 4 m (22 ft × 13 ft)	26.5 m × 25 m (87 ft × 82 ft)
Length	1, 573 m (5, 160 ft)	1, 573 m (4, 900 ft)
Width-to-length ratio	0.013	0.056

over the years for what exactly this "similar" flow should be. Otherwise, designing a physical model with "similar" flow is the art behind experimental fluid mechanics.

For this physical model, the objective was to scale down some public space characterized by a long conduit such that the fluid flow in the physical model was somehow "similar" to the fluid flow in this public space. To do this, first the geometries of typical public spaces being studied were compared. For example, in Table 3.1, geometries are presented for a vehicle transport tunnel and for a passenger airport terminal. What can be gathered from the dimensions presented in this table is that our system being studied is significantly longer than it is wide (or tall). Thus, our physical model must also be the same.

Ideally, the experimental setup would be able to match some sort of non-dimensional number also represented by these public spaces. If we take the velocity that can sometimes be felt in a subway passenger tunnel to be 10 m/s [*Yang et al.* (2007)] and our characteristic length to be approximately the same height as the Detroit-Windsor car tunnel (so $l = 7\,\mathrm{m}$, our Reynold's number at 15 °C with air as the ambient fluid such that our density is $\rho = 1.225\,\mathrm{kg/m^3}$, and dynamic viscosity is $\mu = 1.98 \times 10^{-5}\,\mathrm{N \cdot s/m^2}$ is:

$$Re = \frac{(1.225\,\mathrm{kg/m^3})(10\,\mathrm{m/s})(7\,\mathrm{m})}{1.98 \times 10^{-5}\,\mathrm{N \cdot s/m^2}} = \mathbf{4.3 \times 10^6}. \tag{3.2}$$

For our experiment to match this Reynolds number, then we will need to determine the velocity for a given characteristic length. As will be discussed in Section 3.3.4, our height/width for the test section will be 0.6 m for both. Thus for a $Re = 4.3 \times 10^6$

our velocity will need to be:

$$U = \frac{Re\mu}{\rho l} = \frac{(4.3 \times 10^6)(1.98 \times 10^{-5}\,\mathrm{N \cdot s/m^2})}{(1.225\,\mathrm{kg/m^3})(0.6\,\mathrm{m})} = 116\,\mathrm{m/s}. \tag{3.3}$$

Since this experiment for this dissertation was to show feasibility of real-time control, rather than matching exactly to an actual physical scenario, having the prototype have a velocity of 116 m/s seemed too high. Thus, matching the Reynolds number was not necessarily the goal when creating a “similar” flow at the experimental level. Instead, having a test section that was significantly longer than it was wide or high was the main goal, and a velocity inside the test section match that of the maximum velocity one might feel in an actual public space (e.g. 10 m/s) was followed instead.

3.3.2 Design Constraints

For the wind tunnel developed for this research, three different guides were followed [*Mehta and Bradshaw* (1979); *Mehta* (1979); *Bradshaw and Pankhurst* (1964)]. Based on these guides, design decisions were made with an effort to minimize fluid separation and maximum fluid uniformity. However, there existed some obvious added constraints for this wind tunnel–mainly cost, space, construction capabilities, and timeline–that also influenced design decisions.

3.3.2.1 Space

The wind tunnel was built in the wave basin within the construction laboratory of the Civil and Environmental Engineering Department. The total dimensions of the space that were available for this physical model were 20 ft × 45 ft. Fortunately, this space allowed for a long test section to be integrated into the design that will eventually allow for feedback flow control (more than one iteration of the sensing/control). The disadvantage to this space was the significant amount of dust produced by the

Table 3.2: Cost breakdown of physical model.

Item	**Approximate Cost**
Test Section Glass & Acrylic	$\$4,000$
Wood and Miscellaneous Construction Materials	$\$2,000$
Centrifugal Fan	$\$3,000$
Aluminum Honeycomb	$\$1,000$
Smoke Generator for Visualization and for 'Contaminant'	$\$3,000$
Wire Cloth for Settling Chamber and Wide Angle Diffuser	$\$3,000$
National Instruments Real-Time Controller	$\$8,000$
Compressed-Air Components for Control System	$\$4,500$
Ethernet CCD Camera and Lens	$\$1,500$

structures laboratory that often interfered with visualization efforts.

3.3.2.2 Cost

For the total physical model (including the control system discussed in Section 3.4), the budget was $\approx$ $\$30,000$. The wind tunnel was constructed in-house by laboratory technicians, thus, labor was not included in the budget. Instead, the budget was split in half between expenses for the control system and for the wind tunnel. For the wind tunnel, the budget was split quite evenly through the cost for the test section, the fan, the wire cloth, and the smoke generator. The expense breakdown of the physical model can be seen in Table 3.2.

3.3.2.3 Construction Capabilities

The construction capabilities of the wind tunnel mainly influenced the orientation of the wind tunnel. It was necessary to build the contraction section, settling chamber, and diffusers out of wood as any other material would need outside contracting work (other materials often used include fiberglass and metal, with fiberglass the preferred material). Thus, because these components were made of wood, it was important that the test section be rectangular, as a circular system would be far more difficult to build with wood. This also lead to the decision to make the wind tunnel a *blower* wind tunnel (i.e. fan at upstream end of test section) rather than a *draw-down* wind tunnel (i.e. fan at downstream end of test section) as the draw-down tunnel would have required a rectangular-to-annular transition section, that would have again been too difficult to construct with wood. One of the difficulties when constructing a rectangular wind tunnel is the unwanted vortices produced in the corners. To mitigate these vortices, it is standard practice to use fillets in the test section, which was done with this wind tunnel.

3.3.2.4 Timeline

Design of the physical model began the summer of 2012, and construction the following summer 2013. The complete physical model (control system and wind tunnel) was completed in June 2016. All materials for the physical model needed to be purchased by the end of June 2013. Many of the items purchased for the control system were decided upon prior to completion of the wind tunnel construction, which has lead to some limitations in the control systems implementation, and that will be discussed in Section 3.6.

3.3.3 Wind Tunnel Orientation

The wind tunnel built for this research is a *blower* wind tunnel. Draw-down wind tunnels are the standard types of tunnels built and used in aerodynamic applications; however, usually the application of these wind tunnels is to study the fluid flow over some scaled-down component of a vehicle. The test sections in these types of wind tunnels are usually quite short, with an almost equivalent ratio between the width/height of the test section to its length. Additionally, there is not usually a fluid injected into the boundary of the test section for these types of applications, so the upstream and downstream flow are not necessarily examined. Thus, the ultimate objective for this wind tunnel was very different than standard uses of wind tunnels. Therefore, a draw-down wind tunnel was not required. Instead, for ease of construction and economical concerns, a blower wind tunnel was built.

Another factor in the design of a wind tunnel is the air speed. This wind tunnel operates at low enough speed to be considered a *low speed* wind tunnel, sometimes referred to as a *subsonic* wind tunnel, which have a low Mach number (test section speed $< 134\,\mathrm{m/s}$ and Mach number < 0.4.)

Another design decision for the wind tunnel was whether it would be *open-looped* versus *closed-loop*. The benefits of closed-loop designs are such that air is recirculated and long tests can take place. Additionally, there is the possibility of having two test sections with a closed loop design. For the same reasons stated above, an open-looped wind tunnel was used because it was more economical, easier to build, and the benefits were not necessary to this research.

The computer-aided drawing of the wind tunnel designed for this work (without the fan) can be seen in Figure 3.3. The different components of the blower wind tunnel are the test section, wind tunnel orientation, fan, wide-angle diffuser, settling chamber, contraction section, and exit diffuser; these components can be seen in the schematic in Figure 3.2, which is based of the work of *Mehta and Bradshaw* (1979). The first decision to be made after the orientation was decided is what the flow must look like in the test section, as that is what guided the needs of the rest of the components.

3.3.4 Test Section

The goal for the wind tunnel is such that air enters the test-section with near uniformity in the flow, and absent of flow separation. To allow for multiple iterations of feedback flow control, the test section was to be made such that it was as long as possible. For the space provided, this allowed for a design of a test section of

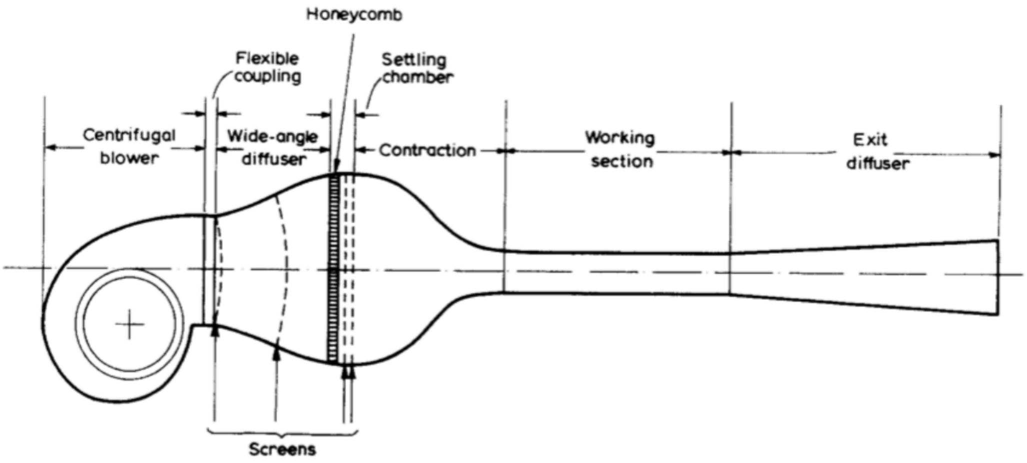


Figure 3.2: Basis of physical model wind tunnel design from *Mehta and Bradshaw* (1979).

potentially over 20 ft in length (≈ 6 m). The cross-sectional area of the test section would influence the size of the components upstream of test section as at least a 6 : 1 ratio is necessary between the cross-sectional area of the settling chamber to the test section. It was decided that a 2 ft × 2 ft cross-sectional area would be used for the test section, in part due to the increase in length of the contraction section were the cross-sectional area any longer, and also in part due to the economical save it would be to have the sides of the test section be multiples of 2 ft.

The material used for the test section is clear such that optical sensors can be used. Specifically, a combination of acrylic plexiglass and tempered glass is used: tempered glass is on the side facing the camera sensor to improve the optics, while plexiglass was used for the other three sides so as it could be cut to install the vacuum nozzles. Altogether, there will be at least three sections that together will yield a total length of at least 24 ft (≈ 7 m). Each section was 8 ft (≈ 2.5 m) in length as the sheets of acrylic plexiglass and tempered glass came in sheets that are 8 ft × 4 ft in size (≈ 2.5 m × 1.25 m). This long length will allow for multiple sets of sensors/actuators to be installed for feedback control. For the experimental results discussed in Section

3.5, only two sections of the test section were used.

Even though the velocity of the air is uniform at the test section entrance, as the air continues through the test-section, there is boundary layer growth eventually creating more of a parabolic flow profile before exiting the test section. While boundary layer growth is generally unfavorable for wind tunnels, for this research, the focus is more on the feasibility of carrying out detection/control, and thus, the boundary layer growth was not the greatest concern. Although, it will need to be taken into consideration when trying to match any experimental data to computational.

3.3.5 Centrifugal Fan

A picture of the centrifugal fan to push air through the wind tunnel can be seen in Fig. 3.4. The fan provides a flowrate of up to 2000 ft^3/s (which for the given cross-sectional area of the test section approximates up to $\approx 2.5\,\mathrm{m/s}$ velocity in the test section). A variable frequency driver is also used to allow for the fan's flowrate to be controlled down to one-tenth its maximum flowrate. The air used will be the ambient air inside of the structure laboratory. A table showing the different test section velocities for the frequency provided to the fan can be seen in Table 3.3. The centrifugal fan also provides a static pressure of ≈ 1 wg H_2O ($\approx 249\,\mathrm{Pa}$).

3.3.6 Wide-Angle Diffuser

Once air enters the wind tunnel via a fan, it first travels through an expansion section connecting the fan to the settling chamber; this expansion section is known as the wide-angle diffuser. It is necessary to have an expansion with a large area ratio. But the difficulty lies in reducing the length over which this expansion occurs, which subsequently reduces the pressure loss. A wide-angle diffuser allows for this area expansion to occur over a shorter length. However, with such a short length, there is an increased risk for separation in the fluid. Therefore, there must also be a means

Table 3.3: Frequency of the fan corresponding to velocity in test section.

Motor (Hz)	Test Section Velocity, ft/s (m/s)
5.0	0.69 (0.21)
10.0	1.39 (0.42)
15.0	2.08 (0.64)
20.0	2.78 (0.85)
25.0	3.47 (1.06)
30.0	4.17 (1.27)
35.0	4.86 (1.48)
40.0	5.56 (1.69)
45.0	6.25 (1.91)
50.0	6.94 (2.12)
55.0	7.64 (2.33)
60.0	8.33 (2.54)

of boundary control, such as metal screens, to minimize this risk for separation.

There are two main design decisions that must be made for the wide-angle diffuser: (i) the length of expansion (and subsequent angle), and (ii) the number of screens. The two decisions are interdependent. A chart compiled from previously built wind tunnels was used to determine the number of screens necessary to minimize separation of flow for the diffuser expansion angle. For this physical model, the wide-angle diffuser is 5 ft long ($\approx$ 1.5 m) with 3 screens total. The screens used were the same screens decided upon for the settling chamber and discussed in greater detail below.

3.3.7 Settling Chamber

The next section, the settling chamber, contains a metal honeycomb used to literally straighten the flow, reducing any turbulent swirl in the flow; and is then followed by a series of metal screens that are used to minimize the differences in the flow field, which in turn improves the uniformity of the flow. The settling chamber's cross-sectional area was chosen as 5 ft $\times$ 5 ft ($\approx$ 1.5 m $\times$ 1.5 m) due to a necessary contraction ratio occurring between it and the test section, discussed further below. The settling chamber has a much greater cross-sectional area than the test section so that the velocity of the fluid is much lower to reduce the pressure losses in the flow as it moves through the honeycomb and screens.

An aluminum honeycomb that is 5 ft × 5 ft (≈ 1.5 m × 1.5 m) in area is used for the settling chamber. The cell length of each honeycomb should be about $6-8$ times its diameter [*Mehta and Bradshaw* (1979)]. Thus, a honeycomb with a cell length of 3 in (≈ 75 mm) with a cell size of 3/8 in (≈ 9.5 mm) is used.

For the screens (also known as wire cloth) used in the settling chamber and the wide-angle diffuser, the major design decisions are that: (i) the open area is greater than 57% and (ii) the mesh count (i.e. number of wires per lineal inch) be as large as possible [*Bradshaw and Pankhurst* (1964)]. Screens with a 67.4% open area, and 0.0075 in (≈ 0.2 mm) wire diameter were selected for this physical model. The same screens are used for both the settling chamber and the wide-angle diffuser discussed above.

3.3.8 Contraction Section

The final part of the wind tunnel design before the test section is the contraction section. The purpose of the contraction section is two-fold: (i) to increase the mean velocity of the flow after it moves through the screens and honeycomb in the settling chamber, and (ii) to reduce the mean and fluctuating velocity variations to a smaller fraction of the average velocity [*Barlow et al.*]. The contraction section connects the 5-feet by 5-feet (≈ 1.5 m × 1.5 m) settling chamber to the 2 ft × 2 ft (≈ 0.6 m × 0.6 m) test section. The standard contraction ratios used in wind tunnel designs are between $6-9$, thus for this physical model there is a ratio of 6.25.

The major design decision made with the contraction ratio is the contraction curvature. As the fluid flows in the contraction section, there is a high risk for either separation or boundary layer growth to occur. Therefore, the curvature of the contraction section must be chosen such that separation does not occur and boundary

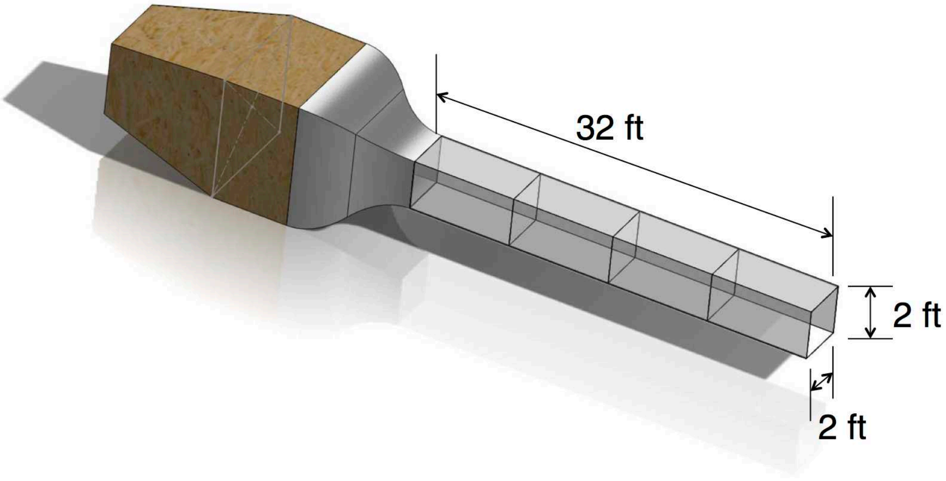


Figure 3.3: CAD drawing of the physical model for construction.

layer growth is minimal. There does not exist a standard contraction section curvature to ensure no separation occurs and boundary layer growth is minimal; however, many studies have examined what contraction section curvature works better than others. A cubic spline was chosen for the contraction section built in this wind tunnel and is based on the guidelines provided in *Bell and Mehta* (1988).

3.3.9 Exit Diffuser

For wind tunnels with large velocities, there exists a need to reduce the speed at the end of the test section before the air in the wind tunnel reaches the ambient environment, thus, minimizing disturbance. To reduce this speed, an exit diffuser is usually used that gradually increases the cross-sectional area in which the air is flowing (i.e. for a constant flow rate, the larger the cross-sectional area, the lower the velocity). Because the speeds for this wind tunnel are relatively small, the exit diffuser was determined to be unnecessary.

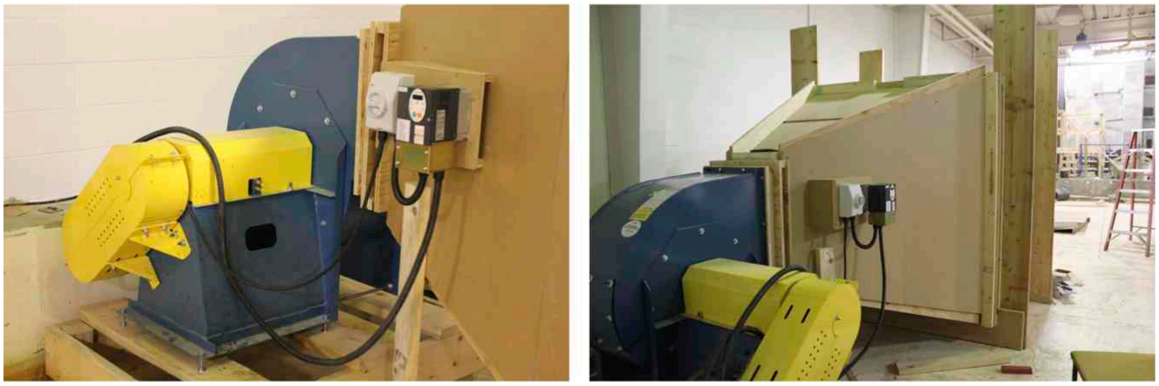


Figure 3.4: Picture of centrifugal fan for physical model.

3.4 Real-time Control System

A schematic of the control system designed and implemented in the long test-section conduit can be seen in Fig. 3.5. The control system is currently for a single time horizon. However, the system was designed such that it can be extended to include another time horizon, allowing for future implementation of feedback control. Unlike the wind tunnel, there did not exist a standard design for this control system, thus much of it was designed provisionally. Therefore, the initial results for this portion of the physical model were about feasibility of real-time control, rather than carrying out an actual controlled experiment.

3.4.1 Contaminant

The 'contaminant' being using for the physical model is propylene glycol smoke (i.e. the smoke used in everyday fog machines). The smoke generator was designed by Aerolab LLC and can be seen in Fig. 3.6. The smoke generator is usually used in wind tunnels for the purpose of flow visualization. However, for this physical model, the smoke generator creates a plume of white smoke that is injected into the ambient wind tunnel flow. A picture of the plume created by this smoke machine can also be seen in Fig. 3.6.

3.4.2 Sensors

The visual sensor that will be used to detect the plume of smoke is a Manta GigE Vision camera from Allied Vision Technologies. The camera was chosen such that: (i) it was compatible with a lens with a long frame to capture most of the length of the wind tunnel test section (ii) can take multiple pictures in a short time frame, and (iii) have the ability to send those pictures to the controller with minimal time lag. The camera sends the pictures via an ethernet cable to the controller. It has the ability to capture images at a rate of 30 frames per second with a resolution of 1292×964. Furthermore, the camera can capture an area of $8\,\mathrm{ft} \times 4\,\mathrm{ft}$ into its frame. An image of the camera can be seen in Fig. 3.7.

3.4.3 Actuators

The actuators used in the control system shown in Fig. 3.5 are operated by a compressed air system. Compressed air is stored in a 60 gal ($0.23\,\mathrm{m}^3$) receiver tank supplied by the building's compressed air system. The compressed air tank is connected to pressure regulators. The pressure regulators are connected to the actuators, and are powered and controlled by the connected controller.

The company used to supply the actuators is EXAIR Corporation that specializes in compressed air-operated products. Two types of actuators will be used for this project and can be seen in Fig. 3.8. For the vacuum nozzle, the compressed air is used to create a vacuum inside of the vacuum nozzle, which can then be used to draw the contaminant out. For the air knife, the compressed air is used to create a sheet of air flowing in a downward direction. The air knives will be used to section off the test section, mimicking air curtains in buildings. The velocity that the vacuum

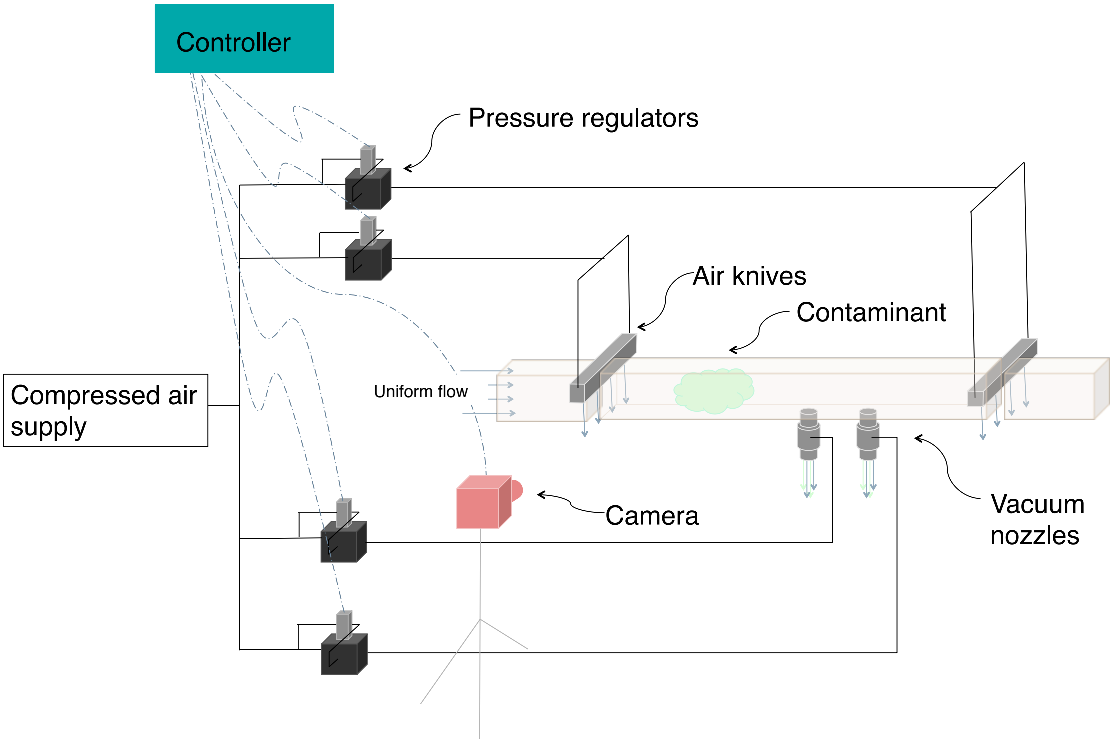


Figure 3.5: Schematic of physical model control system implemented for a single time horizon.

nozzles create for a given pressure is not known; therefore, the vacuum nozzles must be calibrated.

3.4.4 Controller

The controller to be used for this physical model is a National Instruments (NI) PXIe-8102 embedded system, and can be seen in Figure 3.9. It is stored inside of a NI controller chassis. The controller has an ethernet connection card that allows the camera sensor to be hooked up to it and process the pictures in real-time. Furthermore, the controller has connections to send a voltage to the pressure regulators, which in turn control the pressure sent to the vacuum nozzles and air knives. The software that will be used to program the controller is the NI LabVIEW Real-Time Module, which is an extension of the LabVIEW Core Software.

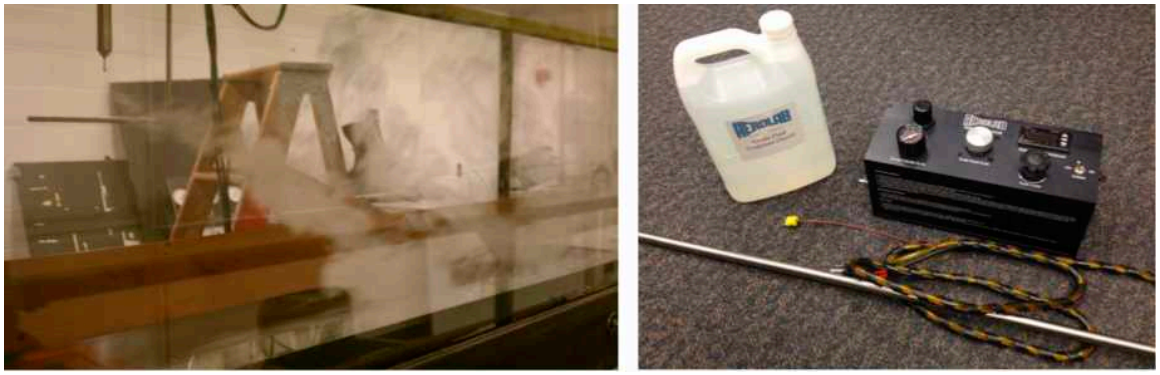


Figure 3.6: Example smoke plume made by the smoke generator system on right.

Figure 3.7: Camera used as an image sensor.

3.5 Results

As has been stated, much of the goal of the physical model is to demonstrate feasibility and proof-of-concept of real-time control. Thus, the ultimate goal and

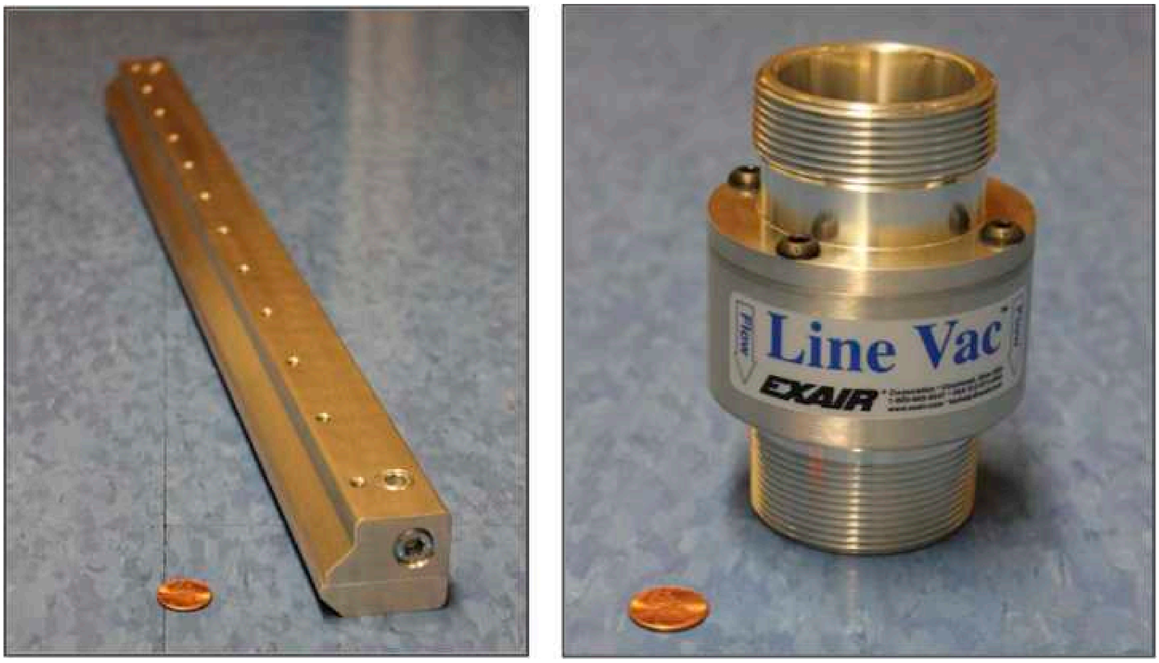
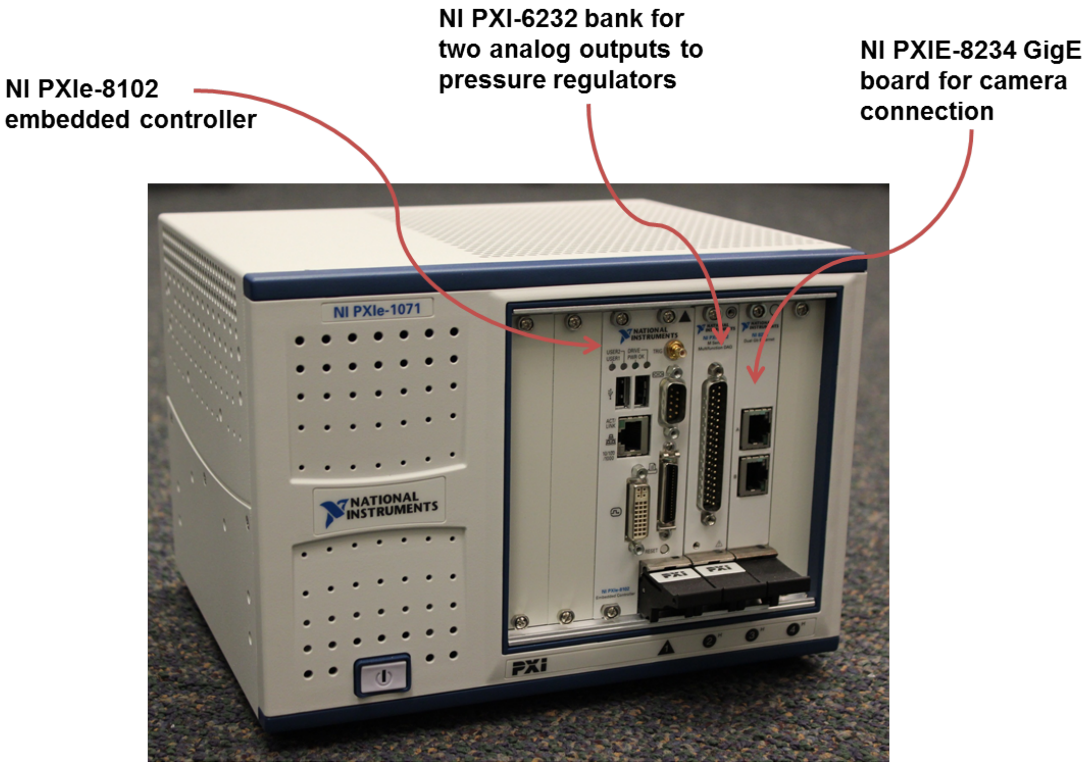


Figure 3.8: The actuators used for the control of the contaminant. On the left is the air curtain; on the right is the vacuum nozzle.

subsequent results are to show that real-time control can occur, without necessarily having fidelity to the CFD flow-control model from Chapter II at this point. To do so, the first tasks were to successfully connect and process the data to and from the controller via the camera and the pressure regulator. Once those were connected and information was successfully collected and sent by the controller, a real-time control experiment was carried out to see if a contaminant could be injected into the test section, read in by the camera, and dispatch a signal to the controller. This experiment was successfully carried out, with some interesting behavior by the contaminant. Thus, a simulation with the CFD flow-control model was carried out similar to the setup of the physical model's test section to verify the qualitative fluid behavior.



NI PXIe-8102 embedded controller

NI PXI-6232 bank for two analog outputs to pressure regulators

NI PXIE-8234 GigE board for camera connection

Figure 3.9: The controller used for this physical model.

3.5.1 Real-time Image Processing

The NI PXI-8232 Gigabit Ethernet port is one of the cards that was purchased to be housed within the real-time controller. Images are able to be taken and sent with 30 fps from the Manta GigE Vision camera to the controller. In order to 'detect' the contaminant within these images, they must be processed by the controller. To demonstrate this filtering, an example of the contaminant plume from Figure 3.6 as it is filtered from color, to greyscale, to black and white can be seen in Figure 3.10.

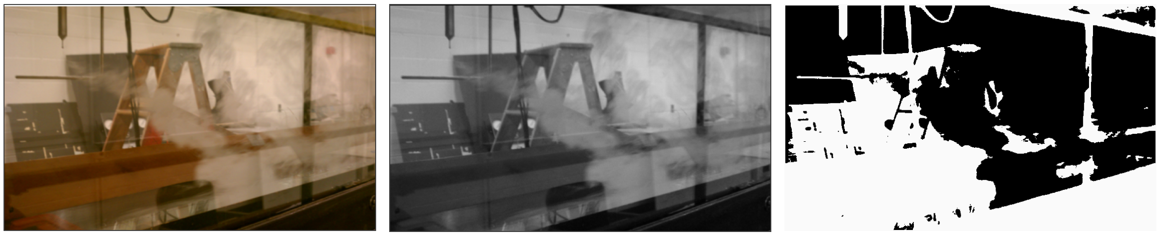


Figure 3.10: The filtering of a color image of the smoke plume, to greyscale, to black-and-white.

The images sent from the camera are greyscale. To detect the contaminant, the images were converted to black and white. This conversion was carried out using the `IMAQdx` function palette from the Labview programming software. A series of functions were used to convert the greyscale image into an array of a binary set of numbers (e.g. 1 for white; 0 for black). A threshold was set such that the images able to filter out possible noise from the camera, and to extract the existence of the contaminant plume as best as possible. The images with the different threshold can be seen in Figure 3.11.

Once a base threshold was decided upon, the controller was programmed such that it read in images continuously, processed the images, and once the sum of the black-and-white array reach a set value higher than the threshold (signifying significant presence of the contaminant), a signal was generated to change settings on the pressure regulator. The final example of the filtered image can be seen in Figure 3.12.

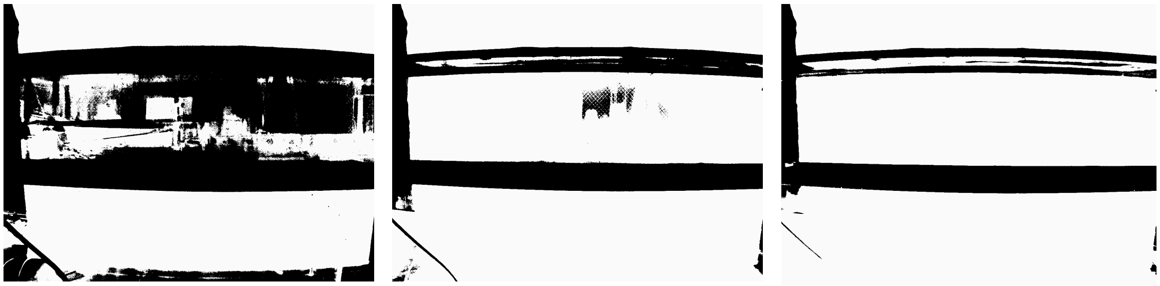


Figure 3.11: The filtering black-and-white images to reduce noise.

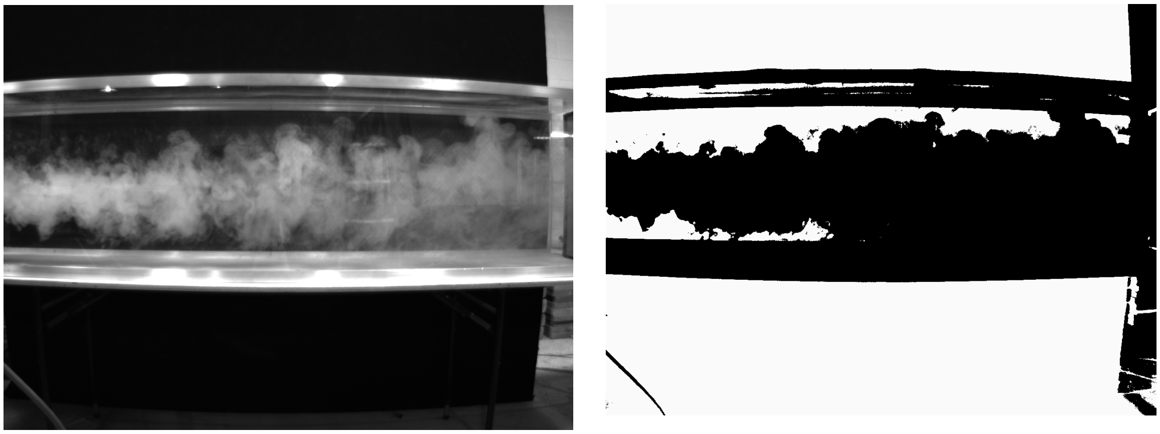


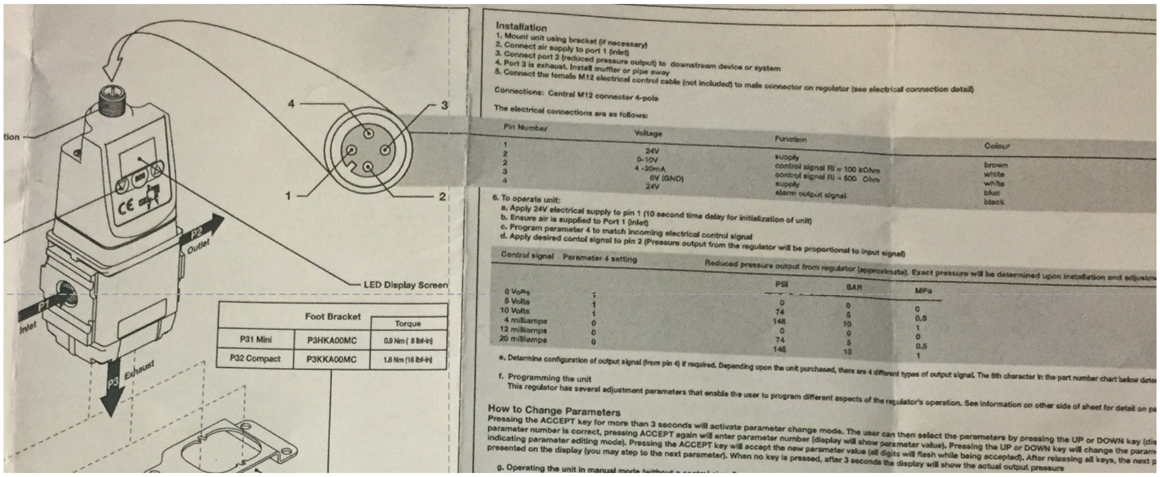
Figure 3.12: A contaminant plume filtered from greyscale to black-and-white.

3.5.2 Real-time Signal Processing

The NI PXI-6232 Multifunction Data Acquisition (DAQ) board was used in conjunction with the real-time controller to send voltage to the pressure regulator, which in turn controlled the compressed air supplied to the vacuum nozzle, and its subsequent flowrate. The PXI-6232 Multifunction board has the ability to send up to two voltage analog signals (up to 10 V each), up to four voltage digital signals (up to 24 V each), and receive signals from up to six digital inputs.

The connector for the pressure regulator can be seen in Figure 3.13. As can be seen from the figure, the regulator has a pin for one digital voltage input (which powers the regulator), and one analog voltage input (which controls the subsequent flowrate of the connected nozzle). The pressure regulator was connected to the DAQ board via one voltage output analog signal, and one voltage output digital signal. A constant voltage was supplied to the board using a standalone controllable voltage supplier.

The signals were sent using the `NI-DAQmx` driver software, and programmed with the subsequent `DAQmx` LabVIEW function palette. Once a case loop received a signal from the image case loop, the analog voltage channel was triggered to supply a voltage of 5 V to the regulator, which essentially turned on the vacuum nozzle.


tion
4
3
1
2
CE
P2
Outlet
P1
Inlet
P3
Exhaust
LED Display Screen

Foot Bracket		Torque
P31 Mini	P3HKA00MC	0.9 Nm (8 lbf-in)
P32 Compact	P3KKA00MC	1.6 Nm (16 lbf-in)

Installation
1. Mount unit using bracket (if necessary)
2. Connect air supply to port 1 (inlet)
3. Connect port 2 (reduced pressure output) to downstream device or system
4. Port 3 is exhaust. Install muffler or pipe away
5. Connect the female M12 electrical control cable (not included) to male connector on regulator (see electrical connection detail)

Connections: Central M12 connector 4-pole

The electrical connections are as follows:

Pin Number	Voltage	Function	Colour
1	24V	supply	brown
2	0-10V	control signal Ri = 100 kOhm	white
2	4 -20mA	control signal Ri = 500 Ohm	white
3	0V (GND)	supply	blue
4	24V	alarm output signal	black

6. To operate unit:
a. Apply 24V electrical supply to pin 1 (10 second time delay for initialization of unit)
b. Ensure air is supplied to Port 1 (inlet)
c. Program parameter 4 to match incoming electrical control signal
d. Apply desired contol signal to pin 2 (Pressure output from the regulator will be proportional to input signal)

Control signal	Parameter 4 setting	Reduced pressure output from regulator (approximate). Exact pressure will be determined upon installation and adjustm		
		PSI	BAR	MPa
0 Volts	1	0	0	0
5 Volts	1	74	5	0,5
10 Volts	1	148	10	1
4 milliamps	0	0	0	0
12 milliamps	0	74	5	0.5
20 milliamps	0	148	10	1

e. Determine configuration of output signal (from pin 4) if required. Depending upon the unit purchased, there are 4 different types of output signal. The 8th character in the part number chart below deter

f. Programming the unit
This regulator has several adjustment parameters that enable the user to program different aspects of the regulator's operation. See information on other side of sheet for detail on pa

How to Change Parameters
Pressing the ACCEPT key for more than 3 seconds will activate parameter change mode. The user can then select the parameters by pressing the UP or DOWN key (dis parameter number is correct, pressing ACCEPT again will enter parameter number (display will show parameter value). Pressing the UP or DOWN key will change the param indicating parameter editing mode). Pressing the ACCEPT key will accept the new parameter value (all digits will flash while being accepted). After releasing all keys, the next p presented on the display (you may step to the next parameter). When no key is pressed, after 3 seconds the display will show the actual output pressure

g. Operating the unit in manual mode

Figure 3.13: The wired connector for the pressure regulator.

3.5.3 Real-time Control Feasibility Experiment

An experiment was carried out to see if the Image and Signal Processing could work successfully in conjunction with the wind tunnel and corresponding controller. The initial setup of the experiment can be seen in Figure 3.14. Screenshot images from the experiment can be seen in Figure 3.15. The threshold described in Section 3.5.1 was set such that the stream of smoke from the smoke generator could fully develop before the nozzle was triggered. For this experiment, the frequency for the centrifugal fan was 12 Hz, which translates to about 1.7 ft/s (≈ 0.5 m/s). The experiment lasted for a total of about 5 min (although the controller was triggered within a minute of starting the experiment). While it is difficult to discern from the photos, the experiment proceeds such that the smoke generator begins to supply a continuous plume of smoke (see part (a) of Figure 3.15) Eventually, once enough smoke fills the test section that is being sensed by the camera, the control system is triggered (see part (b) of Figure 3.15). As a result of the control, significant mixing occurs at the downstream end of the test section, and disruptions in the flow from the vacuum nozzle can be seen on the upstream end of the test section (see parts (c) and (d) of Figure 3.15). A more detailed view of the upstream disruptions can be seen in Figure 3.16.

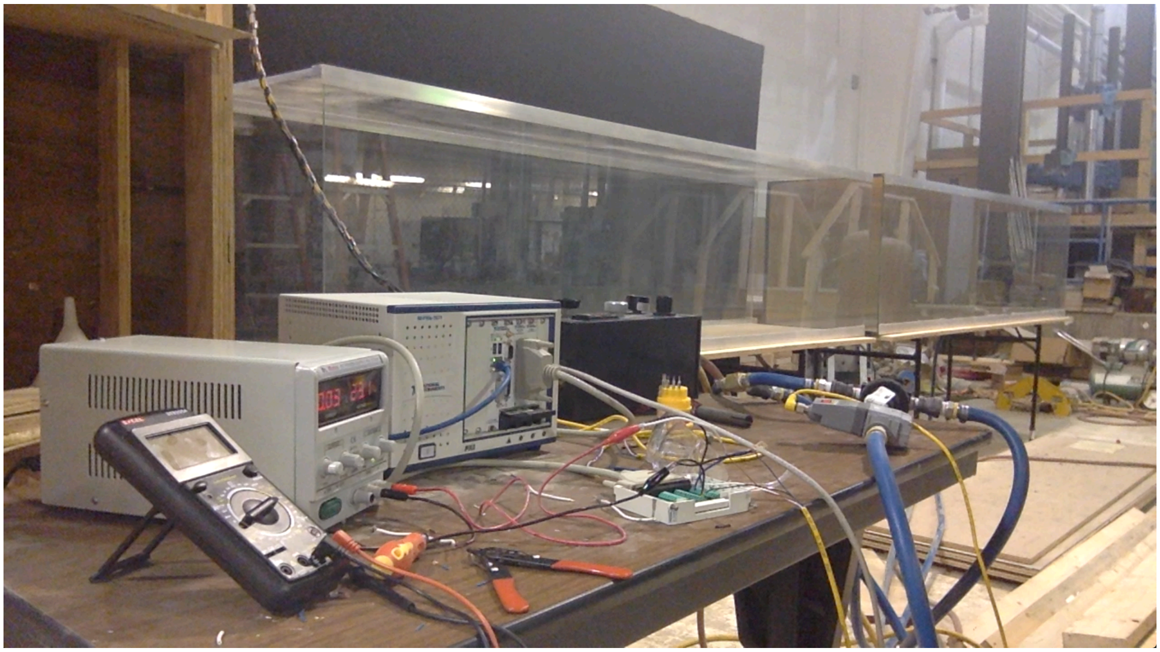


Figure 3.14: The experimental setup for the physical model. The controller is connected to the camera and pressure regulator, which controls the vacuum nozzle at the downstream end of the tunnel.

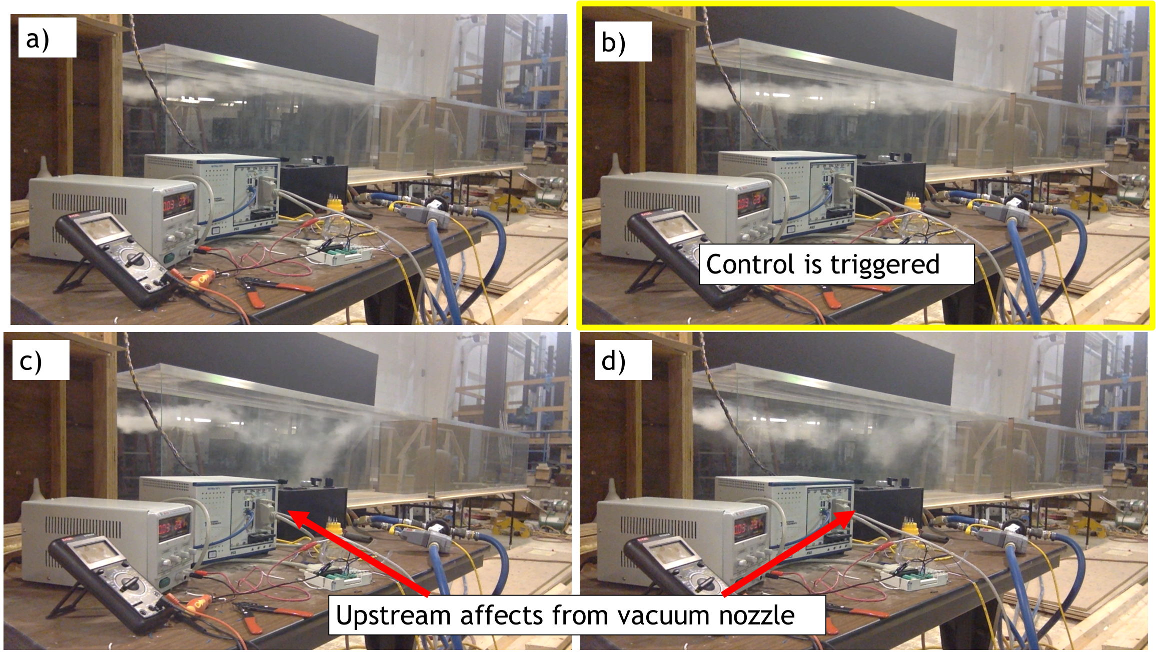


Figure 3.15: Screenshots of the physical experiment. In picture (a), the smoke is just starting to be generated; for picture (b) enough smoke is generated that the control action is triggered; in pictures (c) and (d) there are two types of fluid phenomena occurring as a result of the vacuum nozzle: disturbances at the upstream end of the test section and mixing at the downstream end.

3.5.4 CFD Flow-Control Applied to Physical Model

Two of the interesting observations of the fluid behavior from the experiment are: (i) the vacuum nozzle influences the upstream behavior of the smoke to a certain extent, and (ii) while the nozzle is not successful at completely removing the contaminant; instead it creates significant mixing with the ambient air in the downstream end of test section.

To see how these qualitative observations matched up between the physical model and the CFD flow-control model, a simulation of the experimental setup of the physical model was carried out using the CFD flow-control model. Now it must be stated the geometries are different for both examples; the flowrate out of the vacuum nozzle is unknown, so the flowrate for the port in the CFD flow-control model was a guess;

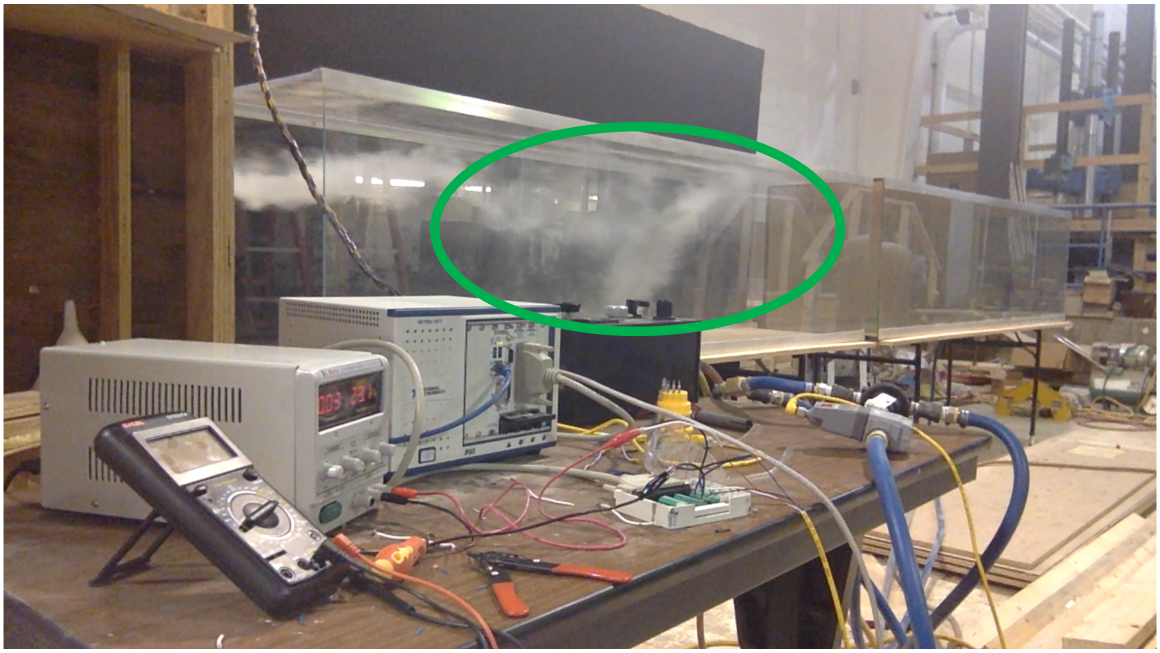


Figure 3.16: Disruption upstream of flow.

and the contaminant in the CFD flow-control model is not assumed to be smoke, and actually is assumed to be neutrally buoyant, which is the the case for the propylene glycol in the experiment. However, even with these

The CFD flow-control model was ran with a single downstream port on the bottom of the domain. The contaminant was released into the ambient flow continuously, creating a constant stream of contaminant. Because the flowrate of the nozzle for the physical model is unknown, deciding upon the flowrate of the nozzle for the CFD flow-control model was somewhat tricky. Thus, there were two runs that were carried out: (i) one with a port velocity range similar to those from Section 2.4 (i.e. $0 - 3\,\mathrm{m/s}$), and (ii) one with a significantly higher port velocity range of $20 - 40\,\mathrm{m/s}$.

The results from these two runs can be seen in Figures 3.17 and 3.18. As can be seen, the upstream influence from the nozzle is similar to that of the physical model. Additionally, for the high port velocity range, mixing creating what looks like uniform distribution of the contaminant in the downstream end is also apparent.

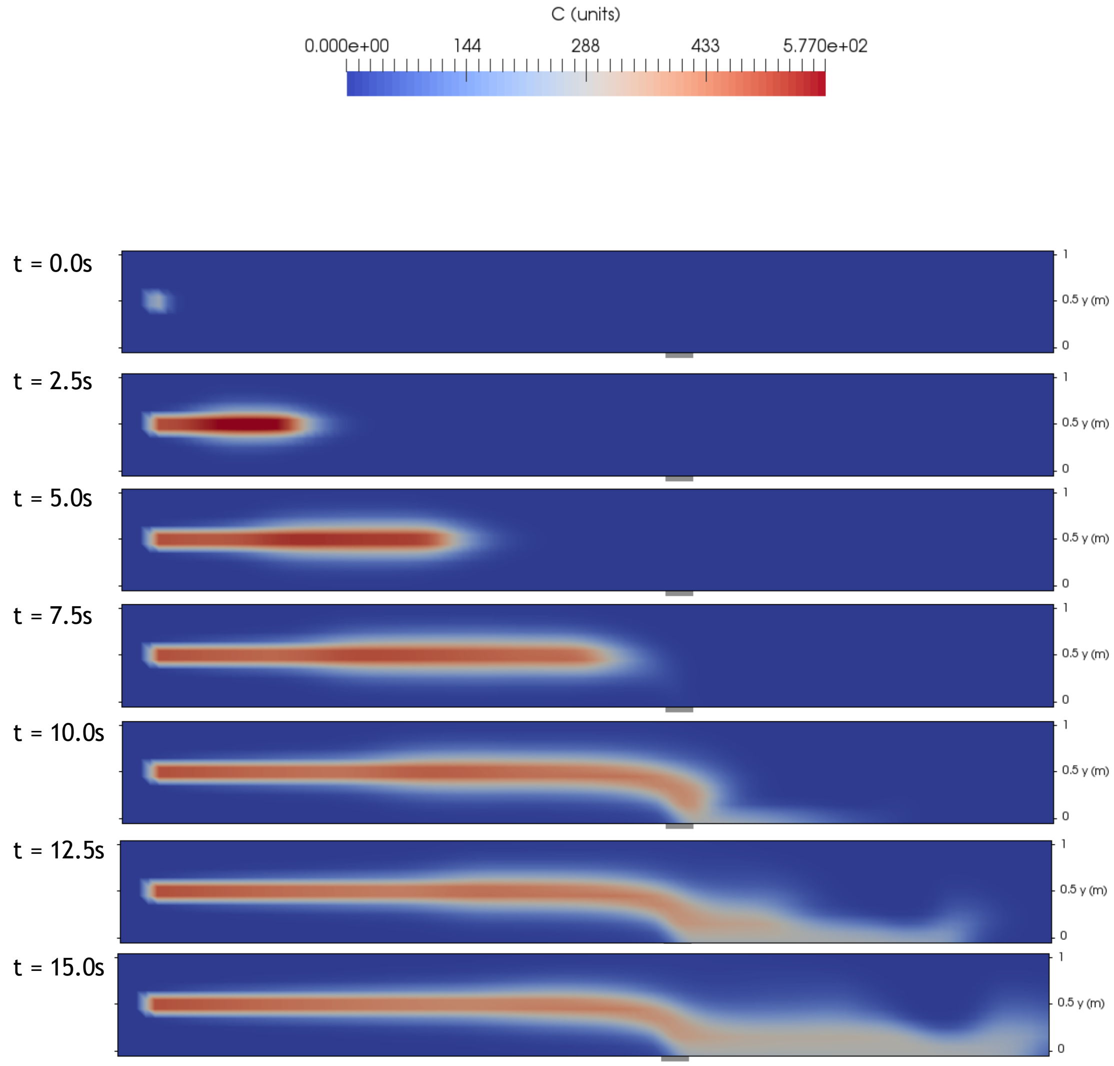


Figure 3.17: One port with continuous contaminant. The control port can only create a transpiration control action at a low velocity ($\leq 3\,\mathrm{m/s}$).

3.6 Discussion

The physical prototype and the real-time control system were implemented successfully, however, there still exists quite a bit of limitation. The main limitation to the prototype is the nozzle that draws the contaminant out. Because the nozzle functions by being supplied a large amount of compressed air, it creates a high velocity that significantly disrupts the flow in the test section, rather than simply drawing

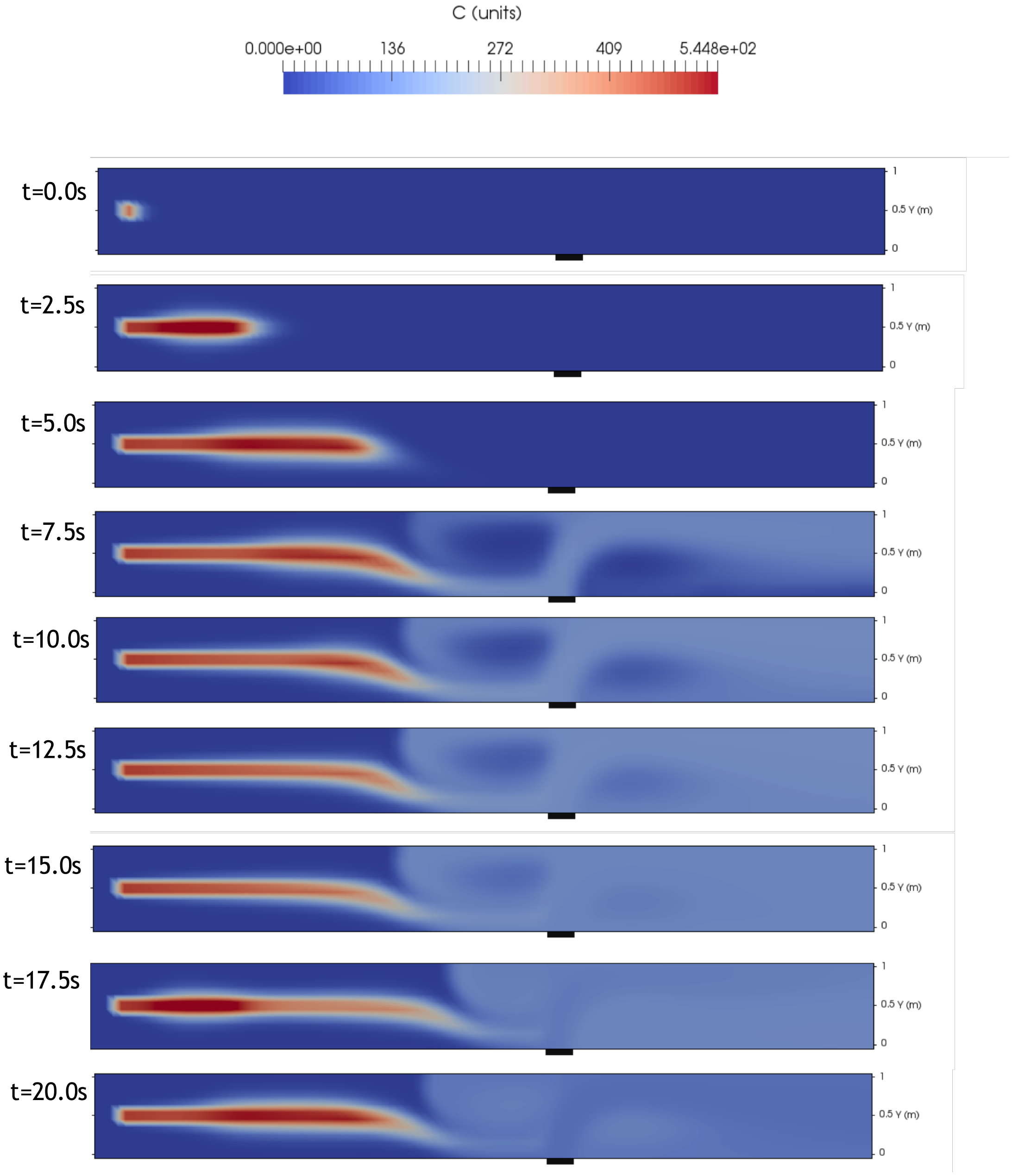


Figure 3.18: One port with continuous contaminant. The control port can only create a transpiration control action at a high velocity (20 – 40 m/s).

out the contaminant. Part of the reason this design is limited is that the items for the prototype were purchased before the wind tunnel was actually built. Boundary control via air is a much more difficult problem than with water as the ambient fluid.

Thus, in the future an improved control system should be considered.

Further analysis needs to be carried out regarding similarity of fluid flow between the prototype, the computational model, and the actual public spaces being studied. As was seen earlier in the chapter, matching Reynolds numbers exactly for the three analyses might not yield the information that is pertinent to this research. Instead, questions regarding what time scales and length scales of the different components of the experimental setup – particularly when compared to the actual public spaces – should be further analyzed and possibly adjusted. For example, is the length scale of the width of the test section what is important, or is the length scale of the turbulent disruption what should be taken into consideration.

It is also important to note how the CFD flow-control model qualitatively is able to capture some of the disturbances seen in the prototype from the high velocity nozzle. It will be interesting to compare these two models more (physical and computational) after actual measurements are made on the flow field within the wind tunnel.

CHAPTER IV

Evacuation Dynamics

4.1 Motivation

Many different computational models have been developed in order to simulate humans and crowds as they evacuate a space. These models have assisted a countless number of public safety teams in search and rescue operations, emergency training scenarios, and in building safety design [*Aguirre et al.* (2011)]. Moreover, some computational egress models have incorporated evacuation environments that are dynamic and threatening, and with which agents must interact (e.g. earthquakes) [*El-Tawil and Aguirre* (2010)].

This research seeks to build upon this work by developing an evacuation ABM that is coupled with a CFD flow-control model of a spreading contaminant within a public infrastructure system: the CFD flow-control model is able to "sense" the location of the humans evacuating and "control" the contaminant via boundary actuators to minimize contaminant exposure to evacuees. We look at public spaces defined by a long conduit which allows us to assume unidirectional ambient fluid flow. We initialize the agents in our evacuation ABM with one trait that we defined to influence their egress decision-making. This research demonstrates how basic decision making at the human-level influences aggregate behavior of our human population, which in turn influences the CFD flow-control that is carried out in such a space.

4.2 Background of Agent-based Modeling

From an engineering standpoint, modeling how human's behave can be a jarring and formidable experience. Moving from physics to people is quite a leap, a sentiment best summarized with a statement posed by the physicist *Gell-Mann* (1995):

> *"Imagine how hard physics would be if atoms could think."*

This statement epitomizes the challenge in studying human behaviors. Traditionally, if engineers were to study a system that involved humans, the most comfortable and obvious approach might be to develop some sort of mathematical model that can be used to describe the overarching system, and fit this model to the observed behavior. Actually, that has been a strategy by many seeking to bridge their engineering with the human components that might influence it. However, at best, these sorts of models are limited in their reach and fallacious in their application. Therefore, there exists a need to bring in a more realistic method to model systems that involve human behavior. Agent-based modeling is one technique used throughout this research that allows for heterogeneity in decision-making among its agents at the micro-scale, giving way to the broader macro-scare patterns of the population as a result.

4.2.1 What are Complex Systems?

ABM is a technique used to model what are known as *complex systems*. A system is *complex* when it has a diverse set of interacting *agents*–or parts–whose collective behavior exceed and transcend the capabilities of the constituent agents. As laid out by *Boccara* (2010), a complex system more or less has the following traits:

1. It is a system consisting of individual, interacting *agents*,
2. The system exhibits *emergence*, or rather self-organizing, collective behavior that might be difficult to distinguish solely by observing a single agent's behavior,

3. There is no central controller that gives rise to this emergent behavior, thus this self-organizing occurs from the bottom up, rather than top-down.

Some examples of complex systems include sparrows flocking together; fish swimming in a school; an audience giving a standing ovation, or possibly the wave at a sporting event; racial self-segregation in an urban neighborhood; or cells as they develop into an embryo[1].

One of the most famous examples of complexity is that of foraging and maintenance in an ant colony. As *Boccara* (2010) describes it (based on the work of *Gordon* (1996)), worker ants of the harvester ant species perform one of four tasks: (i) foraging along cleared trails in the nest to collect seeds; (ii) performing maintenance in the nest by clearing sand or vegetation; (iii) patrolling the nest by responding to damage to the nest or protecting it from invasion of other ant species; and (iv) collecting and sorting the nest's waste. What Gordon and his team showed is that the allocation of these four tasks is not set, and instead is continually adjusting. For example, when researchers placed a toothpick near the entrance to the nest, the number of ants performing maintenance on the nest significantly increased. Additionally, when mounds of seeds were placed near the nest, the number of ants foraging also significantly increased. But the fascinating aspect of this example is that there was no central controller that decided upon these tasks, nor would it have been possible for any macro-level agent, such as the queen ant, to have allocated tasks in a way that led to such cooperative, efficient, and emergent behavior amongst so many ants. Instead, the organization among the ants occurred at the micro-level, with each ant behaving based on their perception of the chemical and tactile signals communicated to them by other agents. Because this emergent behavior was a result of the individual agents in the system, it is *complex*.

[1] The list of different complex systems is endless, and spans social, economic, ecological, biological, political, political etc. realms. For examples of different complex systems, the reader is directed toward computational tutorials on agent-based modeling, such as Repast Simphony [*North et al.* (2013)] or Netlogo [*Tisue and Wilensky* (2004)].

4.2.2 The Case for Agent-based Modeling

What is present in all of these examples described above is that micro-level behavior and interactions amongst the agents within the model are what lead to the broader aggregate behavior. The field of complex systems spans beyond the behavioral dynamics of the systems described above, and the ability to describe the many other aspects of complex systems within this text would be a vain attempt. Instead, the reader is directed to other texts such as *Gilbert* (2008).

But behavioral dynamics is one of the fields that has been completely revolutionized by agent-based modeling, a method that came about from complex systems. Rather than develop a set of governing equations, or a different high-level mathematical means, to describe the system and its corresponding emergent behavior, ABM builds a model computationally from the bottom-up, via the accretion of the agents.

At this point, to better explain ABM, it would be a useful exercise to juxtapose ABM with CFD. They are similar in that they both use models to describe the observed world around; however, they contrast in both the development and application of these models. CFD uses a set of partial differential equations (e.g. Navier-Stokes equations) to describe the motion of a fluid. Because these equations cannot necessarily be solved outright, a computational model is developed that estimates the solution to these equations. If the computational model is from the Eulerian perspective (i.e. domain-level perspective), the estimates occur at nodes or volumes throughout the domain; if the computational model is from the Lagrangian perspective (i.e. particle-level perspective), the estimates occur for each of the hypothetical particles in the flow. Either way, the behavior of the fluid at these nodes or particles is dependent on the overarching equations.

Instead, for ABM, the model itself consists of the individual agents, and a set of simplified, diverse behavior individually prescribed, of which they can adapt and change with time. These agents are implemented into a computational model, and they move through space and/or time, interacting with each other, eventually giving way to the broader, emergent behavior. CFD is a tool in order to better understand the underlying theory of fluids; while ABM is tool to better understand human behavior[2].

The development of ABM, and the computational theory behind it, can first be seen in the mid-20th century, beginning with John von Neumann's self-replicating machine, which in turn motivated the development of cellular automata, in which a system is described by discrete grid cells able to take on a finite number of states that change values overtime based on the states of neighboring cells. The most famous example of cellular automata is *The Game of Life* by John Conway *Gardner* (1970). In this "game," a grid of square cells have one of two states: alive or dead. Each cell then progresses in time and may or may not change state based on a set of four basic rules concerning the state of the other cells it "interacts with" in its Moore neighborhood[3]. What is interesting about this cellular automata model is that based on the initial states of the cells, the model will either completely die, continuously evolve, or develop a constant state[4].

Conway's Game of Life actually lead to what is one of the first social science ABMs by *Schelling* (1980). Schelling developed a model similar to the Game of Life in which agents take on a "state," which in this case is a racial demographic. Again, each agent changes state based on rules, their individual prescribed preferences, and the corresponding state of their neighbors–thus, they are interacting with their neighbors. The model showed that even when the cells were seeded with a small overall preference and inclination to be nearer to other cells of the same state, it could ulti-

[2] Interestingly, some of the earliest models for simulating human evacuation used fluid dynamics concepts and their corresponding mathematical equations *Zheng et al.* (2009).

[3] To try out this "Game of Life," the reader is directed to the following website: `http://www.bitstorm.org/gameoflife/`.

[4] Interestingly, Conway's Game of Life has developed somewhat of a cult following, with followers changing or adding rules to find different emergent patterns in the grid, e.g. `http://www.conwaylife.com/`.

mately lead to total segregation of the cell population.

During the 1980s, Robert Axelrod was a seminal researcher who brought ABM to political science *Axelrod* (1981), while Craig Reynold's developed one of the first biological ABM recreating the phenomena of "flocking" of birds *Reynolds* (1987).

Soon after these initial models, ABM eventually extended into arguably all disciplines after accessible open-source software codes and corresponding tutorials for widespread use were developed in the 1990s and 2000s, namely Swarm (of the Santa Fe Institute of the study of Complex Adaptive Systems *Terna et al.* (1998)), Repast (University of Chicago), and Netlogo (Northwestern University). Today, ABM has become ubiquitous to a variety of, if not all disciplines *Macal* (2016).

4.3 Evacuation and Agent-based Modeling

Most investigation into human evacuation strategies occurs *a posteriori* as disasters that induce these types of evacuation responses are unpredictable, and unethical to experimentally create. Thus, social science researchers usually resort to interviewing evacuees after an egress. Although follow-up investigations are not necessarily the norm, nor always possible after events, the investigations that do occur have yielded considerable insight into the decision making processes of evacuees. While there are many underlying behavioral patterns that exist for all types of emergency evacuations, there are also unique behaviors that emerge depending on the type of emergency evacuees are facing. For example, one of the most interesting evacuation characteristics observed is that when evacuating a building, humans are more likely to move toward an exit where others are also evacuating, even if a quicker exit is available; this phenomenon is called *herding Pan* (2006).

Evacuation due to the spread of a hazardous contaminant is one of the least studied evacuation scenarios because of the rarity of such events. However, one common scenario that can be related to hazardous contaminant release is the need for evacuation during a fire inside a building. A plume of smoke created by fires can be considered to behave similarly enough that the evacuation of humans during such a scenario is a useful comparison to make. Of course, one limitation to such a comparison is that smoke is usually able to be sensed by building occupants in a variety of ways (e.g. smell, sight, feel, breath), whereas, other potential contaminants (e.g. biological, chemical) may not be perceived.

Many egress models have been developed with different behaviors amongst agents. *Pan* (2006) developed one of the more comprehensive agent-based models for evacuation out of a building, which was named MASSEgress. According to *Pan* (2006), the evacuation of a crowd can be modeled as the culmination of three levels: the agent, interaction amongst individuals, and group behavior. Usually, the most efficient form of evacuation occurs when an individual can make clear decisions. Disruptive, or nonadaptive, behavior is behavior by an individual that worsens her/his ability to evacuate. It is a result of high stress levels that may diminish the full functioning of one's senses. When the agents follow the model framework described above, the collective result may lead to emergent behavior. In *Pan* (2006), this emergent behavior was the top level of the behavior hierarchy. *Pan* (2006) allowed for three types of emergent behavior to occur in their model: competition, queuing, and herding.

Another egress model by *Pelechano and Badler* (2006) emphasizes an individual's role within a group during the evacuation. Individuals could take on one of three roles: trained leaders, untrained leaders, and untrained non-leaders (i.e. followers). A trained leader would have complete knowledge of a building's internal structure; an untrained leader may not have previous experience, but can handle stress better than others; and a follower is a person that might panic during an emergency situation and become incapably of making her/his own decision.

The one form of emergent behavior that occurred in the *Pelechano and Badler* (2006) model were the multiple clusters of agents that developed from the leader-follower hierarchy. The higher percentage of leaders in the population led to a larger number of groups with smaller amounts of agents in each group. The larger amounts of leaders will smaller groups looking for exits actually led to more efficient evacuation because the groups could communicate with each other. The lower percentage of leaders with larger group sizes had the opposite effect.

Recently, *Fang* (2015) has been developing an ABM that incorporates and emphasizes the importance of social and socio-psychological factors of agents as they evacuate. In particular, they develop a model in which agents will find their "familial" members before evacuating a space together.

One of the most important applications for evacuation models is their use in real emergency situations, and their potential to save lives in such a situation. *El-Tawil and Aguirre* (2010) discuss the potential these evacuation models have in training search and rescue teams for survivors in collapsed buildings. They have used data from earthquake disasters and building occupants as they evacuate during these disasters. In the future, they plan to couple that data with a computer model simulating the collapse of a building. This combination of both the evacuation of occupants and the collapsing of a building are what can be used to assist search and rescue teams during and after a disaster such as an earthquake. Furthermore, their approach is novel in the area of evacuation models in that the agents in their model must now interact with a dynamic environment as they evacuate.

One of the ABM models most potent to this research is the model developed by *Gwynne et al.* (2001), who improved upon an evacuation model called buildingEXODUS by implementing agent behavior unique to an environment with smoke from a fire. The three behaviors implemented for agents in this sort of environment are:

- Occupant prior exit knowledge

- Functionality when moving through smoke

- **Confrontation with smoke barrier**.

The occupant prior exit knowledge has a similar effect on agent behavior that was present in both the *Pan* (2006) and *Pelechano and Badler* (2006) models. For the most part, it is believed that agents will try to exit to their most familiar route, not necessarily the route that is the least distance. *Gwynne et al.* (2001) additionally used agent behavior such that agents must decide whether to continue moving toward their desired exit even if there is a contaminant barrier affecting that route, or whether they must change their desired exit.

Gwynne et al. (2001) also implemented agent functionality that is affected by exposure to the fire cloud. Agents' temperament as well as physical behavior were negatively affected. Further, agents were unable to make clear decisions.

The model developed in this paper particularly seeks to implement decision making of agents when faced with a contaminant cloud. According to *Gwynne et al.* (2001), when faced with a contaminant barrier, agents will either continue through the contaminant or move away and adjust evacuation route. This type of agent behavior is what this dissertation seeks to explore. However, instead of the contaminant barrier being smoke, an arbitrary contaminant is considered (as was done in Chapter II).

4.4 Results

There are three major results that are discussed throughout this section. First, an initial evacuation ABM is developed in which the contaminant simply diffuses. With this initial model, a comparison of two different decision styles implemented into the model computationally is developed and compared. An experiment was carried out where these for these two models in which there were run for a total of 1080 times. The results demonstrate how sensitive a model of micro-level human behavior is to the means by which it is programmed.

Based on the original model, a new evacuation ABM is developed that mimics the space of the CFD flow-control model developed in Chapter II. A contaminant is injected into the space that is loyal to the physical model from the CFD flow-control model and with which that the evacuees must interact.

Finally, the new evacuation ABM is coupled with the CFD flow-control model such that the models feed data back and forth to each other that influence the behavior of the phenomena in the models.

4.4.1 A Comparison of Two Different Decision Styles

The original evacuation ABM was programmed such that agents were able to employ one of two decision make styles: (i) a *discrete* decision style or (ii) *combination* decision style. Both decision styles incorporated the agent's pre-assigned risk tolerance, but the decision on how to move was implemented differently for the two.

Three types of measures were put into place in order to analyze the emergent behavior. First, the percent of agents evacuated was measured. The second measurement was the percent average contaminant exposure. The average contaminant exposure at each time step for each agent was divided by the average contaminant for each cell in the world. The third measurements are with the three measurements were plotted on the same graph.

Discrete Strategy. For the discrete decision, the *risk tolerance* is what determines how the agent will choose from the other two options. Again, a new random number is computed between the values of 0.0 and 1.0. If the risk tolerance for an agent is greater than the random number computed, the agent will move to the open cell closest to the exit (i.e. it will risk being exposed to more contaminant). Otherwise, the agent will move to the open cell that has the least amount of contaminant. Snapshots of the program with a high *risk tolerance* mean versus of a low *risk tolerance* mean can be seen in Figures 4.1 and 4.2. For this case, 120 agents were present in a space the size of 71 × 71 cells. As can be seen, when there is a higher risk tolerance mean, the agents are able to evacuate quickly while also exposing themselves to more contaminant. The opposite occurs with a low risk tolerance mean.

Combination Strategy. For the combination decision, an agent's move was dependent on a defined "score" that was calculated at each step. The score is calculated based on a combination of the agent's *risk tolerance*, the cell's normalized distance to the exit, and the normalized contaminant concentration. The *normalized distance* is the distance a cell is from the exit divided by the maximum length of the world. The *normalized contaminant concentration* is the amount of contaminant in the cell divided by the maximum contaminant that is allowed into the system per cell.

Each step, a random number is chosen between 0.0 and 1.0. If the risk tolerance is greater than the random number, then the following combination is used to calculate the score:

$$score = \frac{c_{norm}^2}{r_t} + \frac{d_{norm}}{1 - r_t} \tag{4.1}$$

where c_{norm} is the normalized contaminant value, d_{norm} is the normalized distance, and $risk$ is the risk tolerance.

If the risk tolerance is not greater than the random number, the following is used to calculate the score:

$$score = \frac{d_{norm}^2}{r_t} + \frac{c_{norm}}{1 - r_t}. \tag{4.2}$$

Snapshots of the program with a high *risk tolerance* mean versus of a low *risk tolerance* mean can be seen in Figures 4.3 and 4.4. For this case, 120 agents were

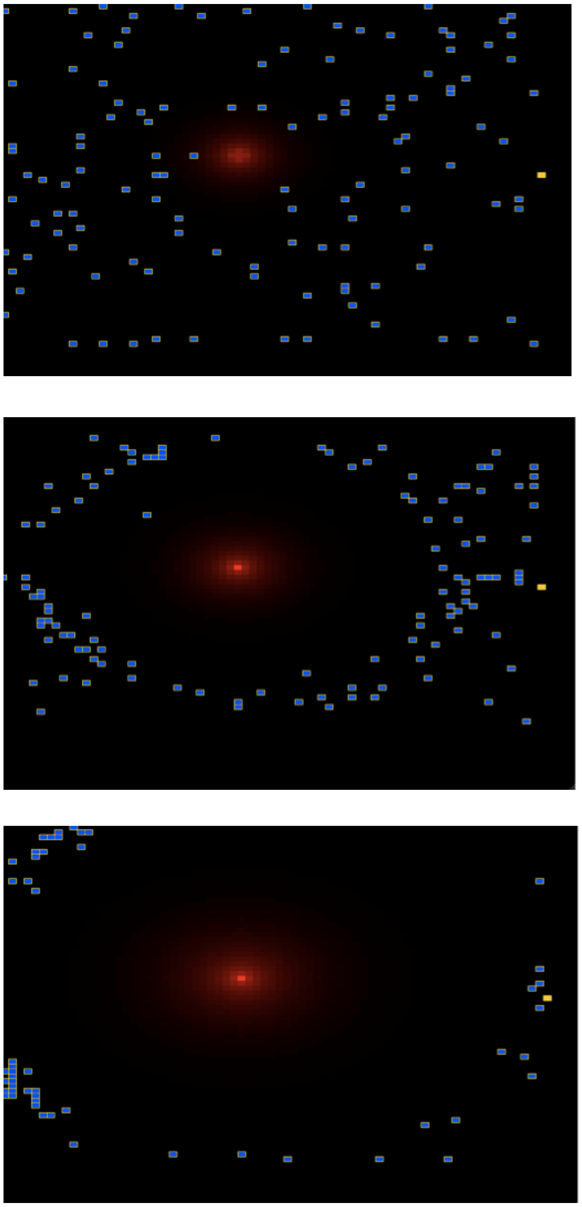


Figure 4.1: Evacuation patterns when agents have a lower risk tolerance mean using first (discrete) decision strategy (risk tolerance mean = 0.2).

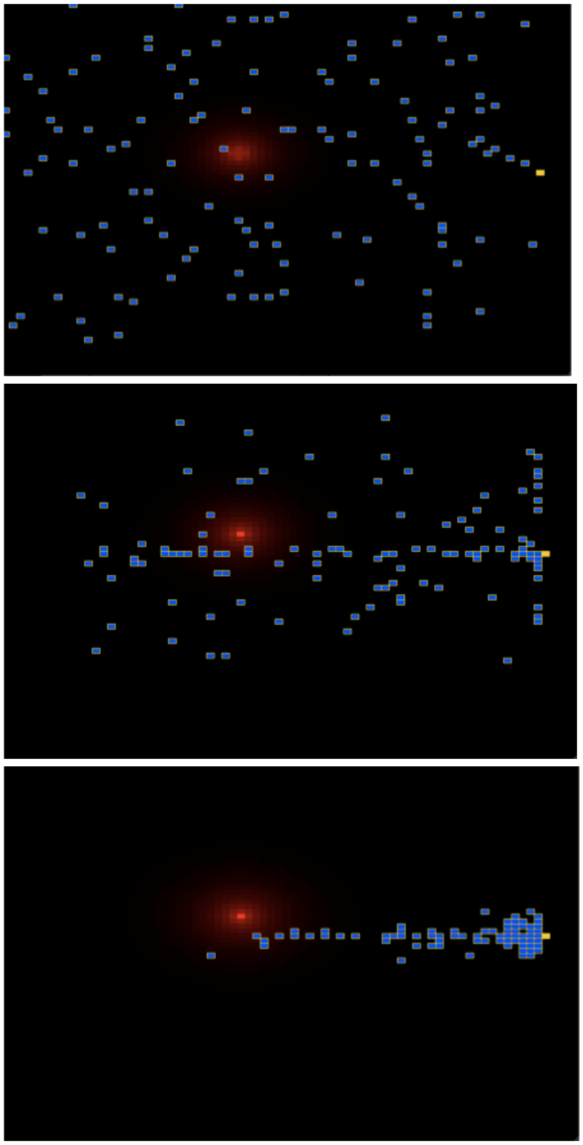


Figure 4.2: Evacuation patterns when agents have a higher risk tolerance mean first (discrete) decision strategy (risk tolerance mean = 0.8).

> present in a space the size of 71 × 71 cells. As can be seen, the evacuation time is much quicker using this decision strategy. Further, exposure to contaminant is somewhat reduced. The only time the contaminant exposure is not reduced when compared to Decision Strategy 1, is when the risk tolerance is low.

As can be seen, for the discrete decision with low risk tolerance, the percent of agents evacuated grows slowly, however, the percent contaminant exposure is small. In some cases, some agents are unable to even evacuate. The opposite is true for the runs with a high risk tolerance. As can be seen, agents have a high percent contaminant exposure, however a quick evacuation time. Agents using this strategy with high risk tolerance all are able to evacuate. However, some agents are exposed to more contaminant than the average cell contains. For the combination decision, it can be seen that evacuation time is quicker when compared with the low risk tolerance for the first decision strategy. However, contaminant exposure is somewhat higher. When compared to high risk tolerance of the first decision strategy, the agents have less exposure to contaminant, but slower evacuation time. These results were expected.

However, when comparing the second decision strategy for low and high risk factor, there does not seem to be much of a difference in behavior. Evacuation time is similar, as well as exposure to contaminant. This behavior may be due to the algorithm that was employed when the agents had to choose between cells based on the calculated score from equations (4.1) and (4.2). The equation where the agent takes into consideration the contaminant exposure as well as the distance from the exit may not vary the weight of each consideration enough when there are different risk tolerance means. Experiments were used in order to verify this hypothesis.

4.4.1.1 Experimental Analysis

Experiments were run in order to compare the discrete decision and the combination decision. Further, the experiments were used to examine the combination

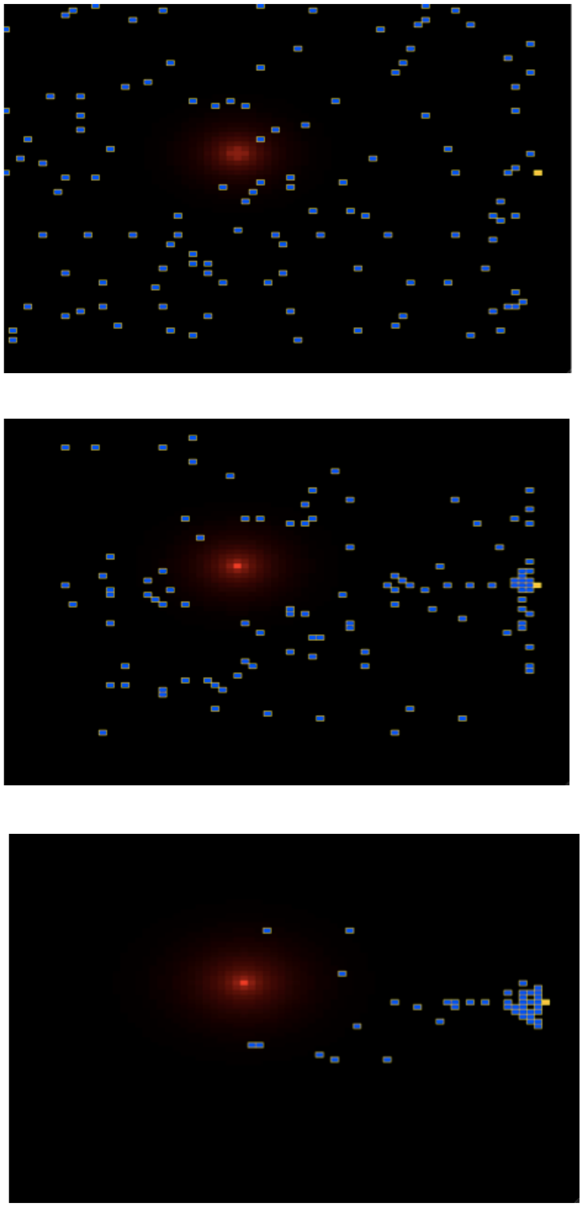


Figure 4.3: Evacuation patterns when agents have a lower risk tolerance mean using second (combination) decision strategy (risk tolerance mean = 0.2).

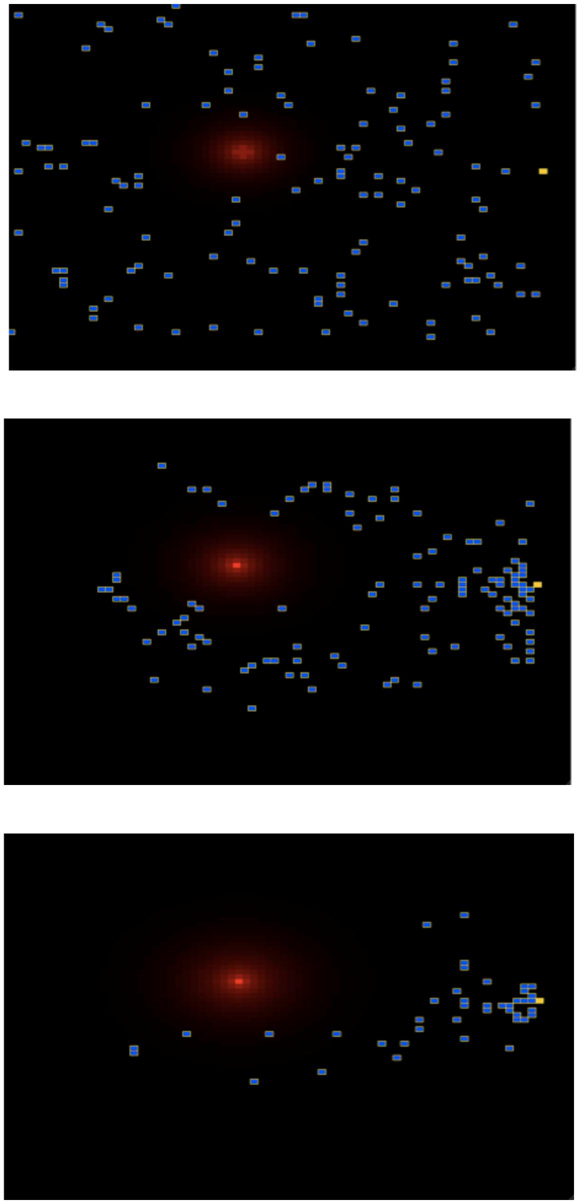


Figure 4.4: Evacuation patterns when agents have a higher risk tolerance mean second (combination) decision strategy (risk tolerance mean = 0.8).

decision to see if the different risk tolerance values led to distinct differences in agent behavior. Twenty runs were made each for risk tolerance means of 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, and 0.90 and corresponding standard deviations of 0.00, 0.10, and 0.20. These runs were carried out using both the discrete and combination decision making strategies. Thus, there was a total of 1080 runs. Each run the size of the world was 71 × 71 cells with 120 agents.

Contaminant Exposure. The first results from the experiments examined the contaminant exposure. The average contaminant exposure was taken for each time step across the twenty runs for each case. The maximum contaminant exposure for each case was found. These maximum contaminant exposure values were then plotted against each other and can be seen in Figure 4.5. As was expected, for the discrete decision, the maximum contaminant exposure increases with risk tolerance. Further, the maximum contaminant exposure seems to stay constant until a risk tolerance value of about 0.5.

What is even more interesting is how the maximum contaminant exposure seems to stay constant for all of the combination decision. Further, the maximum contaminant exposure is about the same value as the discrete decision for low risk tolerance means.

Evacuation time. Each run was carried for 1000 time steps. The percent evacuation was averaged for each case for each time step. The plots of the average percent evacuation time for each case can be seen in Figures 4.6, 4.7, 4.8, 4.9, 4.10, and 4.11. As can be seen and as was expected, for the discrete decision, the percent evacuation increases much faster with higher risk tolerance means. Further, for some cases, not all of the agents are even able to evacuate within 1000 time steps.

For the combination decision, all of the cases allow the agents to evacuate within about 300 time steps. Further, there does not seem to be a distinction between

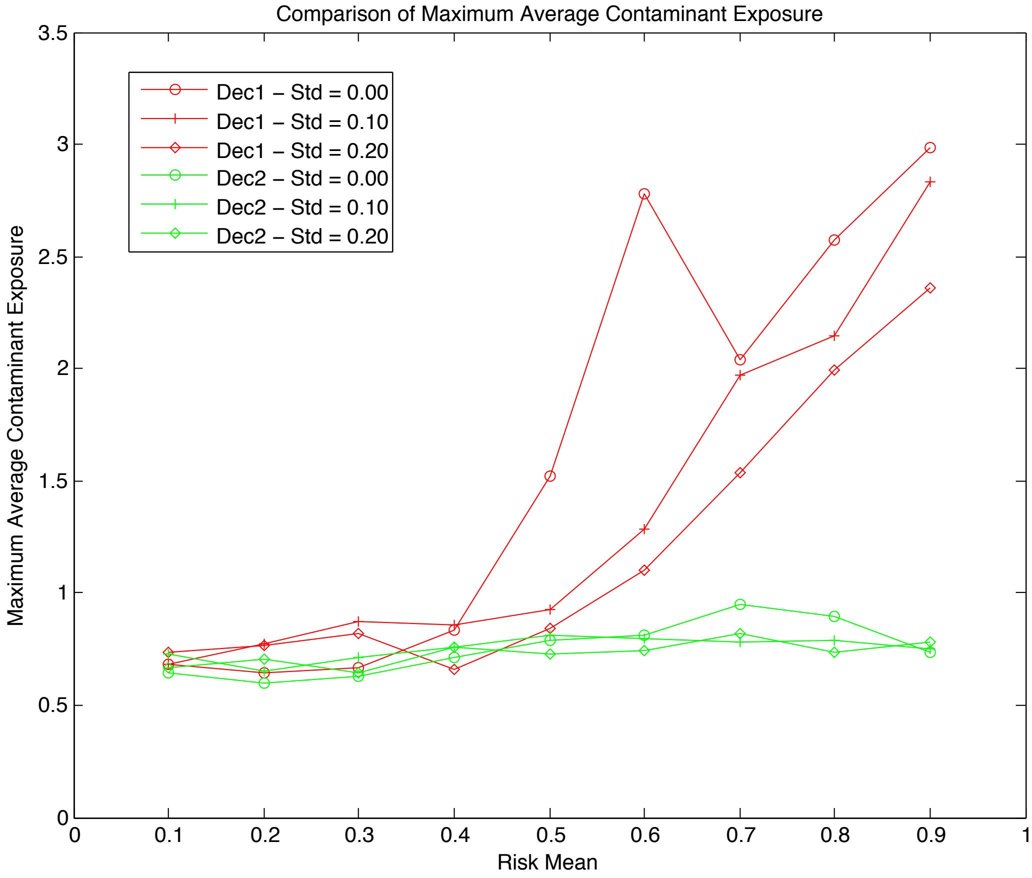


Figure 4.5: The maximum average contaminant exposure for all of the cases in the experiment.

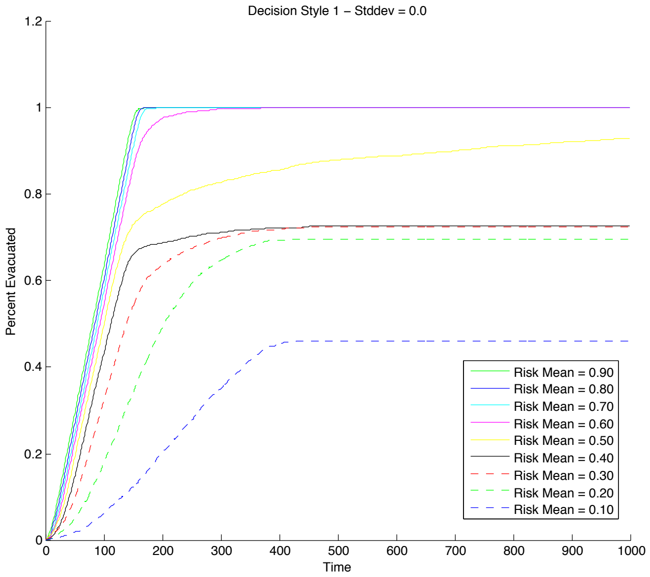


Figure 4.6: The percent evacuated versus time for decision strategy 1 with risk tolerance standard deviation of 0.00.

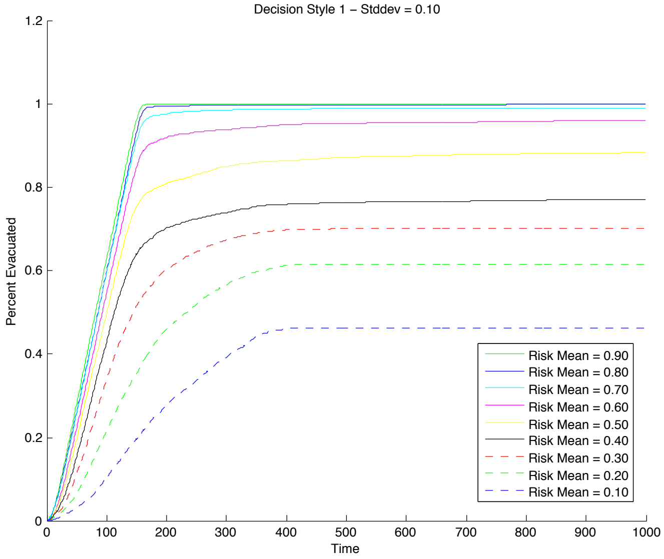


Figure 4.7: The percent evacuated versus time for decision strategy 1 with risk tolerance standard deviation of 0.10.

the evacuation time and the risk tolerance value in the same manner as the first decision making strategy. This observation seems to confirm the hypothesis

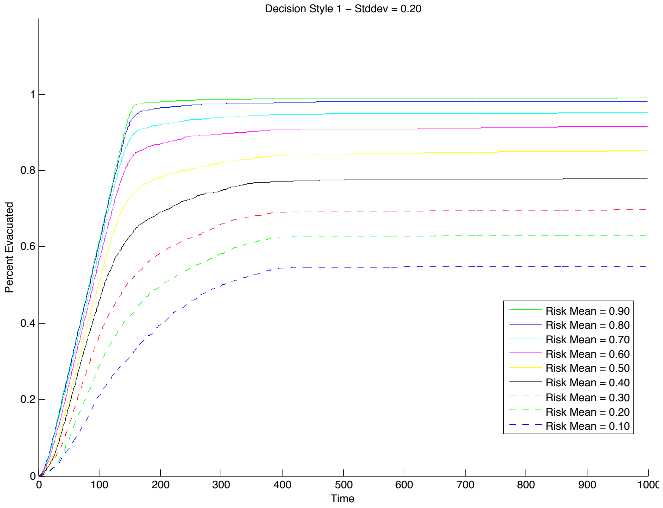


Figure 4.8: The percent evacuated versus time for decision strategy 1 with risk tolerance standard deviation of 0.20.

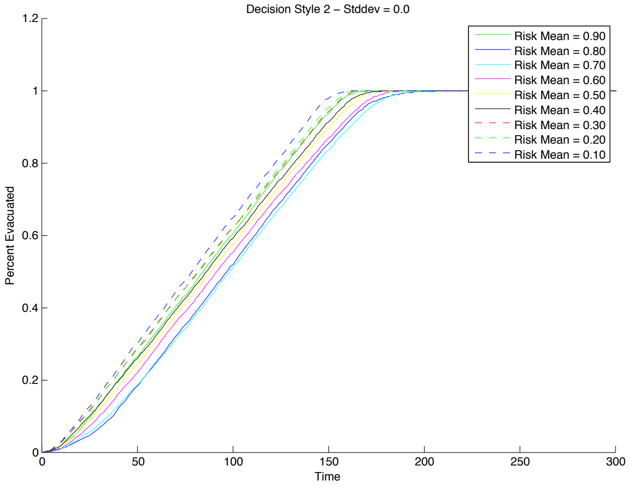


Figure 4.9: The percent evacuated versus time for decision strategy 2 with risk tolerance standard deviation of 0.00.

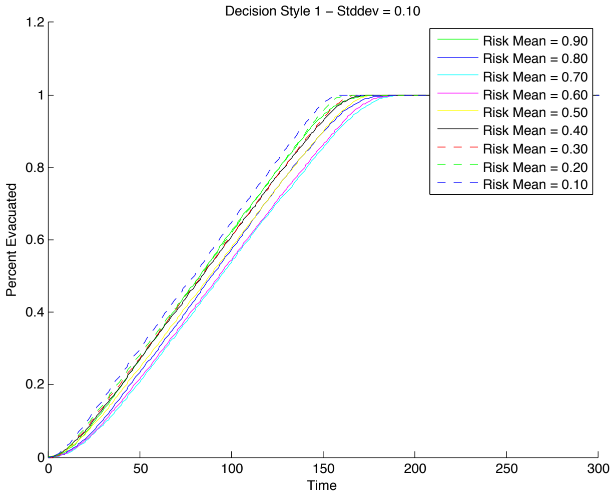


Figure 4.10: The percent evacuated versus time for decision strategy 2 with risk tolerance standard deviation of 0.10.

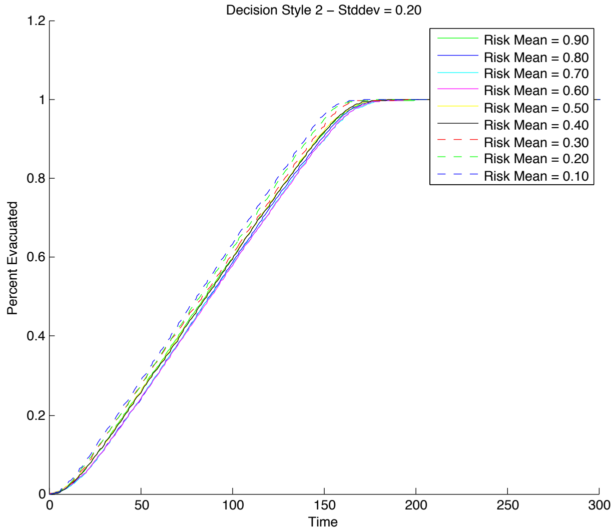


Figure 4.11: The percent evacuated versus time for decision strategy 2 with risk tolerance standard deviation of 0.20.

above that believes the combination decision algorithm might not be optimal in allowing the agent behavior to vary with regards to the risk tolerance.

What is notable about this experiment is that the combination decision does not have as distinct aggregate behavior as the discrete decision for the sweep of risk tolerance means. There are a few points to make about this. First, the idea for the combination decision was somewhat arbitrary, and was carried out in the early portions of this research with the naive desire to complicate the model. While it is not to say whether or not a discrete decision style implemented here is more "correct" than the combination decision style. To make that conclusion, the model would need to be compared against data from an actual evacuation event. However, even then, at this point it is unknown if humans make decisions via discrete thoughts, or a linear combination of thoughts. Instead, what it demonstrates is that at the agent-level, the way decisions are made *affects aggregate behavior*. What may seem like an innocuous implementation of a "fancier" calculation for egress, actually has a large influence on the emergent behavior at hand.

Thus, it was based on these results that the iteration of the evacuation ABM would keep the decision making process as basic as possible, and agents would make decisions via the *discrete* method. It was also a sobering message that as an engineer, the desire to make a model as all-encompassing as possible should be met with a critical eye–that complicatedness *is not equivalent* to complexity.

4.4.2 Evacuation ABM coupled with CFD Flow-Control Model

One of the major ideas of complex systems is that emergent behavior does not occur in a vacuum. That is for any complex system to be studied, a successful model is one that is able to isolate a phenomena even within its broader contextual system. Furthermore, only bring in an aspect of the broader context when it is fully understood. For example, if we return to the ants from Section 4.2, the model is able to describe an isolated series of aggregate behavior of the ant colony. However, this model in no way able to capture *everything* that is occurring with this ant colony. For example, it doesn't consider how seasonal changes influence it, etc. Even though it is obvious that the *rest of the world* could in someway affect it. However, because the *rest of the world* (i.e. the broader context) is not necessarily well-understood by the modeler, it is not brought in. Another component to this system should only be brought in when it is also well understood.

It is this exact sentiment that makes the research approach in this dissertation quite unique. The underlying physical processes that are occurring within the broader context of the evacuation at hand *are understood.* Thus, combining the two models only strengthens the investigation into both topics.

Eventually we rebuilt the original model developed in Section 4.4.1 using a more recent version of Repast, and coupled with the CFD flow-control model (but without any control). Using the discrete decision style, two screenshots of the example runs with average high and low risk tolerances can be seen in Figure 4.12 and Figure 4.13, respectively. The space is a 100×10 long conduit space with one exit at the middle. The space is occupied with 100 agents each. The agents are yellow, the exit is blue (in the middle of the space), and the contaminant is red. As can be seen, the model behaves quite similarly to the earlier model employing the same discrete decision making: for the model with a low risk tolerance, not all of the agents are able to evacuate; for the high risk tolerance, the agents are indeed able to evacuate.

The model was then closely coupled with the CFD flow-control model via the ZeroMQ messaging system that flags each respective program to wait until the other finishes running for a given window of time. Again, using the discrete decision style, two screenshots of the example runs with average high and low risk tolerances can be seen in Figure 4.14 and Figure 4.15, respectively. For these runs, the space is again 100×10 long conduit space with one exit at the middle. The space is occupied with

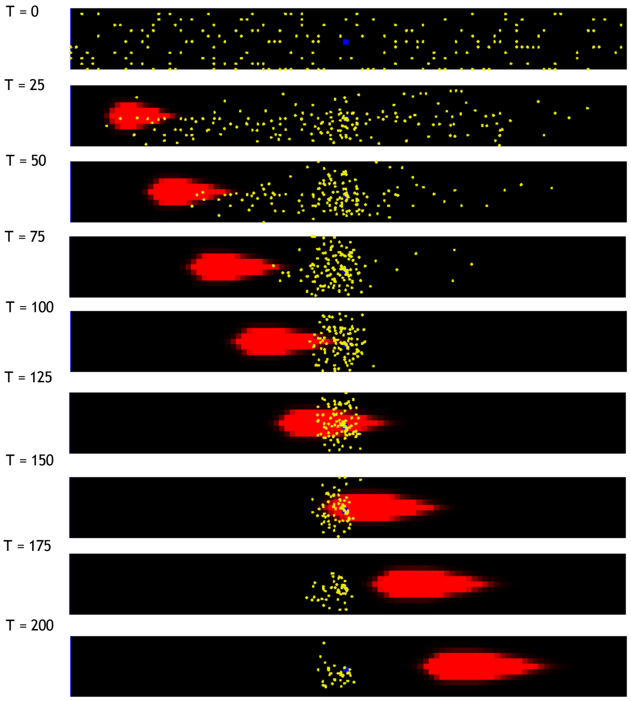


Figure 4.12: The average risk tolerance is 0.8, thus as can be seen, evacuees are moving toward the exit rather than away from the contaminant.

150 agents each. For the prediction horizon, we set $T_f = 20$, $T_a = 5$, and $T = 10$.

4.5 Discussion

An evacuation ABM is successfully developed and coupled with a CFD flow-control model, demonstrating a unique ability to combine a physical with a social model. The implications for this coupling can potentially be far reaching for implementation of a real-time flow control system in an actual public space. It may be possible to better understand how the control of a fluid will impact the evacuees (e.g. a certain type of control could lead to higher mixing, which may cause greater contaminant exposure to evacuees).

To improve upon the current model, one of the major steps that can be taken is a sweep of parameters for the evacuation ABM when coupled with the CFD flow-control model. Some parameters that can be studied include the density of occupants in the

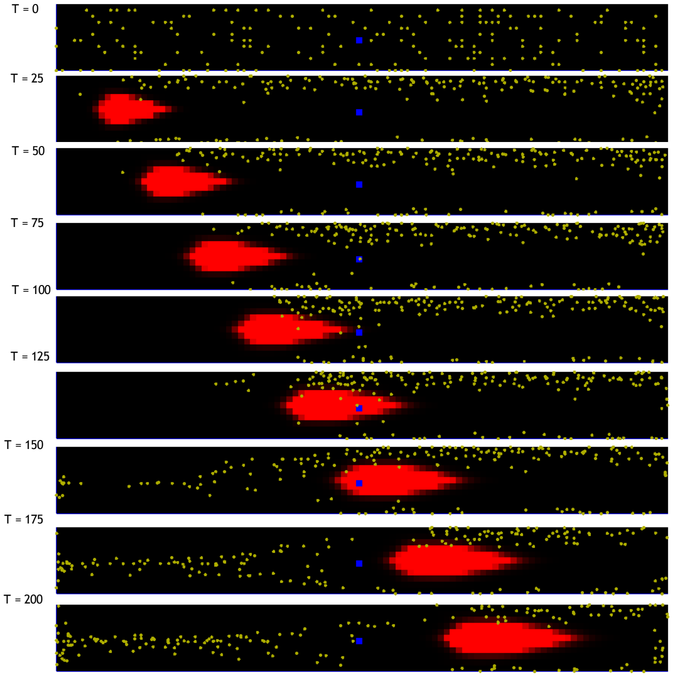


Figure 4.13: The average risk tolerance is 0.2, thus as can be seen, evacuees are moving away from the contaminant rather than toward the exit.

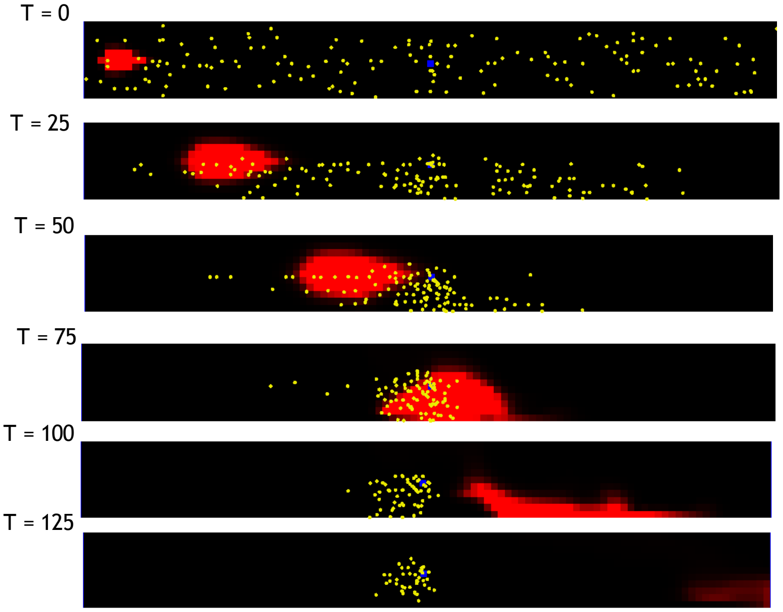


Figure 4.14: The average risk tolerance is 0.8, thus as can be seen, evacuees are moving toward the exit rather than away from the contaminant.

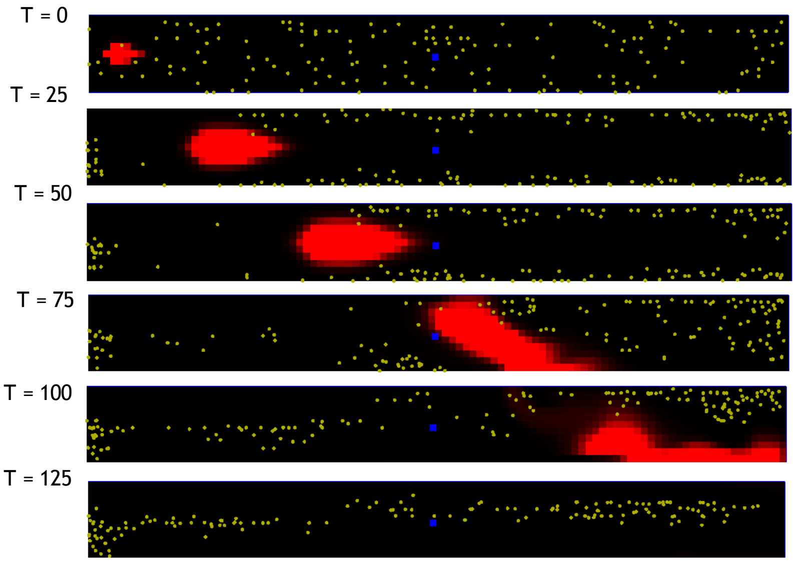


Figure 4.15: The average risk tolerance is 0.2, thus as can be seen, evacuees are moving away from the contaminant rather than toward the exit.

space and the location of the exit. Additionally, measures for contaminant exposure to the evacuees (both at the individual time step, and for the whole simulation) should be implemented to better understand how the control of the contaminant influences overall contaminant exposure to the population.

Again, another question arises regarding the length and time scales of the evacuation ABM when coupled with the CFD flow-control model. Currently, the grid sizes of the evacuation ABM match that of the CFD flow-control model (i.e. one person in the evacuation ABM is equal to one computational element in the CFD flow-control model). However, a more refined computational grid might be necessary if more detailed fluid measurements such as turbulence are taken into consideration for occupant exposure and health.

CHAPTER V

Conclusion

5.1 Summary of Results

In summary, this research develops a comprehensive means to address hazardous contaminant release in public spaces where human's are present. In particular, a CFD flow-control model has been developed that is capable of detection and mitigation of a contaminant via boundary ports along a domain. The CFD is developed using the open source OpenFOAM software suite, and employing the Navier-Stokes equations for incompressible viscous transient fluid flow; the contaminant is modeled using a basic scalar transport equation. The CFD OpenFOAM model is coupled to the DAKOTA optimization suite such that it minimizes the contaminant at given protection points for a specific time horizon using a quasi-Newton optimization scheme. An iterative feedback flow control algorithm is implemented for the CFD modeling and optimization using model predictive control. This research has shown different types of control that occur based on different location patterns of protected points. Additionally, this research has applied the CFD flow-control model to both the prototype and evacuation model developed.

A laboratory-scale experimental prototype was built using a blower wind tunnel as its design basis such that uniform flow is provided at the entryway to a test section. This wind tunnel's test section can reach > 7 m in length, and has a 0.37 m^2

cross-sectional area. A control system has been design such that it injects a 'contaminant' (propylene glycol) at the entrance of the test section, is detected via a CCD digital camera optical sensor, and is controlled via a compressed air regulated vacuum nozzle. The controller that reads in the images from the camera and sends voltage signals to the regular is a National Instruments PXIe Real-Time controller. This controller processes the images into a black and white photo, and once the sum of the pixels on this black and white photo is greater than a specified threshold, a signal is sent to the electronic pressure regulator to turn on the vacuum nozzle and draw out the contaminant. The controller is programmed using the National Instruments LabVIEW Real-time software and programming language. The CFD flow-control model is run such that it qualitatively reinforces the behavior of the contaminant of the wind tunnel when it is drawn out.

An evacuation ABM is developed using the Repast software suite and the Java language. A contaminant is injected into a grid occupied by agents who make decisions every time step based on their propensity to expose themselves to a contaminant if it moves them closer to the exit (what is deemed here as an agent's *risk tolerance*). Simulations are carried out with the model to demonstrate how the distribution of this risk tolerance value to the broader population of agents leads to aggregate population behavior. Namely, for a population with an overall higher risk tolerance, the population evacuates quicker, but exposes itself to a higher amount of contaminant; likewise, a population with an overall lower risk tolerance exposes itself to less contaminant, but might not completely evacuate the space. The CFD flow-control model is coupled to the evacuation ABM such that the CFD flow-control controls the contaminant to minimize exposure to agents in the evacuation ABM, and the agents in the evacuation ABM potentially make egress decisions on the new location and concentration of the contaminant.

5.2 Future Work

As with any dissertation, the results that emerge from the research carried out are often only a fragment to what become the more significant questions from that research. Thus, I like to think of the "next steps" for this work as falling within three different areas: the *bettering* of what has already been accomplished; the *applying* of what has been developed to actual civil systems; and the *furthering* of this work to what I believe are the next major research questions within the framework of this dissertation.

5.2.1 Bettering

The bettering of this project is quite difficult as from one perspective it seems as though the changes and improvements to the work are endless. However, there are some specific improvements that could be made to all that was presented throughout this dissertation that would enhance to overall research basis.

- *Parameter Sweeps.* Because so much of the work presented throughout this dissertation was "proof-of-concept," some parameter sweeps for the CFD flow-control model, the physical prototype, *and* the evacuation ABM would bring out the sensitivities to all three parts of the research.

- *PIV Experiments.* To obtain a general understanding of how the underlying wind tunnel itself performs (i.e. uniform velocity at the test section entrance), and to also better understand the fluid dynamics influences from the vacuum nozzle, particle image velocimetry could be used to measure velocity for a given cross-section in the physical model.

- *Additional terms to the CFD flow-control cost functional.* Currently the CFD flow-control model only controls to minimize a contaminant at specific points.

However, it might be beneficial to include another parameter, or added constraint, especially if related to the prototype or evacuation ABM influence.

5.2.2 Applying

As with most engineering research, a question always exists as to how to apply this research to the underlying engineering problem at hand. This particular research has endless applications in civilian systems, particularly as concern for non-localized security threats is continually on the rise. Information from city safety officials, or transportation engineers regarding the systems that are vulnerable to hazardous releases of chemicals would best inform the local physical and social components of their systems that this research could be applied to.

In particular, I am interested to know what exactly the fluid control possibilities are inside of a building using boundary control. How capable is an HVAC system as being able to draw out a contaminant?

For further application of this research, similitude analysis would need to be carried out across the evacuation ABM, the physical model, and the CFD flow-control model. While to a certain extent, the geometry is scaled across the three, time scales should also be considered.

5.2.3 Furthering

5.2.3.1 Real-time

I most admire the research being carried out in the area of Machine Learning. In particular, the real-time sensing and control of water systems by my committee member has inspired me to consider how to include adaptive methods along with the CFD flow-control model as it is deployed in real-time. Additionally, I am interested in using and building upon techniques in this area by those specifically investigating machine learning with regard to fluid mechanics [*Balajewicz et al.* (2015); *Duriez et al.*

(2014)].

5.2.3.2 Optimal Control of Evacuation.

As optimal control was applied to the CFD flow-control model, I am also interested in how it would apply to a bottom-up ABM model such as the evacuation ABM from this research. If optimal control were possible, there exists the possibility to use optimal control and communicate to evacuees optimal evacuation routes.

5.2.3.3 The Potential of Sensors.

The project that this research was a part initially included a component for in-house fabricated sensors to be used with the ability to detect a wide array of chemicals. While those sensors were never deployed, the possibilities for expanding this research regarding sensors is particularly timely.

BIBLIOGRAPHY

BIBLIOGRAPHY

Abergel, F., and R. Temam (1990), On some control problems in fluid mechanics, *Theoretical and Computational Fluid Dynamics*, *1*(6), 303–325.

Adams, B. M., K. R. Dabley, M. S. Eldred, L. P. Swiler, W. Bohnhoff, J. P. Eddy, and D. M. Vigil (2009), Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantifcation, and sensitivity analysis version 5.2 user's manual.

Aguirre, B., S. El-Tawil, E. Best, K. B. Gill, and V. Fedorov (2011), Contributions of social science to agent-based models of building evacuation, *Contemporary Social Science*, *6*(3), 415–432.

Alvarez-Vzquez, L., A. Martnez, M. Vzquez-Mndez, and M. Vilar (2009), An application of optimal control theory to river pollution remediation, *Applied Numerical Mathematics*, *59*(5), 845–858, doi:http://dx.doi.org/10.1016/j.apnum.2008.03.027.

Amsden, A. A., and F. H. Harlow (1970), A simplified mac technique for incompressible fluid flow calculations, *Journal of Computational Physics*, *6*(2), 322–325.

Anderson, J. D., and J. Wendt (1995), *Computational fluid dynamics*, vol. 206, Springer.

Axelrod, R. (1981), The emergence of cooperation among egoists, *American political science review*, *75*(02), 306–318.

Balajewicz, M., D. Amsallem, and C. Farhat (2015), Projection-based model reduction for contact problems, *International Journal for Numerical Methods in Engineering*.

Bale, J. (2000), The changing face of football: Stadiums and communities, *Soccer and Society*, *1*(1), 91–101.

Barlow, J., W. Rae, and A. Pope (), Low-speed wind tunnel testing, 1999, *Jhon Wiley & Sons, Canada*.

Bearman, P., and J. Harvey (1993), Control of circular cylinder flow by the use of dimples, *AIAA journal*, *31*(10), 1753–1756.

Bell, J. H., and R. D. Mehta (1988), Contraction design for small low-speed wind tunnels.

Berggren, M. (1998), Numerical solution of a flow-control problem: Vorticity reduction by dynamic boundary action, *SIAM Journal on Scientific Computing*, *19*(3), 829–860, doi:10.1137/S1064827595294678.

Bewley, T., and P. Moin (1994), Optimal control of turbulent channel flows, *Active Control of Vibration and Noise*, pp. 221–227.

Bewley, T., H. Choi, R. Temam, and P. Moin (1993), Optimal feedback control of turbulent channel flow.

Bewley, T. R., P. Moin, and R. Temam (2001), DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms, *Journal of Fluid Mechanics*, *447*(2), 179–225.

Boccara, N. (2010), *Modeling complex systems*, Springer Science & Business Media.

Bradshaw, P., and R. Pankhurst (1964), The design of low-speed wind tunnels, *Progress in Aerospace Sciences*, *5*, 1–69.

Branscomb, L. M., R. D. Klausner, et al. (2002), Making the nation safer: the role of science and technology in countering terrorism, *Committee on Science and Technology for Countering Terrorism, National Research Council*.

Camelli, F., and R. Löhner (2004), Assessing maximum possible damage for contaminant release events, *Engineering computations*, *21*(7), 748–760.

Carlton, J. (2001), Of microbes and mock attacks: Years ago, the military sprayed germs on us cities, *Wall Street Journal*.

Chen, K., X. Jin, and J. Zhao (2009), Design and characteristics of a large boundary-layer wind tunnel with two test sections, in *The Seventh Asia-Pacific Conference on Wind Engineering*.

de Villiers, E. (2006), The potential of large eddy simulation for the modeling of wall bounded flows, Ph.D. thesis, Imperial College of Science, Technology and Medicine.

Des Jardins, J. (2010), *The Madame Curie complex: The hidden history of women in science*, Feminist Press at CUNY.

Dolgin, E. (2013), Syrian gas attack reinforces need for better anti-sarin drugs, *Nature medicine*, *19*(10), 1194–1195.

Duriez, T., V. Parezanović, L. Cordier, B. R. Noack, J. Delville, J.-P. Bonnet, M. Segond, and M. Abel (2014), Closed-loop turbulence control using machine learning, *arXiv preprint arXiv:1404.4589*.

El-Tawil, S., and B. Aguirre (2010), Search and rescue in collapsed structures: engineering and social science aspects, *Disasters*, *34*(4), 1084–1101.

Evison, D., D. Hinsley, and P. Rice (2002), Chemical weapons, *British Medical Journal*, *324*(7333), 332.

Falkenrath, R. A., R. D. Newman, and B. A. Thayer (1998), *America's Achilles' heel: nuclear, biological, and chemical terrorism and covert attack*, MIT Press.

Fang, J. (2015), Computational study of social interactions and collective behavior during human emergency egress, Ph.D. thesis, University of Michigan.

Fang, J. S., and S. El-Tawil (2013), Egress simulation incorporating social relationships by using scalar field method, *Advanced Materials Research*, *639*, 1313–1316.

Farell, C., and A. K. Iyengar (1999), Experiments on the wind tunnel simulation of atmospheric boundary layers, *Journal of wind engineering and industrial aerodynamics*, *79*(1), 11–35.

Ferziger, J. H., and M. Peric (2012), *Computational methods for fluid dynamics*, Springer Science & Business Media.

Gad-el Hak, M. (1989), Flow control, *Applied mechanics reviews*, *42*(10), 261–293.

Gad-el Hak, M., A. Pollard, and J.-P. Bonnet (2003), *Flow control: fundamentals and practices*, vol. 53, Springer Science & Business Media.

Gao, R., A. Li, W. Lei, Y. Zhao, Y. Zhang, and B. Deng (2012), Study of a proposed tunnel evacuation passageway formed by opposite-double air curtain ventilation, *Safety science*, *50*(7), 1549–1557.

Gardner, M. (1970), Mathematical games: The fantastic combinations of john conway?s new solitaire game ?life?, *Scientific American*, *223*(4), 120–123.

Gell-Mann, M. (1995), *The Quark and the Jaguar: Adventures in the Simple and the Complex*, Macmillan.

Gilbert, G. N. (2008), *Agent-based models*, 153, Sage.

Gordon, D. M. (1996), The organization of work in social insect colonies, *Nature*, *380*(6570), 121–124.

Graves Jr, R. (1982), Computational fluid dynamics-the coming revolution, *Astronautics Aeronautics*, *20*, 20–28.

Grune, L., and J. Pannek (2011), *Nonlinear Model Predictive Control Theory and Algorithms*, Communications and Control Engineering,0178-5354, Springer-Verlag London Limited, London.

Gwynne, S., E. Galea, P. Lawrence, and L. Filippidis (2001), Modelling occupant interaction with fire conditions using the buildingexodus evacuation model, *Fire Safety Journal*, *36*(4), 327–357.

Henderson, D. A. (1999), The looming threat of bioterrorism, *Science*, *283*(5406), 1279–1282.

Issa, R. (1986), Solution of the implicitly discretised fluid flow equations by operator-splitting, *Journal of Computational Physics*, *62*, 40–65, doi:10.1016/0021-9991(86) 90099-9.

Jameson, A. (1988), Aerodynamic design via control theory, *Journal of Scientific Computing*, *3*(3), 233–260.

Jameson, A. (1995), Optimum aerodynamic design using CFD and control theory, *AIAA paper*, *1729*, 124–131.

Jameson, A. (1999), Re-engineering the design process through computation, *Journal of Aircraft*, *36*(1), 36–50.

Jameson, A., L. Martinelli, and N. A. Pierce (1998), Optimum aerodynamic design using the NavierStokes equations, *Theoretical and Computational Fluid Dynamics*, *10*(1-4), 213–237.

Jasak, H. (1996), Error analysis and estimation for the finite volume method with applications to fluid flows, Ph.D. thesis, Imperial College of Science, Technology and Medicine.

Juergensmeyer, M. (2005), *Terror in the mind of God: The global rise of religious violence*, Taylor & Francis.

Katopodes, N. D. (2009), Control of sudden releases in channel flow, *Fluid Dynamics Research*, *41*(6), 24.

Katopodes, N. D., and M. Piasecki (1996), Site and size optimization of contaminant sources in surface water systems, *Journal of Environmental Engineering-Asce*, *122*(10), 917–923, doi:10.1061/(asce)0733-9372(1996)122:10(917).

Kolmogorov, A. N. (1941), The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers, in *Dokl. Akad. Nauk SSSR*, vol. 30, pp. 301–305, JSTOR.

Lions, J. (1966), Controle optimal dc syste mes gouverne's par des equations aux de'rivées partielles, dunod, gauthier villars, paris, 1968.(english translation by sk mitter, grundlehren, springer 170, 1971.)[2] quelques m éthodes dc résolution des proble mes aux limites non line'aires, dunod, gauthier villars, paris, 1969.(english translation by le van, holt, rinehart, and winston, new york, 1972.)'[3] sur le controle optimal dc systemes de'crits par des equations aux dérivées partielles linéaires.(i) equations elliptiques,(11) equations dévolution, *CR. Acad. Sci. Paris*, *263*, 713–715.

Macal, C. (2016), Everything you need to know about agent-based modelling and simulation, *Journal of Simulation*, *10*(2), 144–156.

Mansfield, M. (2016), Hillsborough: the power of the people is so much stronger than the people in power, *Socialist Lawyer*, (73), 18–23.

Mays, L. W., D. Koutsoyiannis, and A. N. Angelakis (2007), A brief history of urban water supply in antiquity, *Water Science and Technology: Water Supply*, *7*(1), 1–12.

Mehta, R. (1979), The aerodynamic design of blower tunnels with wide-angle diffusers, *Progress in Aerospace Sciences*, *18*, 59–120.

Mehta, R. D., and P. Bradshaw (1979), Design rules for small low speed wind tunnels, *The Aeronautical Journal (1968)*, *83*(827), 443–453.

Meza, J. C., R. A. Oliva, P. D. Hough, and P. J. Williams (2007), Opt++: An object-oriented toolkit for nonlinear optimization, *ACM Trans. Math. Softw.*, *33*(2), doi: 10.1145/1236463.1236467.

Moin, P., and T. Bewley (1994), Feedback control of turbulence, *Appl. Mech. Rev*, *47*(6), S3–S13.

Moukalled, F., L. Mangani, and M. Darwish (2015), *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab*, vol. 113, Springer.

Murakami, H. (2010), *Underground: The Tokyo gas attack and the Japanese psyche*, Vintage.

North, M. J., N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko (2013), Complex adaptive systems modeling with repast simphony, *Complex adaptive systems modeling*, *1*(1), 1.

Okumura, T., K. Suzuki, A. Fukuda, A. Kohama, N. Takasu, S. Ishimatsu, and S. Hinohara (1998), The tokyo subway sarin attack: disaster management, part 1: Community emergency response, *Academic Emergency Medicine*, *5*(6), 613–617.

Pan, X. (2006), Computational modeling of human and social behaviors for emergency egress analysis., Ph.D. thesis, Stanford University.

Pan, X., C. S. Han, K. Dauber, and K. H. Law (2007), A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations, *Ai & Society*, *22*(2), 113–132.

Pelechano, N., and N. I. Badler (2006), Modeling crowd and trained leader behavior during building evacuation, *Departmental Papers (CIS)*, p. 272.

Pelechano, N., K. O'Brien, B. Silverman, and N. Badler (2005), Crowd simulation incorporating agent psychological models, roles and communication, *Tech. rep.*, DTIC Document.

Piasecki, M., and N. D. Katopodes (1997a), Control of contaminant releases in rivers. II: optimal design, *Journal of Hydraulic Engineering*, *123*(6), 493503.

Piasecki, M., and N. D. Katopodes (1997b), Control of contaminant releases in rivers. i: Adjoint sensitivity analysis, *Journal of Hydraulic Engineering*, *123*(6), 486492.

Pope, S. B. (2001), Turbulent flows.

Potter, M. C., D. C. Wiggert, and B. H. Ramadan (2011), *Mechanics of Fluids SI Version*, Nelson Education.

Prandtl, L. (1904), Uber flussigkeits bewegung bei sehr kleiner reibung, *Verhaldlg III Int. Math. Kong*, pp. 484–491.

Quarantelli, E. L. (2001), The sociology of panic.

Quarteroni, A., and G. Rozza (2003), Optimal control and shape optimization of aorto-coronaric bypass anastomoses, *Mathematical Models & Methods in Applied Sciences*, *13*(12), 1801–1823.

Reynolds, C. W. (1987), Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH computer graphics*, *21*(4), 25–34.

Richardson, W. (1993), Identifying the cultural causes of disasters: An analysis of the hillsborough football stadium disaster, *Journal of Contingencies and Crisis Management*, *1*(1), 27–35.

Sanders, B. F., and N. D. Katopodes (1999), Control of canal flow by adjoint sensitivity method, *Journal of Irrigation and Drainage Engineering-Asce*, *125*(5), 287–297, doi:10.1061/(asce)0733-9437(1999)125:5(287).

Sanders, B. F., and N. D. Katopodes (2000), Adjoint sensitivity analysis for shallow-water wave control, *Journal of Engineering Mechanics-Asce*, *126*(9), 909–919.

Schelling, T. C. (1980), *The strategy of conflict*, Harvard university press.

Scraton, P. (1999), Policing with contempt: The degrading of truth and denial of justice in the aftermath of the hillsborough disaster, *Journal of Law and Society*, *26*(3), 273–297.

Settles, G. S. (2006), Fluid mechanics and homeland security, *Annu. Rev. Fluid Mech.*, *38*, 87–110.

Shanno, D. F. (1970), Conditioning of quasi-newton methods for function minimization, *Mathematics of Computation*, *24*(111), 647–656, doi:10.2307/2004840, times Cited: 1011 1032.

Shih, Y.-C., A.-S. Yang, and C.-W. Lu (2011), Using air curtain to control pollutant spreading for emergency management in a cleanroom, *Building and Environment*, *46*(5), 1104–1114.

Sreedharan, P., M. D. Sohn, W. W. Nazaroff, and A. J. Gadgil (2011), Towards improved characterization of high-risk releases using heterogeneous indoor sensor systems, *Building and Environment*, *46*(2), 438–447.

Szinicz, L. (2005), History of chemical and biological warfare agents, *Toxicology*, *214*(3), 167–181.

Terna, P., et al. (1998), Simulation tools for social scientists: Building agent based models with swarm, *Journal of artificial societies and social simulation*, *1*(2), 1–12.

Tisue, S., and U. Wilensky (2004), Netlogo: A simple environment for modeling complexity, in *International conference on complex systems*, vol. 21, pp. 16–21, Boston, MA.

Uehara, K., S. Murakami, S. Oikawa, and S. Wakamatsu (2000), Wind tunnel experiments on how thermal stratification affects flow in and above urban street canyons, *Atmospheric Environment*, *34*(10), 1553–1562.

Vanderplaats, G. (1973), Conmin: A program for constrained function minimization (1973) user's manual, *NASA TMX-62282*.

Versteeg, H. K., and W. Malalasekera (2007), *An introduction to computational fluid dynamics: the finite volume method*, Pearson Education.

Wang, B. (2015), Real-time control of solute plume in closed conduit flow, Ph.D. thesis, The University of Michigan.

Wang, B., A. G. Stefanopoulou, and N. D. Katopodes (2013a), Model development for real time optimal control in pipe lines, *American Control Conference (ACC), 2013*, pp. 5536–5541.

Wang, B., A. Warnock, A. G. Stefanopoulou, and N. D. Katopodes (2013b), Model development for real time optimal control in pipe lines, in *American Control Conference (ACC), 2013*, pp. 5536–5541.

Warnock, A. (2013a), Automatic detection and control of hazardous plumes in wall-bounded flow systems, Ph.D. thesis, University of Michigan.

Warnock, A. M. (2013b), Automatic detection and control of hazardous plumes in wall-bounded flow systems, Ph.D. thesis, University of Michigan.

Washabaugh, K., M. J. Zahr, and C. Farhat (2016), On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries, in *54th AIAA Aerospace Sciences Meeting*, p. 1814.

Wendt, J. (2008), *Computational fluid dynamics: an introduction*, Springer Science & Business Media.

Wu, Y., and M. A. Bakar (2000), Control of smoke flow in tunnel fires using longitudinal ventilation systems–a study of the critical velocity, *Fire Safety Journal*, *35*(4), 363–390.

Yang, H., L. Jia, L. Yang, and P. Huang (2007), Numerical simulation of the impact of ventilation mode on subway platform air distribution, in *Proceedings of Building Simulation*.

Zheng, X., T. Zhong, and M. Liu (2009), Modeling crowd evacuation of a building based on seven methodological approaches, *Building and Environment*, *44*(3), 437–445.